

# The 4<sup>th</sup> International Conference on Artificial Intelligence in Information and Communication

 **IEEE ICAIIC 2022** 

February 21 (Mon.) ~ 24 (Thur.), 2022, Shilla Stay Jeju, Jeju Island, Korea

<http://icaaic.org>



## Proceedings

### Organized by



### Technical Co-Sponsored by



### Patrons



Institute of Information & Communications  
Technology Planning & Evaluation



- Society Safety System Forum
- Internet of Energy Research Center (Kookmin University)
- Center for ICT & Automotive Convergence (Kyungpook National University)
- AI Mobility Research Institute (Kookmin University)

## Table of Contents



Committee .....	3
Message from Organizing Chairs.....	9
Message from TPC Chairs .....	10
Program Matrix for ICAIC 2022 .....	11
Keynote Speech .....	13
Tutorial.....	15
Oral Sessions.....	18
Venue .....	28
Travel Information.....	29
proceedings.....	33

## Committee

### International Advisory Committee

Ramjee Prasad	Aarhus Univ., Denmark
Pascal LORENZ	Univ. of Haute Alsace, France
Zhisheng Niu	Tsinghua Univ., China
Ilyoung Chong	HUFS, Korea
Tomoaki Ohtsuki	Keio Univ., Japan
Robert F. Karlicek	RPI, USA
Md. Nurunnabi Mollah	KUET, Bangladesh
Joel Rodrigues	Inatel, Brazil
Myung Jong Lee	CUNY, USA
Hsi-Pin Ma	National Tsing Hua Univ., Taiwan
Honggang Wang	Univ. of Massachusetts, USA
Myung Joon Kim	ETRI, Korea
Young Sam Kim	KETI, Korea
Kiseong Lim	KILT, Korea
Jong-Seon No	Seoul National Univ., Korea
Young-Han Kim	Soongsil Univ., Korea
Makoto Naruse	Univ. of Tokyo, Japan

### Steering Committee

Yeong Min Jang	Kookmin Univ., Korea
Takeo Fujii	Univ. of Electro-Comms, Japan
Dong Seog Han	Kyungpook National Univ., Korea
Seung Hyong Rhee	Kwangwoon Univ., Korea
Seong-Ho Jeong	HUFS, Korea
Periklis Chatzimisios	ATEITHE, Greece
Xin Wang	Fudan Univ., China
Sang-Jo Yoo	Inha Univ., Korea
Honggang Zhang	Zhejiang Univ., China
Myungsik Yoo	Soongsil Univ., Korea
Won Cheol Lee	Soongsil Univ., Korea
Juan Carlos Cano	Technical Univ. of Valencia, Spain
Jungwoo Lee	Seoul National Univ., Korea
Heung-Kook Choi	Inje Univ., Korea
Sanghyun Ahn	Univ. of Seoul, Korea
Sang-Chul Kim	Kookmin Univ., Korea

Takaya Yamazato	Nagoya Univ., Japan
Ki-Hyung Kim	Ajou Univ., Korea
Hyoung Jun Kim	ETRI, Korea
Kyubok Lee	KETI, Korea
Seung Chan Bang	ETRI, Korea
Song Chong	KAIST, Korea
Dongsung Kim	Kumoh National Univ., Korea
Young-Joo Suh	POSTECH, Korea
Jongwon Kim	GIST, Korea
Linyang Song	Peking Univ., China

### Organizing Committee

#### Honorary Conference Chairs

Dong Seog Han	Kyungpook National Univ., Korea
Pascal LORENZ	Univ. of Haute Alsace, France
Ilyoung Chong	HUFS, Korea
Honggang Wang	Univ. of Massachusetts, USA
Hsi-Pin Ma	National Tsing Hua Univ., Taiwan
Joel Rodrigues	Inatel, Brazil
Hyoung Jun Kim	ETRI, Korea

#### General Chairs

Yeong Min Jang	Kookmin Univ., Korea
Takeo Fujii	Univ. of Electro-Comms, Japan

#### Area Chairs

Sang Min Yoon	Kookmin Univ., Korea
Kenji Doya	OIST, Japan
Toshihisa Tanaka	Tokyo Univ. of Agriculture and Tech., Japan
Insoon Sohn	Dongguk Univ., Korea
Ilwoo Lee	ETRI, Korea
Oh-Soon Shin	Soongsil Univ., Korea
Naoki Wakamiya	Osaka Univ., Japan

#### Regional Chair

Peer Peter	Ljubljana Univ., Slovenia
------------	---------------------------

## Committee

Organizing Vice-Chairs	
Kenta Umebayashi	Tokyo Univ. of Agriculture and Tech., Japan
Dongkyun Kim	Kyungpook National Univ., Korea
Sunwoong Choi	Kookmin Univ., Korea
Celimuge Wu	The Univ. of Electro-Communications, Japan
Workshop Chairs	
Sung-Rae Cho	Chung-Ang Univ., Korea
Hong Kook Kim	GIST, Korea
Mianxiong Dong	Muroran Institute of Tech., Japan
Special Session Chairs	
Xiaoyan Wang	Ibaraki Univ., Japan
Wansup Cho	Chungbuk National Univ., Korea
Jiwoong Choi	DGIST, Korea
Eun-Seok Ryu	Gachon Univ., Korea
International Liaison Chair	
Jong-Ho Lee	Soongsil Univ., Korea
International Journal Chairs	
Junhee Seok	Korea Univ., Korea
Yujin Lim	Sookmyung Women's Univ., Korea
Registration Chairs	
Min Young Kim	Kyungpook National Univ., Korea
Pyungsoo Kim	KPU, Korea
Local Arrangement Chairs	
Jaeyong Choi	Univ. of Guam, USA
DoHyun Kim	Jeju National Univ., Korea
Sukchan Kim	Pusan National Univ., Korea
Masato Saito	Univ. of the Ryukyus, Japan
Mai Ohta	Fukuoka Univ., Japan
Publication Chairs	
Jung Hoon Lee	HUFS, Korea
Sangjoon Park	ETRI, Korea
Publicity Chairs	
Joohyun Lee	Hanyang Univ., Korea
Kazuto Yano	ATR, Japan
Mostafa Zaman Chowdhury	KUET, Bangladesh
Yoshikazu Washizawa	The Univ. of Electro-Communications, Japan
Haeyoung Lee	University of Surrey, UK

Patronage Chairs	
Hyun-Woo Lee	ETRI, Korea
Byeongho Choi	KETI, Korea
Hye Young Park	Kyungpook National Univ., Korea
Finance Chairs	
Osamu Takyu	Shinshu Univ., Japan
Wooyong Lee	ETRI, Korea
Young-Seok Choi	Kwangwoon Univ., Korea
Web Chair	
Joon Won Choi	Hanyang Univ., Korea
Soochahn Lee	Kookmin Univ., Korea
Technical Program Committee	
TPC Chairs	
Seokjoo Shin	Chosun Univ., Korea
Youn-Hee Han	Korea University of Technology and Education, Korea
Mikio Hasegawa	Tokyo Univ. of Science, Japan
Benaoumeur Senouci	North Dakota State University, USA
TPC Co-Chairs	
Joongheon Kim	Korea Univ., Korea
Jihoon Lee	Sangmyung Univ., Korea
TPC Vice Chairs	
Takayuki Nishio	Kyoto Univ., Japan
Ohyun Jo	Chungbuk National Univ., Korea
Wooyeol Choi	Chosun Univ., Korea
TPC Members	
Muhammad Afzal	Sejong University, Korea (South)
Sandeep Agrawal	RJIT Tekanpur, India
Ijaz Ahmad	Chosun University, Korea (South)
Taqdir Ali	University of British Columbia, Canada
Mohamad Yusoff Alias	Multimedia University, Malaysia
Esraa Saleh Alomari	Wasit University, Iraq
Gayan Amarasuriya	Southern Illinois University, USA
Beongku An	Hongik University, Korea (South)
Ali Balador	Mälardalen University, Sweden
Vo Nguyen Quoc Bao	Posts and Telecommunications Institute of Technology, Vietnam

## Committee

Filipe Cardoso	ESTsetubal/Polytechnic Institute of Setubal and INESC-ID, Portugal
Chinmay Chakraborty	Birla Institute of Technology, Mesra, India
Woong Cho	Daegu Catholic University, Korea (South)
Hyun-Ho Choi	Hankyong National University, Korea (South)
Ji-Woong Choi	DGIST, Korea (South)
Jun Won Choi	Hanyang University, Korea (South)
Peter Choi	Akamai Technologies, USA
Sunwoong Choi	Kookmin University, Korea (South)
Wooyeol Choi	Chosun University, Korea (South)
Yong-Hoon Choi	Kwangwoon University, Korea (South)
Young Choi	Regent University, USA
Li-Der Chou	National Central University, Taiwan
Mostafa Zaman Chowdhury	Khulna University of Engineering & Technology, Bangladesh
Theofilos Chrysikos	University of Patras, Greece
Kwangsue Chung	Kwangwoon University, Korea (South)
Yeonho Chung	Pukyong National University, Korea (South)
Renato de Moraes	Federal University of Pernambuco (UFPE), Brazil
Amine Dhraief	University of Manouba, Tunisia
Trung Duong	Colorado State University Pueblo, USA
Zbigniew Dziong	École de technologie supérieure, University of Quebec, Canada
Yee Loo Foo	Multimedia University, Malaysia
Tapio Frantti	Finnish Research and Engineering, Finland
Vasilis Friderikos	King's College London, United Kingdom (Great Britain)
Takeo Fujii	The University of Electro-Communications, Japan
Alireza Ghasempour	University of Applied Science and Technology, USA
Debasis Giri	Haldia Institute of Technology, India
Weihan Goh	Singapore Institute of Technology, Singapore
Javier Gozálvez	Universidad Miguel Hernandez de Elche, Spain
Zygmunt Haas	Cornell University, USA
Majed Haddad	University of Avignon, France
Dong Seog Han	Kyungpook National University, Korea (South)
Youn-Hee Han	Korea University of Technology and Education, Korea (South)

Mikio Hasegawa	Tokyo University of Science, Japan
Ibrahim Hokelek	TUBITAK BILGEM, Turkey
Shih-Cheng Horng	Chaoyang University of Technology, Taiwan
Sayed Jahed Hussini	Western Michigan University, USA
Nguyen Huu Thanh	Hanoi University of Science and Technology, Vietnam
Euseok Hwang	Gwangju Institute of Science and Technology, Korea (South)
Ganguk Hwang	KAIST, Korea (South)
Takeshi Ikenaga	Kyushu Institute of Technology, Japan
Eun-Jin Im	Kookmin University, Korea (South)
Keisuke Ishibashi	International Christian University, Japan
Yeong Min Jang	Kookmin University, Korea (South)
Seong-Ho Jeong	Hankuk University of Foreign Studies, Korea (South)
Anxiao Andrew Jiang	Texas A&M University, USA
Yutaka Jitsumatsu	Kyushu University, Japan
Ohyun Jo	Chungbuk National University, Korea (South)
Changhee Joo	Korea University, Korea (South)
Jingon Joung	Chung-Ang University, Korea (South)
Moonsoo Kang	Chosun University, Korea (South)
Akimitsu Kanzaki	Shimane University, Japan
Eiji Kawai	National Institute of Information and Communications Technology, Japan
Wajahat Khan	University of Derby, United Kingdom (Great Britain)
Dong Seong Kim	Kumoh National Institute of Technology, Korea (South)
Dongkyun Kim	Kyungpook National University, Korea (South)
Haesik Kim	VTT Technical Research Centre of Finland, Finland
Hwangnam Kim	Korea University, Korea (South)
Hwasung Kim	Kwangwoon University, Korea (South)
Hyunbum Kim	Incheon National University, Korea (South)
Jeong Kim	Kyung Hee University, Korea (South)
JongWon Kim	GIST (Gwangju Institute of Science & Technology), Korea (South)
Joongheon Kim	Korea University, Korea (South)
Junsu Kim	Korea Polytechnic University, Korea (South)
Ki-Hyung Kim	Ajou University, Korea (South)
Ki-Il Kim	Chungnam National University, Korea (South)

## Committee

Kwangju Kim	ETRI, Korea (South)
Kyeong Soo Kim	Xi'an Jiaotong-Liverpool University, China
Su Min Kim	Korea Polytechnic University, Korea (South)
Sunwoo Kim	Hanyang University, Korea (South)
Taewoon Kim	Hallym University, Korea (South)
Yeongkwun Kim	Western Illinois University, USA
Teruaki Kitasuka	Hiroshima University, Japan
Nattapong Kitsuwat	The University of Electro-Communications, Japan
Haneul Ko	Korea University, Korea (South)
Ren-Song Ko	National Chung Cheng University, Taiwan
Young-Bae Ko	Ajou University, Korea (South)
Nobuyoshi Komuro	Chiba University, Japan
Eisuke Kudoh	Tohoku Institute of Technology, Japan
Sungoh Kwon	University of Ulsan, Korea (South)
Taesoo Kwon	Seoul National University of Science and Technology, Korea (South)
Edmund Lai	Auckland University of Technology, New Zealand
Kwok-Yan Lam	Nanyang Technological University, Singapore
Chaewoo Lee	Ajou University, Korea (South)
Gyu Myoung Lee	Liverpool John Moores University, United Kingdom (Great Britain)
Hyang-Won Lee	Konkuk University, Korea (South)
HyungJune Lee	Ewha Womans University, Korea (South)
Jack Y. B. Lee	The Chinese University of Hong Kong, Hong Kong
Jeong Woo Lee	Chung-Ang University, Korea (South)
Jihoon Lee	Sangmyung University, Korea (South)
Joohyun Lee	Hanyang University, Korea (South)
Jung Ryun Lee	Chung-Ang University, Korea (South)
Kyunghan Lee	Seoul National University, Korea (South)
SuKyoung Lee	Yonsei University, Korea (South)
Sungchang Lee	Hankuk Hangkong University, Korea (South)
Chi-Yu Li	National Yang Ming Chiao Tung University, Taiwan
Hyuk Lim	Gwangju Institute of Science and Technology, Korea (South)
Wansu Lim	Kumoh National Institute of Technology, Korea (South)
Yujin Lim	Sookmyung Women's University, Korea (South)

Chun-Cheng Lin	National Yang Ming Chiao Tung University, Taiwan
Bing-Hong Liu	National Kaohsiung University of Science and Technology, Taiwan
Feng Liu	Shanghai Maritime University, China
Huey-Ing Liu	Fu-Jen Catholic University, Taiwan
Miguel López-Benítez	University of Liverpool, United Kingdom (Great Britain)
Pascal Lorenz	University of Haute Alsace, France
Pavel Loskot	ZJU-UIUC Institute, China
Eng Lua	NEC Laboratories Singapore, Singapore
Hsi-Pin Ma	National Tsing Hua University, Taiwan
Md Mainul Islam Mamun	University of Missouri-Kansas City, USA
Ganapathy Mani	Purdue University, USA
Pietro Manzoni	Universitat Politècnica de València, Spain
Nobuhiko Miki	Kagawa University, Japan
Jeonghoon Mo	Yonsei University, Korea (South)
Bongkyo Moon	Dongguk University, Korea (South)
Ioannis Moscholios	University of Peloponnese, Greece
Amitava Mukherjee	Globsyn Business School, Kolkata, India
Osamu Muta	Kyushu University, Japan
Lilian Mutalemwa	Chosun University, Korea (South)
Woongsoo Na	Kongju National University, Korea (South)
Seung Yeob Nam	Yeungnam University, Korea (South)
Jad Nasreddine	Rafik Hariri University, Lebanon
S H Shah Newaz	Universiti Teknologi Brunei (UTB), Brunei Darussalam
Devarani Ningombam	Gandhi Institute of Technology and Management (GITAM) University, India
Wonjong Noh	Hallym University, Korea (South)
Toshiro Nunome	Nagoya Institute of Technology, Japan
JongTaek Oh	Hansung University, Korea (South)
Hiraku Okada	Nagoya University, Japan
Eiji Okamoto	Nagoya Institute of Technology, Japan
Kenko Ota	Nippon Institute of Technology, Japan
Carlos Palau	Universitat Politècnica Valencia, Spain
Hyungbae Park	University of Central Missouri, USA
Hyunggon Park	Ewha Womans University, Korea (South)

## Committee

Hyunhee Park	Myongji University, Korea (South)
Hyunho Park	ETRI, Korea (South)
Jaehyun Park	Pukyong National University, Korea (South)
Kyung-Joon Park	DGIST, Korea (South)
Al-Sakib Khan Pathan	United International University, Bangladesh
P k Paul	Raiganj University, India
Shuping Peng	Huawei Technologies, China
Jae-Young Pyun	Chosun University, Korea (South)
Tony Q. S. Quek	Singapore University of Technology and Design, Singapore
Hassaan Khaliq Qureshi	National University of Sciences and Technology, Pakistan
Ilkyeun Ra	University of Colorado Denver, USA
Redha Radaydeh	Texas A&M University-Commerce, USA
Nuno Rodrigues	Instituto Politécnico de Bragança, Portugal
Byeong-hee Roh	Ajou University, Korea (South)
Heejun Roh	Korea University, Korea (South)
Roberto Rojas-Cessa	New Jersey Institute of Technology, USA
Eun-Seok Ryu	Sungkyunkwan University (SKKU), Korea (South)
Ansa S	Bits Pilani K K Birla Goa Campus, India
Yatendra Sahu	Maulana Azad National Institute of Technology, Bhopal, India
Surasak Sanguanpong	Kasetsart University, Thailand
Chathura Sarathchandra	InterDigital Europe, United Kingdom (Great Britain)
Vrajesh Sharma	I. K. Gujral Punjab Technical University, India
Stavros Shiaeles	University of Portsmouth, United Kingdom (Great Britain)
Kuei-Ping Shih	Tamkang University, Taiwan
Choonsung Shin	Chonnam National University, Korea (South)
Dongwan Shin	New Mexico Tech, USA
Oh-Soon Shin	Soongsil University, Korea (South)
Seokjoo Shin	Chosun University, Korea (South)
Soo Young Shin	Kumoh National Institute of Technology, Korea (South)
Yoan Shin	Soongsil University, Korea (South)
Paulo Simões	University of Coimbra, Portugal
Seppo Sirkemaa	, Finland

Jaewoo So	Sogang University, Korea (South)
Jungmin So	Sogang University, Korea (South)
Insoo Sohn	Dongguk University, Korea (South)
Arun Kumar SP	Google, Ireland
Andrej Stefanov	IBU Skopje, Macedonia, the former Yugoslav Republic of
Young-Joo Suh	Pohang University of Science and Technology (POSTECH), Korea (South)
Norrozila Sulaiman	University Malaysia Pahang, Malaysia
Weiping Sun	Samsung Research, Korea (South)
Osamu Takyu	Shinshu University, Japan
Yuuichi Teranishi	NICT, Japan
Weitian Tong	Georgia Southern University, USA
Kazuya Tsukamoto	Kyushu Institute of Technology, Japan
Ihsan Ullah	Korea University of Technology and Education, Korea (South)
John Vardakas	Iquadrat Informatica, Spain
Athanasios V. Vasilakos	Lulea University of Technology, Sweden
Dario Vieira	EFREI, France
Naoki Wakamiya	Osaka University, Japan
Sheng-Wei Wang	Tamkang University, Taiwan
You-Chiun Wang	National Sun Yat-Sen University, Taiwan
Zheng Wang	Qingdao University, China
Charles H.-P. Wen	National Yang Ming Chiao Tung University, Taiwan
Carlos Becker Westphall	Federal University of Santa Catarina, Brazil
Michal Wodczak	Samsung Electronics, Poland
Longfei Wu	Fayetteville State University, USA
Yik-Chung Wu	The University of Hong Kong, Hong Kong
Qin Xin	University of the Faroe Islands, Faroe Islands
Li Xu	Fujian Normal University, China
Nariyoshi Yamai	Tokyo University of Agriculture and Technology, Japan
Kazuto Yano	ATR, Japan
Chia-Hung Yeh	National Sun Yat-Sen University, Taiwan
Joonhyuk Yoo	Daegu University, Korea (South)
Myungsik Yoo	Soongsil University, Korea (South)
Seokhoon Yoon	University of Ulsan, Korea (South)

## Committee

Joo-Sang Youn	Donggeui University, Korea (South)
Ji-Hoon Yun	Seoul National University of Science and Technology, Korea (South)
Rachid Zagrouba	College of Computer Science and Information Technology, Saudi Arabia
Sherali Zeadally	University of Kentucky, USA
Juzi Zhao	San Jose State University, USA
Natasa Zivic	University of Siegen, Germany



## Message from Organizing Chairs

Welcome to ICAIIIC 2022, the Fourth International Conference on Artificial Intelligence in Information and Communication, organized by the Korean Institute of Communications and Information Sciences (KICS) and technically cosponsored by IEEE Communications Society (ComSoC) and IEICE-CS. The ICAIIIC conference is pursuing a premier international forum to provide a great opportunity for exchanging the state-of-the-art research advances in artificial intelligence in information and communication and future technologies and expanding the research community.

We would like to welcome you to Jeju Island! Jeju Island is a popular vacation spot for Koreans and foreigners. Jeju Volcanic Island and Lava Tubes were inscribed on the World Heritage list. The island offers visitors a wide range of activities: hiking on Halla-san or Olle-Gil, catching sunrises and sunsets over the ocean, riding horses, touring all the locales from a favorite television K-drama, or just lying around on the sandy beaches. We have prepared an exciting program for you in ICAIIIC 2022. The safety and well-being of all conference participants is our priority. The emergence of the Omicron variant has forced us to change the venue from Guam, USA to Jeju Island, Korea. ICAIIIC has therefore taken the difficult decision to limit the number of participants in the venue. The conference will be a combination of online and offline events.

We would like to express our sincere gratitude to all committee members and referees who made tremendous contributions to this event. On behalf of the ICAIIIC steering committee and on behalf of all attendees, we thank the President of KICS AI Society, professor Dong Seog Han, for producing such an excellent program. Thanks to the tireless efforts of the Technical Program Committee Chairs, Professors Seokjoo Shin, Youn-Hee Han, Mikio Hasegawa, Benaoumeur Senouci, and all TPC members, ICAIIIC 2022 is packed with an excellent mix of technical sessions. We do hope that you will take this unique opportunity to attend the technical sessions, meet the authors, and foster greater collaboration with other researchers. The Organizing Committee put a lot of effort to make this conference greatly successful and enjoyable. In addition, if you have additional time, please do not miss the chance to tour around Jeju Island. As you walk around Jeju Island in February, you'll spot the small, white apricot blossoms on trees.

We look forward to seeing you in Jeju Island and online! We also wish your active participation in the future event.



**Yeong Min Jang**  
Kookmin Univ., Korea



**Takeo FUJII**  
The Univ. of Electro-Comms, Japan

## Message from TPC Chairs

It is our great pleasure to welcome all of you to Jeju Island, Korea, from Feb. 21 to 24, 2022, for the 4<sup>th</sup> International Conference on Artificial Intelligence in Information and Communication (ICAIC). ICAIC has addressed all aspects of artificial intelligence (AI), computing, networking, communications, and their convergence. This ICAIC 2022 will also be a successful conference covering a wide range of topics on various AI technologies and many forms of information and communication systems with AI.

This year we have received 202 paper submissions electronically from 19 countries in the world. Many of the papers were submitted from the Asia/Pacific region, and also an increasing number of submissions were made from Europe and North America. By a rigorous review process, all papers have been reviewed by at least three independent reviewers. After the reviews and discussions, we have selected 92 technical papers for presentation at the conference. The accepted technical papers were organized into 18 technical oral sessions, which will be held in 2 parallel tracks. The program is designed to provide a broad range of AI technologies, including AI for information and communications technology, AI for image processing and multimedia, AI for Data Analysis, Big Data and Cloud, AI for eHealth and Diagnosis, AI Applications for Information System, AI Foundation, AI for Control and Decision. We also invited world-class leading researchers for keynote speeches and tutorials, and they will give us wonderful talks.

As you may be aware, the World Health Organization officially declared the novel coronavirus COVID-19 a pandemic. This global health crisis is a unique challenge that has impacted many members of the ICAIC 2022. We would like to express our concern and support for all the members of the ICAIC 2022 community, our professional team, our families and all others affected by this outbreak.

Along with the contributions of prominent authors from around the world, we believe that this year's valuable and interesting program is possible by the commitment of the technical program members. We are indebted to all of the 233 TPC members for their active participation and precious time. We would also like to thank our sponsors, KICS, IEEE Communications Society, and IEICE Communications Society, for their kind support of this successful event. We express our deepest gratitude to the Organizing Committee Chairs, Prof. Yeong Min Jang, Prof. Takeo Fujii, and Prof. Dong Seog Han, for their continued support and guidance. We hope that all of you will enjoy the splendid program of ICAIC 2022 as well as the beautiful scenery and charm of Jeju.



**Seokjoo Shin**  
Chosun Univ.,  
Korea



**Youn-Hee Han**  
Korea University of  
Technology and Education,  
Korea



**Mikio Hasegawa**  
Tokyo Univ. of Science,  
Japan



**Benaoumeur Senouci**  
North Dakota State University,  
USA

## Program Matrix for ICAIIIC 2022

February 21, 2022 (Monday)		
Room	Room A (Zoom A)	Room B (Zoom B)
10:00~12:00	ICAIIIC Organizing Committee Meeting	
14:30~15:00	Registration	
15:00~16:00	Tutorial Session I - Prof. Katsuya Suto (University of Electro-Communications) Title: Deep Learning and Its Applications to Radio Map Construction Chair: Prof. Mikio Hasegawa (Tokyo University of Science)	
16:00~16:30	Break	
16:30~17:30	Session 1A: Information and Communications Technology I Chair: Prof. Wooyeol Choi (Chosun University)	Session 1B: AI for Image Processing and Multimedia I Chair: Prof. Jung Hoon Lee (Hankuk University of Foreign Studies)

February 22, 2022 (Tuesday)		
Room	Room A (Zoom A)	Room B (Zoom B)
09:00~09:30	Registration	
09:30~10:50	Session 2A: Information and Communications Technology II Chair: Prof. Senouci Ben (North Dakota State University)	Session 2B: AI for Image Processing and Multimedia II Chair: Prof. Dong Seog Han (Kyungpook National University)
10:50~11:00	Break	
11:00~11:10	Opening and Plenary Session Chair: Prof. Sang-Chul Kim (Kookmin University, Korea)	
	Opening Session	Opening (Prof. Yeong Min Jang, General Chair)
		Welcome Speech I (Prof. Yoan Shin, President of KICS)
		Welcome Speech II (Prof. Lingyang Song, Chair of IEEE ComSoc Cognitive Network TC)
11:10~11:40	Keynote Speech I - Prof. Jinchang Ren (Robert Gordon University) Title: AI Enabled Smart Data for Smart Cities	
11:40~12:10	Keynote Speech II - Dr. Seong-Ju Kang (Vice Chair of Korea Intelligent IoT Association (KIoT)) Title: AI Policy in Korea and Its Implication for Responding COVID-19 Pandemic	
12:10~14:00	Break	
14:00~15:00	Tutorial Session II - Prof. Yexiang Xue (Purdue University) Title: Knowledge Embeddings to Attack Multi-stage Inference Problems in Reasoning, Learning, and Decision Making Chair: Prof. Wansu Lim (Kumoh National Institute of Technology)	
15:00~15:20	Break	
15:20~16:40	Session 3A: Information and Communications Technology III Chair: Dr. Hui Han (Fraunhofer Institute)	Session 3B: AI for Image Processing and Multimedia III Chair: Dr. Eric Xue (University of Toronto)

## Program Matrix for ICAIC 2022

February 23, 2022 (Wednesday)		
Room	Room A (Zoom A)	Room B (Zoom B)
09:00~09:30	Registration	
09:30~10:50	Session 4A: Information and Communications Technology IV Chair: Prof. Dongkyun Kim (Kyungpook National University)	Session 4B: AI for eHealth and Medical Diagnosis I Chair: Prof. Micheal Tee (University of the Philippines Manila)
10:50~11:10	Break	
11:10~12:30	Session 5A: AI for Image Processing and Multimedia IV Chair: Prof. Hong Qin (University of Tennessee at Chattanooga)	Session 5B: AI for eHealth and Medical Diagnosis II Chair: Prof. Min Young Kim (Kyungpook National University)
12:30~14:00	Break	
14:00~14:50	Tutorial Session III - Prof. Dong Seog Han, (Kyungpook National University) Title: Facial Emotion Recognition with Deep Learning Chair: Prof. Youn-Hee Han (Korea University of Technology and Education)	
14:50~16:10	Session 6A: AI Foundation Chair: Dr. Deepesh Agarwal (Kansas State University)	Session 6B: AI for Control and Decision I Chair: Prof. Eunkyung Kim (Hanbat National University)
16:10~16:30	Break	
16:30~17:50	Session 7A: AI Applications for Information Systems I Chair: Dr. Ali Rizwan (Qatar University)	Session 7B: AI for Control and Decision II Chair: Dr. Adhitya Bantwal Bhandarkar (University of New Mexico)

February 24, 2022 (Thursday)		
Room	Room A (Zoom A)	Room B (Zoom B)
09:00~09:30	Registration	
09:30~10:50	Session 8A: AI Applications for Information Systems II Chair: Prof. Anteneh Girma (University of the District of Columbia)	Session 8B: AI for eHealth and Medical Diagnosis III Chair: Prof. Pyungsoo Kim (Korea Polytechnic University)
10:50~11:10	Break	
11:10~12:30	Session 9A: AI Applications for Information Systems III Chair: Prof. Jeong Gon Kim (Korea Polytechnic University)	Session 9B: AI for Data Analysis, Big Data and Cloud Chair: Prof. Ihsan Ullah (Korea University of Technology and Education)
12:30~12:35	Closing Remark (Prof. Dong Seog Han, Kyungpook National Univ., President of KICS AI Society)	

## Keynote Speech

February 22, 2022 (Tuesday) 11:10 ~ 11:40

**Keynote Speech I** (Prof. Jinchang Ren / Robert Gordon University)

**AI Enabled Smart Data for Smart Cities**

### Abstract

Smart cities feature artificial intelligence enabled automation and decision making, which rely heavily on big data analytics based smart data applications. In this talk, applications of smart data for smart cities will be focused, including in particular big data analytics of crime and traffic data, which are from our most recent work in these areas. Useful algorithms and tools will be introduced, especially for data visualisation and demonstration, where valuable findings and conclusions in terms of the trend development, periodic terms and holiday events, will be presented to facilitate the advancement of smart cities.



### Biography

Jinchang Ren (M'05, SM'17) received his B. E. degree in Computer Software, M.Eng. in Image Processing, D.Eng. in Computer Vision, all from Northwestern Polytechnical University, Xi'an, China. He was also awarded a Ph.D. in Electronic Imaging and Media Communication from the University of Bradford, Bradford, U.K. Currently he is a chair Professor of Computing Sciences, National Subsea Centre, School of Computing, Robert Gordon University, Aberdeen, UK. His research interests focus mainly on hyperspectral imaging, image processing, computer vision, big data analytics and machine learning. He has published 300+ peer reviewed journal/conferences articles, and acts as an Associate Editor for several international journals including IEEE Trans. Geoscience and Remote Sensing and J. of the Franklin Institute et al.

## Keynote Speech

February 22, 2022 (Tuesday), 11:40 ~ 12:10

**Keynote Speech II** (Dr. Seong-Ju Kang / Vice Chair of Korea Intelligent IoT Association (KIoT))  
**AI Policy in Korea and Its Implication for Responding COVID-19 Pandemic**

### Abstract

Artificial intelligence(AI) is domination in various areas such as financing, commerce, manufacturing, and public management. Since the event of AlphaGo in 2016 many governments have developed a set of policy to promote AI in those areas including Korea.

They formed an advisory group to develop policies, including academia and business. In December of 2019 they announced an AI strategy based on recommendations from the advisory group. According the strategy, there are three pillars and nine directions. First pillar is to establish the ecology of AI such as 5G infrastructure, institution building and start-ups. The second is to utilize the potential of AI via human resource development, application to various sectors and AI-based digital government. The third is to realize human-centered AI by securing jobs and setting up new ethics. In addition these strategies could also contribute to respond COVID-19 pandemic.



### Biography

Seong Ju Kang received the M.A. degree in public management from Syracuse University, and studied at Pennsylvania State University for doctoral. He spent more than three decades in IT policy arena such as AI, 5G, IoT, cyber security and digital government in Korean government and OECD. His research interests include digital transformation, metaverse, and blockchain. He is actively involving in academic activities as well such as green network technology, mobile computing, and deep learning.

## Tutorial

February 21, 2022 (Monday), 15:00 ~ 16:00

**Tutorial Session I** (Prof. Katsuya Suto, University of Electro-Communications)  
**Deep Learning and Its Applications to Radio Map Construction**

### Abstract

Radio map plays a key role in decision-making in 6G systems, i.e., resource management for cell-free wireless networks, spatial spectrum sharing, intelligent reflecting surface (IRS). However, it remains an open challenge. Deep learning (especially image-driven deep learning) has been developing as a promising solution to express the complex radio propagation features in the urban area using feature extraction from 3D maps of cities. The approach learns the correlation between the building features and propagation features to recognize the reflection and diffraction by the buildings. By use of rapid advancement of GPU, it achieves high estimation accuracy with low computation time.

The main objective of this tutorial is to provide a fundamental background of deep learning and then show how to address practical challenges in radio map construction. In particular, we first give a tutorial of deep learning used in radio map construction to provide comprehensive knowledge to the audiences. We then give the current research trend together with implementation details to have a better understanding. Finally, we introduce our proposed methods for path loss modeling, spatial interpolation, and spatial extrapolation.



### Biography

Katsuya Suto received the B.Sc. degree in computer engineering from Iwate University, Morioka, Japan, in 2011, and the M.Sc. and Ph.D. degrees in information science from Tohoku University, Sendai, Japan, in 2013 and 2016, respectively. He has worked as a Postdoctoral Fellow for Research Abroad, Japan Society for the Promotion of Science, in the Broadband Communications Research Lab., University of Waterloo, ON, Canada, from 2016 to 2018. He is currently an Assistant Professor with the Graduate School of Informatics and Engineering, the University of Electro-Communications, Tokyo, Japan. His research interests include mobile edge computing, cognitive radio, green wireless networking, and deep learning. He received the Best Paper Award at the IEEE VTC2013-spring, the IEEE/CIC ICC2015, the IEEE ICC2016, and the IEEE Transactions on Computers in 2018.

## Tutorial

February 22, 2022 (Tuesday), 14:00 ~ 15:00

**Tutorial Session II** (Prof. Yexiang Xue, Purdue University)

### Knowledge Embeddings to Attack Multi-stage Inference Problems in Reasoning, Learning, and Decision Making

#### Abstract

Problems at the intersection of reasoning, optimization, and learning often involve multi-stage inference and are therefore highly intractable. I will introduce a novel computational framework, based on embeddings, to tackle multi-stage inference problems. As a first example, I present a novel way to encode the reward allocation problem for a two-stage organizer-agent game-theoretic framework as a single-stage optimization problem. The encoding embeds an approximation of the agents' decision-making process into the organizer's problem. We apply this methodology to eBird, a well-established citizen-science program for collecting bird observations, as a game called Avicaching. Our AI-based reward allocation was shown highly effective, surpassing the expectations of the eBird organizers and bird conservation experts. As a second example, I present a novel constant approximation algorithm to solve the so-called Marginal Maximum-A-Posteriori (MMAP) problem for finding the optimal policy maximizing the expectation of a stochastic objective. To tackle this problem, I propose the embedding of its intractable counting subproblems as queries to NP-oracles subject to additional XOR constraints. As a result, the entire problem is encoded as a single NP-equivalent optimization. The approach outperforms state-of-the-art solvers based on variational inference as well as MCMC sampling on probabilistic inference benchmarks, deep learning applications, as well as on a novel decision-making application in network design for wildlife conservation. Lastly, I will talk about how the embeddings of phase-field modeling in an end-to-end neural network allow us to learn partial differential equations governing the dynamics of nanostructures in metallic materials under extreme heat and irradiation conditions.



#### Biography

Dr. Yexiang Xue is an assistant professor at the Department of Computer Science at Purdue University, USA. The goal of Dr. Xue's research is to bridge large-scale constraint-based reasoning and optimization with state-of-the-art machine learning techniques to enable intelligent agents to make optimal decisions in high-dimensional and uncertain real-world applications. More specifically, Dr. Xue's research focuses on scalable and accurate probabilistic reasoning techniques, statistical modeling of data, and robust decision-making under uncertainty. Dr.

Xue's work is motivated by key problems across multiple scientific domains, ranging from artificial intelligence, machine learning, renewable energy, materials science, crowdsourcing, citizen science, urban computing, ecology, to behavioral econometrics. Dr. Xue focuses on developing cross-cutting computational methods, with an emphasis on the areas of computational sustainability and scientific discovery. Dr. Xue received several NSF grants, Purdue's seed of success award, Cornell's Ph.D. dissertation award, and the IAAI Innovative application award. He published over 45 papers at top-tier CS conferences, and also journal articles including in Science, Nature Communications, the communications of ACM, Materials Research Society Communications, and the Artificial Intelligence magazine.



## Tutorial

February 23, 2022 (Wednesday), 14:00 ~ 14:50

**Tutorial Session III** (Prof. Dong Seog Han, Kyungpook National University)

### Facial Emotion Recognition with Deep Learning

#### Abstract

Facial emotion recognition (FER) is vital for interactive robots detecting users' feelings. The solid performance of the FER requires a well-designed neural network and a reliable FER dataset. Managing the FER dataset is highly effective in having a solid performance than redesigning the neural network. These days, many FER researchers are more focused on designing a deep learning model without thoroughly inspecting the FER dataset samples. In addition, the FER without improper pre-processing of the FER dataset could cause degrading the deep learning model's performance even with a well-designed neural network. Some FER datasets contain irrelevant facial images or unnecessary features, confusing a deep neural network's training. In this tutorial, we demonstrate how properly pre-process the FER dataset to enhance the overall quality of the FER dataset and improve the performance of FER's training.



#### Biography

Dong Seog Han received the B.S. degree in electronic engineering from Kyungpook National University (KNU), Daegu, Korea, in 1987, and the M.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 1989 and 1993, respectively. From 1987 to 1996, he was with Samsung Electronics Company Ltd., where he developed the receiver chipset for HDTV.

Since 1996, he has been with the School of Electronics Engineering, KNU, as a faculty and is currently a full Professor. He was a courtesy Associate Professor with the Department of Electrical and Computer Engineering, University of Florida, in 2004. He was the Director of the Center of Digital TV and Broadcasting, Institute for Information Technology Advancement (IITP), from 2006 to 2008. He is currently directing the Center for ICT & Autonomous Convergence, KNU, since 2011. His main research interests include intelligent signal processing and autonomous vehicles.

## Oral Sessions

### Oral Session 1A: Information and Communications Technology I / February 21, 2022 (Monday)

Chair: Prof. Wooyeol Choi (Chosun University)

16:30-17:30, Room A/Zoom A

- 1A-1 **Classification and Discretization of Shadowing Toward Low Storage Radio Map**  
*Keita Katagiri (The University of Electro-Communication, Japan); Takeo Fujii (The University of Electro-Communications, Japan)*
- 1A-2 **Countering DNS Vulnerability to Attacks Using Ensemble Learning**  
*Love Allen Ahakonye and Cosmas Ifeanyi Nwakanma (Kumoh National Institute of Technology, South Korea); Simeon Ajakwe (Kumoh National Institute of Technology, Gumi, South Korea); Dong Seong Kim and Jae Min Lee (Kumoh National Institute of Technology, South Korea)*
- 1A-3 **NetMD-Network Traffic Analysis and Malware Detection**  
*Sampath Kumar Katherasala and Sri Manvith Vaddeboyina (Tata Consultancy Services, India); Ajay Therala (TATA Consultancy Services, India)*
- 1A-4 **Defect Information Synthesis via Latent Mapping Adversarial Networks**  
*Seunghwan Song and Jun-Geol Baek (Korea University, South Korea)*
- 1A-5 **FFDNet Based Channel Estimation for Multiuser Massive MIMO System with One-Bit ADCs**  
*Md. Habibur Rahman, Md. Shahjalal and Md. Osman Ali (Kookmin University, Korea); ByungDeok Chung (ENS. Co. Ltd, Korea); Yeong Min Jang (Kookmin University, South Korea)*

### Oral Session 1B: AI for Image Processing and Multimedia I / February 21, 2022 (Monday)

Chair: Prof. Jung Hoon Lee (Hankuk University of Foreign Studies)

16:30-17:30 Room B/Zoom B

- 1B-1 **iVoiding: A Thermal-Image based Artificial Intelligence Dynamic Voiding Detection System**  
*Yu-Chen Chen (Kaohsiung Medical University, Kaohsiung, Taiwan); Jian-Ping Su (Southern Taiwan University of Science and Technology, Taiwan); Cheng Han Tsai (Southern Taiwan University of Science and Technology, Tainan, Taiwan); Ming-Che Chen and Wan-Jung Chang (Southern Taiwan University of Science and Technology, Taiwan); Wen-Jeng Wu (Kaohsiung Medical University, Kaohsiung, Taiwan)*
- 1B-2 **Determining Jigsaw Puzzle State from an Image based on Deep Learning**  
*Ijaz Ahmad, Suk-seung Hwang and Seokjoo Shin (Chosun University, South Korea)*
- 1B-3 **Growth Estimation Sensor Network System for Aquaponics using Multiple Types of Depth Cameras**  
*Ryota Murakami and Hiroshi Yamamoto (Ritsumeikan University, Japan)*
- 1B-4 **Image Synthesis with Single-type Patterns for Mixed-type Pattern Recognition on Wafer Bin Maps**  
*Yunseon Byun (Korea University, Republic of Korea, South Korea); Jun-Geol Baek (Korea University, South Korea)*
- 1B-5 **Evaluating Opcodes for Detection of Obfuscated Android Malware**  
*Saneeha Khalid (Bahria University Islamabad Pakistan, Pakistan); Faisal Bashir Hussain (Bahria University, Islamabad, Pakistan)*

## Oral Sessions

### Oral Session 2A: Information and Communications Technology II / February 22, 2022 (Tuesday)

Chair: Prof. Senouci Ben (North Dakota State University)

09:30-10:50 Room A/Zoom A

- 2A-1 **Procedural Generation of Game Levels and Maps: A Review**  
*Tianhan Gao (Northeastern University of China, China); Jin Zhang and Qingwei Mi (Northeastern University, China)*
- 2A-2 **Similarity-based Local Feature Extraction for Wafer Bin Map Pattern Recognition**  
*Jieun Kim and Jun-Geol Baek (Korea University, South Korea)*
- 2A-3 **Body Segmentation Using Multi-task Learning**  
*Julijan Jug and Ajda Lampe (University of Ljubljana, Faculty of Computer and Information Science, Slovenia); Vitomir Štruc (Faculty of Electrical Engineering, University of Ljubljana, Slovenia); Peter Peer (University of Ljubljana, Faculty of Computer and Information Science, Slovenia)*
- 2A-4 **Aerial Supervision of Drones and Birds using Convolutional Neural Networks**  
*Vivian Ukamaka Ihekoronye (Kumoh National Institute of Technology, South Korea); Simeon Ajakwe (Kumoh National Institute of Technology, Gumi, South Korea); Dong Seong Kim and Jae Min Lee (Kumoh National Institute of Technology, South Korea)*
- 2A-5 **Performance Analysis of UAV-based Array Antenna Arrangement for Target Detection**  
*Ji-Hyeon Kim, Soon-Young Kwon and Hyoung-Nam Kim (Pusan National University, South Korea)*

### Oral Session 2B: AI for Image Processing and Multimedia II / February 22, 2022 (Tuesday)

Chair: Prof. Dong Seog Han (Kyungpook National University)

09:30-10:50 Room B/Zoom B

- 2B-1 **Image Prediction for Lane Following Assist using Convolutional Neural Network-based U-Net**  
*Byung Chan Choi (LIG Nex1, South Korea); Jaerock Kwon (University of Michigan - Dearborn, USA); Haewoon Nam (Hanyang University, South Korea)*
- 2B-2 **Forward and Backward Warping for Optical Flow-Based Frame Interpolation**  
*Joi Shimizu, Heming Sun and Jiro Katto (Waseda University, Japan)*
- 2B-3 **Performance Improvement Method of the Video Visual Relation Detection with Multi-modal Feature Fusion**  
*Kwangju Kim and Pyong-Kun Kim (ETRI, South Korea); Kil-Taek Lim (Electronics and Telecommunications Research Institute, South Korea); Jong Taek Lee (ETRI, South Korea)*
- 2B-4 **A high-speed driver behavior detection deep learning system using the amount of change in contrast between frames**  
*Min Woo Yoo and Dong Seog Han (Kyungpook National University, South Korea)*
- 2B-5 **Intelligent Receiver for Optical Camera Communication**  
*Ida Bagus Krishna Yoga Utama (Kookmin University, Korea); Md. Habibur Rahman (Kookmin University, Korea); ByungDeok Chung (ENS. Co. Ltd, Korea); Yeong Min Jang (Kookmin University, Korea)*

## Oral Sessions

### Oral Session 3A: Information and Communications Technology III / February 22, 2022 (Tuesday)

Chair: Dr. Hui Han (Fraunhofer Institute)

15:20-16:40 Room A/Zoom A

- 3A-1 Interference analysis study for coexistence between C-V2X and Wi-Fi 6E in the 6GHz band  
*YoungWoon Kim (Soongsil University, South Korea)*
- 3A-2 Neural Architecture Search for Real-Time Driver Behavior Recognition  
*Jaeho Seong and Dong Seog Han (Kyungpook National University, South Korea)*
- 3A-3 Smart Anomaly Detection: Deep Learning modeling Approach and System Utilization Analysis  
*Ben Senouci (North Dakota State University (NDSU), USA); Mourad Bouache (Intel USA, USA)*
- 3A-4 An Evaluation Framework for Machine Learning Methods in Detection of DoS and DDoS Intrusion  
*Temechu G Zewdie and Anteneh Girma (University of the District of Columbia, USA)*
- 3A-5 A study on the application of mission-based cybersecurity testing and evaluation of weapon systems  
*Dongkyoo Shin and Ikjae Kim (Sejong University, South Korea)*

### Oral Session 3B: AI for Image Processing and Multimedia III / February 22, 2022 (Tuesday)

Chair: Dr. Eric Xue (University of Toronto)

15:20-16:40 Room B/Zoom B

- 3B-1 Grey Wolf Optimizer-Based Automatic Focusing for High Magnification Systems  
*Islam Helmy and Wooyeol Choi (Chosun University, South Korea)*
- 3B-2 Research and examination on implementation of super-resolution models using deep learning with INT8 precision  
*Shota Hirose, Naoki Wada, Jiro Katto and Heming Sun (Waseda University, Japan)*
- 3B-3 Mitigating Overflow of Object Detection Tasks Based on Masking Semantic Difference Region of Vision Snapshot for High Efficiency  
*Heuijee Yun (Kyungpook National University, South Korea); Daejin Park (Kyungpook National University (KNU), South Korea)*
- 3B-4 Calibration-Net: LiDAR and Camera Auto-Calibration using Cost Volume and Convolutional Neural Network  
*An Duy Nguyen and Myungsik Yoo (Soongsil University, South Korea)*
- 3B-5 Granular Analysis of Pretrained Object Detectors  
*Eric Xue (University of Toronto, Canada); Tae Soo Kim (Johns Hopkins University, USA)*

## Oral Sessions

### Oral Session 4A: Information and Communications Technology IV / February 23, 2022 (Wednesday)

Chair: Prof. Dongkyun Kim (Kyungpook National University)

09:30-10:50 Room A/Zoom A

- 4A-1 Irregular Repetition Slotted ALOHA Scheme with Multi-Packet Reception in Packet Erasure Channel  
*Chundie Feng (Chongqing University, China); Xuhong Chen (China Development Bank, China); Zhengchuan Chen, Zhong Tian and Yunjian Jia (Chongqing University, China); Min Wang (Chongqing University of Posts and Telecommunications, China)*
- 4A-2 Two-Policy Cooperative Transfer for Alleviation of Sim-to-Real Gap  
*Liangdong Wu (Institute of Automation, Chinese Academy of Sciences, China)*
- 4A-3 Graph Neural Network-based Clustering Enhancement in VANET for Cooperative Driving  
*Hang Hu (City College of New York, USA); Myung Lee (City University of New York, City College, USA)*
- 4A-4 Machine Learning-Based Power Loading for Massive Parallel Gaussian Channels  
*Min Jeong Kang and Jung Hoon Lee (Hankuk University of Foreign Studies, South Korea)*
- 4A-5 Enhanced Semi-persistent scheduling (e-SPS) for Aperiodic Traffic in NR-V2X  
*Malik Muhammad Saad, Muhammad Ashar Tariq, Md. Mahmudul Islam, Muhammad Toaha Raza Khan, Junho Seo and Dongkyun Kim (Kyungpook National University, South Korea)*
- 4A-6 Target Detection using U-Net for a DTV-based Passive Bistatic Radar System  
*Ji-Hun Park, Do-Hyun Park and Hyoung-Nam Kim (Pusan National University, South Korea)*

### Oral Session 4B: AI for eHealth and Medical Diagnosis I / February 23, 2022 (Wednesday)

Chair: Prof. Micheal Tee (University of the Philippines Manila)

09:30-10:50 Room B/Zoom B

- 4B-1 Privacy-preserving collaborative machine learning in biomedical applications  
*Wonsuk Kim and Junhee Seok (Korea University, South Korea)*
- 4B-2 Computer Code Representation through Natural Language Processing for fMRI Data Analysis  
*Jaeyoon Kim (Korea University, South Korea); Una-May O'Reilly (MIT, USA); Junhee Seok (Korea University, South Korea)*
- 4B-3 A Machine Learning Approach in Evaluating Symptom Screening in Predicting COVID-19  
*Geoffrey A. Solano (University of the Philippines Manila, Philippines); Marc Jermaine Pontiveros (University of the Philippines Manila & University of the Philippines Diliman, Philippines); Michael L. Tee (University of the Philippines Manila, Philippines)*
- 4B-4 A Study on the Clinical Effectiveness of Deep Learning CAD Technology  
*Ju-Hyuck Han, Hyun-Woo Oh and Woong-Sik Kim (Konyang University, South Korea)*
- 4B-5 Fake Data Generation for Medical Image Augmentation using GANs  
*Donghwan Kim, Jaehan Joo and Suk Chan Kim (Pusan National University, South Korea)*

## Oral Sessions

### Oral Session 5A: AI for Image Processing and Multimedia IV / February 23, 2022 (Wednesday)

Chair: Prof. Hong Qin (University of Tennessee at Chattanooga)

11:10-12:30 Room A/Zoom A

- 5A-1 **Vision Anomaly Detection Using Self-Gated Rectified Linear Unit**  
*Israt Jahan, Md. Osman Ali and Md. Habibur Rahman (Kookmin University, Korea); ByungDeok Chung (ENS. Co. Ltd, Korea); Yeong Min Jang (Kookmin University, Korea)*
- 5A-2 **A Comparison of YOLO and Mask-RCNN for Detecting Cells from Microfluidic Images**  
*Mehran Ghafari (University of Tennessee Chattanooga, USA); Daniel Mailman, Parisa Hatami, Trevor Peyton and Li Yang (University of Tennessee at Chattanooga, USA); Weiwei Dang (Baylor Collage of Medicine, USA); Hong Qin (University of Tennessee at Chattanooga, USA)*
- 5A-3 **Multiview Attention for 3D Object Detection in Lidar Point Cloud**  
*Kevin T. Wijaya, Donghee Paek and Seung-Hyun Kong (Korea Advanced Institute of Science and Technology, South Korea)*
- 5A-4 **Multi-scale synergy approach for real-time semantic segmentation**  
*Min Young Kim and Quyen Van Toan (Kyungpook National University, South Korea)*
- 5A-5 **CIAFill: Lightweight and Fast Image Inpainting with Channel Independent Attention**  
*Chung-Il Kim (Korea Electronics Technology Institute, South Korea); Saim Shin (KETI, South Korea); Han-Mu Park (Korea Electronics Technology Institute, South Korea)*

### Oral Session 5B: AI for eHealth and Medical Diagnosis II / February 23, 2022 (Wednesday)

Chair: Prof. Min Young Kim (Kyungpook National University)

11:10-12:30 Room B/Zoom B

- 5B-1 **Anomaly Detection for Alzheimer's Disease in Brain MRIs via Unsupervised Generative Adversarial Learning**  
*Geoffrey A. Solano and Sun Arthur A. Ojeda (University of the Philippines Manila, Philippines)*
- 5B-2 **Heart Disease Prediction Using Adaptive Infinite Feature Selection and Deep Neural Networks**  
*Sudipta Modak, Esam Abdel-Raheem and Luis Rueda (University of Windsor, Canada)*
- 5B-3 **A federated binarized neural network model for constrained devices in IoT healthcare services**  
*Hyeontaek Oh, Jongmin Yu, Nakyoung Kim, Dongyeong Kim and Jangwon Lee (KAIST, South Korea); Jinhong Yang (INJE University & Korea Advanced Institute of Science Technology, South Korea)*
- 5B-4 **Hierarchical User Status Classification for Imbalanced Biometric Data**  
*Nakyoung Kim (KAIST, South Korea); Hyunseo Park (Korea Advanced Institute of Science and Technology (KAIST), South Korea); Gyeong Ho Lee, Jaeseob Han, Hyeontaek Oh and Jun Kyun Choi (KAIST, South Korea)*
- 5B-5 **Increasing Accuracy of Hand Gesture Recognition using Convolutional Neural Network**  
*Gyutae Park and Chandrasegar Vasantha Kumar (Gyeongsang National University, South Korea); JoongGun Park (JD Co., Ltd, South Korea); Jinhwan Koh (Gyeongsang National University, South Korea)*

## Oral Sessions

### Oral Session 6A: AI Foundation / February 23, 2022 (Wednesday)

Chair: Dr. Deepesh Agarwal (Kansas State University)

14:50-16:10 Room A/Zoom A

- 6A-1 **Impacts of Behavioral Biases on Active Learning Strategies**  
*Deepesh Agarwal (Kansas State University, USA); Obdulia Covarrubias and Stefan Bossmann (The University of Kansas Medical Center, USA); Bala Natarajan (Kansas State University, USA)*
- 6A-2 **Effect of the Period of the Fourier Series Approximation for Binarized Neural Network**  
*Seon-Yong Lee, Hee-Youl Kwak and Jong-Seon No (Seoul National University, South Korea)*
- 6A-3 **CMCL: Clustering-based Memory Management for Continual Learning**  
*Jiae Yoon (GIST, South Korea); Hyuk Lim (Gwangju Institute of Science and Technology, South Korea)*
- 6A-4 **TinyML: A Systematic Review and Synthesis of Existing Research**  
*Hui Han (Fraunhofer Institute for Experimental Software Engineering IESE, Germany); Julien Siebert (Fraunhofer Institut for Experimental Software Engineering IESE, Germany)*
- 6A-5 **A Survey of Procedural Content Generation of Natural Objects in Games**  
*Tianhan Gao (Northeastern University of China, China); Jiahui Zhu (Northeastern University, China)*

### Oral Session 6B: AI for Control and Decision I / February 23, 2022 (Wednesday)

Chair: Prof. Eunkyung Kim (Hanbat National University)

14:50-16:10 Room B/Zoom B

- 6B-1 **Reinforcement Learning for Neural Collaborative Filtering**  
*Alexandros I Metsai (My Company Projects, Greece); Konstantinos Karamitsios and Konstantinos Kotrotsios (My Company Projects O. E., Greece); Periklis Chatzimisios (International Hellenic University (Greece), Greece & University of New Mexico (USA), USA); George Stalidis and Kostas Goulianas (International Hellenic University, Greece)*
- 6B-2 **A Survey of Markov Model in Reinforcement Learning**  
*Tianhan Gao (Northeastern University of China, China); Baicheng Chen (Northeastern University & Arlinton University, China); Qingwei Mi (Northeastern University, China)*
- 6B-3 **Fairness Enhancement of TCP Congestion Control Using Reinforcement Learning**  
*Sang-Jin Seo and You-Ze Cho (Kyungpook National University, South Korea)*
- 6B-4 **Merging Reinforcement Learning and Inverse Reinforcement Learning via Auxiliary Reward System**  
*Wadhah Zeyad Tareq and M. Fatih Amasyalı (Yildiz Technical University, Turkey)*
- 6B-5 **Pothole Detection Using Optical Camera Communication**  
*Md. Osman Ali (Kookmin University, Korea); Israt Jahan (Kookmin University, Korea); Raihan Bin Mofidull (Kookmin University, Korea); ByungDeok Chung (ENS. Co. Ltd, Korea); Yeong Min Jang (Kookmin University, Korea)*

## Oral Sessions

### Oral Session 7A: AI Applications for Information Systems I / February 23, 2022 (Wednesday)

Chair: Dr. Ali Rizwan (Qatar University)

16:30-17:50 Room A/Zoom A

- 7A-1 **Sensor Network System for Condition Detection of Harmful Animals by Step-by-step Interlocking of Various Sensors**  
*Keigo Uchiyama and Hiroshi Yamamoto (Ritsumeikan University, Japan); Eiji Utsunomiya (KDDI Research, Japan); Kiyohito Yoshihara (KDDI Research Inc., Japan)*
- 7A-2 **WiFi Positioning by Optimal k-NN in 3GPP Indoor Office Environment**  
*Sung Hyun Oh and Jeong Gon Kim (Korea Polytechnic University, South Korea)*
- 7A-3 **A Study on the improvement of chinese automatic speech recognition accuracy using a lexicon**  
*Minjeong Gu (University of Science and Technology & Electronics and Telecommunications Research Institute, South Korea); Shingak Kang (Electronics and Telecommunication Research Institute, South Korea)*
- 7A-4 **Addressing Data Sparsity with GANs for Multi-fault Diagnosing in Emerging Cellular Networks**  
*Ali Rizwan (University of Glasgow, United Kingdom (Great Britain)); Adnan Abu-Dayya and Fethi Filali (QMIC, Qatar); Ali Imran (University of Oklahoma, USA)*
- 7A-5 **Edge-Computing based Secure E-learning Platforms**  
*Sameer Ahmad Bhat (Gulf University for Science and Technology, Kuwait); Muneer Dar (National Institute of Electronics & Information Technology, Srinagar, India); Saadiya Shah (National Institute of Electronics and Information Technology, Kuwait)*
- 7A-6 **Efficient classification of human activity using PCA and deep learning LSTM with WiFi CSI**  
*Sang-Chul Kim and Yong-Hwan Kim (Kookmin University, South Korea)*

### Oral Session 7B: AI for Control and Decision II / February 23, 2022 (Wednesday)

Chair: Dr. Adhitya Bantwal Bhandarkar (University of New Mexico)

16:30-17:50 Room B/Zoom B

- 7B-1 **MARL-based Optimal Route Control in Multi-AGV Warehouses**  
*Ho-Bin Choi, Ju-Bong Kim, Chang-Hun Ji, Ihsan Ullah and Youn-Hee Han (Korea University of Technology and Education, South Korea); Se Won Oh (ETRI, South Korea); Kwi-Hoon Kim (Korea National University of Education, South Korea); Cheol Sig Pyo (ETRI, South Korea)*
- 7B-2 **DDPG-Edge-Cloud: A Deep-Deterministic Policy Gradient based Multi-Resource Allocation in Edge-Cloud System**  
*Arslan Qadeer (City College of New York, CUNY, USA); Myung Lee (City University of New York, City College, USA)*
- 7B-3 **A Study on Update Frequency of Q-Learning-based Transmission Datarate Adaptation using Redundant Check Information for IEEE 802.11ax Wireless LAN**  
*Kazuto Yano, Kenta Suzuki and Babatunde Ojetunde (Advanced Telecommunications Research Institute International (ATR), Japan); Koji Yamamoto (Kyoto University, Japan)*



## Oral Sessions

- 7B-4 User Coverage Maximization for a UAV-mounted Base Station Using Reinforcement Learning and Greedy Methods  
*Adhitya Bantwal Bhandarkar (University of New Mexico, USA); Sudharman K Jayaweera (University of New Mexico & Bluecom Systems, USA); Steven Lane (Air Force Research Laboratory, USA)*
- 7B-5 SVR-based Blind Equalization on HF Channels with a Doppler Spread  
*Soon-Young Kwon, Ji-Hyeon Kim and Hyoung-Nam Kim (Pusan National University, South Korea)*

### Oral Session 8A: AI Applications for Information Systems II / February 24, 2022 (Thursday)

Chair: Prof. Anteneh Girma (University of the District of Columbia)

09:30-10:50 Room A/Zoom A

- 8A-1 Resolving Camera Position for a Practical Application of Gaze Estimation on Edge Devices  
*Linh Van Ma and Tin Trung Tran (Gwangju Institute of Science and Technology, South Korea); Moongu Jeon (Gwangju Institute of Science and Technology (GIST), South Korea)*
- 8A-2 Throughput Prediction by Radio Environment Correlation Recognition Using Crowd Sensing and Federated Learning  
*Satoshi Nakaniida and Takeo Fujii (The University of Electro-Communications, Japan)*
- 8A-3 Thermal Array Sensor Resolution-Aware Activity Recognition using Convolutional Neural Network  
*Goodness Oluchi Anyanwu, Cosmas Ifeanyi Nwakanma, Adinda Riztia Putri, Jae Min Lee and Dong Seong Kim (Kumoh National Institute of Technology, South Korea)*
- 8A-4 An Investigation on Deep Learning-Based Activity Recognition Using IMUs and Stretch Sensors  
*Nguyen Thi Hoai Thu and Dong Seog Han (Kyungpook National University, South Korea)*
- 8A-5 Comparative analysis of solar power generation prediction system using deep learning  
*So-yeong Kim and Eun-ji Lee (Chosun University, South Korea); Uttam Khatri (Chosun University, Nepal); Seokjoo Shin, Ji-In Kim and Goo-Rak Kwon (Chosun University, South Korea)*

### Oral Session 8B: AI for eHealth and Medical Diagnosis III / February 24, 2022 (Thursday)

Chair: Prof. Pyungsoo Kim (Korea Polytechnic University)

09:30-10:50 Room B/Zoom B

- 8B-1 Multi-head CNN and LSTM with Attention for User Status Estimation from Biometric Information  
*Hyunseo Park (Korea Advanced Institute of Science and Technology (KAIST), South Korea); Nakyoung Kim, Gyeong Ho Lee, Jaeseob Han and Hyeontaek Oh (KAIST, South Korea); Jun Kyun Choi (Korea Advanced Institute of Science and Technology (KAIST), South Korea)*
- 8B-2 An Explainable Computer Vision in Histopathology: Techniques for Interpreting Black Box Model  
*Subrata Bhattacharjee, Hwang-Byn Yeong, Kobiljon Ikromjanov, Rashadul Islam Sumon, Hee-Cheol Kim and Heung-Kook Choi (Inje University, South Korea)*
- 8B-3 Whole Slide Image Analysis and Detection of Prostate Cancer using Vision Transformers  
*Kobiljon Ikromjanov, Subrata Bhattacharjee, Hwang-Byn Yeong, Rashadul Islam Sumon, Hee-Cheol Kim and Heung-Kook Choi (Inje University, South Korea)*

## Oral Sessions

8B-4 A Generative Adversarial Network Approach to Metastatic Cancer Cell Images  
*Seohyun Lee, Hyuno Kim and Hideo Higuchi (The University of Tokyo, Japan); Masatoshi Ishikawa (University of Tokyo, Japan); Ryuichiro Nakato (Institute of Quantitative Bioscience, Japan)*

8B-5 UIRNet: Facial Landmarks Detection Model with Symmetric Encoder-Decoder  
*Savina Colaco, Young Jin Yoon and Dong Seog Han (Kyungpook National University, South Korea)*

### Oral Session 9A: AI Applications for Information Systems III / February 24, 2022 (Thursday)

Chair: Prof. Jeong Gon Kim (Korea Polytechnic University)

11:10-12:30 Room A/Zoom A

9A-1 Design and Analysis of an Efficient Energy Sharing System among Electric Vehicles using Evolutionary Game Theory  
*MD Rizwanul Kabir, Muhammad Mutiul Muhaimin and Abrar Mahir (Islamic University of Technology (IUT), Bangladesh); Habibul Khondokar Kabir (Islamic University of Technology (IUT), Japan)*

9A-2 GAN-based Data Augmentation for UWB NLOS Identification Using Machine Learning  
*Duc Hoang Tran (Kookmin University, Korea); ByungDeok Chung (ENS. Co. Ltd, Korea); Yeong Min Jang (Kookmin University, Korea)*

9A-3 BER Minimization by User Pairing in Downlink NOMA Using Laser Chaos-Based MAB Algorithm  
*Masaki Sugiyama, Aohan Li and Zengchao Duan (Tokyo University of Science, Japan); Makoto Naruse (The University of Tokyo, Japan); Mikio Hasegawa (Tokyo University of Science, Japan)*

9A-4 Hybrid Energy Management Systems based on Edge Processing for Electric Transportation Applications  
*Henar Canilang (Kumoh National Institute of Technology, South Korea); Danielle Jaye S. Agron (Kumoh National Institute of Technology, South Korea, South Korea); Wansu Lim (Kumoh National Institute of Technology, South Korea)*

9A-5 Studies on Intelligent Curation for the Korean Traditional Cultural Heritage  
*Jae-Ho Lee (Electronics and Telecommunications Research Institute: ETRI, South Korea); Hee-Kwon Kim and Chan-woo Park (Electronics and Telecommunications Research Institute, South Korea)*

### Oral Session 9B: AI for Data Analysis, Big Data and Cloud / February 24, 2022 (Thursday)

Chair: Prof. Ihsan Ullah (Korea University of Technology and Education)

11:10-12:30 Room B/Zoom B

9B-1 Community Detection with Graph Neural Network using Markov Stability  
*Chao Wang (Xidian University, China); Shunjie Yuan (University of Xidian, China)*

9B-2 Exploiting Heterogeneous Monitoring Data for Spatiotemporal Algal Bloom Prediction  
*Taewhi Lee and Miyoung Jang (ETRI, South Korea); Jang-Ho Choi (Electronics and Telecommunications Research Institute, South Korea); Jong Ho Won and Jiyong Kim (ETRI, South Korea)*

9B-3 Three-dimensional Data Outlier Detected by Angle Analysis  
*Zhongyang Shen (China Mobile, China)*

## Oral Sessions

- 9B-4 Identification and Analysis of COVID-19-related Misinformation Tweets via Kullback-Leibler Divergence for Informativeness and Phraseness and Biterm Topic Modeling  
*Thomas Daniel S. Clamor and Geoffrey A. Solano (University of the Philippines Manila, Philippines); Nathaniel Oco (De La Salle University, Philippines); Jasper Kyle Catapang (University of Birmingham, United Kingdom (Great Britain)); Jerome Cleofas (De La Salle University, Philippines); Iris Thiele Isip-Tan (University of the Philippines Manila, Philippines)*
- 9B-5 Blockchain based Secure Data Exchange between Cloud Networks and Smart Hand-held Devices for use in Smart Cities  
*Muneer Dar (National Institute of Electronics & Information Technology, Srinagar, India); Sameer Ahmad Bhat (Gulf University for Science and Technology (GUST), Mehref, Kuwait)*

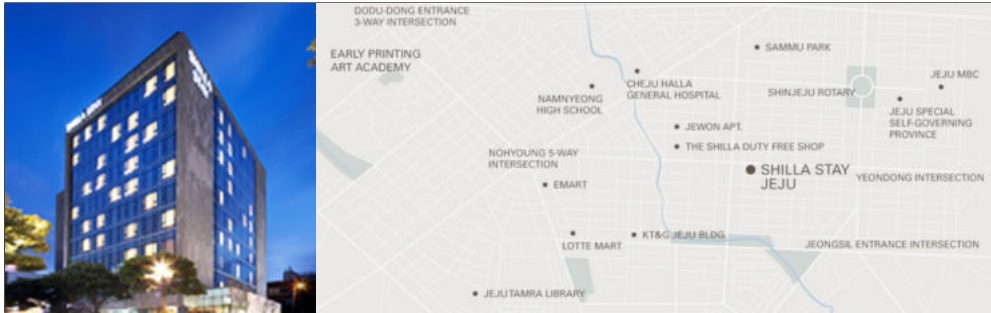
## Venue

### Shilla Stay Jeju

Address: 100 Noyeon-ro, Jeju, Jeju Island

Phone: +82-64-717-9000

<https://shillastay.com/jeju/index.do>



## Travel Information

### Hallasan National Park

Hallasan stands out at the center of South Korea's southernmost island, boasting exquisite landscapes due to its varied volcanic topography and vegetation distribution ranging vertically through the subtropical, temperate, frigid and alpine zones. The special nature of this area led to its being designated and managed as a national park in 1970, a UNESCO Biosphere Reserve in 2002, a World Natural Heritage Site in 2007. Muljangori Oreum registered as a Ramsar Wetland in 2008.



### Jeju Olle

"Olle" [Ole] is the Jeju word for a narrow pathway that is connected from the street to the front gate of a house. Hence, "Olle" is a path that comes out from a secret room to an open space and a gateway to the world. If the road is connected, it is linked to the whole island and the rest of the world as well. It has the same sound as "Would you come?" in Korean, so Jeju's "Olle" sounds the same as 'Would you come to Jeju?'



Jeju Olle's founder Suh, Myungsook used to be the chief editor of a weekly news magazine. She worked hard as a journalist, but after struggling to reach her dream job for twenty years and then being at the top of her profession for another fifteen years, she needed a rest. She was exhausted and her mind felt empty, so she set herself a new goal. She hoped that she could walk the road to Santiago (Camino de Santiago – 800km pilgrimage from France to Spain). Eventually she made her pilgrimage in September, 2006. She met a British woman at the end of the trip and they promised to share their comforts and happiness on the road with others when they returned to their homelands....

### Udo (Cow Islet)

The island was named "Udo" or "Cow Island" as its contours look like a cow lying down on the ground. There are 8 scenic wonders of Udo: day and night (Judan-myeongwol and Yahang-eobeom), sky and earth (Cheonjin-gwansan and Jidu-cheongsas), front and back (Jeonpo-mangdo and Huhae-seokbyeok), and east and west (Dongan-gyeonggul and Seobin-baeksa)



The island was named "Udo" or "Cow Island" as its contours look like a cow lying down on the ground.

There are 8 scenic wonders of Udo: day and night (Judan-myeongwol and Yahang-eobeom), sky and earth (Cheonjin-gwansan and Jidu-cheongsas), front and back (Jeonpo-mangdo and Huhae-seokbyeok), and east and west (Dongan-gyeonggul and Seobin-baeksa)

## Travel Information

### Mysterious Road (Dokkaebi Road)

On Mysterious Road (or Bugaboo Road), a parked car on a slight hill road rolls uphill instead of going down hill. This is a result of an optical illusion in which the lower part looks higher because of its surrounding environment.



### Geomun Oreum

The eroded valley of lava that erupted from the middle of the crater is the largest on Jeju Island. On one side is a 4km oval valley.

On the southeast ranch site, there are many conical hills with lava detritus which are volcanic cones without craters. The Geomi Oreum in Songdang-ri, Gujwa-eup is also called the East Geomun Oreum to distinguish it from this West Geomun Oreum.



Local residents call it Geomul Chang (Geomeol Chang) or the Geomun Oreum since it looks black when covered with forest. However, according to a scholars etymological study, "Geomun" originates from "Gam or Geom" during the Ancient Joseon Era which means "God". Therefore, "Geomun Oreum" means "Holy mountain". The forest is thick with *Pinus thunbergii* and Japanese cedars. It is a multiple-shaped volcanic cone. On the top of the mountain, there is a large crater with a small peak with a horse hoof-shaped crater that widens to the northeast.

### Manjang Cave

Manjang Cave, situated at Donggimnyeong-ri, Gujwa-eup, North Jeju, 30 kilometers east of Jeju City, was designated as Natural Monument No. 98 on March 28, 1970. The 7,416-meter long cave has been officially recognized as the longest lava tube in the world. The annual temperature inside the cave ranges from 11°C to 21°C, thus facilitating a favorable environment throughout the year. The cave is also academically significant as rare species live in the cave. Created by spewing lava, "the lava turtle", "lava pillar", and "Wing-shaped Wall" look like the work of the gods. It is considered to be a world class tourist attraction.



## Travel Information

### Sanbang Cave buddhist temple

A Buddhist statue is enshrined in a cave on the southwestern slope of Mt. Sanbang. With a height of 5 m, a length of 10 m and a width of 5 m, it is not known when this statute was carved. It is said to have been Heil residence during the Goryeo Dynasty and the calligrapher Chusa Jeong-hee Kim often visited here to contemplate life. Water drips from the ceiling and some Jeju people say the water droplets are tears of goddess Sanbangdeok who guards the mountain.



Legend has it that this daughter of Mt. Sanbang was gorgeous.

She fell in love with a youth named Goseong but an official in town who had a crush on her confiscated his property and falsely put him into exile. Disappointed and despairing of the world rife with sins, Sanbangdeok returned to the Sanbang cave, turned herself into a rock, and continues to weep to this day.

### Seopjikoji

Jutting out at the eastern seashore of Jeju Island, Seopji-Koji is one of the most scenic views with the bright yellow canola and Seongsan Sunrise Peak as a backdrop.



The pristine beauty of Jeju can be seen in Seopji-koji. Sinyang Beach, a meadow filled with canola flowers, peacefully grazing Jeju ponies, a rocky sea cliff, and a towering legendary large rock (Sunbawe) all combine to make nature's masterpiece. Unlike the other coastal areas of Jeju, it has red volcanic rock (songi) and strangely-shaped rocks that at low tide transform this area into a breath-taking stone exhibition gallery.

Seopji-Koji has become a movie and drama location hotspot with *Gingko Bed 2*, *The Uprising of Lee Jae Su*, *Thousand Day Night*, and *All In*

-A location shot was taken at Seopji-Koji to portray a picturesque scene of seaside home where the actress Jin Sil Choi lived in the movie *Gingko Bed 2* (*Danjeokbiyeonsu*). It is also well-known as the shooting location for the TV drama *All In*, *The Hyeopja beacon mound and lighthouse* attracts a lot of tourists.

### 5.16 Road's Forest Tunnel

Highway 5.16 was the first national road in Jeju which directly linked Jeju City to Seogwipo City, reducing the travel time between the southern and northern parts of the island to less than one hour. Naturally forming a tunnel, a line of trees on each side along the highway stretches for about 1 km just south of Seongpanak. Snow in winter and the brilliant red and yellow leaves in autumn make this drive a mystical experience.



## Travel Information

### Seongsan Ilchulbong (Sunrise Peak)

99 rocky peaks surround the crater like a fortress and the gentle southern slope connected to water is a lush grassland.

On the grassland at the entrance of Sunrise Peak, you can enjoy horseback riding. Breathtaking scenic views while taking a rest in the middle of climbing up the peak such as Mount Halla, the deep blues of the ocean, the multi-colored coast line, and the picturesque neighboring villages will become unforgettable memories.



### Cheonjiyeon Waterfall

The waterfall falls from a precipice with thundering sounds, creating white water pillars. It has the name Cheonjiyon, meaning 'the heaven and the earth meet and create a pond'. At 22 m in height and 12 m in width, the waterfall tumbles down to the pond to produce awe-inspiring scenery. The valley near the waterfall is home to *Elaeocarpus sylvestris* var. *ellipticus*, which is Natural Monument No. 163, *Psilotum nudum*, *Castanopsis cuspidata* var. *sieboldii*, *Xylosma congestum*, *Camellia* and other subtropical trees. This place is also famous as home to the eel of *Anguilla mauritiana*, which is Natural Monument No. 27 and is active primarily at night. The Chilspri Festival is held in every September at the falls.>





**Proceedings**

# **ICAIIIC 2022**

## **Proceedings**

# Classification and Discretization of Shadowing Toward Low Storage Radio Map

Keita Katagiri<sup>†</sup> and Takeo Fujii<sup>†</sup>

<sup>†</sup> Advanced Wireless and Communication Research Center (AWCC), The University of Electro-Communications  
1-5-1 Chofugaoka, Chofu, Tokyo 182-8585, Japan  
Email: {katagiri, fujii}@awcc.uec.ac.jp

**Abstract**—This paper considers the shadowing classifier and shadowing discretization to create a low storage radio map. We have confirmed that the data size of the radio map can be reduced while high estimation accuracy keeps via the former method. However, the data size has not been evaluated using the latter method that simply discretizes a shadowing realization in each position. Thus, this paper discusses the effectiveness of the two methods in terms of the data size and estimation accuracy. The verification results show that the shadowing classifier can skillfully predict the radio propagation compared with the discretization-based method. Meanwhile, if the slight degradation of the estimation accuracy is tolerated, the discretization-based method may be useful.

**Index Terms**—Radio map, classification, radio propagation, discretization

## I. INTRODUCTION

Several researchers are studying a radio map as a tool to skillfully grasp the radio propagation [1], [2]. The radio map often visualizes the average received signal power in each reception position by processing measured datasets collected by mobile terminals. The created radio map is stored as the statistical data in a cloud server and is provided to a transmitter and receivers. The site specific radio propagation can be precisely estimated via the radio map.

However, because the conventional radio map stores the average received signal power in each position, the enormous data may be registered according to the communication area range. To reduce the registered data size, it is necessary to unify average received signal power values in a certain area range. However, if this area range is too wide, the average received signal power may be inaccurately calculated owing to the fluctuation of the radio propagation. Obviously, there is the trade-off between the estimation accuracy and the area range for which the average power is derived.

It is well known that there is empirically the spatial correlation in the shadowing realization [3]. The reference [3] has clarified that the correlation coefficient can be approximately expressed as the exponential decay model in terms of the moving distance. This fact means that the shadowing realizations may be similar in neighborhood position.

Focusing on this property, we have proposed the shadowing classifier [4] to reduce the registered data size of the radio map. This method creates  $K$  shadowing realizations by quantizing measured data. Then, a quantized shadowing is assigned to each location according to the proposed objective function that

minimize the error between the quantized shadowing and instantaneous received signal power values. We have confirmed that the shadowing classifier can construct the accurate radio map with low registered data size.

However, if the communication area range is wide, the conventional classifier may take the long time to assign a shadowing to each position. This is because the processing load becomes enormous owing to the calculation of the objective function in each position. Here, as a simpler method, we can consider the discretization of the shadowing realization in each position. This method simply discretizes a shadowing based on the rounding without using the objective function. As a result, the registered data size may be reduced with low processing load compared to the shadowing classifier.

This paper discusses the effectiveness of the shadowing discretization in terms of the data size and estimation accuracy. The verification results show that the shadowing classifier can skillfully predict the radio propagation compared with the discretization-based method. Meanwhile, if the slight degradation of the estimation accuracy is tolerated, the discretization-based method may be useful.

The remainder of this paper is organized as follows. Sect. II mentions the conventional radio map and its issue. After that, the shadowing classifier and the discretization are presented in Sects. III and IV. Then, Sect. V explains the comparative methods for the shadowing classification. After the 3.5GHz band datasets are elucidated in Sect. VI, the verification results are described in Sect. VII. Finally, we conclude this paper in Sect. VIII.

## II. CONVENTIONAL RADIO MAP

The radio map is usually constructed based on an actual observation of radio environment. Mobile devices, such as smartphones, observe radio environment information in each location and upload these data to the cloud server. This paper assumes that the instantaneous received signal power and reception position are observation dataset in each position. The cloud server splits the communication area into a square area called a mesh. Then, the radio map is created by averaging the instantaneous received signal power samples in each mesh. Such the construction method is well known as *crowdsourcing* [5], [6]. Crowdsourcing enables us to efficiently construct a radio map in terms of the short observation time. The most

advantage of the radio map is accurate prediction of the path loss and shadowing based on actual observations.

However, since the conventional radio map accumulates the average received signal power along with a mesh code [7] in each mesh, the enormous data may be stored according to the area size. Thus, this paper focuses on the reduction of the registered data size in the radio map based on the classification of the shadowing realizations.

### III. SHADOWING CLASSIFIER

We have proposed the shadowing classifier [4] to reduce the registered data size of the radio map. This classifier constructs  $K$  shadowing realizations by analyzing measured datasets. For the explanation, this paper assumes that there are  $L$  meshes ( $l = 0, 1, \dots, L - 1$ ) in the communication area. The  $k$ -th propagation model ( $k = 0, 1, \dots, K - 1$ ) is given by

$$\bar{P}_k(d_l) = b - 10a \log_{10}(d_l) + \hat{s}_k \quad [\text{dBm}], \quad (1)$$

where  $\bar{P}_k(d_l)$  is the average received signal power estimated by the  $k$ -th model,  $d_l$  [m] denotes the link distance between the transmitter and the  $l$ -th mesh,  $b$  [dBm] is the constant value that includes the transmission power and antenna effects,  $a$  is a path loss index, and  $\hat{s}_k$  [dB] is the  $k$ -th quantized shadowing realization. The cloud server first generates the scatter diagram as the horizontal axis is  $\log_{10}(d_l)$  and the vertical axis is the average received power in each mesh. After that, we can obtain the parameters  $b$  and  $a$  based on the least squares method. This paper assumes that  $b$  and  $a$  are constant in all meshes.

#### A. Quantization of Shadowing

The cloud server first calculates a non-quantized shadowing realization  $s_l$  [dB] in the  $l$ -th mesh as follows:

$$s_l = \bar{p}_l - \hat{p}(d_l), \quad (2)$$

where  $\bar{p}_l$  [dBm] denotes the average received signal power in the  $l$ -th mesh and  $\hat{p}(d_l) = b - 10a \log_{10}(d_l)$  [dBm] is the median path loss. A non-quantized shadowing is calculated in each mesh and the shadowing vector  $\mathbf{s} = (s_0, s_1, \dots, s_{L-1})$  is created.

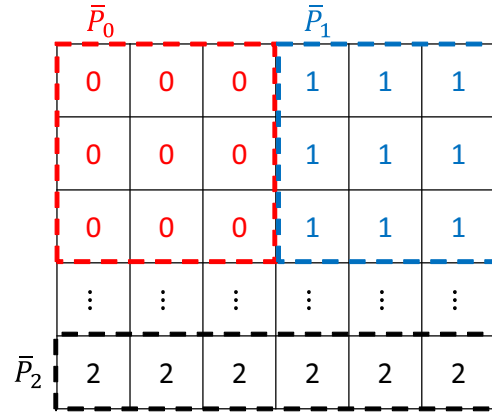
After that, the cloud server obtains the maximum shadowing  $s_{\max} = \max(\mathbf{s})$  [dB] and the minimum shadowing  $s_{\min} = \min(\mathbf{s})$  [dB], and quantizes between  $[s_{\min}, s_{\max}]$  by the step size  $w$  [dB]. Thus, we can formulate the  $k$ -th quantized shadowing realization  $\hat{s}_k$  as,

$$\hat{s}_k = s_{\min} + wk \quad [\text{dB}]. \quad (3)$$

By the processing, the shadowing classifier that consists of different  $K$  shadowing models can be generated.

#### B. Model Assignment

The registered data size can be reduced by assigning the same propagation model to multiple meshes where the radio propagation is similar. For the explanation of the model assignment, this subsection assumes that there are  $n_l$  measurement samples  $\mathbf{p} = (p_1, p_2, \dots, p_{n_l})$  in the  $l$ -th mesh. Here,  $\mathbf{p}$  is the instantaneous received signal power vector and  $p_i$  [dBm]



The same label denotes the same model.

Fig. 1. An overview of the shadowing classification.

( $i = 1, \dots, n_l$ ) denotes the  $i$ -th instantaneous value including small-scale fading. In the radio map construction, the deviation of the average received signal power may be removed by using small mesh size. However, the fluctuation of the small-scale fading may remain a little if the number of samples is too small in the mesh. To consider the effect, the cloud server assigns a propagation model  $\bar{P}_k$  to the  $l$ -th mesh so that the root mean squared error (RMSE) between  $\mathbf{p}$  and  $\bar{P}_k$  is minimized. The shadowing label  $k$  is searched based on the following function,

$$k = \arg \min_{k=0,1,\dots,K-1} \sqrt{\frac{1}{n_l} \sum_{i=1}^{n_l} (p_i - \bar{P}_k)^2}. \quad (4)$$

The cloud server registers the shadowing label  $k$  as an integer in each mesh as illustrated in Fig. 1.

### IV. SHADOWING DISCRETIZATION

This section explains the second method for reducing the registered data size. The cloud server discretizes the shadowing realization  $s_l$  in the  $l$ -th mesh by rounding it to the  $D$ -th decimal places. This calculation is performed in each mesh and each rounded value is stored in the cloud server. The registration contents of this method will be explained in Sect. VII-C. The average received signal power is derived by considering the median path loss based on  $b$  and  $a$  in addition to the discretized shadowing.

The same shadowing can be assigned using the small  $D$ ; however, the estimation accuracy may be slightly degraded owing to the discretization error. Meanwhile, the number of same shadowing models may be small if the large  $D$  is used.

### V. COMPARATIVE METHODS

#### A. Preliminary Descriptions

As the comparative, this paper uses two clustering algorithms: the  $k$ -means++ [8] and Gaussian mixture model (GMM). In radio propagation, the shadowing realization has the spatial correlation property that is empirically established as the exponential decay model [3]. Accordingly, similar

shadowing realizations can be observed in vicinity positions. Fortunately, the  $k$ -means++ assumes that several data are collectively distributed in a certain area; thus, we use the  $k$ -means++ as the first comparison method.

Additionally, the shadowing realization empirically follows the log-normal distribution; that is, this phenomenon matches to the assumed distribution of the GMM. Hence, the GMM is used as the second comparative.

Next, the input data vector is defined as follows:

$$\mathbf{z}_l = (x_l, y_l, s_l), \quad (5)$$

where  $\mathbf{z}_l$  and  $(x_l, y_l)$  are the input data vector and the coordinate value in the  $l$ -th mesh, respectively. The cloud server calculates  $(x_l, y_l)$  based on the mesh code [9].

The coordinate value and shadowing realization are different scales. Therefore, the cloud server standardizes the coordinate value and shadowing realization. Then, a weight is multiplied to each data for changing the impact on the clustering. Note that the standardization is performed using the mean and the standard deviation of each input data.

### B. $k$ -means++

$k$ -means++ clusters the input data by minimizing the following evaluation function  $F$ :

$$F = \sum_{k=0}^{K-1} \sum_{l=0}^{L-1} u_{lk} d_e(\mathbf{z}_l, \mathbf{c}_k), \quad (6)$$

where 1 is substituted into  $u_{lk}$  if  $\mathbf{z}_l$  belongs to the  $k$ -th cluster; otherwise, 0.  $\mathbf{c}_k$  denotes the centroid in the  $k$ -th cluster and  $d_e(\mathbf{z}_l, \mathbf{c}_k)$  is an euclidean distance between  $\mathbf{z}_l$  and  $\mathbf{c}_k$ . Here, the number of clusters  $K$  corresponds to the number of shadowing models in the proposed classifier.

To determine an initial centroid so that an euclidean distance between each centroid is long, the following procedures are performed:

- The cloud server randomly picks  $\mathbf{z}_l$  as the first centroid  $\mathbf{c}_0$  from the  $L$  meshes.
- The following probability is used to select another centroid  $\mathbf{c}_k$  from the  $L$  meshes, where its input data vector is defined as  $\mathbf{z}'_l$ .

$$\frac{\min_{0 \leq k \leq K-1} d_e(\mathbf{z}'_l, \mathbf{c}_k)}{\sum_{l=0}^{L-1} \min_{0 \leq j \leq K-1} d_e(\mathbf{z}_l, \mathbf{c}_j)}. \quad (7)$$

- b). is repeated while  $K$  centroids are chosen.

After the selection of the  $K$  initial centroids, the shadowing realizations are clustered into  $K$  clusters by referring the evaluation function  $F$ . Then, the cloud server averages the shadowing realizations having the same cluster label  $k$  and registers the averaged value and cluster label  $k$  in each mesh.

### C. GMM

GMM consists of the linear combinations of several Gaussian distributions. Each input data vector can be clustered by determining a Gaussian distribution that the data belongs

to. The  $k$ -th probability density function (PDF) of  $V$ -variate Gaussian distribution is defined as,

$$g_k(\mathbf{x}) = \frac{1}{(2\pi)^{V/2} |\boldsymbol{\Sigma}_k|^{1/2}} \exp \left( -\frac{1}{2} (\mathbf{x} - \mathbf{c}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x} - \mathbf{c}_k) \right), \quad (8)$$

where  $\mathbf{c}_k$  is the mean vector of the  $k$ -th distribution that corresponds to the centroid in the  $k$ -means++.  $\boldsymbol{\Sigma}_k$  is the covariance matrix of the  $k$ -th distribution.  $\mathbf{x} = (x_1, \dots, x_V)$  is the random variable vector and  $\boldsymbol{\theta}_k = (\mathbf{c}_k, \boldsymbol{\Sigma}_k)$ . The PDF of GMM is represented as follows:

$$g(\mathbf{x}|\boldsymbol{\Theta}) = \sum_{k=0}^{K-1} \alpha_k g_k(\mathbf{x}|\boldsymbol{\theta}_k), \quad (9)$$

where  $\alpha_k$  is defined as a mixing coefficient of the  $k$ -th distribution and  $\boldsymbol{\Theta} = (\alpha_0, \alpha_1, \dots, \alpha_{K-1}, \boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_{K-1})$ . Then, the log-likelihood function of GMM is given by

$$\begin{aligned} \log g(\mathbf{X}|\boldsymbol{\Theta}) &= \log \prod_{i=1}^N g(\mathbf{x}_i|\boldsymbol{\Theta}) \\ &= \sum_{i=1}^N \log \sum_{k=0}^{K-1} \alpha_k g_k(\mathbf{x}_i|\boldsymbol{\theta}_k), \end{aligned} \quad (10)$$

where  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$  and  $N$  is the number of data.

To maximize the log-likelihood function, Expectation-Maximization (EM) algorithm is generally utilized. We define the latent variable  $r_k$  that is 1 if data  $\mathbf{x}_i$  ( $i = 1, \dots, N$ ) belongs to the  $k$ -th distribution; otherwise, 0. The following procedures are the EM algorithm.

#### E-step

The cloud server calculates the posterior distribution of  $r_k$  as follows:

$$e_k^{(t)}(\mathbf{x}_i) = \frac{\alpha_k^{(t)} g_k(\mathbf{x}_i|\boldsymbol{\theta}_k^{(t-1)})}{\sum_{j=0}^{K-1} \alpha_j^{(t)} g_j(\mathbf{x}_i|\boldsymbol{\theta}_j^{(t-1)})}, \quad (11)$$

where  $e_k^{(t)}(\mathbf{x}_i)$  denotes the posterior distribution of  $r_k$  and  $t$  is the iteration index.

#### M-step

Each parameter is updated as follows:

$$\alpha_k^{(t)} = \frac{1}{N} \sum_{i=1}^N e_k^{(t)}(\mathbf{x}_i), \quad (12)$$

$$\mathbf{c}_k^{(t)} = \frac{\sum_{i=1}^N e_k^{(t)}(\mathbf{x}_i) \mathbf{x}_i}{\sum_{i=1}^N e_k^{(t)}(\mathbf{x}_i)}, \quad (13)$$

$$\boldsymbol{\Sigma}_k^{(t)} = \frac{\sum_{i=1}^N e_k^{(t)}(\mathbf{x}_i) (\mathbf{x}_i - \mathbf{c}_k^{(t)}) (\mathbf{x}_i - \mathbf{c}_k^{(t)})^T}{\sum_{i=1}^N e_k^{(t)}(\mathbf{x}_i)}. \quad (14)$$

The cloud server repeats the above steps until the following function  $O$  converges:

$$O = \sum_{i=1}^N \sum_{j=0}^{K-1} e_j^{(t)}(\mathbf{x}_i) \log \alpha_j^{(t)} g_j(\mathbf{x}_i|\boldsymbol{\theta}_j^{(t)}). \quad (15)$$



Fig. 2. Measurement environment.

TABLE I  
MEASUREMENT SPECIFICATIONS

Transmission power [dBm]	29
Center frequency [GHz]	3.5
Antenna	Ominidirectional
Antenna height of transmitter [m]	17.5
Antenna height of receiver [m]	1.5
Spectrum analyzer	Anritsu MS2712E
Resolution bandwidth [Hz]	300
Walking speed [km/h]	4
Measurement interval [ms]	0.15

## VI. 3.5GHZ BAND DATASETS

To evaluate the validity of the shadowing classification, this paper uses the measured data over 3.5GHz band [10]. The experiment campaign was conducted in Kudanshita Chiyodaku in Tokyo, Japan. There are many buildings in the area; thus, the environment is an urban. Fig. 2 presents the experiment environment and the red square denotes the location of the transmitter. The transmitter sent the continuous wave to each location. Meanwhile, the receiver having the spectrum analyzer recorded the instantaneous received signal power and reception location while walking at about 4 [km/h]. Table I shows the experiment parameters. We used 100423 samples to construct the shadowing classifier.

## VII. EMULATION RESULTS

This section mentions the evaluation results. The comparative methods use the weights of 0.1 and 0.9 for  $(x_i, y_i)$  and  $s_i$ , respectively.

### A. Example of Radio Maps

Fig. 3 indicates the examples of the radio map and shadowing classification. Fig. 3(a) is the true radio map that visualizes the average received signal power in each 10m mesh. Meanwhile, the others are the classified shadowing maps. Here,  $D$  is 4 and  $K$  is 218 that corresponds to  $w = 0.2$  [dB]. These maps clarify that the shadowing realizations can be appropriately classified. For example, around the southeast area in Fig. 3(a), the average received signal power is similar; that is,

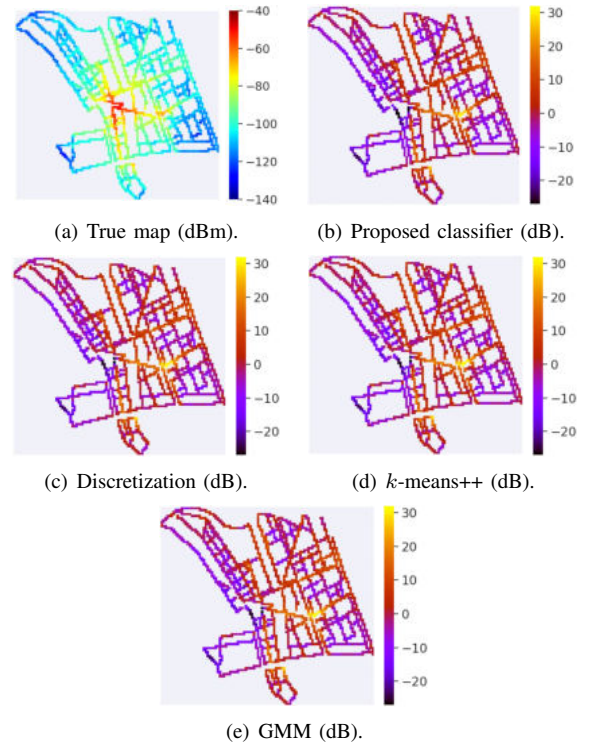


Fig. 3. Examples of the radio map and shadowing classification.

the spatial correlation of the shadowing can be found. Each shadowing map reveals that similar shadowing realizations can be classified in the area.

### B. Estimation Accuracy

This subsection evaluates the prediction accuracy of each method using RMSE as follows:

$$\text{RMSE} = \sqrt{\frac{1}{L_{\text{eva}}} \sum_{i=1}^{L_{\text{eva}}} (p_i - \bar{p}_{i,\text{predicted}})^2} \quad [\text{dB}], \quad (16)$$

where  $p_i$  [dBm] is defined as an instantaneous received signal power in the  $i$ -th mesh.  $\bar{p}_{i,\text{predicted}}$  [dBm] denotes the estimated power in the  $i$ -th mesh, and  $L_{\text{eva}}$  is the number of 10m meshes. We divided 100423 samples into 3 groups and performed the cross-validation.

The prediction accuracy is depicted in Fig. 4. The RMSE of the Radio map is calculated using the median received signal power in each mesh without the shadowing classification. Meanwhile, the Fitted path loss estimates the average received signal power based on the path loss parameters  $b$  and  $a$ . Note that  $D$  is 0 in the Discretization; thus, a shadowing realization is stored as an integer value in each mesh. These numerical values mean that the average RMSE becomes small in the proposed classifier,  $k$ -means++, and GMM as  $K$  increases. The Discretization-based method can skillfully calculate the average received signal power; however, the accuracy saturates around 4.2 [dB] owing to the discretization error. In contrast, the proposed classifier can slightly improve the RMSE because

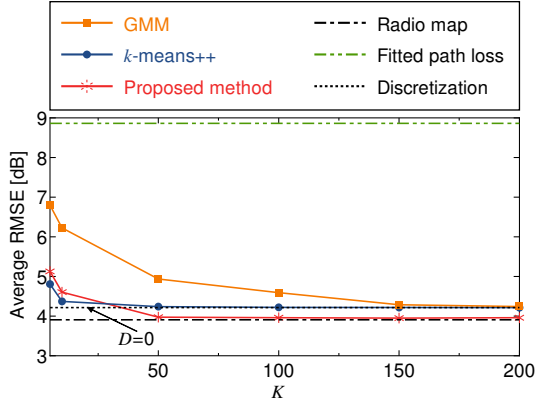


Fig. 4. Estimation accuracy.

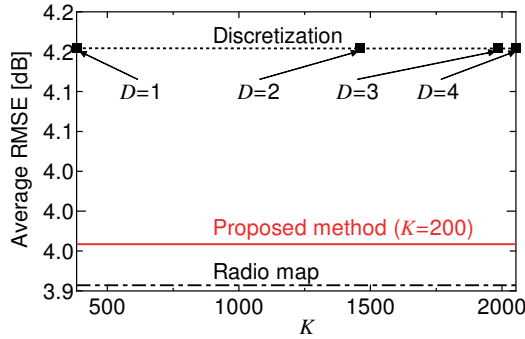


Fig. 5. Average RMSE of the proposed classifier and discretization.

the shadowing realization is classified by considering the small-scale fading.

Additionally, the RMSE between the classifier and discretization is shown in Fig. 5. In the Discretization-based method, the four plots indicate the accuracy using  $D = 1, 2, 3, 4$ , respectively; thus, there are many shadowing models in large  $D$ . Note that the RMSE of the proposed classifier is calculated in  $K = 200$  that corresponds to the right end in Fig. 4. These values show that the RMSE is not improved in the Discretization-based method even if  $D$  increases. Meanwhile, the proposed classifier is superior to the Discretization-based method with small  $K$ . From the results, we argue that the Discretization-based method should not be used in  $D$  is 1 or more because the registered data size unnecessarily increases.

### C. The Registered Data Size

Finally, this subsection calculates the data size to verify the effectiveness of the shadowing classification. Tables II and III present the accumulation items of the proposed classifier and conventional radio map, respectively. The proposed classifier and comparative methods construct  $K$  shadowing models; thus, a shadowing value is managed with the label  $k$  in the shadowing table as shown in Table II(b). These elements are registered for  $K$  models. To reduce the accumulated data size,

TABLE II  
ACCUMULATION ITEMS IN CLASSIFIER AND COMPARISON METHODS

(a) Mesh Table

Element	Type	Size [byte]
Mesh code (10m)	text	11
Shadowing label (Cluster label) $k$	int	4
Total data size per mesh		15

(b) Shadowing Table

Element	Type	Size [byte]
Shadowing label (Cluster label) $k$	int	4
Shadowing realization $\hat{s}_k$	double	8
Total data size per model		12

(c) Single Table

Element	Type	Size [byte]
$b, a$	double	16
Transmitter mesh code	text	16
1st mesh code	text	5
Total data size		37

TABLE III  
ACCUMULATION ITEMS IN THE RADIO MAP

(a) Mesh Table

Element	Type	Size [byte]
Mesh code (10m)	text	11
Average received signal power	double	8
Total data size per mesh		19

(b) Single Table

Element	Type	Size [byte]
1st mesh code	text	5

mesh table as defined in Table II(a) stores the label  $k$  as the integer in each mesh. Although the reference [7] presents the 10m mesh code as text type with 16 [byte], this paper considers the size as 11 [byte] by considering the constant value of the 1st mesh code. The remaining size of 5 [byte] is accumulated in Table II(c) with  $b, a$ , and the transmitter mesh code.

Meanwhile, Table III(a) shows the accumulation items in the conventional radio map. The table registers the average power value with 10m mesh code. The 1st mesh code is registered in Table III(b).

In the shadowing discretization with  $D = 0$ , the shadowing realization is accumulated as the integer; thus, Table II(a) is used with the discretized shadowing value instead of the label  $k$ . Meanwhile, in  $D$  is 1 or more, since the discretized shadowing value is created as the floating point type, Tables II(a) and II(b) are used. In both cases, Table II(c) is utilized to calculate the path loss.

In summary, the total data size of the proposed classifier and comparative methods can be expressed as follows:

$$A_1 = (L \times 15) + (K \times 12) + 37 \text{ [byte]}, \quad (17)$$

where  $A_1$  is the accumulated data size in the proposed classifier and comparative methods. Additionally, the accumulated

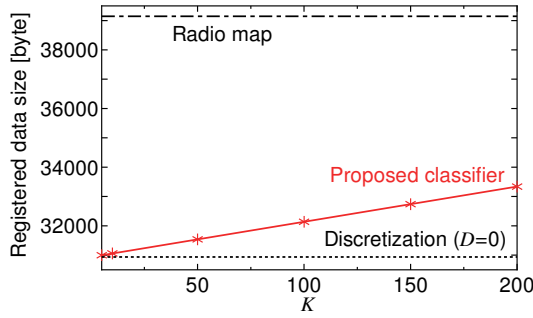


Fig. 6. The accumulated data size.

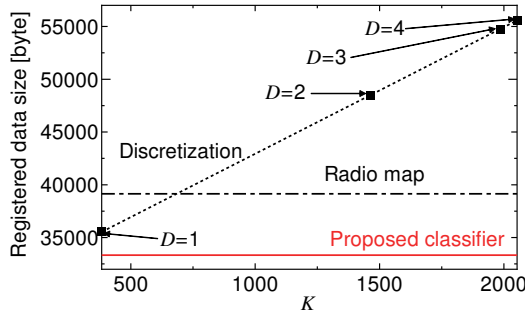


Fig. 7. The registered data size of the proposed classifier and quantization.

data size of the conventional radio map  $A_2$  can be defined as,

$$A_2 = L \times 19 + 5 \quad [\text{byte}]. \quad (18)$$

Finally, the data size of the discretization with  $D = 0$  is given by

$$A_3 = L \times 15 + 37 \quad [\text{byte}], \quad (19)$$

where  $A_3$  is the registered data size of the discretization-based method. Note that  $A_3 = A_1$  in  $D$  is 1 or more.

Fig.6 depicts the data size in each method with  $L = 2060$ . This figure clarifies that the proposed classifier and discretization-based method can decrease the data size around 14.8–20.8 [%] compared to the conventional radio map. If the slight degradation of the RMSE is tolerated, the discretization-based method with  $D = 0$  may be useful in terms of the registered data size. If not, the proposed classifier should be utilized.

The registered data size of the discretization as  $D$  is 1 or more is illustrated in Fig.7. It can be confirmed that the data size increases in an increase of  $D$ . Additionally, since the number of same shadowing realizations is little, the size becomes large compared to the conventional radio map. Thus, in the discretization, the shadowing realization should not be accumulated with floating point type.

### VIII. CONCLUSION

This paper has evaluates the shadowing classification toward to the low storage radio map. As the reduction of the data size,

we have considered three methods: the shadowing classifier, the discretization-based method, and conventional clustering algorithms. The performance evaluation has revealed that the proposed classifier is suitable in terms of the prediction accuracy and data size. Additionally, the discretization-based method with  $D = 0$  is useful if the slight degradation is tolerated.

### ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Numbers 18H01439, 18KK0109, 19J23352.

### REFERENCES

- [1] S. H. Jung and D. Han, "Automated construction and maintenance of Wi-Fi radio maps for crowdsourcing-based indoor positioning systems," *IEEE Access*, vol. 6, pp. 1764–1777, Dec. 2017.
- [2] X. Ying, S. Roy, and R. Poovendran, "Pricing mechanism for quality-based radio mapping via crowdsourcing," in *Proc. 2016 IEEE GLOBE-COM*, Washington, DC, USA, Dec. 2016, pp. 1–6.
- [3] M. Gudmundson, "Correlation model for shadow fading in mobile radio systems," *Electron Lett.*, vol. 27, no. 23, pp. 2145–2146, Nov. 1991.
- [4] K. Katagiri, K. Sato, K. Inage, and T. Fujii, "Experimental verification of shadowing classification for radio map," in *Proc. 92nd IEEE VTC2020-Fall*, Victoria, BC, Canada, Dec. 2020, pp. 1–7.
- [5] X. Wang, M. Umehira, B. Han, P. Li, Y. Gu, and C. Wu, "Online incentive mechanism for crowdsourced radio environment map construction," in *Proc. IEEE ICC 2019*, Shanghai, China, May 2019, pp. 1–6.
- [6] Y. Ye and B. Wang, "Rmapcs: Radio map construction from crowdsourced samples for indoor localization," *IEEE Access*, vol. 6, pp. 24224–24238, Apr. 2018.
- [7] K. Sato, M. Kitamura, K. Inage, and T. Fujii, "Measurement-based spectrum database for flexible spectrum management," *IEICE Trans. Commun.*, vol. E98-B, no. 10, pp. 2004–2013, Oct. 2015.
- [8] Y. Uesugi, K. Katagiri, K. Sato, K. Inage, and T. Fujii, "Clustering for signal power distribution toward low storage crowdsourced spectrum database," *IEICE Trans. Commun.*, vol. E104-B, no. 10, pp. 1237–1248, Oct. 2021.
- [9] R. Hasegawa, K. Katagiri, K. Sato, and T. Fujii, "Low storage, but highly accurate measurement-based spectrum database via mesh clustering," *IEICE Trans. Commun.*, vol. E101-B, no. 10, pp. 2152–2161, Oct. 2018.
- [10] K. Katagiri, K. Onose, K. Sato, K. Inage, and T. Fujii, "Highly accurate prediction of radio propagation using model classifier," in *IEEE 89th VTC2019-Spring*, Kuala Lumpur, Malaysia, Apr. 2019, pp. 1–5.

# Countering DNS Vulnerability to Attacks Using Ensemble Learning

Love Allen Chijioke Ahakonye, Cosmas Ifeanyi Nwakanma, Simeon Okechukwu Ajakwe\*,  
Jae Min Lee, and Dong-Seong Kim  
*IT Convergence Engineering, Kumoh National Institute of Technology Gumi, South Korea*  
(loveahakonye, cosmas.ifeanyi, ljmpaul, dskim) @kumoh.ac.kr  
simeonajlove@gmail.com\*

**Abstract**—The Domain Name System (DNS) is the hub of the cyberspace and communications services which also plays enabling role in the Industrial Internet of Things (IIoT) and transmission at large. DNS enciphering in HyperText Transfer Protocol Secure (HTTPS) as DoH did not eliminate vulnerability and intrusion into critical systems. This study proposed a time-efficient Ensemble Learning (EL) model for countering DNS vulnerability to attack. The proposed EL candidate incorporates feature selection capability in extracting relevant features for enhanced model optimization. The simulation results showed that the proposed EL candidate effectively mitigates vulnerability, classifying DNS traffic into Non-DoH, Malicious DoH and Benign-DoH. The proposed model outperforms other compared state-of-art EL techniques with a combined advantage of accuracy and training time of 99.5% and 13.96s.

**Index Terms**—AI, DNS, Ensemble Learning, IIoT,

## I. INTRODUCTION

Amongst the protocols in a network communication system is the Domain Name System (DNS). It serves as the internet directory. Online information access is through the DNS. It is also one of the early network protocols with high vulnerability and diverse security flaws constantly exploited. DNS exploitation is invariably a domain of eminent attention for cyber-security researchers. However, delivering privacy and safeguarding DNS demands and acknowledgement remains a daunting endeavor as intruders employ advanced intrusion strategies for exploiting DNS vulnerability [1].

Some of the DNS attacks are domain lock-up attacks, DNS hijacking, DNS spoofing, DNS Tunneling. In pursuit of security improvement, DNS has become more relevant recently by providing validation and approval to some internet resources. Nevertheless, the recent development of DNS falls short of guaranteeing requisite security to users. Hence, DNS security has been a widely researched topic in the cyber-security domain. The National Institute of Standards and Technology (NIST) issued a document with recommendations for the safe deployment of DNS to prevent security challenges [2].

To reduce the DNS vulnerabilities associated with security and data processing, the Internet Engineering Task Force (IETF) initiated DNS over HTTPS (DoH) in RFC8484 [1]. DoH is a set of codes that strengthens security and counters attacks by encoding DNS mistrust and forwards in a hidden channel such that there is no data obstruction in transit. How-

ever, the inadequacy of an illustrative dataset is a challenge in evaluating the approach for securing DoH traffic in a network topology. Distinct perspectives for DNS vulnerability such as Internet Protocol (IP)/domain boycott and removing suspected DNS packets to attain DNS restriction have been applied [3]–[5]. Most researchers criticize DoH for making DNS tunnels difficult for attack detection.

There have been efforts at protecting network systems; this attempt includes network intrusion detection systems (NIDS), the use of firewalls to minimize the issues of unlawful access, etc. Consequently, the authors in [1] attempted resolving the problem of dataset insufficiency. The study proposed an approach to secure a model dataset. It is for the analysis, testing and evaluation of DoH traffic in hidden channels. The focus was on deploying DoH in an application to take hold of benign and malicious DoH traffic. This new protocol improved privacy and DNS security. However, the issue of DNS vulnerability persists, hence, the need for an Artificial Intelligence (AI) countering measure.

AI has supported in strengthening the performance of detection techniques in NIDS. Considering this, numerous authors have utilized AI techniques for vulnerability and attack mitigation. For instance, authors [6]–[8] proposed various intrusion detection frameworks. Presently industrial innovations such as the industrial internet of things (IIoT) and machine learning (ML) have revolutionized daily life and influenced various sectors. IIoT has become ubiquitous as it is applicable in different areas, including communication and the industry generally. ML has played enabling role in the development of Intrusion Detection Systems (IDS). Such application includes attack and vulnerability detection, which has found use in IIoT.

The use of ML for enabling attack and vulnerability detection is still a contending issue. Thus, this study investigated the DoH traffic and non-DoH traffics for an efficient IDS. Also, an investigation into different ensemble learning (EL) prospects for an efficient, accurate and time-aware countering system for DNS vulnerability to attacks. Recent research works show that ML would not only improve the detection rate but would also reduce the computational time [9].

This work has the following goals:

- To deduce an efficient AI technique for IDS in terms of the combined advantage of accuracy and least training time using the CIRA-CIC-DoHBrw-2020 datasets.



- To utilize Python to determine the best EL candidate with respect to DNS vulnerability to attacks.
- To propose an EL architecture to counter DNS vulnerability, achieve high accuracy of detection with reduced complexity and less training time.

The paper arrangement is thus: Section II is the summary of existing works detailing IIoT Intrusion detection and various approaches of EL and establishing research gaps. In Section III, problem formulation, which involves a brief description of EL and the best AI candidate, has been discussed. Section IV describes the performance evaluation with the comparison of accuracy plot and time-efficiency of the proposed model, Section V is the conclusion of the paper.

## II. RELATED WORKS

### A. Intrusion Detection System applying Ensemble Learning (EL)

Indications are replete in literature to deduce that conventional ML approaches may not be reliable in manipulating intricate data, such as high-dimensional data, noisy (usually from industrial environment) and imbalanced data. To solve this challenge, various works on EL schemes are now accessible to ensure a blend of data mining, fusion and modeling into a consolidated scheme [10].

EL is commonly a collective classifier system. It requires the combination and training of collective learners known as base learners, to determine a learning problem. The EL framework can be homologous or diverse ensembles reliant on the type of base learners constituting the scheme. While the homologous have base learners, the diverse ensemble comprises of individual unique learners or simply called component learners. Significantly, most studies on EL schemes are focused on weak learners, thus, base learners are often referred to as weak learners. Fig. 1 depicts the fusion of base learners for an improved outcome, where the outcome is anticipated to be of an enhanced performance in comparison with base learners 1, 2, 3 ...K. The base learners can be perceived as those learners



Fig. 1. Flow of Ensemble Learning showing the fusion of base learners

whose accuracy in terms of binary classification ability is marginally within 50%, [10].

EL techniques uses a blend of distinct classifiers for detection, which has facilitated different operations and improved

performance in IoT systems. This yields enhanced performance for various attack types and protocols used in IoT networks. The study by [11] proposed a modern ensemble IDS approach for attack defence on Ethernet Consist Networks of trains. Though this system delivered superior results, the approach can be complex with a lack of computational speed. Authors [12] proposed an AdaBoost EL scheme for mitigating malicious activities, especially HTTP, MQTT and DNS attacks from botnet attacks in IoT networks. The approach holds good performance accuracy when compared to other models. However, the computation time is not efficient for a time-critical system. Recently, the works of [13] proposed a novel scheme named EISStream for detecting concept drift utilizing traditional ML approach and EL. This approach uses only optimum classifier to vote for decision by using the majority voting scheme. Authors [14] in a recent study attempted AI techniques for mitigating attacks in SCADA Systems.

IIoT are real-time systems which are time-critical and as such a vital factor in its design. Regardless of the value improvement of these authors, the techniques lacks consideration for EL technique for mitigating DNS vulnerability and time complexity for proposed EL in attack detection. Moreover, all enumerated studies did not establish an efficient and time-aware EL. Hence, this study presents the application of an effective EL approach for countering DNS attacks.

## III. METHODOLOGY

### A. DNS Traffic Dataset

This dataset is made of recent attack features as enumerated in [1]. Below is a brief overview of the composition of the dataset. Benign-DoH: This is a non-malicious DoH activities gathered from websites that uses HTTPS, and tagged Benign-DoH. In an effort generate sufficient data to stabilize the dataset, numerous webpages from Alexa were surfed. Non-DoH: This data was created with similar method as in Benign-DoH by employing browsers such as Google Chrome and Mozilla Firefox. Malicious-DoH: DNS channeling mechanism like DNSCat2, Iodine and dns2tcp were used to create malicious DoH data. This mechanism sends encoded TCP data in DNS query. Particularly, the mechanism generate channels of encoded traffic. Consequently, DNS query is made by applying traffic layer-encoded HTTPS application to dedicated DoH servers.

### B. Ensemble Learning Framework For DNS

An EL approach is presented to detect vulnerability which reveal IIoT networks through the DNS codes by examining the DNS codes, as displayed in Fig 2. The architecture comprises of three stages, viz: a feature lay, feature selection, and EL techniques. The feature lay is simply the features of the DNS traffic dataset. Following is the Pearson Coefficient Correlation (PCC). It is used for choosing the lowest correlated features with likely features of benign, Non-DoH and malicious patterns as seen in equation 1, with the option of variables that has a high correlation value threshold of between +/-1. Lastly, the EL technique used for countering the vulnerability in

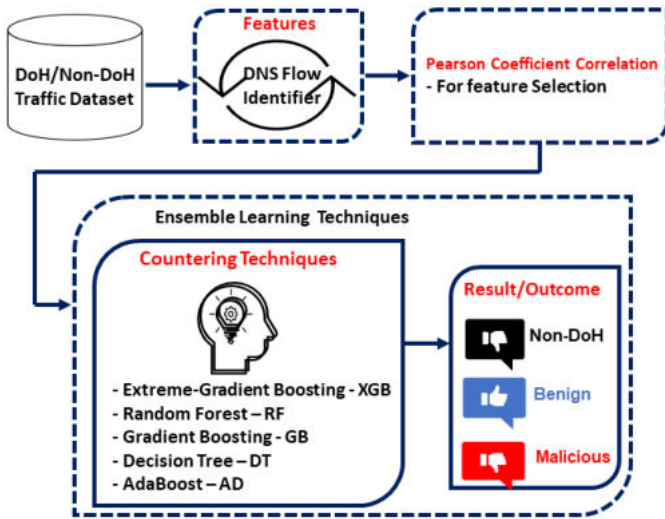


Fig. 2. Proposed Framework for Countering DNS Vulnerability

DNS thereby classifying into Non-DoH, benign and malicious. The dataset was split into training and testing to conduct the experimental evaluation, where 80% of data is used for training and 20% for testing.

$$Q = \frac{\sum (\alpha_i - \hat{\alpha}) (\beta_i - \hat{\beta})}{\sqrt{\sum (\alpha_i - \hat{\alpha})^2 (\beta_i - \hat{\beta})^2}}, \quad (1)$$

Feature selection is vital in IDS for ridding unnecessary features and choosing the relevant ones that supports segregation of DNS traffic into benign, Non-DoH and malicious. It improves the general performance of the system, lowering computational cost, eliminating information redundancy and enhances accuracy and also helps in the analysis of network data normality. The training specifications are as presented in Table I. This study focused on state-of-the-art EL candidates such as Extreme-Gradient Boosting (XGB), Gradient Boosting (GB), AdaBoost (AD), Random Forest (RF) and Decision Trees (DT) EL classifiers.

TABLE I  
ENSEMBLE LEARNING TRAINING PARAMETERS

Parameter	Remark
Observations	226406 samples
Predictors	11
Classes	3
No of Trainable Classifiers	5
Model type	Decision Tree Classifier
Result Presentation type	Response plot
Training time	13.960 sec
No of Splits	10
Random State	42
Cross-validation	kfold

## IV. PERFORMANCE EVALUATION

### A. Parameter Metrics

To determine an efficient EL technique for countering DNS vulnerability to attacks, the CIRA-CIC-DoHBrw-2020 dataset was evaluated utilizing XGB, GB, AD, RF and DT EL candidates. See Figs 3 and 4 for the performance comparison of the evaluated models. The performance of the prospective EL model was compared with the studies by [11]–[13] using performance evaluation metrics summarized by equations (2) and computation time. The authors in [13] achieved a higher accuracy. However they did not consider computational time as an important factor for such a system. Thus, a research gap which necessitated this new approach.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \quad (2)$$

where  $FP$ ,  $TP$ ,  $TN$  and  $FN$  represents False Positive, True Positive, True Negative and False Negative respectively.

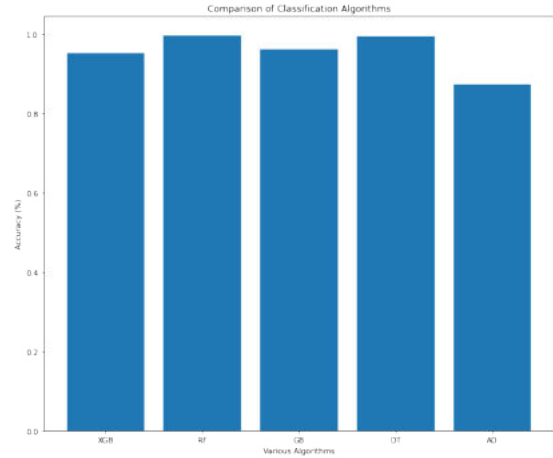


Fig. 3. Comparison of Accuracy Performance of Various EL Techniques

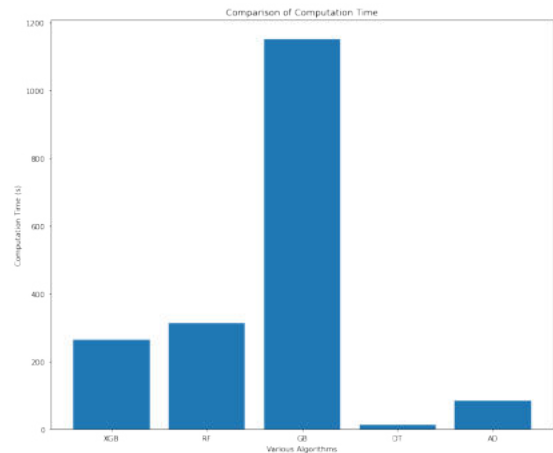


Fig. 4. Comparison of Computation Time of Various EL Techniques

## B. Experimental Environment

Training and testing of the proposed EL scheme was carried out on Google Colab using various Keras and scikit learn libraries. All experimentation can be administered on a laptop designed by NVIDIA GeForce GTX 1080Ti, 11G memory, Intel Xeon E5-1650 CPU 3.60-GHz processor, with windows 10 64 bit operating system.

## C. Comparison of the Performance of some EL Techniques

This study centres on the choice of an efficient EL technique for countering DNS vulnerability to attacks, with effort at comparing the performance of various EL techniques in review literature. Highlighting the achievements of the utilization of EL techniques such as AdaBoost EL (AD-EL), ensemble IDS (E-IDS) and ElStream. Table II, gives an illustration of the achievements based on computation time, accuracy, precision and recall.

TABLE II  
PERFORMANCE COMPARISON OF VARIOUS ENSEMBLE TECHNIQUES

Models	Acc. (%)	Prec (%)	Recall (%)	Comp. Time (ms)
DT	99.3	99.2	99.3	13.96
AD	87.2	87.5	86.9	84.38
AD-EL [12]	99.54	98.62	98.93	150.8
XGB	95.1	95.7	95.1	263.36
GB	96.0	96.0	95.6	1149.92
RF	99.5	99.4	99.6	313.23
Elstream [13]	99.99	99.95	99.97	-
E-IDS [11]	97	96.8	97.5	-

## D. Trade-off of Computation Time and Accuracy

It is vital to note that the efficient performance of a model is not solely based on accuracy, rather on a combination of performance evaluation metrics as portrayed in this study. Time is an important factor for time-critical system as DNS security, hence requires the mitigation technique to act as swift as possible. Since any time lapse in mitigating a security breach could lead to fatality in exposure/loss of vital information. On this note, a trade-off between accuracy and time is used as performance metrics. The DT compensated its accuracy (99.3%) with computation time (13.96ms), while other models (AD-EL 99.54%) had longer computation time.

## V. CONCLUSION

This work evaluated various EL candidates leveraging DNS traffic data. The result shows that the performance of the proposed EL DT exceeded other states of the art EL candidates such as RF, XGB, GB and AD for the efficient mitigation of DNS vulnerability, as can be seen in the least computation time of 13.96ms and accuracy of 99.3%. The specific, practical significance of this ascertainment is in the decision of an efficient EL candidate for IDS, basically where the preference is time-efficiency. For future directions, expanding the comparison by unveiling the capability and flexibility of the DT parameters to more current cyber-security datasets looks promising.

## ACKNOWLEDGMENT

This research work was supported by Priority Research Centers Program through NRF funded by MEST (2018R1A6A1A03024003) and the Grand Information Technology Research Center support program (IITP-2021-2020-0-01612) supervised by the IITP by MSIT, Korea.

## REFERENCES

- [1] M. MontazeriShatoori, L. Davidson, G. Kaur, and A. H. Lashkari, "Detection of DoH Tunnels using Time-series Classification of Encrypted Traffic," in *The 5th IEEE Cyber Science and Technology Congress, Calgary*, 06 2020.
- [2] "Secure Domain Name System (DNS) Deployment Guide, author=Chandramouli, Ramaswamy and Rose, Scott," *NIST Special Publication*, vol. 800, pp. 81–2, 2006.
- [3] M. MontazeriShatoori, L. Davidson, G. Kaur, and A. Habibi Lashkari, "Detection of DoH Tunnels using Time-series Classification of Encrypted Traffic," in *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCCom/CyberSciTech)*, 2020, pp. 63–70.
- [4] A. Nadler, A. Aminov, and A. Shabtai, "Detection of Malicious and Low Throughput Data Exfiltration over the DNS Protocol," *Computers and Security*, vol. 80, pp. 36–53, 2019.
- [5] C. Patsakis, F. Casino, and V. Katos, "Encrypted and Covert DNS Queries for Botnets: Challenges and Countermeasures," *Computers and Security*, vol. 88, p. 101614, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S016740481831321X>
- [6] G. C. Amaizu, C. I. Nwakanma, S. Bhardwaj, J. M. Lee, and D. S. Kim, "Composite and Efficient DDoS Attack Detection Framework for B5G Networks," *Computer Networks*, vol. 188, p. 107871, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1389128621000438>
- [7] M. Teixeira, T. Salman, M. Zolanvari, R. Jain, N. Meskin, and M. Samaka, "SCADA System Testbed for Cybersecurity Research Using Machine Learning Approach," *Future Internet*, vol. 10, no. 8, p. 76, Aug 2018. [Online]. Available: <http://dx.doi.org/10.3390/fi10080076>
- [8] A. H. Mirza, "Computer Network Intrusion Detection Using Various Classifiers and Ensemble Learning," in *2018 26th Signal Processing and Communications Applications Conference (SIU)*, 2018, pp. 1–4.
- [9] S. O. Ajakwe, C. I. Nwakanma, D. S. Kim, and J. M. Lee, "Intelligent and Real-Time Smart Card Fraud Detection for Optimized Industrial Decision Process," in *2021 Korean Institute of Communication and Sciences Summer Conference*, vol. 75, 2021, pp. 1368–1370. [Online]. Available: [www.dbpia.co.kr/journal/articleDetail?nodeId=NODE10587528](http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE10587528)
- [10] X. Dong, Z. Yu, W. Maharani, Cao, Y. Shi, and Q. Ma, "A Survey on Ensemble Learning," *Frontiers of Computer Science*, vol. 14, no. 8, pp. 241–258, 2020. [Online]. Available: <https://doi.org/10.1007/s11704-019-8208-z>
- [11] C. Yue, L. Wang, D. Wang, R. Duo, and X. Nie, "An Ensemble Intrusion Detection Method for Train Ethernet Consist Network Based on CNN and RNN," *IEEE Access*, vol. 9, pp. 59 527–59 539, 2021.
- [12] N. Moustafa, B. Turnbull, and K. R. Choo, "An Ensemble Intrusion Detection Technique Based on Proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4815–4830, 2019.
- [13] A. Abbasi, A. R. Javed, C. Chakraborty, J. Nebhen, W. Zehra, and Z. Jalil, "ElStream: An Ensemble Learning Approach for Concept Drift Detection in Dynamic Social Big Data Stream Learning," *IEEE Access*, vol. 9, pp. 66 408–66 419, 2021.
- [14] L. A. C. Ahakonye, C. I. Nwakanma, J. M. Lee, and D. S. Kim, "Evaluating Artificial Intelligence Mitigation Techniques for Countering Attack on Smart Factory SCADA Network," in *2021 2nd Korea Artificial Intelligence Conference (KAI)*, 2021.

# NetMD- Network Traffic Analysis and Malware Detection

Sampath Kumar Katherasala<sup>1</sup> Vaddeboyina Sri Manvith<sup>2</sup> Ajay Therala<sup>3</sup> Manjari Murala<sup>4</sup>

<sup>1,2,3</sup>Tata Consultancy Services Limited, Hyderabad, India <sup>4</sup>IIT Hyderabad, India

<sup>1</sup>sampath.katherasala@tcs.com <sup>2</sup>srimanvith.v@tcs.com <sup>3</sup>ajay.therala@tcs.com <sup>4</sup>muralamanjari@gmail.com

**Abstract**— *In this digitally connected world, data networks play a crucial role in the communication field. As there is a massive growth in data exchange, transactions and other sensitive data need to be secured. Networks must be safeguarded from malware attacks for flawless transmission of data between devices. Some of the harmful consequences of malware attacks are gaining administrative control and data breaches. Malware detection has become a significant task to get rid of those dreadful consequences and make networks secure. In this paper, we implement machine learning algorithms against the malware detection datasets NetML and CICIDS2017, and the traffic classification dataset non-vpn2016 dataset. The results are very promising and have been validated with the results obtained in the NetML Network Traffic Analytics Challenge 2020, organized by ACANETS. The overall score on the CICIDS2017 and non-vpn2016 datasets outperformed the baseline results published in the challenge, against all the five tracks (top, mid, and fine annotations).*

**Keywords**— *Machine Learning, Malware Detection, NetML, CICIDS2017, non-vpn2016.*

## I. INTRODUCTION

In recent years, there has been an enormous spike in the number of electronic devices connected to the internet. With the increasing number of connected devices, malware attacks transferred over the network are also increasing regularly.

Malware is malicious software created with the intent of gaining illegal access to computers and other network devices. Malicious software can be sent over the network through web links or emails. When the user clicks the link with malicious code, these attacks run in the background and lead to a data breach. These breaches may include the loss of the victim's confidential data. There are various types of malware attacks. Malware attacks such as spyware unknowingly steal the user's information; while Ransomware encrypts the user data until the victim makes a payment. Key loggers record everything that the victim inputs to collect sensitive information about the user, such as passwords. The Botnet gains access to the system and can control the computer system remotely. Root-privilege acquisition acquires administrative rights and installs malicious applications and runs them in the background. Adware servers throw advertisements by looking at a victim's browsing history, becoming a threat to the network. It is of the utmost priority to eliminate these attacks and safeguard the devices and network from malware. In this data-driven world, machine learning (ML) and deep learning (DL) algorithms are evolving and helping to find solutions to various problems in every sector. Researchers started employing ML and DL in malware detection tasks due to the development of complex algorithms and the amount of data available on the web for analysis. These ML and DL

Algorithms can be used in Network Traffic Analysis (NTA) tasks (Malware detection, traffic classification, etc.).

Researchers have employed various supervised and unsupervised learning techniques in network traffic analysis tasks. They found that supervised techniques are more effective than unsupervised techniques. Supervised techniques help in learning the patterns and predicting the class of data packets (malignant or benign).

As the dataset plays a crucial role in the performance of the ML model, the network research community needs a comprehensive, open, and up-to-date dataset for obtaining effective classification results. To address this issue, the NetML challenge introduced three datasets: NetML, CICIDS2017, and non-vpn2016. These datasets contain nearly 1.3 million labeled flows and provide researchers with a benchmarking platform to evaluate their approaches and contribute to their research on NTA.

This paper is organized into five sections. Section II provides an overview of related work on various malware detection and traffic classification problems. Section III describes the different datasets used in the challenge. Section IV explains the methodology and algorithms implemented for enhancing the results. Section V analyses the results obtained and also compares and contrasts them with the baseline results published in the challenge. Some interesting conclusions are presented in Section VI. In the appendix, we present some of the relevant exploratory data analysis of the network traffic flow features.

## II. RELATED WORK

In the literature, we see many research articles published on network traffic analysis, malware detection, and also coupled with machine learning algorithms. In this paperwork, we predominantly focus on malware detection, and network traffic classification. A performance-based comparison approach was proposed by Rahim et al. [1], on the NSL-KDD dataset. They have evaluated the results based on Support Vector Machine (SVM), Random Forest (RF), and Extreme Learning Machine (ELM) algorithms on full, half, and  $\frac{1}{4}$  data. They concluded that RF outperforms other approaches. Jiang et al. [2] worked on DoS and found that only fourteen classes of the CICIDS2017 dataset were considered. They compared the original CICIDS2017 features with the newly proposed features for neural networks. Ullah et Mahmoud proposed a two-level model [3]. They used a Decision tree to classify the traffic as an attack or normal at the first level and identified the type of attack at the second level using a random forest, after SMOTE-based data augmentation and edited KNN. Their

two-level method has been tested on the UNSW-NB15 and CICIDS2017 datasets. A two-step approach was proposed by Ustebay et al. [4]. on the CICIDS2017 dataset. The most useful features were identified using the Recursive Feature Elimination (RFE) technique, and these features are used for training the neural network model. The models developed do not perform well when all types of attacks are considered. Otherwise, performance results are not that great.

Arnaud et al. [5] have worked on the MLP algorithm and have built a golden model. They used neural networks to evaluate the results on the CICIDS2017 dataset. Their approach provided better results on the complete dataset and good performance on training the model without the IP addresses and destination port features. A new classification model called Arc margin was developed by Xiaojun Wang et al. [6], which closely maps network traffic samples from the same category. They experimented on three datasets: non-vpn2016, CICIDS2017, and CICIDS2012, obtaining precision and recall values of 0.9857, 0.9853 on non-vpn2016, 0.9934, and 0.9933 on CICIDS2017, and 0.9971, and 0.9971 on the CICIDS2012 dataset.

A model based on LSTM and CNN for network traffic classification was proposed by Feifei Hu et al. [7] on the non-vpn dataset. They have obtained a precision, recall, and f1 score of 97.4, 97.5, and 96.8 respectively on the non-vpn dataset. M. Lopez-Martin et al. [8] have experimented and shown that usage of a single deep learning technique (such as CNN or RNN) compared to combinations of techniques such as (CNN+LSTM, RNN+LSTM) has more advantages in classifying network traffic.

Onur Barut et al. [9] have performed research on the importance of NTA in application classification and malware detection. They have generated three datasets, namely, NetML, CICIDS2017, and non-vpn2016, and implemented several machine learning algorithms like Random Forest, SVM, and MLP on these datasets. They have presented challenge baseline results on seven different tracks. However, they did not perform data balancing techniques. There is a scope to improve the performance of proposed ML models.

All of these related works have motivated us, and we were able to outperform the NetML Network Traffic Analytics Challenge 2020 baseline results and leaderboard participants organised by the ACANETS challenge [12].

### III. DATASETS

There have been a plethora of research attempts to analyze and classify network traffic using a variety of datasets. Nevertheless, with the open datasets we have in computer vision research, such as ImageNet and COCO, it is very difficult to find a comprehensive dataset for researchers in networking. However, in the recent work [9], to enable data-driven machine learning-based network flow analytics, they introduced a benchmark traffic dataset, known as NetML, curated from open sources for malware detection and network traffic classification. They have released the traffic flow features and different levels of annotations, aiming to

present a common dataset for the research community. This dataset consists of three different sets in which two data samples are prepared for malware detection, NetML and CICIDS 2017, and one dataset is for traffic classification, vpn2016.

#### Malware Detection Datasets: NetML & CICIDS2017

NetML and CICIDS2017 are the two datasets created with the raw traffic captured from the Stratosphere IPS [10] website and the Canadian Institute of Cybersecurity (CIC) [11], respectively. These datasets are captured for detecting malware. Both datasets were further divided into top and fine-grained annotations. In the top-level annotation of NetML [9] and CICIDS2017 [9] datasets, captured traffic data is classified as benign or malware, as shown in Table I and Table II. At the top level, if a packet is classified as malware, then in fine-grained annotation the type of malware is classified. In the NetML dataset, twenty different malware classes are there, such as Dridex, Trickster, Ursnif, etc., as shown in Table III. In the CICIDS 2017 dataset, there are seven different types of malware classes, such as portScan, DoS, infiltration, etc., that are classified in the fine-grained annotation as shown in Table IV.

TABLE I. CLASS DISTRIBUTION OF NETML - TOP DATASET

Class	Number of Samples
benign	311273
malware	75995

TABLE II. CLASS DISTRIBUTION OF NETML - FINE-GRAINED DATASET

Class	Number of Samples
benign	242661
malware	198455

TABLE III. CLASS DISTRIBUTION OF CICIDS2017 - TOP DATASET

Class	Adload	Artemis	Downware	CCleaner	Cobalt
Samples	75995	57796	30442	37271	31458
Class	PUA	Dridex	Emotet	HTBot	
Samples	8238	18627	15767	15289	
Class	TrickBot	Trickster	Ramnit	Sality	Tinba
Samples	4074	4020	8139	6162	4732
Class	BitCoinMiner		Trojan Downloader		Miner Trojan
Samples	45907		3849		8482
Class	MagicHound		WebComp anion	Ursnif	benign
Samples	9208		400	1379	33

TABLE IV. CLASS DISTRIBUTION OF CICIDS2017 - FINE DATASET

Class	DDoS	DoS	ftp-patator	infiltration
Samples	198455	122430	36136	23806
Class	portScan	ssh-patator	webAttack	benign
Samples	3168	1972	1617	53532

#### Traffic Classification Dataset: non-vpn2016

The non-vpn2016 dataset is collected from NetML Network Traffic Analytics Challenge 2020 organized by ACANETS [12]. This dataset primarily emphasizes application

classification. Three levels of annotation are assigned to this dataset in which top-level annotation groups the captured traffic data into seven classes, namely P2P, chat, audio, email, file\_transfer, tor, and video, as shown in Table V. Mid-level annotation consists of eighteen different applications (Gmail, Google, Netflix, Skype, Youtube, Facebook, etc..) as shown in Table VI. Fine-level annotation of thirty-one low-level classes in an application (skype\_video, facebook\_audio, tor\_google, tor\_twitter, etc..) as shown in Table VII. In the non-vpn2016, dataset four sets of features were extracted. They are Meta Data Features, TLS Features, DNS Features, and HTTP Features. Protocol-specific features are extracted only if the flow contains packets with any one of the protocols, whereas metadata features are extracted for any kind of flow.

TABLE V. CLASS DISTRIBUTION OF NON-VPN2016 - TOP DATASET

Class	P2P	audio	chat	email
Samples	992	121930	6772	4300
Class	file_transfer	tor	video	
Samples	25896	128	3693	

TABLE VI. CLASS DISTRIBUTION OF NON-VPN2016 - MID DATASET

Class	aim	email	facebook	ftps	gmail
Samples	1011	12477	106596	1995	1089
Class	google	hangouts	icq	netflix	scp
Samples	9	113940	1056	822	450
Class	sftp	skype	spotify	torrent	twitter
Samples	453	140133	546	2496	12
Class	vimeo	voipbuster	youtube		
Samples	1095	7047	1968		

#### IV. METHODOLOGY

In this work, we have divided our methodology into preprocessing and classification of algorithms. Data preprocessing is the process of applying transformations on raw data to clean the data. We apply various preprocessing techniques, such as scaling and balancing. We have scaled the data using the Standard Scaler. The datasets we experimented on have a high-class imbalance, as shown in Fig. [a, c, e, g, i, k, l]. Machine learning models may suffer from bias due to unbalanced data. Hence, to avoid bias problems, we have also experimented with balancing techniques as shown in Fig. [b, d, f, h, j]. These help in either upsampling or downsampling the number of samples in each class such that each class contains an equal number of samples. Techniques such as Undersampling, Oversampling, and SMOTE were experimented with. Of these, SMOTE gave better results.

In the next part, we have employed various Machine Learning algorithms; Random Forest, Support Vector Machine, Logistic Regression, Naïve Bayes, Adaboost, CatBoost, and XGBoost on

- unscaled and unbalanced dataset
- unscaled and balanced dataset
- scaled and unbalanced dataset
- scaled and balanced dataset

Out of all these, the performance results are good against scaled and balanced datasets.

TABLE VII. CLASS DISTRIBUTION OF NON-VPN2016 - FINE DATASET

Class	aim_chat	email	facebook_audio
Samples	346	4372	63349
Class	gmail_chat	hangouts_audio	hangouts_chat
Samples	415	38201	366
Class	ftps_up	skype_chat	scp_up
Samples	176	4880	84
Class	scp_down	sftp_down	sftp_up
Samples	89	98	67
Class	facebook_chat	skype_audio	facebook_video
Samples	433	17149	357
Class	icq_chat	hangouts_video	ftps_down
Samples	353	1231	535
Class	netflix	youtube	voipbuster
Samples	348	735	2754
Class	vimeo	torrent	tor_youtube
Samples	437	1016	104
Class	tor_vimeo	tor_twitter	tor_google
Samples	17	4	3
Class	tor_facebook	spotify	skype_video
Samples	3	207	584

#### V. RESULTS

The NetML and CICIDS2017 datasets are used for malware detection problems, while the nonvpn2016 dataset is used for traffic classification problems. For all the datasets, we have calculated validation accuracy based on 20% validation data as shown in Figure 1. Top Annotations of both NetML and CICIDS2017 datasets are for binary classification problems. Hence, we have calculated the True Positive Rate and False Alarm Rate for assessing the performance of the model. Fine annotation of the NetML and CICIDS2017 datasets, the non-vpn2016 dataset is used for traffic classification purposes, and the performance metrics F1 score and mAP score are evaluated for assessing the model performance.

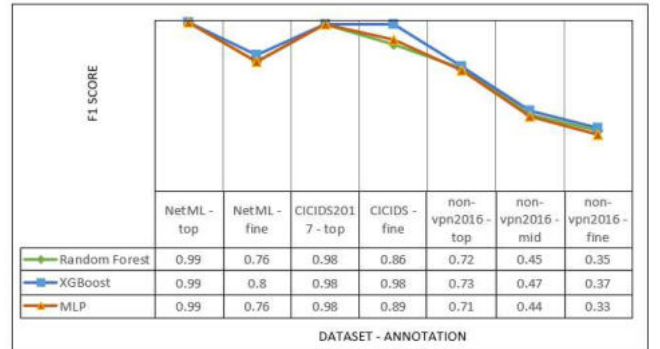


Fig. 1. validation accuracies on 20% dataset

For all the annotations of the three datasets, XGBOOST performed better than Random Forest and MLP.

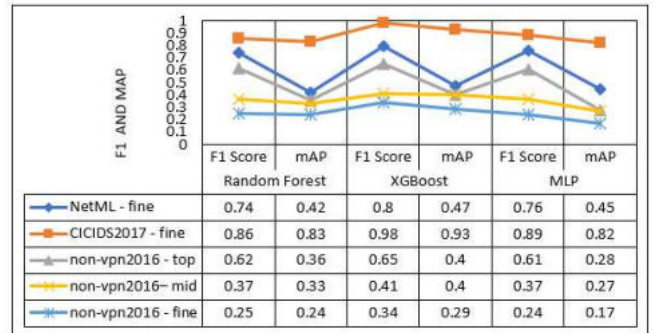


Fig. 2. F1 score and Mean Absolute Precision

As shown in Figure 2, the F1 score and mAP values of the fine annotation of NetML are 0.80 and 0.47, respectively, and those of CICIDS2017 are 0.98 and 0.93 respectively. The F1 score and mAP values of the top, mid, and fine annotations of the non-vpn2016 dataset are 0.65 and 0.40; 0.41 and 0.40; 0.34 and 0.29 respectively. XGBOOST performed better than Random Forest and MLP, for both F1 and mAP.

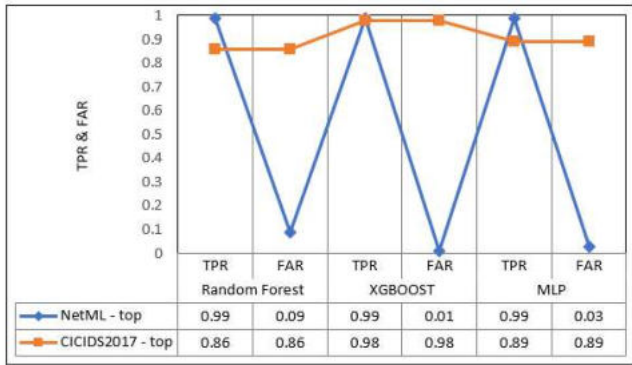


Fig. 3. TPR and FAR scores/values

In Figure 3, The TPR and FAR values of the top annotation of NetML are 0.99 and 0.01 respectively, and those of CICIDS2017 are 0.98 and 0.98 respectively. XGBOOST performed better than Random Forest and MLP on TPR and FAR.

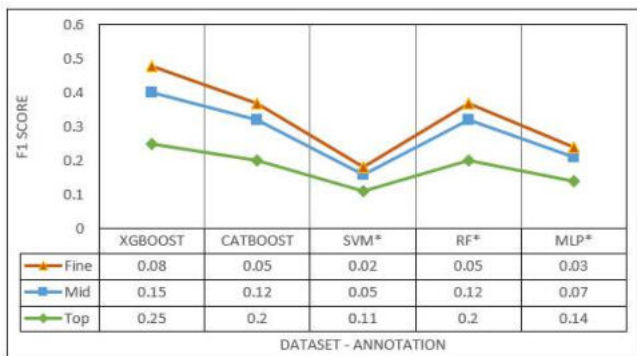


Fig. 4. Challenge Scores of the non-vpn2016 dataset

As shown in Figure 4, for all the annotations of the non-vpn2016 dataset, XGBOOST (top - 0.25, mid - 0.15, fine - 0.08) performed better than other algorithms. Our experimental results outperformed the ACANETS baseline results, as shown in Fig. [o, p, q].

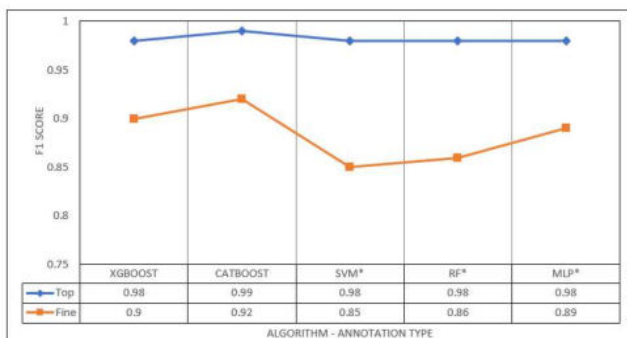


Fig. 5. Challenge Scores of the CICIDS2017 dataset

From Figure 5, it is emphasized that for all the annotations of the CICIDS2017 dataset, CATBOOST (top - 0.99, fine - 0.92) performed better than other algorithms. Our experimental results outperformed the ACANETS baseline results, as shown in Fig. [m, n].

## VI. CONCLUSION

Secure networks help in defending against malware attacks and safeguarding computer systems from data breaches. Detecting malware is required to protect the network from malicious activities and make networks secure. In this paper, we have performed malware detection on NetML, CICIDS2017, and non-vpn2016 datasets by applying machine learning algorithms. In this experiment, we applied various machine learning algorithms such as Random Forest, SVM, Naïve Bayes, XGBoost, CatBoost, and MLP. We have published our results in the NetML Network Traffic Analytics Challenge 2020, organized by ACANETS. The results are outperformed against the baseline results on the challenge leaderboard. Metrics calculated for evaluation of our work include FAR and TPR for binary classification, mAP, and F1 score for multi-class classification. As per the performance analysis, we found that XGBoost performed well for all annotations on the non-vpn2016 and NetML datasets. CatBoost performed well on CICIDS2017 - top and MLP performed well on fine annotation.

## REFERENCES

- [1] Ahmad, M. Basher, M. J. Iqbal and A. Rahim, "Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection," in IEEE Access, vol. 6, pp. 33789-33795, 2018, DOI: 10.1109/ACCESS.2018.2841987.
- [2] J. Jiang et al., "ALDD: A Hybrid Traffic-User Behavior Detection Method for Application Layer DDoS," 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), 2018, pp. 1565-1569, DOI: 10.1109/TrustCom/BigDataSE.2018.00225.
- [3] I. Ullah and Q. H. Mahmoud, "A Two-Level Hybrid Model for Anomalous Activity Detection in IoT Networks," 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC), 2019, pp. 1-6, DOI: 10.1109/CCNC.2019.8651782.
- [4] S. Ustebay, Z. Turgut and M. A. Aydin, "Intrusion Detection System with Recursive Feature Elimination by Using Random Forest and Deep Learning Classifier," 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT), 2018, pp. 71-76, DOI: 10.1109/IBIGDELFT.2018.8625318.
- [5] Rosay, Arnaud & Carlier, Florent & Leroux, Pascal. (2020). MLP4NIDS: An Efficient MLP-Based Network Intrusion Detection for CICIDS2017 Dataset, DOI:10.1007/978-3-030-45778-5\_16.
- [6] Mo, Chen & Xiaojuan, Wang & Mingshu, He & Lei, Jin & Javeed, Khalid & Wang, Xiaojun. (2020). A Network Traffic Classification Model Based on Metric Learning. Computers, Materials & Continua, DOI:10.32604/cmc.2020.09802.
- [7] Hu, Feifei & Zhang, Situo & Lin, Xubin & Wu, Liu & Liao, Niandong & Song, Yanqi. (2021). Network Traffic Classification Model Based on Attention Mechanism and Spatiotemporal Features. DOI: 10.21203/rs.3.rs-353938/v1.
- [8] Lopez-Martin, Manuel & Carro, Belén & Sanchez-Esguevillas, Antonio & Lloret, Jaime. (2017). Network Traffic Classifier With Convolutional and Recurrent Neural Networks for Internet of Things. IEEE Access. DOI:10.1109/ACCESS.2017.2747560.
- [9] NetML: A Challenge for Network Traffic Analysis <https://arxiv.org/abs/2004.13006>
- [10] Stratosphere. 2015. Stratosphere Laboratory Datasets. (2015). <https://www.stratosphereips.org/datasets-overview> [Online; accessed 12-March-2020].

- [11] Iman Sharafaldin, Arash Habibi Lashkari, and Ali Ghorbani. 2018. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. 108–116.
- [12] NetML Network Traffic Analytics Challenge 2020: <https://eval.ai/web/challenges/challenge-page/526/overview> ; Leaderboard Results: non-vpn2016 Dataset : Top annotation : <https://eval.ai/web/challenges/challenge-page/526/leaderboard/1471> ; Mid annotation: <https://eval.ai/web/challenges/challenge-page/526/leaderboard/1472> ; Fine annotation <https://eval.ai/web/challenges/challenge-page/526/leaderboard/1473> ; CICIDS2017 Dataset : Top annotation: <https://eval.ai/web/challenges/challenge-page/526/leaderboard/1474>; Fine annotation : <https://eval.ai/web/challenges/challenge-page/526/leaderboard/1475>

APPENDIX

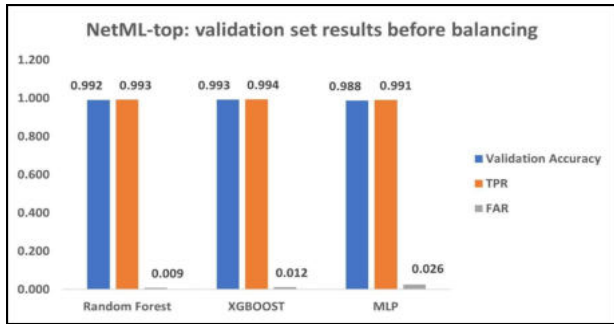


Fig. a. NetML-top: validation set results before balancing

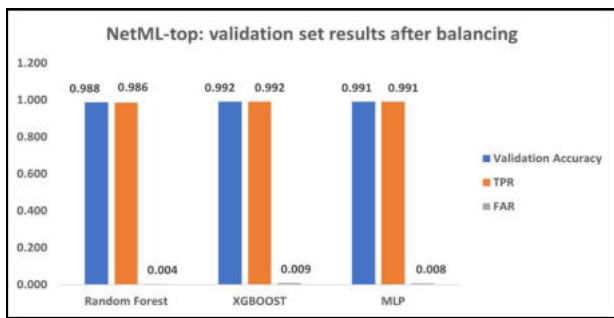


Fig. b. NetML-top: validation set results after balancing

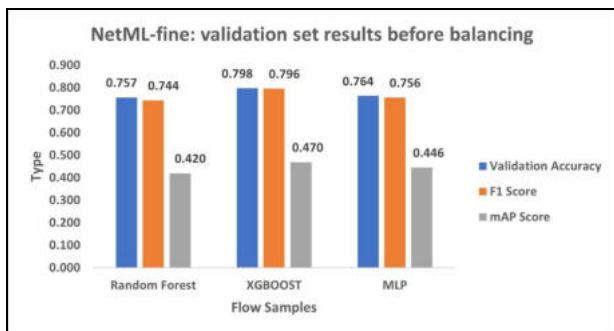


Fig. c. NetML-top: validation set results before balancing

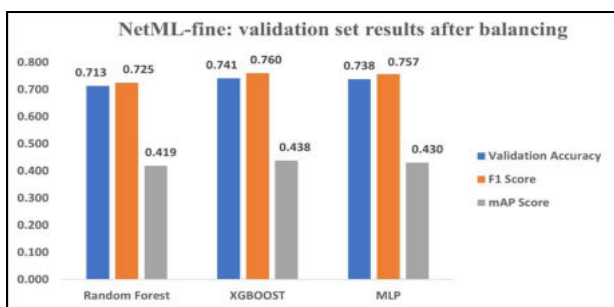


Fig. d. NetML-fine: validation set results after balancing

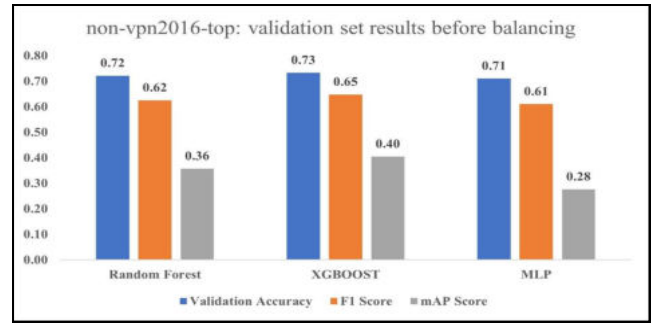


Fig. e. non-vpn2016-top: validation set results before balancing

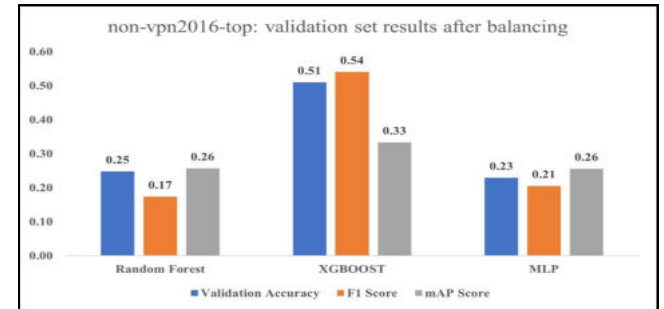


Fig. f. non-vpn2016-top: validation set results after balancing

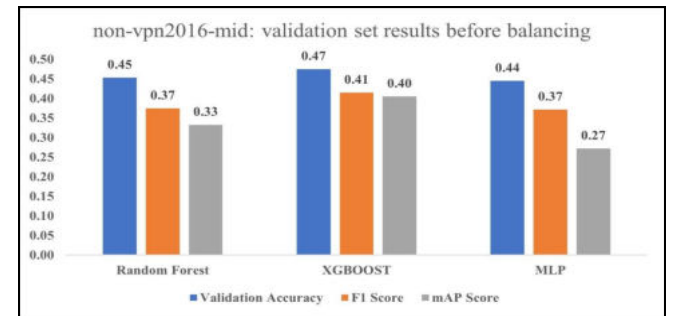


Fig. g. non-vpn2016-mid: validation set results before balancing

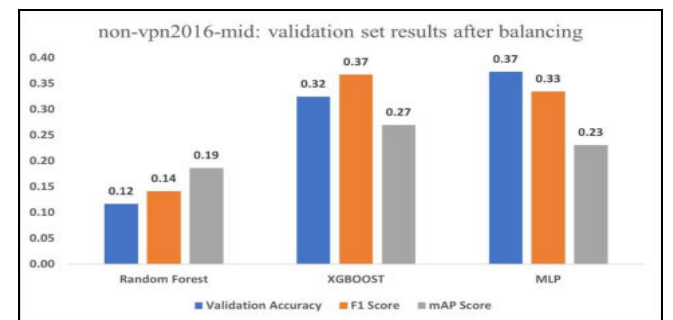


Fig. h. non-vpn2016-mid: validation set results after balancing

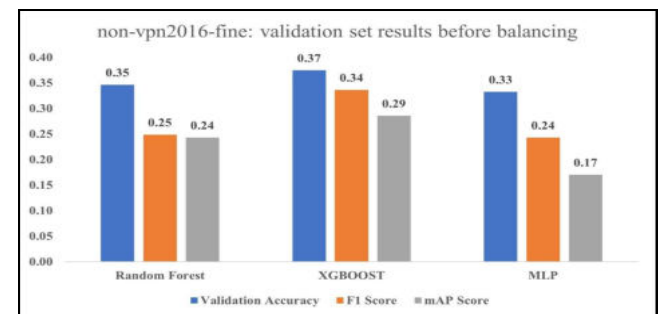


Fig. i. non-vpn2016-fine: validation set results before balancing



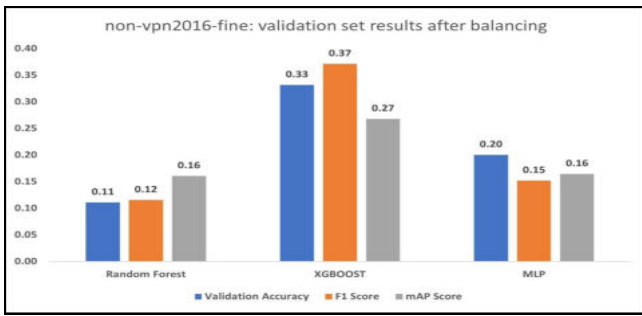


Fig. j. non-vpn2016-fine: validation set results after balancing

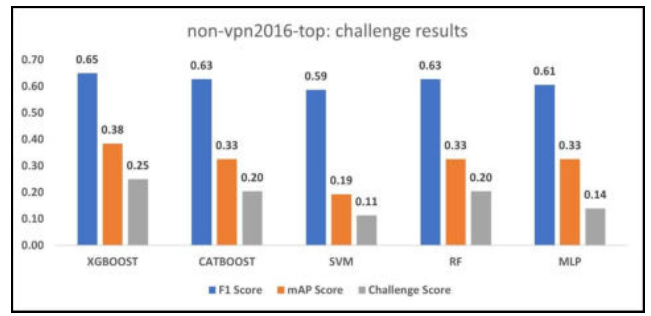


Fig. o. non-vpn2016-top: challenge results

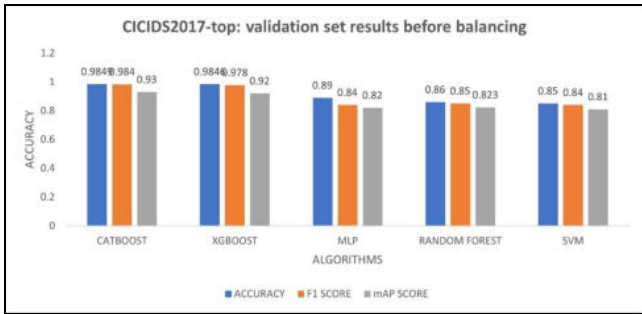


Fig. k. CICIDS2017-top: validation set results before balancing

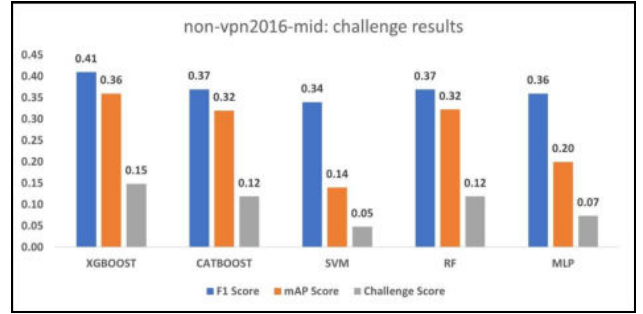


Fig. p. non-vpn2016-mid: challenge results

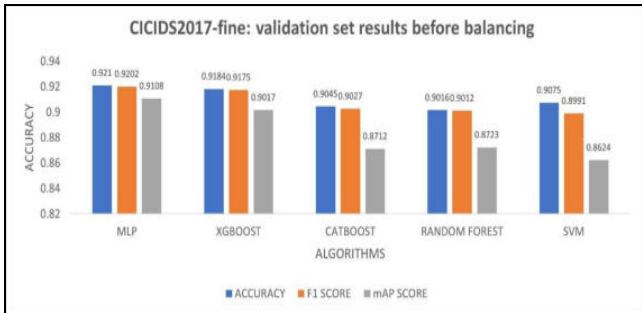


Fig. l. CICIDS2017-fine: validation set results before balancing

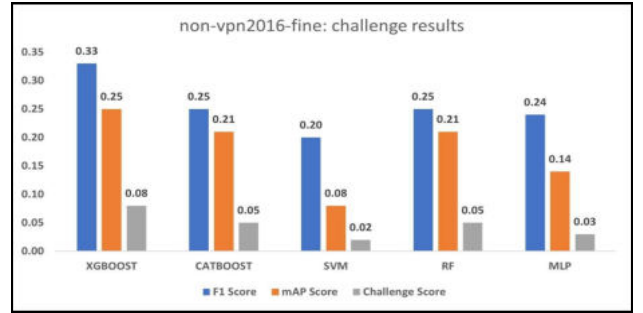


Fig. q. non-vpn2016-fine: challenge results

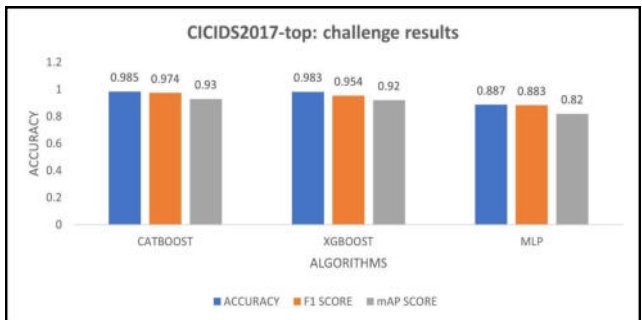


Fig. m. CICIDS2017-top: validation set results after balancing

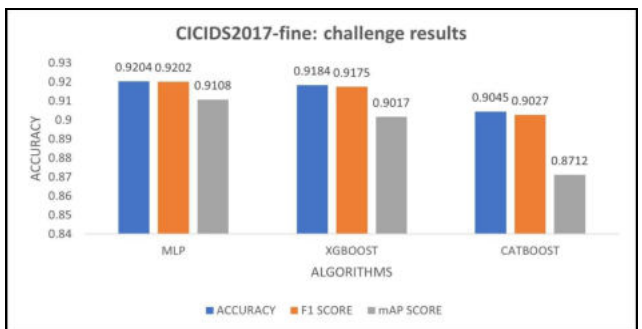


Fig. n. CICIDS2017-top: validation set results after balancing

# Defect Information Synthesis via Latent Mapping Adversarial Networks

Seunghwan Song

Department of Industrial and Management Engineering  
Korea University  
Seoul, South Korea  
ss-hwan@korea.ac.kr

Jun-Geol Baek\*

Department of Industrial and Management Engineering  
Korea University  
Seoul, South Korea  
jungeol@korea.ac.kr

**Abstract**— This research presents a new image synthesis methodology for automated visual inspection (AVI) in steel manufacturing process. We develop a novel methodology, termed Latent Mapping Adversarial Networks. As the end product of the manufacturing process is directly linked to economic factors, various methods are being utilized to improve the quality of the product. Among them, the defect detection steps carried out in advance are important as it greatly impacts productivity. However, new challenges have emerged for several reasons. First, it requires prior knowledge of the expert to define the defect image and perform detection. To alleviate this problem, various companies have started utilizing AVI to reduce this dependence on domain knowledge. Secondly, defect detection is an arduous task since fewer defect images are available compared to normal images. This underlying problem leads to a classification model that is biased toward the majority class, which degrades the final performance. In this paper, we propose a method to synthesize defect images to solve the above-mentioned problems. Inspired by StyleGAN, we build mapping networks for latent space of the generator. Through this, we can synthesize defect images of various sizes in the manufacturing process. In addition, we experiment to find the most suitable loss function to solve the common problems of Generative Adversarial Networks (GAN). We also optimized the proposed method in terms of convergence and computation speed by estimating the size of optimal latent space. The experimental results using quantitative metrics illustrate the improved performance of the proposed methodology. As a result, it is now possible to solve the quality problem and increase productivity by reducing misclassification in the model through AVI experiments using the generated images

**Keywords**—Automated visual inspection, generative adversarial networks, latent mapping, mapping network, synthesize defect

## I. INTRODUCTION

Manufacturing process technologies are becoming increasingly fragmented and complex. The end product of the manufacturing process is directly related to economic factors as it affects productivity. Therefore, if defects in the product are not detected in advance, the cost of processing defective products occurs, affecting the entire manufacturing process [1]. Recently, there is a continuous rise in the demand for improvement in surface and shape quality of steel products [2]. Among them, detecting defects in advance to control defects in manufacturing is essential as it directly affects productivity and business competitiveness.

Defect refers to physical and chemical failures caused due to certain problems in the manufacturing process, facility, or manufacturing environment. Steel is manufactured through various processes such as rolling and forging. In this process, defects such as crazing, inclusion, pitted surface, rolled-in scale, and scratch occur as shown in Fig. 1 [3].

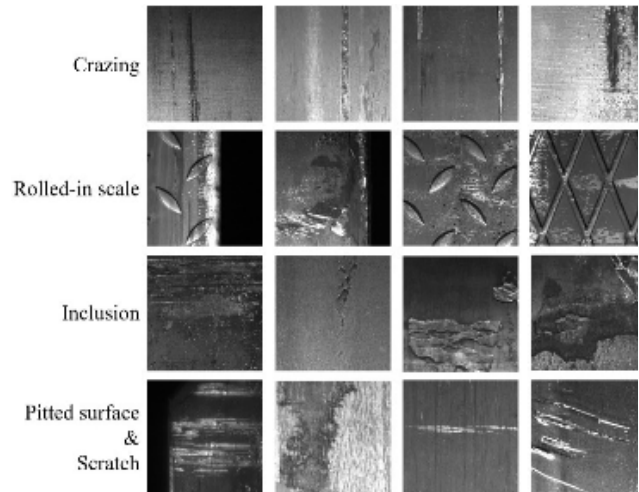


Fig. 1. Types of defects in the steel manufacturing process.

Defect detection for steel surfaces is an important step in ensuring the quality of industrial production. Steel surface defect detection undergoes 3 preliminary steps as shown in Fig. 2. First is the inspection step: Through this step, defects on the steel surface are detected by inspection tools [4]. Second is the review step: In this step, images of detected defects are captured by a specific tool. Third is the detection step: Detecting and classifying the types of defects according to the captured images. Steel surface defect detection processes allow the engineer to perform cause analysis and defect control. However, this visual inspection requires great reliance on the experience and the ability of individual engineers. Additionally, this process is usually done manually in the industry, making it unreliable and time-consuming. Therefore, automated visual inspection (AVI) targeting surface quality emerges as a standard configuration for steel manufacturing mills to improve product quality and promote production efficiency[5]. AVI, which performs classification through Convolutional Neural Networks (CNN), is not only widely applied to steel manufacturing process, but to glass, fiber, and semiconductor production process as well [6].

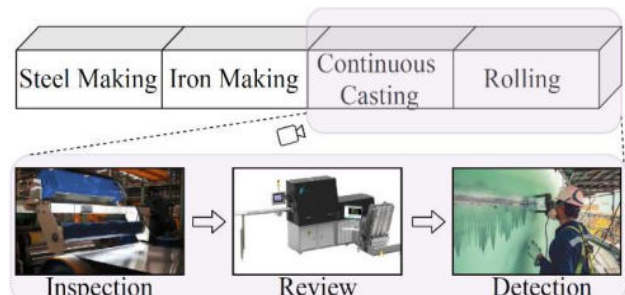


Fig. 2. Steel manufacturing process and defect detection steps.

\* Corresponding author--Tel: +82-2-3290-3396; Fax: +82-2-3290-4550

Although the deep learning-based AVI model shows excellent classification performance for numerous defect types, it inherits two practical problems in steel manufacturing process. First, the frequency of defect data occurrence is extremely low that very little data exist to be used for deep learning model development [7]. In general, enough training data for both defect and normal class is required to improve the classification performance of a deep learning model [8]. However, in the actual industry, the number of defective data is minimal compared to that of normal data. Were if the AVI was conducted with collected data alone, the imbalance may lead to the lower learning rate of defect types and to degradation of performance. Therefore, it is necessary to balance normal and defective class. Class imbalance refers to the substantial proportional difference of each class in the total dataset. When the class distribution is unbalanced, the model is trained with a bias toward the majority class, classifying well for the class with a lot of data, but the opposite for the minority class. Furthermore, the imbalanced class distribution can also lead to serious type II errors. Therefore, preprocessing for class imbalance is essential in improving the overall classification performance in defect detection.

Second, the steel defect data has consisted of defects of various sizes. When using a generative model to solve the imbalance problem, large-sized defects can be easily generated with a simple generator. However, generation of small-sized defects is greatly influenced by which type of generative model is used [5]. Especially, a sophisticated generator is essential in situations where the defect size of the final product is about 0.2 mm, such as in the cold rolling process [9]. This study proposes a novel deep learning model for synthesizing defect data in steel manufacturing process. The proposed method generates a defective image similar to that of the real one, constructing effective training data for detecting detailed defective patterns.

In this study, we propose latent mapping adversarial networks to overcome two practical problems in steel manufacturing process. Our methodology is inspired by Style-based Generative Adversarial Networks (StyleGAN), which exhibits state-of-the-art in the data generation field [10]. The proposed method uses mapping networks in the latent space of the generator network. As the latent space goes through the mapping network, it becomes possible to learn the disentanglement of the training data distribution. This is the first step in the direction of explicit learning on real data. Mapping networks allow for a sophisticated generation of small-size defects. Our methodology also cares about the stability of learning. We use the Wasserstein distance as a distribution distance metric instead of the Jensen-Shannon (JS) divergence. The Wasserstein distance solves problems such as vanishing gradient and mode collapse witnessed in vanilla GAN [11]. The advantages of using the Wasserstein distance are discussed in section III.

To demonstrate the performance of the proposed method, images of flat steel plates are used in the production process. The data generation aspect of the proposed method was first evaluated by the quantitative evaluation metric, Fréchet Inception Distance (FID), and visual results. Also, the second evaluation is performed on the classification performance through a simple CNN structure [12]. Finally, to reduce the computational cost, we performed a task to find the optimal latent space and mapping network size for the data used in the experiment.

In summary, our contributions are as follows:

- Solve the data imbalance problem by generating steel defect data through the proposed method. Based on the quantitative evaluation metric, visual results, and classification results, we confirm the excellent results of the generation model.
- Set up the optimal potential space and mapping network to achieve the highest efficiency in the optimal time.

The rest of this paper is organized as follows. In Section II, we introduce previous works on AVI. Also we take a look at the background of our study and review some previous work. In Section III, we describe the proposed methodology. In Section IV, the performance of the steel surface defect dataset is evaluated using the proposed method. Finally, conclusions based on the experimental results and directions for future research in Section V.

## II. RELATED WORK

In Section I, two problems needed to be solved by this study were shown. This section deals with the underlying class imbalance problem. Numerous solutions which have been proposed to solve class imbalance problems in AVI can be divided into two methods, a method of correcting the model itself and a method of directly processing the data [13]. In the former, similar to active learning or kernel-based methods, data instances of different classes are treated differently. As of the latter, the direct processing of data utilizes methods such as sampling or data generation to directly control the number of instances.

Sampling is a method to correct the bias between classes in data with an overwhelmingly small proportion of abnormal data compared to normal data. Representative methods to deal with class imbalance are oversampling and undersampling. Oversampling is a method of creating new data of the minority class in order to even the class ratio, and undersampling is a method of removing existent data of the majority class to match the ratio. As undersampling reduces the number of sample data from the majority class, it has the advantage of reducing model training time. However, it also has the chance to distort data features by removing crucial information. As for oversampling, the risk of data distortion is relatively small due that it creates new data while preserving the original data information. The oversampling methods mainly used for AVI include random oversampling, synthetic minority oversampling technique (SMOTE) [14], and adaptive synthetic sampling approach (ADASYN) [15]. Random oversampling increases the number of minority class data by randomly selecting and replicating a sample from the minority class. SMOTE synthesizes data by selecting random data belonging to a minority class and randomly selecting among the closest top  $k$  number of data. ADASYN is a method of adaptively synthesizing  $k$  number of data from marginal minority data according to the number of majority classes after calculating the ratio of data of a majority class. This oversampling method for image data has the problem of generating an image with a low resolution.

To simplify this problem, we used the technique of handling the raw image itself. This method is called data augmentation and is commonly being used as model regularization technique in recent studies [16]. Some common

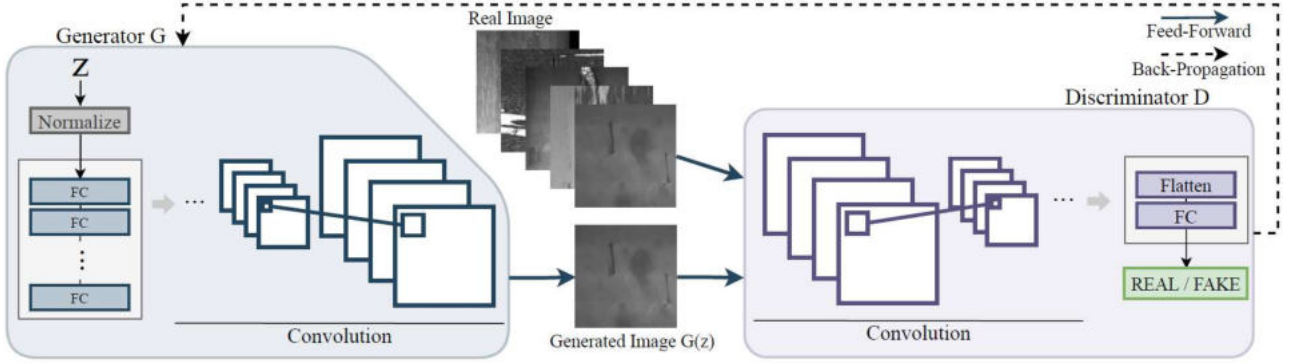


Fig. 3. Our latent mapping adversarial networks framework in manufacturing defect synthesis.

augmentation methods include flipping the image vertically or horizontally, shifting the image vertically or horizontally, and slightly rotating or zooming the image. This method helps the training model to be robust to small changes in the image. However, simple geometric transformations do not significantly change the characteristics of the image, making it impossible to identify more features in the image.

Among data generation methods, GAN is an algorithm that is of great interest [17]. GAN generates data based on distribution and mainly exhibits excellent performance in image generation. Some prior researched GAN are as follows: There was a ball-bearing failure detection method through Deep Convolutional GAN (DCGAN) [18]. In addition, a wafer defect image was adaptively generated using Conditional GAN (CGAN) [19]. Progressive Growing of GAN (PGGAN) increased the model training speed by gradually increasing the generator and discriminator and produced a good quality image [20]. In addition, to address the shortcomings such as vanishing gradient or mode collapse of GAN, Wasserstein GAN (WGAN) has been proposed [21]. This study compares and applies existing GAN-based generation models whose data generation performance has already been verified and finds an optimal generation model suitable for field data application. The focus of the proposed method generates effective learning data for detecting detailed defects.

### III. LATENT MAPPING ADVERSARIAL NETWORKS

This section describes the framework of the latent mapping adversarial networks, an approach to solving the imbalance problem for defect images. Fig. 3 is a schematic diagram of the overall structure of the proposed method. GAN is a neural network in which the generator and the discriminator learn adversarial to each other. The generator is trained to generate an image that is similar to the real image, while the discriminator is trained to discriminate between the real image and the generated image. The components of the proposed method are as follows: (1) Generator: Improved the quality of data generation by adopting mapping network structure for latent space. (2) Discriminator and Loss Function: By using Wasserstein distance with gradient penalty applied, addressed the imbalanced loss function problem occurring when the discriminator is backpropagated. Mapping network for latent space is discussed in Section III. A. Similarly, the imbalanced loss function is discussed in Section III. B.

#### A. Mapping Network for Latent Space

Defects occurring on the steel surface have a significant influence on the quality of the final steel product. Thus, it is

crucial to correctly detect defects to ensure the quality of the final product and prevent the delivery of defective products to customers. However, as a result of imbalance in the steel surface defect data, the increase in misclassification of such data leads to deterioration of the classification performance. Therefore, there exists a need for an oversampling method that generates defect data.

In this study, the mapping network structure for the latent space was used to improve the quality of the generated data. The latent space of a well-trained GAN model has linear subspaces which permit direct variation adjustment[10]. However, direct control of latent space  $z$  is impossible since  $z$  of a vanilla GAN tends to form the training data into a single Gaussian distribution. The mapping network overcomes this problem by impeding the latent space  $z$  from entering the generator as an input value. Instead, we input  $w$  passed through the mapping network as input value to the generator. While latent space  $z$  cannot accurately match the feature distribution of the training data,  $w$  can because it undergoes a nonlinear transformation through the mapping network. Therefore, the disentanglement characteristic of  $w$ , suited for the train data, leads to improved data generation. In summary, in the structure of the vanilla GAN model generator, the latent space is passed through the mapping network composed of fully connected layers.

This approach may seem very simplistic. To learn the distribution of data, we used GAN. When we generate a noise vector and put it as an input to the GAN, we can generate random images similar to our training data but not present in the training data. However, it is not easy to create a random image with the desired characteristics. The reason we get this result is that  $z$  is all related to any other feature. One of the reasons why an axis is entangled usually happens when the degree of  $z$  is not sufficient. The mapping network makes the axes disentangle by making the  $z$  degree sufficient. Therefore, it achieves a performance improvement in terms of generative for the training data.

#### B. Imbalanced Loss Function

Existing oversampling methods do not use data distribution. Additionally, the GAN problems, vanishing gradient and mode collapse, have a detrimental effect on the quality of the generated data [21]. Vanishing gradient refers to a problem that occurs when the discriminator learns to perfection, as in (1). If the discriminator  $D$  is perfect, the loss function of GAN will approach zero and the gradient will not be obtained in the learning process.

$$D(x) = \mathbf{1}^v x \in p_r, D(x) = \mathbf{0}^v x \in p_g \quad (1)$$

In Equation (1),  $p_r$  is denotes the distribution of the real data, and  $p_g$  denotes the distribution of the generated data.

Mode collapse, another characteristic problem of GAN, is when the generator always outputs the same result during the learning process due to GAN using JS divergence as a distance metric. In this study, 1-Wasserstein is used as the distance metric instead of the JS divergence to deviate from the problems of gradient loss and mode collapse. However, the 1-Wasserstein distance has a problem where the weight is clipped. Also, gradient penalty (GP) technique is used to solve the weight clipping problem of 1-Wasserstein [11]. Therefore, the imbalance loss function, WGAN-GP, is expressed as (2), and it is learned in the direction of minimizing this constraint.

$$L = \mathbb{E}_{x \sim p_r}[D(x)] - \mathbb{E}_{z \sim p_g}[D(z)] + \lambda \mathbb{E}_{\hat{x} \sim \mathbb{P}_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2] \quad (2)$$

where  $\hat{x} = tx + (1 - t)z$  with  $0 \leq t \leq 1$

In Equation (2),  $x$  denotes actual data and  $z$  denotes data generated in the latent space. The remainder of (2) denotes the part for the gradient  $\nabla$  of the discriminator  $D$  with  $\hat{x}$  uniformly sampled between  $x$  and  $z$  at the ratio of  $t$ . When the L2 Regularization (L2 Norm) of this gradient has a value other than 1, it is optimized by giving a penalty as much as  $\lambda$ . Consequently, by manipulating the loss function to have a meaningful value when the two distributions do not overlap in a low-dimensional manifold can solve the loss of slope and mode collapse problems.

#### IV. EXPERIMENTAL RESULTS AND DISCUSSION

All experiments are performed using the Pytorch software package and Scikits Learn(Sklearn), Panda library, together with Python3 language, running on a Desktop with Intel(R) Core(TM) i7-9700K CPU 3.60GHz, 32GB RAM with NVIDIA GeForce RTX 3080 10GB. For comparative purposes, we also implement some of the other leading GANs using Pytorch.

##### A. Datasets

The data used for performance verification in this study is acquired from Severstal: steel manufacturing process. This data is collected by a high-frequency camera capturing images of flat sheet steel during the production process. This dataset is usually subjected to defect location and type prediction found in steel manufacturing. The dataset contains a single class of defect type data, multiple class of defect type data, and non-defect type data. Fig. 4 shows an example of the data used in the experiment. Steel defect data is a schematic diagram of each class of defect data consisting of tiny defects to large defects. In this study, image data of  $256 \times 1600$  was cropped into a square image of size  $256 \times 256$ , tailored to be utilized as an input value.

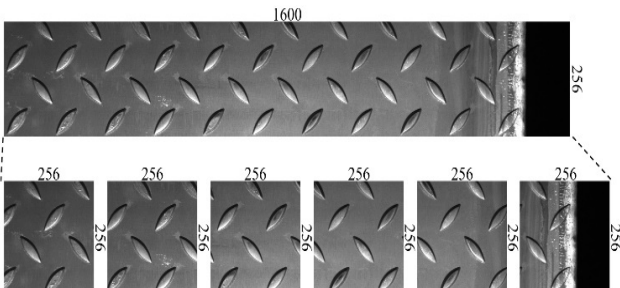


Fig. 4. Example of the cropping steel defect images ( $256 \times 1600$  to  $256 \times 256$ ).

##### B. Experimental Design

The preprocessed dataset was partitioned at a ratio of  $Data_{train}:Data_{test} = 7:3$ . The experiment consists of two main steps. The role of the first stage experiment is to verify the generator model of the proposed method. The superior performance of the proposed method is demonstrated in comparison with other GAN-based generator models. Each GAN layer was uniformly composed of 5 layers, and 100 dimensions were used for the latent space. For an optimization function, RMSProp, which is frequently used in the GAN model, was used.

The second stage experiment finds the optimal latent space size and mapping network. By structuring part of the proposed method with mapping network, the proposed method was able to acquire disentanglement features. Through the proposed model, the optimal size of the initial latent space and the mapping network were experimentally discovered. All experiments were evaluated by the quantitative evaluation metric FID. During data division, the seed was changed and the average value of the ten performed results was used as the final metric.

##### C. Performance Measurement Metric

Manufacturing data is primarily comprised of normal data. However, in many cases, abnormal data is more critical in defect control than normal data. This imbalance becomes a problem as it leads to an increase in the misclassification error rate of abnormal data, consequently degrading overall classification performance. In this study, the oversampling method, a method of randomly generating abnormal data using the GAN model, is used to solve the imbalance of abnormal data.

Early GAN was accompanied by problems such as instability of learning and mode collapse, exhibiting difficulties in performance evaluation [17]. To address such problems, the development of various GAN models on top of inception scores (IS) and FID using the inception model, have made evaluating the performance of GAN possible [12]. The inception model, widely used for transfer learning and fine-tuning, is a CNN model that pre-trained ImageNet data. ImageNet consists of 1,000 class and 1.2 million images. When an image is inputted into the model, the inception model outputs probability vectors belonging to each 1,000 class. Using the generated image as an input value to the inception model, it can calculate the IS as shown in (3).

$$Inception\ Score = \exp(\mathbb{E}_{z \sim p_g} KL(p(y|z) || p(y))) \quad (3)$$

In Equation (3),  $p(y|z)$  is the conditional class distribution and  $p(y)$  is the marginal class distribution. The inception score can have a value between 1 or more and 1,000 or less but is usually around 2. However, the inception score encompasses the disadvantage of not using the real data distribution. In this study, the shortcomings of the IS are overcome by FID which is a measure of the difference between the two normal distributions, as shown below.

$$FID = \|m - m_w\|_2^2 + Tr(C + C_w - 2(CC_w)^{1/2}) \quad (4)$$

Smaller FID means better quality, and  $(m, C)$  and  $(m_w, C_w)$  denotes the mean and covariance of the distribution between the generated image and the real image. As it is

widely accepted that FID captures the quality of generated data better than IS, this study adopts FID as a measure to assess the quality of the generated image.

#### D. Experimental Results

1) *Performance Compared to Generative Model:* The proposed method differentiates by using mapping network structure and imbalanced loss function (WGAN-GP) to improve the quality of the data. The latent space used in previous GAN models displayed difficulties in avoiding entanglement due to its tendency to follow the probability density of the training data. However, we use a mapping network to solve this problem and exhibit the disentanglement of latent space. Thus, making direct adjustments to changes possible.

Table I shows the results of comparing the proposed method with vanilla GAN, DCGAN, and DCGAN+WGAN-GP. Baseline is vanilla GAN, and DCGAN which deep convolutional structure is added to the baseline. DCGAN+WGAN-GP is the loss function of DCGAN changed to WGAN-GP. The proposed method adds mapping network composed of 8 fully connected layers to the previous methods. The reason for constructing the control group as follows is to check the effect of each method briefly. This also shows the gradual evolution of the GAN-base model.

TABLE I. COMPARISON AVERAGE FID OF GENERATIVE MODELS

Method	FID
Baseline (GAN)	105.17
DCGAN	31.47
DCGAN+WGAN-GP	26.64
Proposed Method	<b>15.81</b>

By Table I, it was confirmed that the proposed method showed excellent performance in terms of average FID. As each method was sequentially added, it was possible to confirm that the FID improved sequentially as well. Fig. 5 is a visual result of the comparison of the real image with the generation results of each method.

As a result of comparing the creation results of each method as a 5x5 matrix, it is difficult to recognize a large difference when visually confirmed. As shown in Table 2, we applied the generated data to the classification task. To do this task, we used a simple fully convolutional network (FCN) [22]. We train FCN algorithm on the generated samples and test the accuracy on the real image. By the classification accuracy, the proposed method generates similarly to the real image in terms of creation.

TABLE II. CLASSIFICATION ACCURACY USING FCN FOR THE RESULTS OF GENERATIVE MODELS.

Method	Classification accuracy
Baseline (GAN)	74
DCGAN	83
DCGAN+WGAN-GP	89
Proposed Method	92

2) *Optimal Latent Space and Mapping Network Size:* The proposed method improved the quality of image generation by adopting the mapping network structure. By doing so, optimization of latent space, where random vectors that

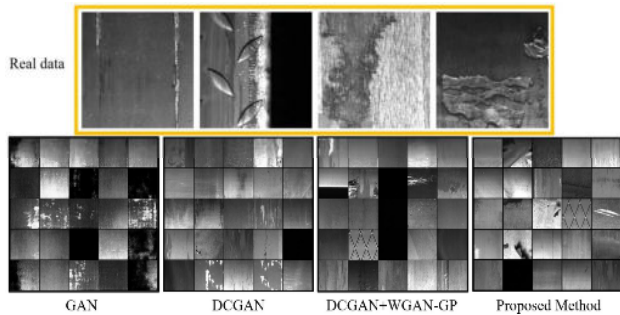


Fig. 5. Visual comparison of the results of each method.

generate images similar to that of real images, was possible. Generally, a sufficiently large latent space can adequately express the characteristics of the real data, leading to the use of 100-dimensional size for general latent space. However, adopting image generation for the steel manufacturing process requires accurate and expeditious processing. Thus, there is a need for image generation that performs well even with a simple structure. As the size of a latent space directly affects the number of parameters, the convergence speed, and computation time, finding the optimal size of the latent space is a necessary task.

In this experiment, we strive to find the optimal latent space size as well as the mapping network. Table 2 shows the results of the experiment where adjustments of mapping network to 0, 2, 4, and 8, with corresponding adjustments to the dimension size of the latent space to 1, 2, 3, 10, 50, and 100 were made.

In Table 3, ‘traditional’ exhibits the result of using latent space of the general GAN without mapping network, and ‘style-based’ indicates the number of mapping networks used. The evaluation is done through the FID, where it is known that the lower the FID, the more similar generated data is to the real data. Up to the 10th dimension, the performance shows a tendency to improve, while thereafter, performance varies with only the slightest difference. Therefore, it can be confirmed that there is not a significant performance difference between conventionally used 100 dimensions and dimensions after the 10 dimensions. Also, the mapping network shows the best performance when it is composed of 8 fully connected layers. Latent space and mapping network size are closely related to computation time. Thus, to accommodate for the need for accurate and expeditious processing characteristics of the steel manufacturing process, the proposed method has consisted of 50 latent spaces and 8 mapping networks. As a result, it was confirmed that the proposed method generates high-quality images.

#### V. CONCLUSION

This study proposed a method to tackle the imbalance that exists in defect detection in the steel manufacturing process. It improved the quality of the generated image by adopting mapping network. At the same time experimented to find the optimal latent space size and mapping network to achieve accurate and expeditious processing in the process. The quality of the generated images was evaluated using quantitative metric FID and visual results, and classification performance.

The method proposed in this study is applicable to AVI problems in the various manufacturing process, especially

TABLE III. COMPARISON AVERAGE FID OF OPTIMAL LATENT SPACE AND MAPPING NETWORK SIZE

Mapping network	Latent space	FID	Mapping network	Latent space	FID
Traditional	1	198.85	Style-based 4	1	139.14
	2	165.11		2	49.52
	3	69.18		3	29.48
	10	43.57		10	7.97
	50	29.21		50	7.22
	100	17.42		100	7.21
Style-based 2	1	161.59	Style-based 8	1	136.32
	2	62.87		2	49.19
	3	32.24		3	25.35
	10	11.87		10	7.46
	50	9.65		50	<b>6.94</b>
	100	7.86		100	<b>7.16</b>

processes with innate imbalance problems and the practicality of the proposed method also makes it highly applicable to various fields other than AVI. For future research, by de-riving image quality evaluation indicators suitable for manufacturing data we will seek to improve the overall quality of the proposed method.

#### ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2019R1A2C2005949, NRF-2021R1A6A3A13045200). Also, this work was supported by Samsung Electronics Co., Ltd (IO201210-07929-01).

#### REFERENCES

- [1] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee, and B. Yin, "Smart factory of industry 4.0: Key technologies, application case, and challenges," *IEEE Access*, 6, 6505-6519, 2017.
- [2] F. Akhyar, C. Y. Lin, K. Muchtar, T. Y. Wu, and H. F. Ng, "High efficient single-stage steel surface defect detection," *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1-4, 2019.
- [3] N. Neogi, D. K. Mohanta, and P. K. Dutta, "Review of vision-based steel surface inspection systems," *EURASIP Journal on Image and Video Processing*, pp. 1-19, 2014.
- [4] J. Wang, Z. Yang, J. Zhang, Q. Zhang, and W. T. K. Chien, "AdaBalGAN: An improved generative adversarial network with imbalanced learning for wafer defective pattern recognition," *IEEE Transactions on Semiconductor Manufacturing*, 32(3), 310-319, 2019.
- [5] Q. Luo, X. Fang, L. Liu, C. Yang, and Y. Sun, "Automated visual defect detection for flat steel surface: A survey," *IEEE Transactions on Instrumentation and Measurement*, 69(3), 626-644, 2020.
- [6] S. Cheon, H. Lee, C. O. Kim, and S. H. Lee, "Convolutional neural network for wafer surface defect classification and the detection of unknown defect class," *IEEE Transactions on Semiconductor Manufacturing*, 32(2), 163-170, 2019.
- [7] Z. J. Xu, Z. Zheng, and X. Q. Gao, "Operation optimization of the steel manufacturing process: A brief review," *International Journal of Minerals, Metallurgy and Materials*, 2021.
- [8] E. Zhang, B. Li, P. Li, and Y. Chen, "A deep learning based printing defect classification method with imbalanced samples," *Symmetry*, 11(12), 1440, 2019.
- [9] D. Kang, Y. J. Jang, and S. Won, "Development of an inspection system for planar steel surface using multispectral photometric stereo," *Optical Engineering*, 52(3), 039701, 2013.
- [10] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401-4410, 2019.
- [11] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein gans," *arXiv preprint arXiv:1704.0028*, 2017.
- [12] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," *arXiv preprint arXiv:1809.11096*, 2018.
- [13] N. Kondo, M. Harada, and Y. Takagi, "Efficient training for automatic defect classification by image augmentation," *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 226-233, 2018.
- [14] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, 16, 321-357, 2002.
- [15] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," *IEEE international joint conference on neural networks*, pp. 1322-1328, 2008.
- [16] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation strategies from data," *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 113-123, 2019.
- [17] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, 27, pp. 2672-2680, 2014.
- [18] J. Viola, Y. Chen, and J. Wang, "FaultFace: Deep convolutional generative adversarial network (DCGAN) based ball-bearing failure detection method," *Information Sciences*, 542, pp. 195-211, 2021.
- [19] M. Mirza, and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.
- [20] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017.
- [21] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," *In International conference on machine learning*, p. 214-223, 2017.
- [22] K. E. Smith, and A. O. Smith, "Conditional GAN for timeseries generation," *arXiv preprint arXiv:2006.16477*, 2020.

# FFDNet Based Channel Estimation for Multiuser Massive MIMO System with One-Bit ADCs

Md. Habibur Rahman, *Student Member, IEEE*, Md. Shahjalal, *Student Member, IEEE*, Md. Osman Ali, Byung Deok Chung\*, and Yeong Min Jang, *Member, IEEE*

Department of Electronics Engineering, Kookmin University, Seoul 02707, Korea

\*ENS. Co. Ltd, Ansan 15655, Korea

Email: rahman.habibur@ieee.org; mdshahjalal26@ieee.org; osman@kookmin.ac.kr; bdchung@ens-km.co.kr; yjang@kookmin.ac.kr

**Abstract**—Low-resolution analog-to-digital converters (ADCs) in massive multiple-input multiple-output (m-MIMO) systems offer significant throughput increase for wireless communication. The low-resolution ADC part limits the performance of the transceiver, especially estimating the channels from highly quantized measurements. Mapping from quantized received signals to channel is very challenging. Since the channel response has the characteristic of sparsity, we have leveraged a fast and flexible denoising convolutional neural network (FFDNet) based channel estimation scheme for the m-MIMO system with one-bit ADCs treating the channel matrix as a 2D natural image in this letter. FFDNet can effectively learn from sufficiently large training datasets and is exploited to estimate the channel in the m-MIMO system equipped with one-bit ADCs. We have investigated the exhibited performance of the FFDNet based channel estimation through simulation results. A significantly reduced normalized mean square error has been achieved in our proposed scheme.

**Index Terms**—Massive MIMO, channel estimation, one-bit ADC, FFDNet, deep learning.

## I. INTRODUCTION

Massive multiple-input multiple-output (m-MIMO) enables the deployment of the base station (BS) equipped with a large number of antenna arrays to ensure the reusing of spectrum resources among multiple users as well as to significantly improve the data transmission rate [1]. However, such benefits turn impractical due to the usage of high-resolution (e.g., 8-12 bits) analog-to-digital converters (ADCs). Having high resolution ADCs in the system leads to high hardware costs and power consumption. Therefore, low resolution ADCs (e.g., one-bit) at the receivers have been considered as a promising solution because of its capability of drastically reducing the power consumption and cost of the m-MIMO systems [2]. The vast potential of one-bit ADCs in m-MIMO systems has attracted significant global attention in the last few years and continues to do so [3].

However, it is extremely challenging to design efficient channel estimation techniques and obtain accurate channel estimation due to the highly quantized measurements accompanied by one-bit ADCs. Accurate prior channel state information (CSI) along with statistical properties of the channel is required for the conventional channel estimation techniques [4]. But, it turns out arduous to manage those prior CSI

and statistical properties when the number of transmitters and receivers is very large especially in the case of m-MIMO systems [5]. Though compressed sensing based techniques are very effective in this case, but utilization of non-linear optimization algorithms increases the complexity while having inadequate performances [6]. Therefore, the feasibility of using these techniques for the one-bit ADCs in m-MIMO systems is decreasing on a large scale.

Recently, the adoption of deep learning (DL) algorithms has been demonstrated to be effective for designing channel estimator and have achieved substantial success. Yang *et al.* [7] studied multilayers perceptron's (MLPs) to learn the uplink to downlink channel mapping in m-MIMO systems. But, MLPs show very poor performances in case of a small amount of data or highly complicated data. A comparison among the estimation performance of DL approaches to generalized approximate message passing (GAMP) in m-MIMO with non-ideal one-bit ADCs depicted in [8]. The simulation results substantiate the robustness of DL approaches to ADC impairments than GAMP approaches. In [9], a DL based channel estimation framework for one-bit m-MIMO has been demonstrated. The authors observed that fewer pilots are required for the same channel estimation performance when more antennas are employed. Balevi and Andrews *et al.* [10] have proposed a two-stage estimation scheme for one-bit massive MIMO by exploiting deep neural networks as well as convolutional neural network. They have obtained 5-10dB gain in channel estimation. To process the average of multiple pilot signal segments, the authors have proposed a segment-average based one-bit massive MIMO channel estimation scheme that utilizes a deep neural network (DNN) in [11]. Their proposed scheme outperforms conventional linear channel estimators. DNN as an autoencoder has been utilized to optimize the training signal for few-bit massive MIMO in [12]. Compared with Busgang-based linear minimum mean squared error channel estimator, their scheme has shown more efficacy. The exploitation of generative adversarial network (GAN) to estimate channels from compressed pilot measurements for one-bit massive MIMO has achieved superior performance over sparse signal recovery methods [13].

However, investigations are being conducted on using deep



convolutional neural network (CNN) for channel estimation currently considering the sparsity features of the channel matrix of m-MIMO systems [14]-[18]. Besides, the changes between adjacent elements in the channel are very subtle. Hence, the channel matrix can be treated as two-dimensional (2D) noise-free natural image [19]. Based on the above-mentioned analyses, it is feasible to utilize the CNN based image denoising network to design an efficient channel estimator for the m-MIMO systems equipped with one-bit ADCs. However, the state-of-the-art CNN based image denoising methods are tailored to specific noise levels, and they work well when the noise is in the trained image. This limits the flexibility and efficiency of the conventional CNN based image denoiser in the practical channel estimation [20].

To overcome the aforementioned drawbacks, we have proposed a fast and flexible denoising convolutional neural network (FFDNet) for the channel estimation in one-bit ADCs equipped multiuser m-MIMO systems. Both denoising performance and computation efficiency have increased the superiority of FFDNet over other state-of-the-art CNN based denoiser. Moreover, FFDNet is very efficient in denoising effectively and flexibly 2D images that are corrupted by additive white Gaussian noise (AWGN). Therefore, it has been considered as promising to apply for realizing the channel estimation into the one-bit ADCs equipped multiuser m-MIMO systems. To this end, we have investigated the performance of exploiting FFDNet for channel estimation demonstrating the simulation results.

## II. SYSTEM MODEL

Consider a single cell one-bit ADCs equipped multiuser m-MIMO system as depicted in Fig. 1, where we have the BS which is equipped with  $N_r$  ( $N_r \gg 1$ ) antennas and  $N_t$  ( $N_t \gg 1$ ) single antenna equipped users. Each antenna includes two one-bit ADCs for the real and imaginary parts, respectively. The channel is assumed to be Rayleigh block-fading channel which stays constant for  $T$  channel uses. If  $N_t$  users transmit pilot sequence with length of  $\tau$  to the BS simultaneously, the received pilot signal  $\mathbf{Y} \in \mathbf{C}^{N_r \times \tau}$  is given by

$$\mathbf{Y} = \sqrt{\rho} \mathbf{H} \mathbf{S} + \mathcal{N} \quad (1)$$

where  $\rho$  is the signal to noise ratio (SNR) during pilot transmission, the channel matrix  $\mathbf{H}$  of  $N_t$  users is  $[\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3, \dots, \mathbf{h}_{N_t}]$  with the dimension of  $\mathbf{H} \in \mathbf{C}^{N_r \times N_t}$ , and  $\mathbf{S}$  is mutually orthogonal pilot sequence from  $N_t$  users with the dimension of  $\mathbf{S} \in \mathbf{C}^{N_t \times \tau}$ . The entries of  $\mathbf{H}$  are independent and  $\mathcal{CN}(0, 1)$  distributed. Furthermore,  $\mathcal{N}$  is also independent and  $\mathcal{CN}(0, \sigma^2)$  distributed, stands for AWGN. The real and the imaginary components of the received signal at each antenna are quantized separately using a one-bit ADC and are represented by the values from the set  $\{1 + j, 1 - j, -1 + j, -1 - j\}$ . Now, the objective of this letter is to recovery the channel matrix  $\hat{\mathbf{H}}$  using the trained FFDNet. A noisy channel image corrupted by AWGN with noise level  $\sigma$  is fed into the trained FFDNet as input. The trained FFDNet gives output a noise free channel image.

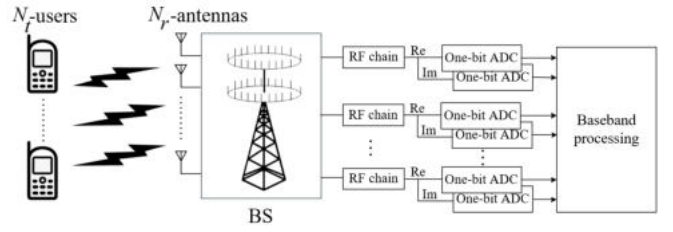


Fig. 1. Diagram of multiuser m-MIMO system with one-bit ADCs.

## III. FFDNET MODEL

### A. Network Architecture

As shown in Fig. 2, the first layer of FFDNet is a reversible downsampling process which takes input noisy channel image  $y$  of size  $N_r \times N_t$ . This layer reshapes the noisy channel image  $y$  into four downsampled sub-images with size  $\frac{N_r}{2} \times \frac{N_t}{2} \times 4$ . The downsampling process can significantly improve the training speed without reducing the modeling ability. After the operation of downsampling, a tunable noise level map with the noise level  $\sigma$  along with the downsampled sub-images to establish  $\tilde{y}$  with a size  $\frac{N_r}{2} \times \frac{N_t}{2} \times (4 + 1)$  as the input of the CNN model. The CNN model comprises three types of layers, such as "Conv+ReLU" with a size  $(3 \times 3 \times 64)$ , "Conv+BN+ReLU" with a size  $(3 \times 3 \times 64)$ , and "Conv" with a size  $(3 \times 3 \times 64)$ . The first layer of the CNN model is "Conv+ReLU" where rectified linear units (ReLU) adds non-linearity to the convoluted output. Middle layers of the CNN model consist of "Conv+BN+ReLU" where BN is added to normalize the layers. "Conv" is used as last layer to regenerate the denoised subimages which leads to reconstruction of denoised channel image  $\tilde{x}$  in turn. Moreover, after each convolution, zero-padding is employed to guarantee that the size of the feature maps is not changed.

### B. Objective Function

The objective of the FFDNet is to generate denoised image  $\tilde{x}$ , which is implicitly defined by [20]

$$\tilde{x} = \mathcal{F}(y, \mathbf{M}; \lambda; \theta)$$

where the noise level map is denoted as  $\mathbf{M}$ ,  $\theta$  is the trainable parameter of CNN model, and  $\lambda$  is used to control the balance between the data fidelity term and the regularization term. But in FFDNet, all the elements of  $\mathbf{M}$  is tunable. As a result, the parameter  $\lambda$  can be neglected. Then, the objective function can be rewritten as

$$\tilde{x} = \mathcal{F}(y, \mathbf{M}; \theta)$$

In order to train the FFDNet, we have acquired noise free 2D channel matrix implementing the multiuser m-MIMO system with one-bit ADCs where the length of pilot sequence  $\tau$  was chosen 8. Then AWGN noise was added to the noise free channel image. In order to keep the balance between complexity and performance, the depth of the network is set

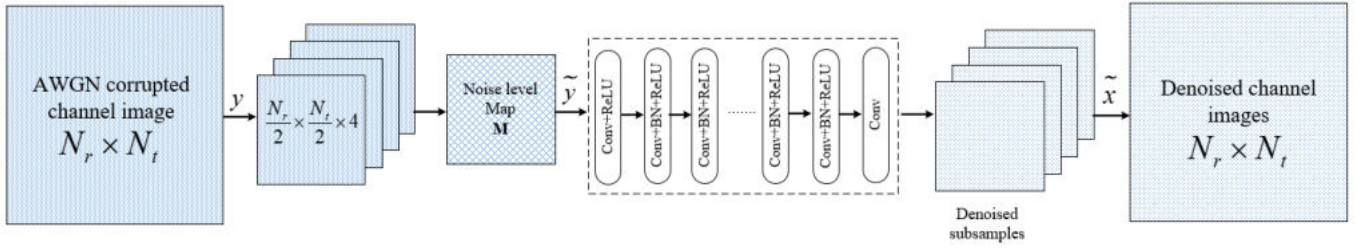


Fig. 2. FFDNet architecture.

as 15. Finally, to optimize the FFDNet during the network training, the loss function is expressed as

$$\mathcal{L}(\theta) = \frac{1}{2N} \sum_1^N \|\mathcal{F}(y, \mathbf{M}; \theta) - x\|^2$$

#### IV. SIMULATION RESULTS

To evaluate the performance of our proposed scheme, we have simulated a multiuser m-MIMO systems with one-bit ADCs in “Matlab 2020b” and generated channel matrices varying the number of antennas  $N_r$  at BS, such as 128, 192, 256 as well as for different SNR values ranging from -10dB to 10dB. The number of user  $N_t$  was kept fixed to 64. Meanwhile, we have added noise level from 2 to 10 to evaluate peak signal to noise ratio (PSNR) performance on channel estimation achieved with FFDNet. Besides, to calculate the the difference between the estimated channel matrix  $\hat{\mathbf{H}}$  and the real channel matrix  $\mathbf{H}$ , we have utilized normalized mean square (NMSE). NMSE is defined as follows

$$NMSE = 10 \log_{10} \left\{ \mathbb{E} \left[ \frac{\|\hat{\mathbf{H}} - \mathbf{H}\|^2}{\|\mathbf{H}\|^2} \right] \right\}$$

After having the datasets, we have divided the datasets into training, testing and validation sets by the ratios of 70%, 20%, and 10%, respectively. All the training and testing programs have been performed in anaconda python 3.7 on a system equipped with 3.80 GHz CPU, 256 GB RAM, and a single NVIDIA Quadro RTX 6000 GPU. To optimize the model, we have used “Adam” optimizer. After training the FFDNet, the network is tested to evaluate noise level sensitivity under different number of antennas at BS. Fig. 3 elucidates the PSNR performance under different noise levels. For a specific noise level map, PSNR value started to decrease rapidly when the number of receiver increases. At noise level 10, the network has achieved 42.08 dB PSNR for 256 number of  $N_r$ . The observed PSNR value is lower when the  $N_r$  is 128 at noise level 10. This is because when the size of transceiver array increases, the channel matrix acquires more obvious sparsity.

In Fig. 4, we have depicted the achieved NMSE varying noise levels. For all combination of the transceiver arrays, the NMSE have started increasing with the increasing of noise levels. Since PSNR performance is better for higher number of

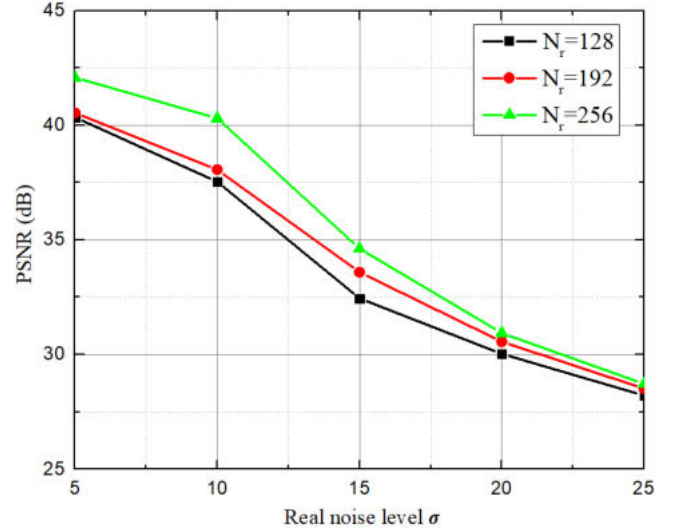


Fig. 3. The PSNR performance of channel estimation.

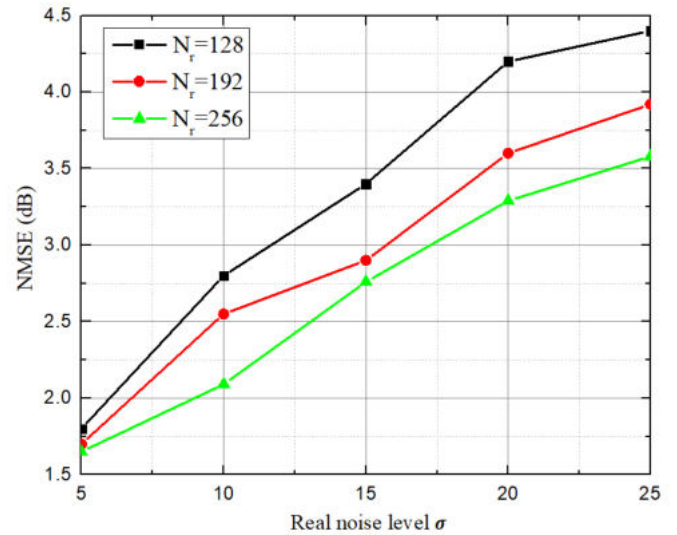


Fig. 4. Performance of NMSE with different number of antennas  $N_r$  at BS.

the receiver array, large transceiver array tends to achieve low NMSE in the channel estimation task. The maximum NMSE is 4.41 dB for the  $N_r$  of 128 at noise level 25 and the minimum NMSE is 1.65dB for the  $N_r$  of 256 at the input noise level 5.

## V. CONCLUSION

In this paper, we have proposed FFDNet for channel estimation in the one-bit ADCs equipped multiuser m-MIMO system. The channel matrix is treated as 2D noise free image considering the sparsity. The trained FFDNet takes the AWGN corrupted noisy image and removes noise to realize the channel. The detail of network architecture and training method for the FFDNet have been demonstrated. The simulation results have also been presented to evaluate the performance. The overall results validate the superiority of FFDNet in channel estimation as well as can be exploited to enhance the existing channel estimation performance significantly of multiuser m-MIMO system with one bit ADCs.

## ACKNOWLEDGMENT

This work was supported by the Technology Development Program (S3098815) funded by the Ministry of SMEs and Startups (MSS, Korea).

## REFERENCES

- [1] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 186–195, Feb. 2014.
- [2] C.-K. Wen, C.-J. Wang, S. Jin, K.-K. Wong, and P. Ting, "Bayes-optimal joint channel-and-data estimation for massive MIMO with low-precision ADCs," *IEEE Trans. Signal Process.*, vol. 64, no. 10, pp. 2541–2556, May 2016.
- [3] Y. Li, C. Tao, G. Seco-Granados, A. Mezghani, A. L. Swindlehurst, and L. Liu, "Channel estimation and performance analysis of one-bit massive MIMO systems," *IEEE Trans. Signal Process.*, vol. 65, no. 15, pp. 4075–4089, Aug. 2017.
- [4] M. Morelli and U. Mengali, "A comparison of pilot-aided channel estimation methods for OFDM systems," *IEEE Trans. Signal Process.*, vol. 49, no. 12, pp. 3065–3073, Dec. 2001.
- [5] X. Chen and M. Jiang, "Adaptive statistical Bayesian MMSE channel estimation for visible light communication," *IEEE Trans. Signal Process.*, vol. 65, no. 5, pp. 1287–1299, Mar. 2017.
- [6] R. Wang, H. He, S. Jin, X. Wang, and X. Hou, "Channel estimation for millimeter wave massive MIMO systems with low-resolution ADCs," in *Proc. IEEE 20th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jul. 2019, pp. 1–5.
- [7] Y. Yang, F. Gao, G. Y. Li, and M. Jian, "Deep learning-based downlink channel prediction for FDD massive MIMO system," *IEEE Commun. Lett.*, vol. 23, no. 11, pp. 1994–1998, Nov. 2019.
- [8] M. Y. Takeda, A. Klautau, A. Mezghani, and R. W. Heath, Jr., "MIMO channel estimation with non-ideal ADCs: Deep learning versus GAMP," in *Proc. IEEE 29th Int. Workshop Mach. Learn. Signal Process. (MLSP)*, Pittsburgh, PA, USA, Oct. 2019, pp. 1–6.
- [9] Y. Zhang, M. Alrabeiah, and A. Alkhateeb, "Deep learning for massive MIMO with 1-bit ADCs: When more antennas need fewer pilots," *IEEE Wireless Commun. Lett.*, vol. 9, no. 8, pp. 1273–1277, Aug. 2020.
- [10] E. Balevi and J. G. Andrews, "Two-stage learning for uplink channel estimation in one-bit massive MIMO," in *Proc. 53rd Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, USA, Nov. 2019.
- [11] R. Zhu and G. Zhang, "A segment-average based channel estimation scheme for one-bit massive MIMO systems with deep neural network," in *Proc. IEEE 19th Int. Conf. Commun. Technol. (ICCT)*, Xi'an, China, Oct. 2019, pp. 81–86.
- [12] D. H. N. Nguyen, "Neural network-optimized channel estimator and training signal design for MIMO systems with few-bit ADCs," *IEEE Signal Process. Lett.*, vol. 27, pp. 1370–1374, 2020.
- [13] E. Balevi, A. Doshi, A. Jalal, A. Dimakis, and J. G. Andrews, "High dimensional channel estimation using deep generative networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 18–30, Jan. 2021.
- [14] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Deep learning-based channel estimation for beamspace mmWave massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 852–855, Oct. 2018.
- [15] H. Huang, J. Yang, H. Huang, Y. Song, and G. Gui, "Deep learning for super-resolution channel estimation and DOA estimation based massive MIMO system," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8549–8560, Sep. 2018.
- [16] M. H. Rahman, M. Shahjalal, M. O. Ali, S. Yoon and Y. M. Jang, "Deep Learning Based Pilot Assisted Channel Estimation for Rician Fading Massive MIMO Uplink Communication System," in *Proc. Twelfth International Conference on Ubiquitous and Future Networks (ICUFN)*, Jeju Island, Republic of Korea, Aug. 2021.
- [17] M. Soltani, V. Pourahmadi, A. Mirzaei, and H. Sheikhzadeh, "Deep learning-based channel estimation," *IEEE Wireless Commun. Lett.*, vol. 23, no. 4, pp. 652–655, Apr. 2019.
- [18] C.-J. Chun, J.-M. Kang, and I.-M. Kim, "Deep learning-based channel estimation for massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 8, no. 4, pp. 1228–1231, Aug. 2019.
- [19] J. Yang, C. Wen, S. Jin and F. Gao, "Beamspace Channel Estimation in mmWave Systems Via Cosparsity Image Reconstruction Technique," *IEEE Transactions on Communications*, vol. 66, no. 10, pp. 4767–4782, Oct. 2018.
- [20] K. Zhang, W. Zuo and L. Zhang, "FFDNet: Toward a Fast and Flexible Solution for CNN-Based Image Denoising," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4608–4622, Sept. 2018.

# iVoiding: A Thermal-Image based Artificial Intelligence Dynamic Voiding Detection System

Yu-Chen Chen<sup>\*\*†</sup>, Jian-Ping Su<sup>†</sup>, Cheng-Han Tsai<sup>†</sup>, Ming-Che Chen, *Member, IEEE*<sup>†\*</sup>, Wan-Jung Chang<sup>†§</sup>,  
*Member, IEEE*, and Wen-Jeng Wu<sup>\*\*†</sup>

<sup>\*</sup> Graduate Institute of Clinical Medicine, College of Medicine,  
Kaohsiung Medical University, Kaohsiung, Taiwan

<sup>†</sup> Department of Urology, Kaohsiung Medical University Hospital,  
Kaohsiung Medical University, Kaohsiung, Taiwan

<sup>†</sup> Department of Electronic Engineering, Southern Taiwan University of Science and Technology, Tainan, Taiwan  
{§allenchang, \*\*jerryhata}@stust.edu.tw

**Abstract**— This paper proposes a thermal-image based artificial intelligence dynamic voiding detection system, designated as iVoiding. iVoiding is composed of a thermal camera, AI recognition platform, and management platform. Furthermore, iVoiding uses deep learning technology to recognize human urination position and dynamically measure the spraying distances and angles of urine flows in the thermal images. The experimental results show that iVoiding provides an objective urine flow measurement that can really be achieved for the purpose of assessing the severity of lower urinary tract symptoms.

**Keywords**—Voiding, Thermal, Lower urinary tract symptoms, Deep Learning

## I. INTRODUCTION

Lower urinary tract symptoms (LUTS), the combinations of serial bothersome symptoms during voiding, is extremely prevalent in men, with rate as high as 62% at any age [1]. This prevalence increases consistently with age, reaching 80.7% in men over 60 years old [2]. In clinical, the severity of LUTS was evaluated by international prostate symptom score (IPSS). There are 7 items in the IPSS questionnaires, including incomplete empty, urinary frequency, intermittency, urgency, weak stream, abdominal straining and nocturia [3], which serves as the first-line tool to allow urologists to better understand the condition of patients' voiding. In the IPSS, we let patients determine how severity their urinary symptoms are in the past month. However, due to the fact that most patients suffered from LUTS are elder, recent studies have shown 30–70% of men could not complete the IPSS because they found the questions too difficult to understand [3]. There are 7 questions in IPSS with the challenges owing to problems with literacy, cognitive ability, visual acuity of the patients and increased time to perform this scoring. It's also hard for elder patients to remember what the voiding pattern really is in the past month, which leads to a recall bias. Visual analogue uroflowmetry (VAUS) score (Fig. 1) was then developed with the advantage of its simple score and easy understanding [4]. Although compared to IPSS, VAUS is more convenient and takes less time to be performed, it is still a subjective questionnaire based on patients' perception of bothersome symptoms and exist a recall bias, especially in older patients who have difficulties with memory. In addition, due to the original limitation of questionnaires, both IPSS and VAUS may not be 100% consistent with the patients voiding. Therefore, an objective tool to timely evaluate the LUTS is necessary.

To address this issue, we consider the need to automatically observe urination status. This paper proposes a novel voiding detection system, designated as iVoiding, which adopts thermal imaging and deep learning technologies. The stream data of voiding can be dynamically measured, and the severity of urinary symptoms can further be objectively assessed according to the measured results.

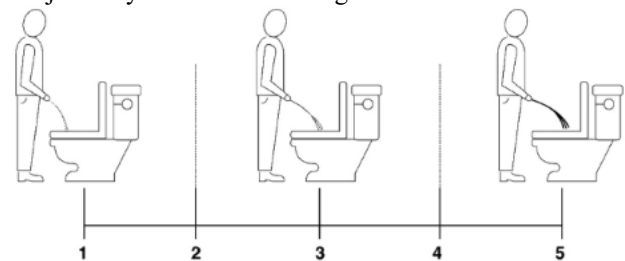


Fig. 1. Visual analogue uroflowmetry score(VAUS). A score with one being slowest stream and least volume, and five being fastest stream and large volume.

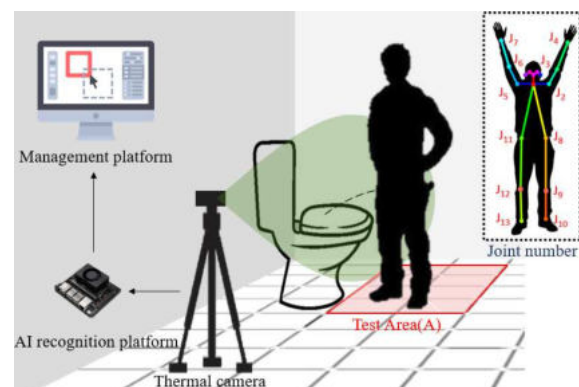


Fig. 2. System architecture of the iVoiding.

## II. SYSTEM ARCHITECTURE

Fig. 2 shows the proposed iVoiding system architecture, which is composed of a thermal camera, AI recognition platform, and management platform. As shown, the iVoiding system captures thermal images in Test Area(A) by the thermal camera, and then converts the human body shapes in the thermal images into 2D human limbs plane coordinates by using the proposed Thermal-Pose module in the AI recognition platform. The 2D human limbs plane coordinates are a set of 18-keypoint (denoted as  $J_i$  with 2D coordinates  $(X_i, Y_i)$  ( $i \in \{0, 1, \dots, 17\}$ )) and 17-edge human backbone coordinates data structure for each body shape. To alleviate shooting angle bias caused by various human height, this

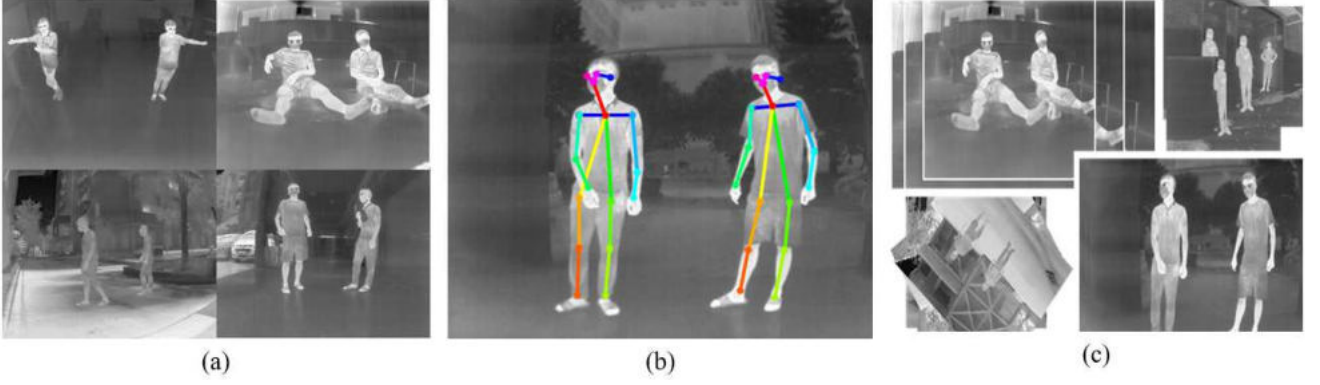


Fig. 3. Thermal-Pose module design. (a) the original dataset ; (b) A total of 18 keypoints with 17 edges annotated in each thermal imaging body pose instance ; (c)the approaches (i.e., scaling, cropping, rotating) applied for data augmentation.

paper considers the coordinates of keypoint  $J_8$  (regarded as the position of left hip) as the reference point of urination position for adjusting vertical height of the thermal camera. Accordingly, the proposed Urine Flow Measurement Algorithm (UFMA) in the AI recognition platform is used to identify the position of voiding and measure the spraying distances and angle of the voiding flow in each thermal image.

#### A. Design of Thermal-Pose Module

The goal of the proposed Thermal-Pose module is to detect 2D human pose of people in the captured thermal images. To implement this module, this paper constructs an original dataset consisting of 12K thermal imaging body pose instances (as in Fig. 3(a)). (Note that the reason for utilizing the thermal images as the training dataset is to achieve thermal imaging-based human pose detection without privacy concerns on the use of RGB images) Each thermal imaging body pose instance in the original dataset is labeled with a total of 18 keypoints (i.e., nose, eyes, ears, neck, shoulders, elbows, wrists, hips, knees, and ankles) according to the body annotations in the COCO keypoint dataset [5] (as in Fig. 3(b)). Furthermore, data augmentation is applied to increase the original dataset size by a factor of 4. This is done by randomly cropping, scaling, and rotating approaches (as in Fig. 3(c)). As a result, the augmented dataset consists of 48K thermal imaging body pose instances annotated with 216K keypoints. The Thermal-Pose module is then built by means of the deep learning-based human pose detection technology (i.e., OpenPose model [6]) trained on the augmented dataset. Consequently, the Thermal-Pose module can monitor the thermal imaging scene of the Test Area(A), in which each detected person will be localized as the 2D human limbs plane coordinates (i.e., 18-keypoint and 17-edge human backbone coordinates data structure).

#### B. Urine Flow Measurement Algorithm (UFMA)

In the proposed UFMA scheme, the urination position is determined in accordance with the 2D human limbs plane coordinates in a thermal image. As shown in Fig. 4, the coordinates of the urination position  $S$  are calculated by combining the x-axis coordinate of the keypoint left wrist (i.e.,  $J_4$ ) and the y-axis coordinate of the keypoint left hip (i.e.,  $J_8$ ) into the 2D coordinates  $S(X_S=Y_8, Y_S=X_4)$ . Furthermore, consider a point  $R$  represents the destination where the urine

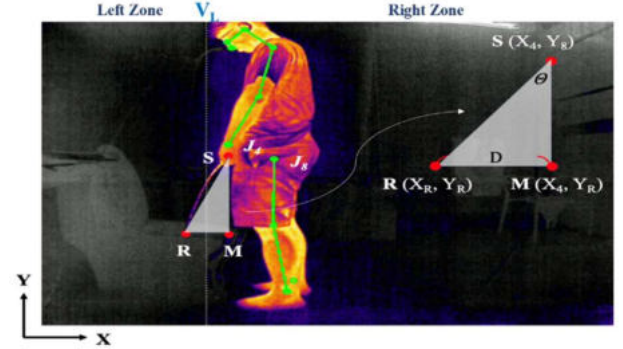


Fig. 4. The parameters denoted in the UFMA scheme.

flow sprays into the toilet. The coordinates of the point  $R$  (denoted as  $R(X_R, Y_R)$ ) can be estimated by observing the location to the left with a higher temperature in the thermal image. Moreover, a point  $M$  with the coordinates  $M(X_M=X_4, Y_M=Y_R)$  can be established by means of the coordinates of  $S$  and  $R$ . By adopting the location information of points  $S$ ,  $R$  and  $M$ , the UFMA algorithm is capable of estimating the spraying distance  $D$  of the urine flow (i.e., the distance between  $R(X_R, Y_R)$  and  $M(X_M, Y_M)$ ) in accordance with the distance formula, i.e.,

$$D = \overline{RM} = \sqrt{|(X_R - X_M)|^2 + |(Y_R - Y_M)|^2} \quad (1)$$

The distance between  $S(X_S, Y_S)$  and  $R(X_R, Y_R)$ , denoted as  $\overline{SR}$ , can be computed by means of the same distance formula, i.e.,

$$\overline{SR} = \sqrt{|(X_S - X_R)|^2 + |(Y_S - Y_R)|^2} \quad (2)$$

As a result, the UFMA algorithm can obtain the angle  $\theta$  of urine flow (i.e., the included angle  $\angle MSR$ ) by means of the inverse sine function taking the ratio of  $D/\overline{SR}$ , i.e.,

$$\theta = \sin^{-1} \frac{D}{\overline{SR}} \quad (3)$$

#### C. Management Platform

After obtaining the distance  $D$  and angle  $\theta$  in a thermal image by the UFMA approach at a time  $t$ , the AI recognition



Fig. 5. The deployment of the proposed iVoiding system in a toilet.

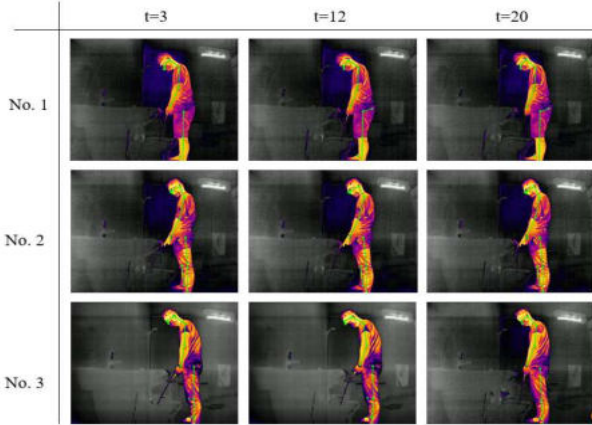


Fig. 6. The thermal snapshots taken from 3 volunteers for the urination test.

platform immediately transmits both  $D$  and  $\theta$  with the time  $t$  toward the management platform. The management platform records the received distance  $D$ , angle  $\theta$ , and time  $t$  in a database, and therefore is capable of providing timely and comprehensive information of urine flows (e.g., a result of spraying distance and angle changes in time). This information can be further analyzed to obtain a novel basis for assessing the severity of urinary symptoms.

### III. EXPERIMENTAL RESULTS

As shown in Fig. 5, the proposed iVoiding system was deployed in a toilet for performance evaluation. To reflect the practical performance of iVoiding, 3 healthy young male volunteers (denoted as No. 1, No. 2, and No. 3) who have been asked to drink water until they felt the urge to urinate were arranged for the experiment. Fig. 6 shows the thermal images of the urination of each volunteer taken at 3rd, 12th, and 20th seconds. It can be seen that the changes of distance and angle of the voiding stream from each volunteer can be observed in the thermal snapshots. Figs. 7 and 8 illustrate the spraying angle and distance measurement performance of the UFMA approach under different amounts of urine. It is observed that irrespective of the amount of urine being voided, during normal urination, the urine streams remain at a certain spraying angle for a while and the stream spraying angles are getting smaller at the ending period of the urination. Furthermore, the normal urine flows continue to spray for a certain distance for a period of time, while the spraying distances are shortened at the ending phase of the voiding.

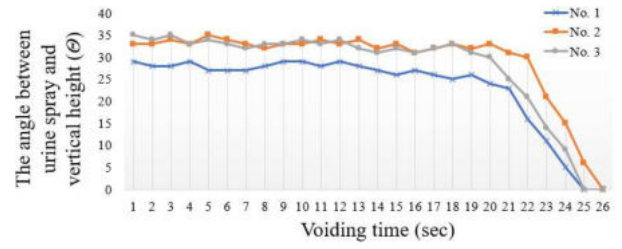


Fig. 7. Spraying distance,  $\theta$ , relative to voiding time,  $t$ .

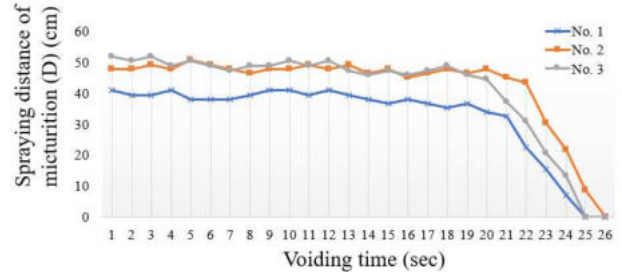


Fig. 8. Spraying distance,  $D$ , relative to voiding time,  $t$ .

### IV. CONCLUSION AND FUTURE WORKS

In this paper, a thermal-image based artificial intelligence dynamic voiding detection system has been proposed, called iVoiding, which can be applied to evaluate the severity of LUTS. The iVoiding can improve the current assessment tools (i.e., IPSS and VAUS questionnaires) by providing an objectively and timely way for observing the urination of patients. The proposed iVoiding system has been implemented to track urination pose by a Thermal-Pose module, which can detect and covert human body shapes into the 2D human limbs plane coordinates in the thermal images. By using the information of 2D human limbs plane coordinates, the iVoiding system can detect urination position and measure the spraying distance and angle of urine flows by a UFMA scheme. Since the iVoiding tracks urination status based on thermal-imaging technology, it can be deployed in a toilet for medical observation without privacy concerns.

For future work, we will cooperate with the urology department at the tertiary referral hospital to apply iVoiding in healthy controls and patients to find out each cut-off value affecting the severity of LUTS.

### REFERENCES

- [1] Irwin DE, *et al.*, "Population-based survey of urinary incontinence, overactive bladder, and other lower urinary tract symptoms in five countries: results of the EPIC study," *Eur Urol*, pp. 1306-1314, 2006
- [2] Thorpe A, *et al.*, "Benign prostatic hyperplasia," *Lancet*, pp. 1359-1367, 2003.
- [3] N Rodrigues Netto Jr, *et al.*, "Latin American study on patient acceptance of the International Prostate Symptom Score (IPSS) in the evaluation of symptomatic benign prostatic hyperplasia," *Urology*, pp. 46-49, 1997.
- [4] Tiwari R, *et al.*, "Prospective validation of a novel visual analogue uroflowmetry score (VAUS) in 1000 men with lower urinary tract symptoms (LUTS)," *World J Urol*, pp.1267-1273, 2020.
- [5] COCO-dataset keypoint evaluation Available at URL: <https://cocodataset.org/#keypoints-eval>.
- [6] Z. Cao, *et al.*, "OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, pp. 172-186, 2021.

# Determining Jigsaw Puzzle State from an Image based on Deep Learning

Ijaz Ahmad<sup>1</sup>, Suk-seung Hwang<sup>2</sup>, and Seokjoo Shin\*

<sup>1</sup>\*Dept. of Computer Engineering, <sup>2</sup>Dept. of Electronic Engineering  
Chosun University

Gwangju, 61452 South Korea

<sup>1</sup>ahmadijaz@chosun.kr, \*sjshin@chosun.ac.kr (Corresponding author)

**Abstract**—The intriguing nature of jigsaw puzzle has captured the attention of researchers for many years. In this paper, we propose a deep learning model to determine different states of jigsaw puzzle from an image. We represent the task as a classification problem where each state of the puzzle is considered as a class. For this purpose, we have proposed a method to generate a dataset that can efficiently represent the jigsaw puzzle states space. The proposed model has 93% accuracy on the test dataset. In addition, we have shown that when the tiles size changes the model is still able to recognize 83% of the states. Though, genetic algorithms (GA) have been successful in solving larger puzzles, they require hand-crafted sophisticated compatibility scores. The computation and memory requirement to store the piecewise compatibility measure increases with the size of the puzzle. As an application, we have shown that the proposed method can be used as a fitness function of GA based jigsaw puzzle solver without using any compatibility measure.

**Keywords**—jigsaw puzzle, deep learning, genetic algorithm

## I. INTRODUCTION

A jigsaw puzzle consists of multiple non-overlapping blocks of an image and the goal is to reassemble all the blocks to reconstruct the original image. Computational jigsaw puzzle assembly has applications in the field of image editing, recovery of shredded documents or photographs, art-piece recovery from shards in archaeology etc. [1]. In the recent years, automatic jigsaw solvers have been vastly improved in terms of number of tiles the puzzle has, tiles size, solution accuracy and amount of manual labor. For the size of the puzzle: [2] proposed a probabilistic approach to solve a puzzle of 432 pieces, [3]'s solver could handle up to 3000 pieces, [4], [5] genetic algorithm based solver to solve larger puzzle of size 22,834 pieces. On the other hand, for the variant of the puzzle: [6] proposed a method to handle puzzle with unknown piece orientation and location, and [7] handled puzzles with missing pieces. [8] proposed a neural network to predict the adjacent pieces and applied shortest path optimization to assemble the puzzle. [9] proposed to use graph connection Laplacian to determine the edge relationships for solving jigsaw puzzles. [1] proposed a deep learning based compatibility measure between two pieces. The metric improves the existing jigsaw puzzle solver accuracy. A somewhat similar approach has been adopted in [10] for solving a real world problem i.e., Portuguese tile panels reconstruction. [11] proposed a GAN-based architecture to solve jigsaw puzzles with eroded boundaries. [12] proposed a GAN-based architecture that utilizes both edge and semantic information to solve jigsaw puzzles efficiently.

In general an automatic jigsaw puzzle solver consists of two steps: a compatibility measure of adjacent tiles and a

strategy to reassemble the compatible tiles. The compatibility measure has been widely studied, e.g., [2] compares five different compatibility measures, among which the dissimilarity function is the most reliable one. For reassembly strategy, greedy algorithms become problematic in case of local optima [1]. To overcome the problem, genetic algorithm (GA) based jigsaw puzzle solver has been proposed in [4]. The GAs have been able to solve larger and harder puzzles; however, they require hand-crafted sophisticated compatibility measures [1]. The compatibility measure for a puzzle of size  $N \times M$  with unknown location requires  $R(N \times M)^2$  size matrix to store all of the pairwise compatibilities for all pieces.  $R = 4$  is the possible position of a tile to be placed in relation to another tile e.g., top, right, bottom or left.

In this paper, we propose a deep learning model to identify a jigsaw puzzle state from a given image. We have posed the problem as a classification problem where each state is a separate class. For this purpose, we propose a method to generate a dataset that can efficiently represent different states of the jigsaw puzzle. Each state is recognized by a score based on the number of tiles being swapped and their original locations. The trained model is able to recognize majority of the jigsaw puzzle states. The misclassified states are the ones where two or more than two tiles have the same texture and are visually imperceptible. Further, we evaluated the efficiency of the model to recognize states when the size of the tiles changes. As an application of the proposed method, we have used the trained model as a fitness function of GA based jigsaw puzzle solver. We assumed a puzzle with unknown tiles position but known dimensions.

## II. PROPOSED METHOD

### A. Jigsaw Puzzle

We define jigsaw puzzle state as two or more than two tiles have swapped their positions as opposed to their appearance in the original image. Numerically, the state can be represented as a sum of differences between each tile in the puzzle and their neighbors (i.e., for two tiles to be neighbors, their Euclidean distance should be 1). The state representation of a jigsaw puzzle with  $r \times c$  tiles  $t$ , where each tile is indexed by  $t_{i,j} = c \times i + j$ , is given as

$$S = \sum_{i=0}^{r-1} \sum_{j=0}^{c-2} d(t_{i,j}, t_{i,j+1}) + d(t_{j,i}, t_{j+1,i}) \quad (1)$$

where,

$$d(x, y) = \begin{cases} 1, & |x - y| \in \{1, c\} \\ 0, & \text{Otherwise} \end{cases}$$

It is an adjacency function that measures if two tiles are adjacent in the original image based on their absolute

difference. The difference value should be equal to 1 or  $c$  for two tiles to be horizontally or vertically adjacent, respectively. The score  $S$  in (1) not only depends on the number of tiles being shuffled but also on their positions they have been swapped from. Let's suppose, only two tiles have swapped their positions. Then, the score obtained is as: for a corner piece being swapped with another corner piece or with its neighbor, then  $S = 4$ . However, if a corner piece is swapped with a piece that is in the same row or column then  $S = 5$ . Similarly,  $S = 6$  for a corner piece being swapped with a piece that is neither in the same row nor in the same column and so on. The score can have a minimum value of 4 and maximum value of  $r^2 + c^2 - (r + c)$ .

### B. Automatic Jigsaw Puzzle Solver

An automatic jigsaw puzzle solver consists of two parts: a measure of piecewise compatibility of adjacent blocks and a strategy to reassemble the blocks in correct order as they would have appeared in the original image. For reassembly strategy, greedy algorithms becomes problematic in case of local optima. To overcome the problem, [4] has proposed genetic algorithm (GA) based jigsaw puzzle solver. The pseudocode for GA is given in Algorithm 1.

**Algorithm 1** Pseudocode for Genetic Algorithm

1. **Generate** initial population  $P$  of  $n$  random chromosomes
2. **while**(stopping GA criteria is false) **Repeat**
3.     **Evaluate**  $P$  using the fitness function
4.      $new\_population = NULL$
5.     copy  $b$  best chromosomes to  $new\_population$
6.     copy  $m$  mutated chromosome to  $new\_population$
7.     **while**  $size(new\_population) \leq n$  **do**
8.          $parent1 = select\ chromosome$
9.          $parent2 = select\ chromosome$
10.          $child = crossover(parent1, parent2)$
11.         add child to  $new\_population$
13.      $P = new\_population$

The first step of the algorithm is to generate an initial population  $P$  with  $n$  random candidates solutions called chromosomes. In line 2, the criteria is the number of generations or how many times the algorithm will repeat the subsequent steps. Next, the entire population is evaluated based on the fitness function, which decides the selection of a particular chromosome to reproduce and pass onto next generation. The new population is generated by the crossover and mutation process operations. The whole process is repeated for a given number of times or when the desired fitness function value has been achieved.

### C. Model

We use ConvNet of the VGG16 [13] model as a feature extractor. Our model consists of a sequence of convolutions followed by batch-normalizations, and max pooling layers. On top of the feature extractor is two dense layers with 512 nodes in the first layer and 22 in the second with softmax activation. The full architecture is shown in Table I. The pixel values are processed to have value between 0 and 1.

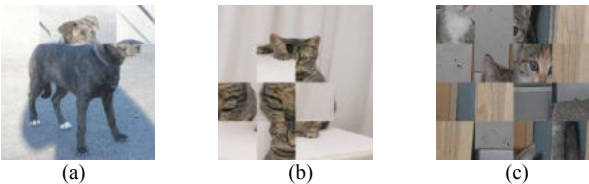


Fig. 1. Examples from the dataset to represent jigsaw puzzle states. (a) label is  $j_2$ , Score = 5. (b) label is  $j_{11}$ , Score = 14. (c) label is  $j_{21}$ , Score = 24.

**TABLE I.** Architecture of the proposed model. Conv3: 3×3 convolution, BN: Batch normalization.

Layer	Shape	Parameters #
Input	224×224×3	0
Conv3+Conv3+BN	224×224×64	39k
Max Pooling2D	112×112×64	-
Conv3+Conv3+BN	112×112×128	221k
Max Pooling2D	56×56×128	-
Conv3+Conv3+Conv3+BN	56×56×256	1.4M
Max Pooling2D	28×28×256	-
Conv3+Conv3+Conv3+BN	28×28×512	5.9M
Max Pooling2D	14×14×512	-
Conv3+Conv3+Conv3+BN	14×14×512	7M
Max Pooling2D	7×7×512	-
Fully Connected	512	12M
Fully Connected	22	11k

**TABLE II.** Accuracy of the model on different size of tiles.

Training	Test	
	56x56	28x28
0.95	0.93	0.83

## III. RESULTS AND DISCUSSION

In this section, we first present our experimental setup. Then, we show the performance of the proposed model to determine a jigsaw puzzle state. Finally, we present an application of the method to be used as a fitness function of genetic algorithm to solve a jigsaw puzzle.

### A. Determining Jigsaw Puzzle State

For this purpose, we have collected 10k color images of size 224 × 224. Next, we have performed the following steps to generate our dataset:

- Divide the images into 4x4 tiles each of size 56x56.
- Generate 10k random indices corresponding to each score value (1), e.g., in our case  $S \in \{4, 5, \dots, 24\}$ , as  $I = \{I_1, I_2, \dots, I_{21}\}$ . Each element of  $I$  has 10k entries and corresponds to a score, i.e., each entry of  $I_1$  has score 4,  $I_2$  has score 5, and so on.
- To generate labelled data, shuffle a copy of the original images using the indices matrix from the previous step.

The dataset consists of 22 classes each with 10k images. The classes labels are  $J = \{j_0, j_1, \dots, j_{21}\}$ , where  $j_0$  is the solved state of the puzzle and the subsequent classes represents states with score equal to the subscript of the label plus 3, e.g.,  $j_1$  is the puzzle state where  $S = 4$ . The examples from the dataset are shown in Fig. 1. The dataset split is: 95% as training, 2.5% as validation and 2.5% as test datasets. We trained the model in a supervised manner using stochastic gradient descend (SGD) with momentum 0.9 for 100 epochs. The initial learning rate was set to 0.1 and then reduced by a factor of 10 after 30 epochs.

In the first experiment, the model was trained and evaluated on the same size of tiles, i.e., 56 × 56. In the second experiment, the trained model was evaluated on smaller tiles size of 28 × 28. We show the accuracy of the model on the train, validation and test dataset in Table II. The model has correctly determine the jigsaw puzzle state 93% of the times in the test dataset. Fig. 2. shows the confusion matrices to represent counts from actual and predicted classes. It can be seen that the model has misclassified the classes corresponding to higher scores with the classes closed by, e.g., class 16 with 17. The reason for this is that the images have regions where the texture is uniform. Two or more tiles from such regions are visually indistinguishable. As for the second



True labels	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
j0	249	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
j1	0	248	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
j2	0	0	250	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
j3	0	2	1	247	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
j4	0	0	0	1	248	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
j5	0	0	0	0	3	241	5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
j6	0	0	0	0	1	2	243	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
j7	0	0	0	0	0	1	8	240	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
j8	0	0	0	0	0	0	5	239	5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
j9	0	0	0	0	0	0	0	2	237	10	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
j10	0	0	0	0	0	0	0	1	5	238	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
j11	0	0	0	0	0	0	0	0	1	8	234	7	0	0	0	0	0	0	0	0	0	0	0	0	0
j12	0	0	0	0	0	0	0	0	0	1	5	228	14	1	0	0	0	0	0	0	0	0	0	0	0
j13	0	0	0	0	0	0	0	0	0	0	7	233	9	1	0	0	0	0	0	0	0	0	0	0	0
j14	0	0	0	0	0	0	0	0	0	0	6	234	7	3	0	0	0	0	0	0	0	0	0	0	0
j15	0	0	0	0	0	0	0	0	0	0	1	10	223	15	1	0	0	0	0	0	0	0	0	0	0
j16	0	0	0	0	0	0	0	0	0	0	6	211	25	5	3	0	0	0	0	0	0	0	0	0	0
j17	0	0	0	0	0	0	0	0	0	0	1	2	13	177	44	13	0	0	0	0	0	0	0	0	0
j18	0	0	0	0	0	0	0	0	0	0	0	1	11	192	45	1	0	0	0	0	0	0	0	0	0
j19	0	0	0	0	0	0	0	0	0	0	0	0	2	11	214	23	0	0	0	0	0	0	0	0	0
j20	0	0	0	0	0	0	0	0	0	0	0	0	0	1	5	221	23	0	0	0	0	0	0	0	0
j21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	240	0	0	0	0	0	0	0	0
j22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 2. Confusion matrices to count actual and predicted classes.

experiment, when the block size is reduced to  $28 \times 28$ , the model is still able to determine 83% of the jigsaw puzzle states. Note that it is necessary to resize the images to meet the input image size requirement of the trained model. We conjecture that the accuracy may improve in this case with the model being trained on mixed sized tiles.

### B. Application—Jigsaw Puzzle Solver based on GA

As an application, we have replaced the fitness function of GA with the proposed jigsaw state recognizer in Algorithm 1. We have generated 40 chromosomes as the initial population and set the number of generations to 30. We have chosen 30% of the top population to be in the successor generation and set the mutation rate to 5%. We show an evolution of the solver how it approaches to the solution state in Fig. 3. The figure shows the average fitness score of the entire population of each generation and score of the best solution in the population. It can be seen that with each generation the solver gets closer to the solution. For visual analysis, we show an example in Fig. 4. The starting state of the puzzle has score  $S = 24$ . The initial number of generation was 30 but the algorithm converges to the solution in 11 generations. For this experiment, we took 30 images randomly from the test dataset. The GA based solver was able to solve the puzzle 85% of the times. The performance may improve by tuning the algorithm parameters, e.g. increasing population size and/or number of generations.

## IV. CONCLUSION AND FUTURE WORK

We proposed a deep learning model to determine jigsaw puzzle states from an image. The model was able to identify 93% of the states in the test dataset. In addition, the model was able to recognize the puzzle states from different size tiled images. As an application, we replaced the fitness function of genetic algorithm based jigsaw puzzle solver with the trained model, and we were able to solve 16 tiles puzzle.

In future, we are interested in extending our method to recognize and solve puzzles with larger number of tiles. For example, the proposed model can be used in the crossover operator to identify good parts in a solution and pass them into next generation.

## ACKNOWLEDGMENT

This research is supported by Basic Science Research Program through the National Research Foundation of Korea

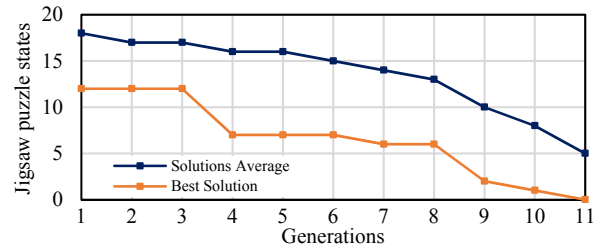


Fig. 3. The evolution of the genetic algorithm to find a solution.



Fig. 4. As genetic algorithm converges to the solution. The original puzzle and the best solution in the generation 1, 4, 7, 9, 10, and 11 (left to right).

(NRF) funded by the Ministry of Education (NRF-2018R1D1A1B07048338).

## REFERENCES

- [1] D. Sholomon, E. David, and N. S. Netanyahu, “DNN-Buddies: A Deep Neural Network-Based Estimation Metric for the Jigsaw Puzzle Problem,” *ArXiv171108762 Cs Stat*, vol. 9887, pp. 170–178, 2016, doi: 10.1007/978-3-319-44781-0\_21.
- [2] T. S. Cho, S. Avidan, and W. T. Freeman, “A probabilistic image jigsaw puzzle solver,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, USA, Jun. 2010, pp. 183–190. doi: 10.1109/CVPR.2010.5540212.
- [3] D. Pomeranz, M. Shemesh, and O. Ben-Shahar, “A fully automated greedy square jigsaw puzzle solver,” in *CVPR 2011*, Colorado Springs, CO, USA, Jun. 2011, pp. 9–16. doi: 10.1109/CVPR.2011.5995331.
- [4] D. Sholomon, O. David, and N. S. Netanyahu, “A Genetic Algorithm-Based Solver for Very Large Jigsaw Puzzles,” in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, Portland, OR, USA, Jun. 2013, pp. 1767–1774. doi: 10.1109/CVPR.2013.231.
- [5] D. Sholomon, O. E. David, and N. S. Netanyahu, “Genetic algorithm-based solver for very large multiple jigsaw puzzles of unknown dimensions and piece orientation,” in *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, Vancouver BC Canada, Jul. 2014, pp. 1191–1198. doi: 10.1145/2576768.2598289.
- [6] A. C. Gallagher, “Jigsaw puzzles with pieces of unknown orientation,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, Jun. 2012, pp. 382–389. doi: 10.1109/CVPR.2012.6247699.
- [7] G. Paikin and A. Tal, “Solving multiple square jigsaw puzzles with missing pieces,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 4832–4839. doi: 10.1109/CVPR.2015.7299116.
- [8] M.-M. Paumard, D. Picard, and H. Tabia, “Deepzle: Solving Visual Jigsaw Puzzles With Deep Learning and Shortest Path Optimization,” *IEEE Trans. Image Process.*, vol. 29, pp. 3569–3581, 2020, doi: 10.1109/TIP.2019.2963378.
- [9] V. Huroyan, G. Lerman, and H.-T. Wu, “Solving Jigsaw Puzzles by the Graph Connection Laplacian,” *SIAM J. Imaging Sci.*, vol. 13, no. 4, pp. 1717–1753, Jan. 2020, doi: 10.1137/19M1290760.
- [10] D. Rika, D. Sholomon, E. (Omid) David, and N. S. Netanyahu, “A novel hybrid scheme using genetic algorithms and deep learning for the reconstruction of portuguese tile panels,” in *Proceedings of the Genetic and Evolutionary Computation Conference*, Prague Czech Republic, Jul. 2019, pp. 1319–1327. doi: 10.1145/3321707.3321821.
- [11] D. Bridger, D. Danon, and A. Tal, “Solving Jigsaw Puzzles With Eroded Boundaries,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 3523–3532. doi: 10.1109/CVPR42600.2020.00358.
- [12] R. Li, S. Liu, G. Wang, G. Liu, and B. Zeng, “JigsawGAN: Auxiliary Learning for Solving Jigsaw Puzzles With Generative Adversarial Networks,” *IEEE Trans. Image Process.*, vol. 31, pp. 513–524, 2022, doi: 10.1109/TIP.2021.3120052.
- [13] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *ArXiv14091556 Cs*, Apr. 2015, Accessed: Mar. 08, 2021. [Online]. Available: <http://arxiv.org/abs/1409.1556>

# Growth Estimation Sensor Network System for Aquaponics using Multiple Types of Depth Cameras

Ryota Murakami

Graduate school of information science and engineering  
Ritsumeikan University  
1-1-1 Noji-higashi, Kusatsu, Shiga 525-8577 Japan  
Email: is0433se@ed.ritsumeai.ac.jp

Hiroshi Yamamoto

Department of information science and engineering  
Ritsumeikan University  
1-1-1 Noji-higashi, Kusatsu, Shiga 525-8577 Japan

**Abstract**—There is an urgent need to automate and improve the efficiency of agricultural work because traditional agricultural work becomes a heavy burden for the elderly who makes up many of the agricultural workers. Therefore, in recent years, aquaponics is attracting attention as a good solution for the problem. Aquaponics is an efficient farming system that combines aquaculture and hydroponics. In the system, the bacteria decompose fish excrement, the plants absorb the decomposed excrement as nutrients, and the purified water returns to a fish tank. This cycle is automated to greatly reduce the amount of consumption of water and fertilizer. In order to keep the accurate cycle, the system should not only manage the environment so that the bacteria can properly decompose fish waste, but also measure the growth condition of the plants and fish accurately. However, the automatic estimation of the growth rate of the fish in the water is a difficult problem because the existing camera-based method cannot easily be applied to the fish living in the water. Therefore, in this study, we propose a new sensor network system with a function to quantify the size of fish as well as plants by utilizing not only a general depth camera that uses infrared rays but also a stereo camera with multiple cameras. Through the performance evaluation, it is confirmed that the proposed method can estimate the leaf area of plants and the standard length of the fish with high accuracy.

**Index Terms**—IoT, aquaponics, depth camera, growth rate, MaskR-CNN

## I. INTRODUCTION

In recent years, the decrease in the number of people working in agriculture has become a serious problem in Japan [1]. The main reason of the problem is a huge burden of the daily farm work, hence there is a need to automate and improve the efficiency of the work. In order to realize efficient work of the agriculture and to obtain stable income, aquaponics is attracting attention as the next generation of circular agriculture. Aquaponics is sustainable agriculture that combines hydroponics and aquaculture to achieve both productivity and environmental friendliness [2]. The overview of the aquaponics system is shown in Fig. 1. This system does not require use of chemical fertilizers because bacteria decompose the fish waste and produce nutrients for the plants. In addition, it is also attracting attention as a sustainable agriculture method because the use of the amount of the water can reduce by about 90 % of the conventional cultivation in the open field [3].

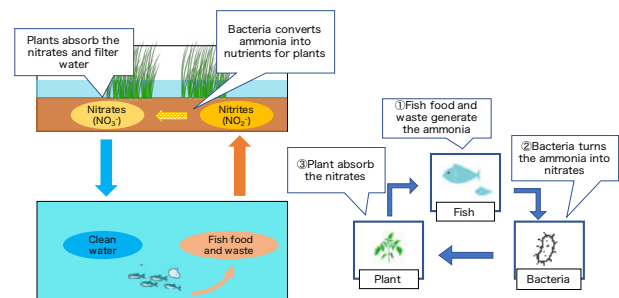


Fig. 1. Overall view of the Aquaponics

Existing study on aquaponics proposes a remote monitoring system of environmental information utilizing multiple sensors [5]. However, it is necessary for the field manager to visually check the growth condition of the fish and the plants because the system is not equipped with a function to automatically observe the growth condition.

Therefore, in this study, we propose a new aquaponics support system using sensor network technology that can automatically estimate the growth condition of the plants and the fish. This system not only measures the environmental information such as water quality composition and temperature but also has the ability to quantify the growth of plants and fish. In order to accurately measure the growth of fish and plants, two different depth cameras are used depending on the environment, and image recognition using deep learning technology is used to accurately estimate the growth.

## II. RELATED WORKS AND OBJECTIVES OF OUR STUDY

### A. Management System for Aquaponics

In the existing study on an environmental observation system for aquaponics, Paul (2019) et al. propose a system that measures the various environmental information (e.g., pH, temperature, and dissolved oxygen) and automatically controls actuators so that the environmental condition is within an appropriate range for the growth of the plants and the fish [6]. In addition, Zheng (2019) et al. propose a control method of the actuators based on the observation results of the growth environment and reduce the power consumption to 42.9% of the previous system by efficiently operating the actuators [7].

These existing studies use ultrasonic distance sensors to measure the height of plants and evaluate the relationship between the number of growing days and the growth condition of the plants. On the other hand, the method cannot accurately quantify the degree of growth because it measures only the height of the plants. In order to accurately quantify the growth status of plants, it is necessary to estimate the leaf area, which is closely related to the amount of harvest. In addition, the field manager should manually observe the growth condition of the fish in the existing system. The existing distance sensor-based method cannot be applied to the observation of the fish in the water because the infrared ray and the ultrasonic used for measuring the distance cannot travel straight through the water.

### B. Image Recognition Using Deep Learning for Growth Estimation of the Plants

Xiaoyang (2019) et al. propose the growth condition estimation system for cucumbers grown in greenhouses. The system acquires RGB images of cucumbers and uses deep learning-based image analysis method, MaskR-CNN, to detect pixels corresponding with cucumbers in the images [4]. By using the MaskR-CNN, the area of the target can be stably estimated by mitigating the effect of the shadow or brightness even in a place where a lot of sunlight enters such as a greenhouse. On the other hand, the method targets only a part of the plants (e.g., fruit) and cannot observe the growth condition of the entire plants including stems and leaves. Therefore, it cannot be applied to leafy vegetables that are mainly grown in aquaponics.

### C. Objectives of Our Study

Existing studies aim to realize a system to measure and visualize the environmental information for aquaponics, and have not been able to accurately estimate the growth condition of the plants and fish. By quantifying the growth condition, it can be confirmed whether the interaction among fish, plants, and bacteria in aquaponics is accurate or not, which may improve the efficiency of the aquaponics. Therefore, in this study, we propose and develop a new sensing system that can accurately quantify the growth condition of plants and fish, and clarify the relationship between the growth condition and the state of the growing environment measured by various sensors. In particular, the proposed system is equipped with a method for estimating the growth of fish living in the water to realize practical and efficient aquaponics. The estimation method uses a stereo camera which can observe the three-dimensional structure of the target by analyzing the parallax between multiple visible light cameras even if the target is the fish living in the water where the conventional measurement method using the distance sensor cannot be leveraged. In addition, the MaskR-CNN is used to accurately identify the pixels corresponding with the target from the images captured by the multiple cameras.

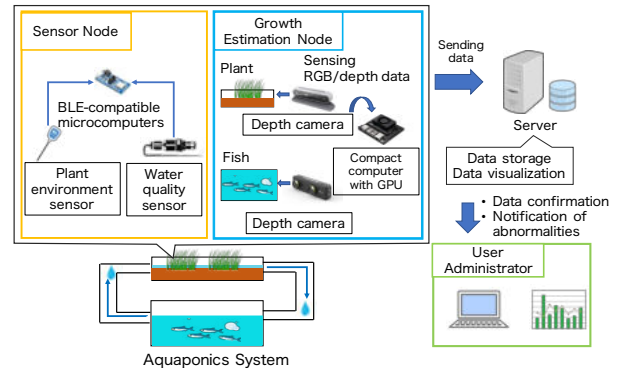


Fig. 2. Overview of the proposed system

## III. PROPOSED MANAGEMENT SYSTEM FOR AQUAPONICS

### A. Overview of Our Proposed System

Figure 2 shows an overview of our proposed system. As shown in this figure, the proposed system consists of sensor nodes, a growth estimation node, and a management server.

The sensor node measures the environmental information of the aquaponics system using multiple sensors, and transmits the data of the measurement results to the growth estimation node using BLE (Bluetooth Low Energy). The growth estimation node uses two types of depth cameras connected to a small computer to acquire depth images and RGB images of plants and fish. For the growth estimation node, the depth camera using the infrared ray and the stereo camera are utilized to estimate the growth condition of the plants and the fish, respectively. In addition, the growth condition is estimated by analyzing the images in the growth estimation node and only the estimation results are sent to the management server so that the amount of data sent to the management server can be greatly reduced. The management server stores and visualizes the data received from the growth estimation node, and the system operator can remotely check the various statuses of the aquaponics through a web application provided by the server.

### B. Design of Sensor Node

The sensor node for measuring the environmental information of the aquaponics is configured using an Adafruit Feather nRF52840 Express, a microcomputer that supports short-range wireless communication via BLE. The system configuration of the sensor node is shown in Fig. 3. To observe the water quality, a pH sensor (pH Kit), a dissolved oxygen sensor (Dissolved Oxygen Kit), and a water temperature sensor (PT-1000 Temperature Kit) manufactured by Atlas Scientific are connected to the sensor node. By measuring the amount of dissolved oxygen in the water, the system operator can check whether the amount of oxygen is enough for the fish or not. The measured pH can be used to check whether the bacteria are correctly converting ammonia contained in fish excrement into nitrate because pH changes based on the amount of ammonia and nitrate in the water. In addition, in order to observe the condition of the environment where the plants

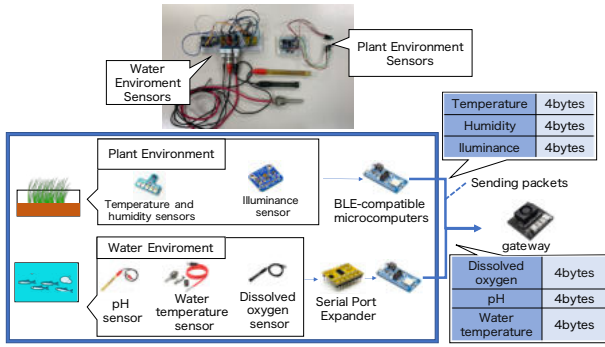


Fig. 3. Device configuration of sensor node

grow, the sensor node consisting of an illuminance sensor (Adafruit TSL2561) and a temperature/humidity sensor module (Adafruit Sensirion SHT31-D) is installed. The sensor node measures the environmental information every 30 seconds and transmits the measured data to the growth estimation node that acts as a gateway, using BLE. Each data measured by the sensor is a floating point number consisting of 4 bytes, and two types of packets are transmitted from the sensor nodes. The first type of packet is consisting of pH, dissolved oxygen, and water temperature data, and the second type is consisting of illumination, temperature, and humidity data.

### C. Design of Growth Estimation Node

The growth estimation node is composed of the single board computer, Jetson Xavier NX, and multiple depth cameras. In the following sections, the procedures of estimating the growth condition of the plants and the fish are explained.

1) *Estimation of the Growth Condition of Plants:* In this study, the surface areas of leaves and stems are estimated as the growth condition of the plants. In order to measure the areas, a depth camera module (Real Sense D415 manufactured by Intel Corporation) is utilized to acquire RGB and depth images of the plants. The depth camera transmits the infrared ray to the target and each pixel of the depth image represents the distance between the depth camera and the part that reflects the ray. In order to estimate the growth condition of the plant, RGB and depth images are obtained hourly and the estimation procedure of the growth condition is performed. After that, the estimation result is sent to the management server. The detailed procedure for estimating the leaf area is described in Section IV-A.

2) *Estimation of the Growth Condition of Fish:* The depth camera using the infrared ray (i.e., Real Sense D415) cannot accurately measure the distance to the target in the water because the attenuation of the infrared ray in the water is much higher than that in the air. Therefore, in order to estimate the growth condition of the fish living in the water, the stereo camera (ZED Mini manufactured by Stereolabs) is utilized to measure the three-dimensional structure of the target based on the parallax between multiple visible light cameras. In order to install the stereo camera in the water to take images of the fish, the camera is sealed in an IP68 grade waterproof container as

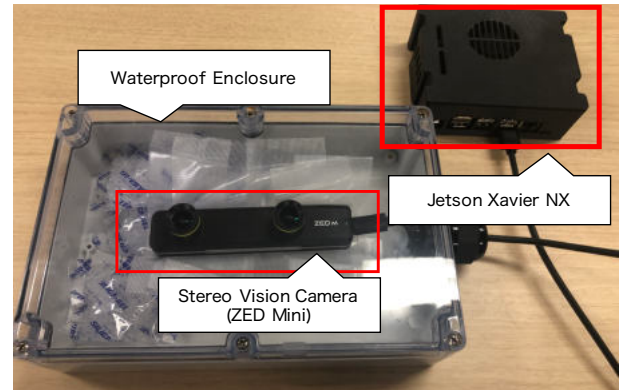


Fig. 4. Waterproof enclosure for depth camera



Fig. 5. Procedure for extracting plants areas

shown in Fig. 4 and is connected to the single board computer through a USB cable. The detailed procedure for estimating the size of the fish is described in Section IV-B.

## IV. ESTIMATION METHOD OF FISH/LEAF GROWTH USING DEPTH CAMERAS

### A. Growth Estimation Method for Plants

In this study, the proposed method quantifies the growth condition of the plants by estimating the leaf area through the analysis of RGB and depth images acquired by the depth camera. The growth estimation node acquires RGB and depth images of  $1280 \times 720$  pixels hourly. In order to estimate the growth of the target plants, the pixels corresponding with the target are extracted from the RGB and the depth images and the area of the target is calculated by analyzing the extracted pixels of the images.

1) *Extraction of Plants Regions:* The procedure for extracting the pixels corresponding with the plants from the captured images is shown in Fig. 5. First, the method extracts only the foreground part by comparing the predetermined threshold (e.g., 40 cm) with each pixel of the depth image. In the next step, the RGB value of each pixel in the foreground part is converted to the HSV value and only the green part of the pixels are identified. HSV is a method of expressing color using three elements (e.g., hue, saturation, and lightness). By converting to the HSV values, the extraction of the specific color can easily be performed. The proposed method focuses on the hue and lightness elements of the HSV values and the pixels with a hue within the range of  $30 \sim 100$  and a lightness within the range of  $40 \sim 255$  are determined as the pixels where the plants are captured.

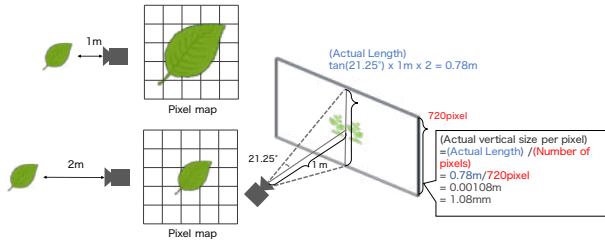


Fig. 6. Estimation of actual area of pixels

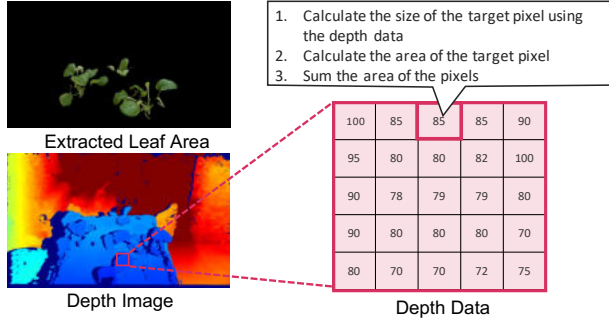


Fig. 7. Estimation of leaf area

2) *Estimation of Leaf Area:* The proposed method estimates the area of the leaf by analyzing the pixels extracted from the RGB and depth images through the previous procedure. Each pixel in the depth image indicates the distance to each part of the target, and the actual area of the part can be calculated based on the viewing angle of the camera and the distance. As shown in Fig. 6, when the distance between the camera and the part of the target changes, the actual area of the leaf corresponding with the single pixel changes. The depth camera (Real Sense D415) used in this study has a horizontal viewing angle of  $69.4^\circ$  and a vertical viewing angle of  $42.5^\circ$  when the shooting resolution is set to  $1280 \times 720$  pixels. According to the setting, it is possible to estimate the area of the single pixel from the distance. As an example, if the distance between the camera and the object is  $1m$  and the vertical viewing angle is  $42.5^\circ$ , the vertical extent of the shooting range is  $0.78m$ . Since this shooting range consists of 720 pixels in the vertical direction, the actual length of the pixels can be calculated to be about  $1.08mm$ . Finally, the area of the leaf is calculated by summing up the estimated area of the pixels as shown in Fig. 7.

### B. Growth Estimation Method for Fish

1) *Definition of Growth Condition of Fish:* In this study, body height and standard length are defined as metrics of the growth condition of the fish. When measuring the metrics, only the body part is focused without including the tail and the caudal fin. In general, these parts of the fish are damaged due to various reasons, hence the total length of the same individual may vary if these parts are considered to estimate the size of the fish. Therefore, by measuring only the body part, the growth condition can stably be estimated.

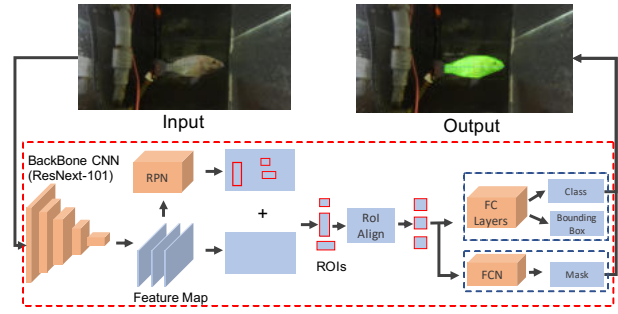


Fig. 8. Structure of the learning model of MaskR-CNN

2) *Construction of a Learning Model for MaskR-CNN:* In this study, we use MaskR-CNN proposed by He et al. to extract pixels including parts of the fish from the RGB images as well as to identify each pixel as an individual fish [8]. By inputting the RGB images into a learning model that has been trained using supervised data prepared in advance, the pixels corresponding with each fish can be output. The network model proposed in this system is shown in Fig. 8.

The learning model is constructed using images taken by a depth camera (i.e., stereo camera) from June to July 2021, and the targets of the images are three tilapias living in the aquarium of an aquaponics system installed in the laboratory where the authors belong. In this experiment, 500 RGB images are captured at different times of the day and are saved in JPEG format with  $1280 \times 720$  pixels. In addition, an image annotation tool, COCO Annotator, is used to set the correct label to the pixels that contain the fish [10]. Examples of the annotation are shown in Fig. 9.

In order to build the learning model that can handle various states of the fish, the training data includes images that are captured in various situations where the fish stays in different postures and multiple fish are overlapping. From the total of 500 images, 300 images are used for training and 100 images are used for testing to build the learning model. The remaining 100 images are used for validation to evaluate the learning model. In this study, the training of the learning model is performed based on the pre-trained model by using the COCO dataset in order to reduce the processing time and the amount of data required for the training [9]. The COCO dataset is large-scale object detection, segmentation and captioning dataset published by Microsoft.

In this study, a ResNeXt-101 proposed by Xie et al. is used as a template of the learning model [11]. The ResNeXt-101 is a type of convolutional neural network consisting of 101 layers and achieves high prediction accuracy and short execution time. Examples of the estimation results by the learning model are shown in Fig. 10. As shown in this figure, the model can accurately identify the pixels corresponding with the fish even when the fish is swimming at various angles in the water.

3) *Estimation of Body Height and Standard Length:* In the proposed method, the contours of the pixels corresponding with the fish estimated by MaskR-CNN in the depth image are used to obtain the body height and the standard length that

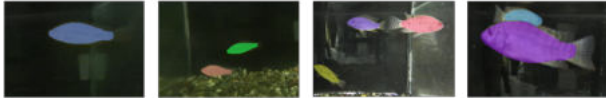


Fig. 9. Examples of annotation of fish

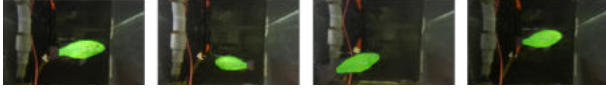


Fig. 10. Examples of detection of fish

are defined as the metrics of the growth condition as shown in Fig. 11. In order to calculate these metrics, an actual length of each pixel should be calculated from the distance data included in the depth image. The median of the distance data in the region corresponding with the fish is used as the representative distance for the instance in order to mitigate the adverse effect of the measurement error of the distance. The representative distance should carefully be decided because there are many reasons that incur the measurement error such as bubbles in the water and light reflection on the water surface. By using this representative distance, the reference length of each pixel is determined.

After that, the line segments corresponding with the body height and the standard length of each instance are extracted from the RGB image. The shape of the group of pixels corresponding with each instance identified by the MaskR-CNN is similar to an ellipse, hence the proposed method derives the bounding rectangle of the ellipse and treats the width and the height of the rectangle as the line segments of the standard length and the body height, respectively. Finally, it is possible to calculate the standard length and the body height by multiplying the reference length.

## V. PERFORMANCE EVALUATION OF PROPOSED SYSTEM

### A. Estimation Accuracy of Growth Condition of Plants

An experimental evaluation is conducted to evaluate the effectiveness of the proposed method for estimating the growth condition of the plants. The target of the growth estimation is a Japanese mustard spinach growing in the aquaponics system in the laboratory where the authors belong, and the leaf area measured manually is compared with the estimation result

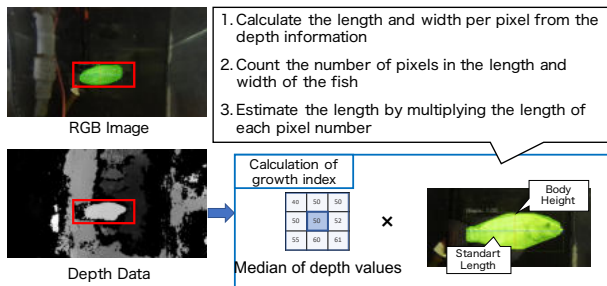


Fig. 11. Procedure of estimation of body height and standard length

TABLE I  
EVALUATION RESULT OF THE GROWTH CONDITION ESTIMATION OF THE PLANTS

ID	correct value (cm <sup>2</sup> )	Average of the estimation results (cm <sup>2</sup> )	RMSE	RMSPE (%)
1	251.9	251.44	2.16	0.86
2	24.5	22.45	2.19	8.96
3	78.25	82.73	4.83	6.16
4	69.38	77.77	9.14	13.2
5	59.0	59.25	3.14	5.43

TABLE II  
NUMBER OF DETECTED LABELS FOR EACH CATEGORY

	GT	TP	FP	FN	Recall	Precision
IoU=0.50	196	192	17	4	0.980	0.919
IoU=0.75	196	187	24	9	0.954	0.886
IoU=0.80	196	174	37	22	0.888	0.825

using the proposed method. For the evaluation, five leaves of different sizes are prepared and the leaf area of each leaf is estimated five times. As the performance measure of the estimation accuracy, we use RMSE (Root Mean Squared Error) and RMSPE (Root Mean Squared Percentage Error). Table I shows the evaluation results. As shown in this figure, the average RMSE of the estimation results of the proposed method is 6.92%. Therefore, we can conclude that the estimation accuracy of the growth condition of the plants by the proposed method is sufficiently high to observe the transition of the growth condition with the growing days.

### B. Estimation Accuracy of the Pixels Corresponding with the Fish

To evaluate the effectiveness of the proposed method for estimating the growth condition of the fish, we evaluate the estimation accuracy of the pixels in the RGB images corresponding with the fish using the MaskR-CNN. The estimation accuracy of the learning model of the MaskR-CNN is evaluated using the 100 images including 196 fish prepared in advance. In this evaluation, we use the IoU (Intersection over Union) which is a measure of how accurately the pixels of the target are detected as a measure of the estimation accuracy of the proposed method. Based on the IoU, the Average Precision (AP) is derived to evaluate the reliability of the object detection by considering the relationship between the recall and the precision [12]. In this study, we evaluate the number of detections (True Negative, False Positive, False Negative) for each label and  $AP_{50}$ ,  $AP_{75}$ , and  $AP_{80}$  are evaluated.

The evaluation results are shown in Tabs. II and III. As shown in these tables, the AP becomes 0.827 even when the threshold of the IoU is set to 0.8 (i.e.,  $AP_{80}$ ), which indicates that the fish can be detected with high accuracy. In order to improve the accuracy, we need to further increase the training data for building the learning model.

TABLE III  
FISH IMAGE RECOGNITION ACCURACY

$AP_{50}$	$AP_{75}$	$AP_{80}$
0.962	0.921	0.827

TABLE IV  
PERFORMANCE EVALUATION OF FISH GROWTH ESTIMATION SYSTEM

	Body Height	Standard Length
Correct Value	3.2(cm)	8.2(cm)
Average of the Estimation Results	3.21(cm)	7.76(cm)
RMSE	0.101	0.504
RMSPE	3.14(%)	6.15(%)

### C. Estimation Accuracy of Standard Length and Height of Fish

In the experimental evaluation, the measured value of the standard length and height of the tilapia is compared with the estimation result. The estimation of the growth condition is performed for one hour, and the 137 estimation results are collected. The estimation accuracy is evaluated by RMSE and RMSPE. The estimation accuracy of the growth condition of the fish is shown in Tab. IV. As shown in this table, the error of the estimation is 3.14% for body height and 6.15% for standard length, which indicates that the proposed method can estimate the growth condition of the fish with high accuracy.

### D. Observation Experiment Using the Proposed Method

Using the proposed method, we observe the changes in the growth rates of plants and fish and confirm that we could accurately obtain the changes in growth rates. Fig. 12 shows the relationship between the number of growing days and the estimation result of the leaf area. In addition, Fig. 13 shows the relationship between the number of growing days and the estimated size of the fish. These results show that our proposed system can clarify the growth condition of plants and fish with the passage of the growing days.

## VI. CONCLUSION

In this study, we have proposed and implemented a system for estimating the growth of plants and fish to support

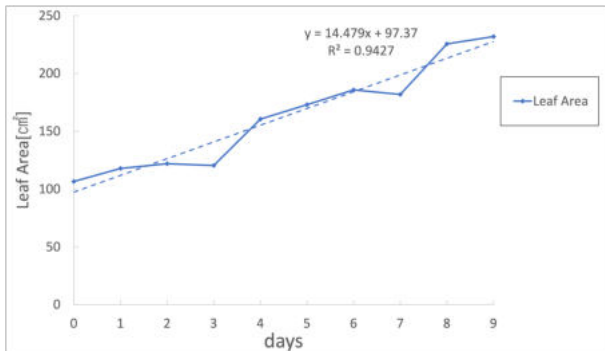


Fig. 12. Observation results of leaf area

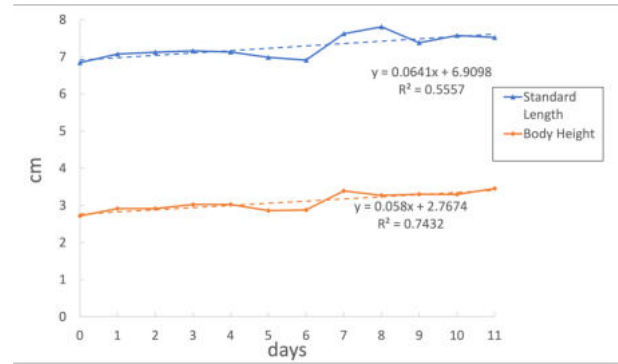


Fig. 13. Observation results of fish

aquaponics operations. The proposed method can automatically measure the size of fish in the water by using a stereo camera and deep learning technology. In the performance evaluation of the system, we have shown that the proposed method can estimate the growth rate of both plants and fish with high accuracy. In addition, we confirmed that we can visualize the growth trend by long-term observation using the proposed method.

In the future study, we will conduct further long-term observations using the proposed system to investigate the effects of the environment on the growth condition.

## REFERENCES

- [1] Ministry of Agriculture, Forestry and Fisheries, the 2020 Census of Agriculture and Forestry, Accessed: Oct. 2021. [Online], Available: <https://www.maff.go.jp/j/press/tokei/census/attach/pdf/201127-1.pdf>
- [2] S. Goddek, B. Delaide, U. Mankasingh, K. Vala Ragnarsdottir, K. Vala Jijakli, H. Jijakli, R. Thorarinsdottir, "Challenges of Sustainable and Commercial Aquaponics," *Sustainability* 7, no. 4: 4199-4224, Apr. 2015.
- [3] M. Manju, V. Karthik, S. Hariharan and B. Sreekar, "Real time monitoring of the environmental parameters of an aquaponic system based on Internet of Things," 2017 Third International Conference on Science Technology Engineering & Management (ICONSTEM), Mar. 2017.
- [4] X.Liu, D.Zhao,W.Jia, W.Ji, C.Ruan, Y.Sun, "Cucumber Fruits Detection in Greenhouses Based on Instance Segmentation," in *IEEE Access*, vol. 7, pp. 139635-139642, Sep. 2019.
- [5] K. He, G. Gkioxari, P. Dollár, R. Girshick, "Internet of Things (IOT)-Based Mobile Application for Monitoring of Automated Aquaponics System," *Information Processing in Agriculture*, Volume 6, Issue 3, Pages 375-385, Aug. 2019.
- [6] Haryanto, M. Ulum, A. F. Ibadillah, R. Alfita, K. Aji R. Rizkyandi, "Smart aquaponic system based Interet of Things (IoT)," *Journal of Physics Conference Series*, Feb. 2019.
- [7] Z. Jie Ong, A. Keong Ng, T. Ya Kyaw, "Intelligent Outdoor Aquaponics with Automated Grow Lights and Internet of Things," *IEEE International Conference on Mechatronics and Automation (ICMA)*, Aug. 2019.
- [8] K. He, G. Gkioxari, P. Dollár, R. Girshick, "Mask R-CNN," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Mar. 2017.
- [9] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona et al., "Microsoft COCO: Common Objects in Context," May. 2014.
- [10] J. Brooks, COCO Annotator, Accessed: Oct. 2021. [Online], Available:<https://github.com/jsbrooks/coco-annotator/>
- [11] K. He, G. Gkioxari, P. Dollár, R. Girshick, "Aggregated Residual Transformations for Deep Neural Networks," *Conference on Computer Vision and Pattern Recognition (CVPR)*, Mar. 2017.
- [12] T. Lin, G. Patterson,M. R. Ronchi, Y.Cui, M.Maire et al., COCO Common Objects in Context, Accessed: Oct. 2021. [Online], Available: <https://cocodataset.org>

# Image Synthesis with Single-type Patterns for Mixed-type Pattern Recognition on Wafer Bin Maps

Yunseon Byun

Department of Industrial and Management Engineering  
Korea University  
Seoul, South Korea  
yun-seon@korea.ac.kr

Jun-Geol Baek\*

Department of Industrial and Management Engineering  
Korea University  
Seoul, South Korea  
jungeol@korea.ac.kr

**Abstract**—To increase the productivity, it is important to manage yield and reduce defects in the semiconductor industry. One of the efforts is to identify defect patterns and control the cause factors that affects the defects. Many engineers inspect the quality of each chip and check the defect pattern on the wafer bin maps. To get the accurate and consistent classification results regardless of the level for domain knowledge or experience of engineers, deep learning-based models have recently been studied. Since most previous studies aim to classify the single-type defect patterns, it is needed to consider the mixed-type defect patterns together. Also, they require a lot of labeled data to train the deep learning-based classification model. However, defects occur extremely rarely in actual manufacturing process. Therefore, the method securing the higher accuracy in a situation where enough labeled data are not given is needed. This paper proposes a deep convolutional generative adversarial network for wafer map synthesis (DCGAN-WS) which generates the mixed-type patterns by synthesizing the single-type pattern and adding the pixel-wise summation. To maintain the characteristics of the binary pixel of the wafer bin maps, a thresholding technique is added. MixedWM38 dataset is used for the experiments, and it was verified that the mixed-type patterns were synthesized well. It helps to construct more robust model for single-type pattern classification and to generate the mixed-type patterns that have not occurred before. In the future, it is expected that this model addresses the problem of the lack of labeled data for defect pattern classification models.

**Keywords**— wafer bin maps, image synthesis, pattern classification, generative adversarial network

## I. INTRODUCTION

In the semiconductor industry, there are several fine and complex production processes such as photolithography, etching, deposition, and cleaning to produce a chip. To increase productivity, the engineers manage yield and control factors that affect some defects. They check whether there are any process issues that cause defects during the manufacturing process for chips. And then, they inspect on each chip through the electrical die sorting (EDS) test, and divide each chip into “pass” or “fail”. Wafer bin map is the result of the EDS test, and it is a map expressed as a binary pixel value according to the “pass” or “fail” of each chip. Defective chips identified as “fail” of each chip

form a specific pattern on the wafer bin map. The EDS test results are expressed on the circular wafer map, and there may be various patterns depending on the position of the defective chips.

Fig. 1 shows normal pattern and several defect patterns such as *Center*, *Donut*, *Edge-Loc*, *Edge-Ring*, *Loc*, *Random*, *Scratch*, and *Near-full*. Each different patterns are caused by different factors. For example, *Center* pattern occurs when the solution is not uniformly polished on the wafer map in chemical mechanical polishing (CMP) process [1]. *Edge-Ring* pattern occurs due to a misalignment between layers in the process for accumulating layers on wafer maps [1]. *Scratch* pattern occurs due to the agglomerated particles and aging of the pad in CMP process [1]. Because the cause factors are different depending on the pattern types, it is important to accurately identify the defect patterns on the wafer bin map and to control the factors for reducing the defects.

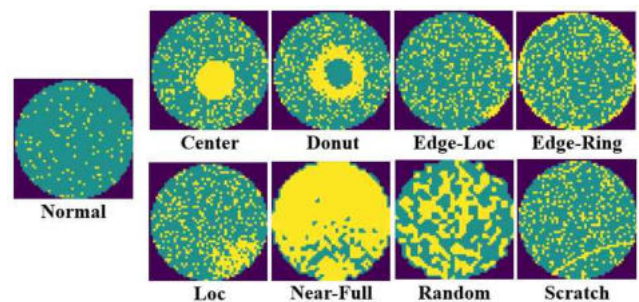


Fig. 1. Examples of the single-type defect patterns

To accurately obtain the consistent classification results regardless of the domain knowledge engineers when classifying defect patterns on wafer bin maps, many studies have been conducted to automatically classify defect patterns based on a model. There are studies that extract the features using transformation technique such as Wavelet transform and Radon transform and classify the patterns using machine learning-based models such as support vector machine (SVM), multilayer perceptron (MLP), and radial basis function (RBF) [2, 3, 4].

\* Corresponding author-Tel: +82-2-3290-3396; Fax: +82-2-3290-4550



When using machine learning-based models for classifying the patterns, there is a difficulty in selecting a proper feature extraction method. To address this problem, deep learning-based models were proposed to perform both feature extraction and classification within a model. In particular, to recognize pattern by processing wafer bin maps as images, Convolutional neural network (CNN)-based model is used because it is known to be effective in image processing [5, 6, 7]. CNN classifies the defect patterns on the wafer bin map with high accuracy.

However, deep learning models such as CNN requires a number of labeled data for training to ensure high accuracy. In practice, it is hard to obtain a lots of labeled data in manufacturing process. Accordingly, unsupervised learning-based models such as density-based spatial clustering of applications with noise (DBSCAN) and self-organized map (SOM) are utilized because they do not require label information [8, 9]. This is useful in situations where labeled data are insufficient, but there is a disadvantage that performance is somewhat degraded compared to when labeled information is used together. Therefore, it is needed to study a model capable of securing high performance in a situation where a lots of labeled data are not given.

Also, there are not only single-type patterns but also mixed-type patterns in the actual manufacturing process. As shown in Fig. 2, the mixed-type patterns can be generated when two or more single-type patterns are mixed. If the mixed-type patterns are not recognized or the mixed-type patterns are incorrectly identified as a single-type pattern, all cause factors will not be controlled and defects continue to occur. Labeled data for the mixed-type patterns are more difficult to obtain because it occurs more rarely than single-type patterns. Mixed-type patterns can be generated through various combinations of single-type pattern, but data of all combinations may not be obtained. In this situation, if only the actually obtained mixed-type patterns are used only for training the models, it is hard to identify the mixed-type patterns when the other patterns appears.

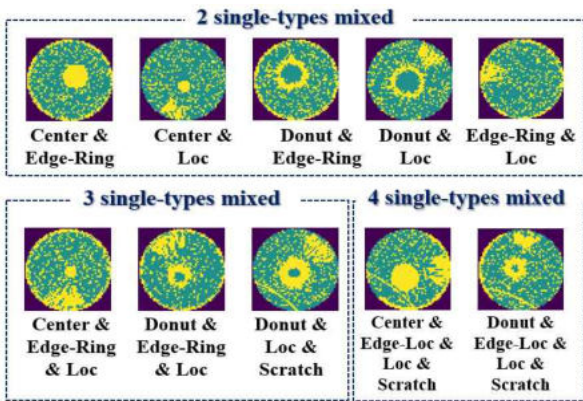


Fig. 2. Examples of the mixed-type defect patterns

Therefore, this paper proposes an image synthesis method that mixed-type patterns on wafer bin maps using only single-type patterns. Through this, a number of labeled samples that are difficult to obtain are generated. It helps to improve the classification performance of the model when a classification model is constructed using the generated data in the future. It

also helps to make a robust model that can detect patterns that have not been mixed previously.

A deep convolutional generative adversarial network-based wafer map synthetic (DCGAN-WS) model is proposed in this paper. The proposed model consists several generators (G) for each single-type patterns for making variational data of single-type patterns and a discriminator (D) for classifying the real patterns and the generated patterns. The generated single-type patterns are used in a synthetic process for making mixed-type patterns. It makes to construct a more robust classification model for the single-type patterns and generates data for various types of the mixed-type patterns. The generated data from DCGAN-WS maintains the characteristics of the binary pixels on the wafer bin maps through the thresholding techniques. Then, the patterns are synthesized through a pixel-wise summation between the variational data considering the various combinations of the single-type patterns. Using the model, some variational data for single-type patterns and various cases for mixed-type patterns can be generated in a desired amount. The proposed method of this paper helps to get all information on single-type patterns and mixed-type patterns in the situation when the single-type patterns are given only. This addresses the problem of the lack of labeled data for defect pattern classification models.

Chapter 2 describes a proposed model, DCGAN-WS for synthesizing the mixed-type patterns. Chapter 3 shows the results of the synthesized maps. Finally, Chapter 4 describes the conclusion.

## II. THE PROPOSED METHOD

This paper proposes an image synthesis method for wafer bin maps, called DCGAN-WS. DCGAN-WS has three steps, generating the variational data, thresholding for maintaining the characteristics of wafer bin maps and pattern synthesis using pixel-wise summation. The detailed description of each part are like the following.

### A. Generating the Variational Data

The first step is to generate variational data for a single-type patterns using a deep convolutional generative adversarial network (DCGAN) model. DCGAN is a variants of the GAN containing deep convolutional layers [10]. It is divided into a generator part that generates new images and a discriminator part that distinguishes the generated images from the real images. The generator is comprised of transposed convolutional layers, batch norm layers, and ReLU activations. The input of the generator is a latent vector  $z$  from a standard normal distribution and the output is a RGB image. The discriminator is comprised of strided convolutional layers, batch norm layers, and Leaky ReLU activations. The input of discriminator is a input image and the output is the probability that the input is from the real data distribution.

Generators of are constructed as many as the number of the single-type patterns. For example, when there are six single-type patterns, the number of generator is six. Each generator makes variational data for each pattern by inserting the latent vector  $z$  as input. Since the generating process from a generator and the

distinguishing process from a discriminator have adversarial learning, the generator can create variational images similar to the actual images that is difficult for the discriminator to distinguish. The overall framework of DCGAN is shown as Fig. 3.

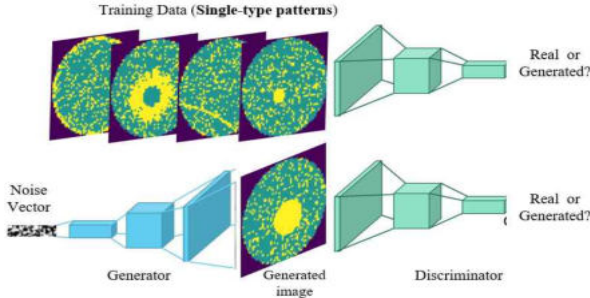


Fig. 3. The overall framework of DCGAN

### B. Thresholding Technique for Wafer Bin Map

The output images of generators from the generator of DCGAN-WS have continuous pixel values on each pixel. However, the original wafer bin maps have only binary pixel values such as 0, 1, and 2 on each chip. Generally, 0 means empty part which the remaining edge part are filled with zero as the circular wafer maps are converted into a rectangular image. 1 means the “pass” or “good” chip in EDS test. 2 means the “fail” or “defective” chip in EDS test, and we have to focus on the defective part which forms a specific defect pattern. Therefore, it is needed to revise the characteristics of the pixel values same as the original wafer maps.

We can set a threshold empirically within the range from 0 to 1. The pixels of the generated images consist of continuous values from 0 to 1. By setting two thresholds which one is for dividing 0 (empty part) and 1 (pass chip) and the other is for dividing 1 (pass chip) and 2 (fail chip). When the thresholds are set, the binary pixel values are assigned to each pixel, as shown in Fig. 4. After thresholding, the defect patterns can be clarified on wafer bin maps. Although the thresholds are defined empirically in this paper, the further studies for appropriate thresholds are necessary because the shape and size of defect patterns can be changed according to the thresholding.

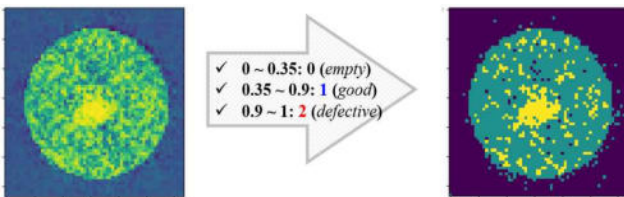


Fig. 4. Result for the thresholding technique

### C. Pattern Synthesis using Pixel-wise Summation

After passing the DCGAN and thresholding, there are many variational images for each single-type pattern. The mixed-type patterns are formed by the combinations of multiple single-type patterns. By combining the generated single-type patterns, the mixed-type patterns can be generated as much as the desired amounts.

To synthesize the single-type patterns, the pixel-wise summation is used on multiple single-type patterns. Pixel-wise summation is to add the pixel values at the same location on wafer bin maps as shown in Fig. 5. When several maps are sampled randomly from the generated maps of single-type patterns, the added values are calculated to the same location on wafer bin maps from the samples. When two single-type patterns are sampled, the mixed-type pattern which is combined by two single-type patterns can be generated. Also, when three single-type patterns are sampled, the mixed-type pattern which is combined by three single-type patterns can be generated. And then, the threshold in Section 2.B is repeated to represent the pixel values as binary type.

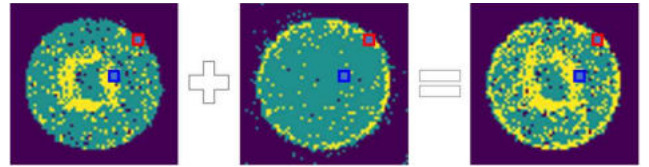


Fig. 5. The result for the pixel-wise summation for mixed-type synthesis

This procedure utilizes the generated variational data from the single-type patterns. Therefore, it helps to make robust model for single-type patterns and synthesize the various mixed-type patterns although it is same defect patterns.

## III. EXPERIMENTALS

### A. Data Description

The dataset for experiments is *MixedWM38* dataset which are used in [11]. The data contain a *Normal* pattern (no defects), 8 single-type defect patterns such as *Center*, *Donut*, *Edge-Loc*, *Edge-Ring*, *Loc*, *Near-Full*, *Random*, *Scratch*, and mixed-type defect patterns which are combined by two or three or four single-type patterns. The number of types for the mixed-type pattern combined by two single-type patterns is 13 and the number of types for the mixed-type pattern combined by three single-type patterns is 11, and the number of types for the mixed-type patterns combined by four single-type patterns is 2. The number of single-type patterns are 1,000 images on each pattern. Because this paper focuses on the defect patters, especially the mixed-type patterns, the target defect patterns are 6 patterns such as *Center*, *Donut*, *Edge-Loc*, *Edge-Ring*, *Loc*, *Scratch*.

### B. Experimental Results

The proposed model, DCGAN-WS, is constructed by three steps, generating the variational data with DCGAN, thresholding for maintaining the characteristics of wafer bin maps, and generating the mixed-type patterns by synthesizing multiple single-type patterns.

In first step, the architecture of DCGAN is same as [10]. When DCGAN is trained, the setting parameters are set like this. The epoch is 20, batch size is 100, learning rate is 0.0001, and the optimizer is Adam. In this step, the generator of DCGAN-WS generates each single-type pattern. The generated maps are shown in Fig. 6. The most generated patterns are similar to the real wafer bin maps. However, *Edge-Loc* and *Scratch* is relatively not similar to the real wafer bin maps. Although actual

*Edge-Loc* has the half of the ring pattern on the edge of the wafer map, the results for *Edge-Loc* contains some circular patterns, not ring shape. Also, the *Scratch* pattern is a linear pattern located randomly on the map. However, the result for the *Scratch* pattern shows the patterns like *Random*. Therefore, further studies for *Edge-Loc* and *Scratch* is needed to improve the performance of generating the single-type patterns, because it affects the quality of the synthesized mixed-type pattern.

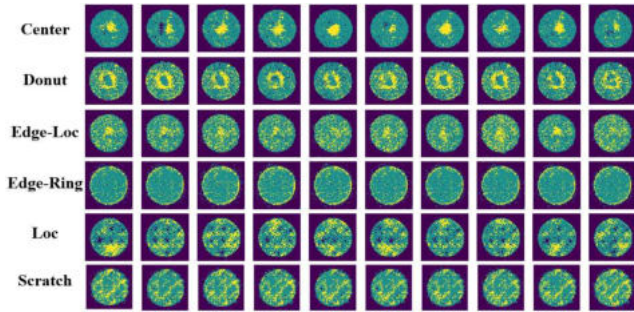


Fig. 6 The results for generating the variational single-type patterns

Using the results in Fig. 5, the variational data for each single-type pattern are utilized to synthesize the mixed-type patterns. By adding the pixel-wise summation to the randomly sampled variational data, several mixed-type patterns can be generated. After the summation, the thresholding technique is repeated to represent the characteristics of wafer bin maps. In this paper, the mixed-type patterns are defined as the synthesized results of two single-type patterns only. Fig. 7 shows the synthesized mixed-type patterns by combining the single-type patterns. *Edge-Loc* and *Scratch* which are not similar to the actual pattern in first step cannot still be mixed well. Therefore, to improve the quality of generated variational data for single-type patterns and the quality of the synthesized mixed-type patterns, it is important to improve the method for generating poor patterns for the single-type.

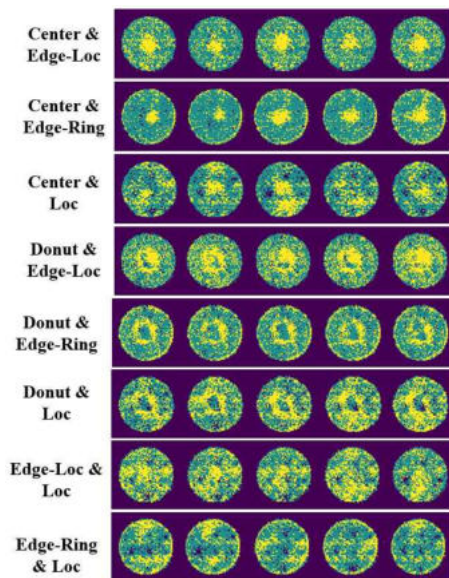


Fig. 7. The synthesized results for the mixed-type patterns

#### IV. CONCLUSION

The studies for automatic defect classification model have been conducted because it is important to check the defect patterns on the wafer bin maps and control the cause factors for yield improvement in semiconductor manufacturing. To address the limitations of the previous deep learning-based model, it is needed to consider the mixed-type defect patterns and secure the higher accuracy in situation where enough labeled data are not given. This paper proposed a deep convolutional generative adversarial network for wafer map synthesis (DCGAN-WS) which generates the mixed-type patterns by synthesizing the single-type pattern. First, DCGAN is applied to generate the variational data of the single-type patterns. The generated maps are sampled randomly and they are added using the pixel-wise summation. To maintain the characteristics of the binary pixel of the wafer bin maps, the thresholding technique is utilized. Using the MixedWM38 dataset as the experimental data, it was verified that the mixed-type patterns were synthesized well. However, some patterns such as *Edge-Loc* and *Scratch* are needed to study more for generating the maps similar to the real patterns. The proposed method helps to construct more robust model for single-type pattern classification although the number of data is lack or there is a class imbalance. Also, It helps to generate the mixed-type patterns that have not occurred before. In the future, it is expected that this model addresses the problem of the lack of labeled data for defect pattern classification models.

#### ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (NRF-2019R1A2C2005949). Also, this work was supported by Brain Korea 21 FOUR and Samsung Electronics Co., Ltd(IO201210-07929-01).

#### REFERENCES

- [1] Kim, J., Lee, Y., & Kim, H., "Detection and clustering of mixed-type defect patterns in wafer bin maps", *Iise Transactions*, 50(2), 99-111, 2018.
- [2] Wang, Y., & Ni, D., "Multi-bin wafer maps defect patterns classification", In 2019 IEEE International Conference on Smart Manufacturing, Industrial & Logistics Engineering (SMILE), 48-52, 2019.
- [3] Liu, S., Chen, F., & Chung, A. S., "Using wavelet transform and neural network approach to develop a wafer bin map pattern recognition model", In International MultiConference of Engineers and Computer Scientists, 2008.
- [4] Su, C. T., Yang, T., & Ke, C. M., "A neural-network approach for semiconductor wafer post-sawing inspection", *IEEE Transactions on Semiconductor Manufacturing*, 15(2), 260-266, 2002.
- [5] Ishida, T., Nitta, I., Fukuda, D., & Kanazawa, Y., "Deep learning-based wafer-map failure pattern recognition framework", In 20th International Symposium on Quality Electronic Design (ISQED), 291-297, 2019.
- [6] Wang, R., & Chen, N., "Defect pattern recognition on wafers using convolutional neural networks", *Quality and Reliability Engineering International*, 36(4), 1245-1257, 2020.
- [7] Kong, Y., & Ni, D., "Recognition and location of mixed-type patterns in wafer bin maps", In 2019 IEEE International Conference on Smart Manufacturing, Industrial & Logistics Engineering (SMILE), 4-8, 2019.

- [8] Jin, C. H., Na, H. J., Piao, M., Pok, G., & Ryu, K. H., "A novel DBSCAN-based defect pattern detection and classification framework for wafer bin map", *IEEE Transactions on Semiconductor Manufacturing*, 32(3), 286-292, 2019.
- [9] Lee, J. H., Yu, S. J., & Park, S. C., "Design of intelligent data sampling methodology based on data mining", *IEEE Transactions on Robotics and Automation*, 17(5), 637-649, 2001.
- [10] Radford, A., Metz, L., & Chintala, S., "Unsupervised representation learning with deep convolutional generative adversarial networks", arXiv preprint arXiv:1511.06434, 2015.
- [11] Radford, A., Metz, L., & Chintala, S., "Unsupervised representation learning with deep convolutional generative adversarial networks", arXiv preprint arXiv:1511.06434, 2015.
- [12] Wang, J., Xu, C., Yang, Z., Zhang, J., & Li, X., "Deformable Convolutional Networks for Efficient Mixed-Type Wafer Defect Pattern Recognition", *IEEE Transactions on Semiconductor Manufacturing*, 33(4), 587-596, 2020

# Evaluating Opcodes for Detection of Obfuscated Android Malware

Saneeha Khalid  
Computer Science Department  
Bahria University  
Islamabad, Pakistan  
01-284172-002@student.bahria.edu.pk

Faisal Bashir Hussain  
Computer Science Department  
Bahria University  
Islamabad, Pakistan  
fbashir.buic@bahria.edu.pk

**Abstract**—Obfuscation refers to changing the structure of code in a way that original semantics can be hidden. These techniques are often used by application developers for code hardening but it has been found that obfuscation techniques are widely used by malware developers in order to hide the work flow and semantics of malicious code. Class Encryption, Code Re-Ordering, Junk Code insertion and Control Flow modifications are Code Obfuscation techniques. In these techniques, code of the application is changed. These techniques change the signature of the application and also affect the systems that use sequence of instructions in order to detect maliciousness of an application. In this paper an 'Opcode sequence' based detection system is designed and tested against obfuscated samples. It has been found that the system works efficiently for the detection of non obfuscated samples but the performance is effected significantly against obfuscated samples. The study tests different code obfuscation schemes and reports the effect of each on sequential opcode based analytic system.

**Index Terms**—obfuscation, opcodes, malware, LSTMs

## I. INTRODUCTION

Android is the most used mobile operating system at present. It provides its users with a large number of useful and entertaining applications. These applications can be downloaded from official Google play store. Although the applications present on the play-store are very beneficial for users; but on the other hand malware writers also create applications with malicious functions. The rate of malware penetration among android applications has increased in recent past. 10.5 million android malware were found in 2019 and 0.48 million new malware were found in 2020 <sup>1</sup>.

These malwares also deploy obfuscation schemes which make their detection more difficult. Obfuscation is a method for changing the structure of code in a way that the original structure is either hidden or changed[12]. Malware writers use obfuscation for generating variants of malwares that have been registered by anti-malware products[2]. This makes them difficult to be detected by commercial anti-malware products. It has been reported by many studies that performance of anti-malware products decrease by a significant percentage against obfuscated samples. [16] [7] [5].

Code Obfuscation techniques are also very effective in deceiving the detection schemes that work on the pattern of code to identify malicious applications. Over the time many schemes have been developed that focus on the analysis of code for generation of features. Many studies [1], [4] and [9] have used code based features and have reported high accuracy for detection of android malware. However it has been reported by [4] that code obfuscation techniques can greatly influence the performance accuracy of systems that work on code based features.

Class encryption, Junk code insertion, Control Flow modifications, and Code Re-ordering are popular code obfuscation techniques[14]. In this paper the effect of these obfuscation techniques on the performance of systems that use opcode based features for designing the classification engine has been critically analyzed. In order to perform this analysis; a classifier that works on opcode based features is designed. It has been observed that the system works well with non obfuscated samples but the efficiency of the system degrades when obfuscated samples are tested. The paper tests four code based obfuscation schemes and reports the efficiency of opcode based system against each obfuscation scheme. Following are the major contributions of the paper:

- An 'opcode sequence' based malware detection engine has been designed. One-gram and two-gram opcode sequences are extracted and LSTM network is trained. This study has analyzed the efficiency of opcodes as features for 2 class (Malicious and Benign) and 4 class (Adware, Ransom, Banking Trojan and Benign) problems.
- Data set for Obfuscated samples is generated. DashO<sup>2</sup> and Obfuscapk<sup>3</sup> is used for applying code obfuscations to malicious samples. Control Flow modifications, Code Reordering and Junk Code Insertions are applied. PrGuard data set is used for Class Encryption based samples.
- The designed system is tested against each obfuscation scheme and results are reported.

The remainder of the paper is organized as follows. Section 2 describes the related work. Section 3 describes the rationale

<sup>1</sup><https://www.statista.com/statistics/680705/global-android-malware-volume/>

<sup>2</sup><https://www.preemptive.com/products/dasho>

<sup>3</sup><https://github.com/ClaudiuGeorgiu/Obfuscapk>

for the study. Section 4 describes the proposed evaluation framework. Section 5 describes the results obtained on obfuscated and non-obfuscated samples and finally conclusion from the study is presented in section 6.

## II. RELATED WORK

This section covers the studies that have used opcode sequences for classification of malicious applications. Amin et al.[1] developed an end to end opcode based system. Dex byte codes are extracted and used to predict the maliciousness of an application. Bi directional LSTM networks are deployed and opcodes are represented using one hot encoding. They have evaluated their results on Drebin , AMD and virus share data sets. However results have not been tested on Obfuscated samples. Pektas et al. [10] extracted instruction call sequences from call graphs and applied deep learning models like LSTMS and CNNs for malware classification. An accuracy of 91.42 is reported but the data set does not contain obfuscated malicious applications.

Chen et al. [4] performed a detailed analysis on the selection of opcodes as features. Only important opcodes are selected and then sequence is formulated. 3-gram opcode sequences are used. Classification is performed using using Random Forest and SVM. Authors have stated it clearly that their system is vulnerable against metamorphic malware samples. Authors in [6] and [15] extracted opcodes from classes.dex file. Opcode patterns are used as images and CNN based networks are used for classification. Authors in [9] worked on raw opcode sequences extracted from dex files. 1-gram , 2-gram and 3-gram sequences have been extracted and tested. CNNs are used for classification.

## III. RATIONALE FOR CURRENT RESEARCH

Some existing studies considered the problem of analyzing the effect of obfuscation on the detection schemes. In this section, the proposed system is compared with existing schemes and the merits of our scheme are discussed. Hammad et al. [7] have analyzed the effect of obfuscation on the performance of anti viruses. It is reported that performance of many anti viruses degrades when obfuscated samples are tested. However they have not analyzed any particular technique used by a detection engine which is specifically affected by a certain obfuscation technique.

Bacci et al. [3] have studied the impact of code obfuscation on static and dynamic machine learning based techniques. They have considered opcodes frequency and system calls as features. It is reported that obfuscation effects the static analysis based methods more adversely than dynamic analysis based methods. This study is the close to our work , however our study has specifically focused on the performance degradation of sequential deep classifiers.

To the best of our knowledge, this is the first work which has investigated the effect of Code obfuscation schemes on the performance of deep sequential classifiers given static feature set. We have explained the effect of each code obfuscation scheme on the sequential classification systems and have also

reported the performance degradation metrics against each obfuscation scheme separately.

## IV. PROPOSED EVALUATION FRAMEWORK

The purpose of our evaluation framework is to measure the effectiveness of opcode based analytic system on non-obfuscated and obfuscated samples. For this purpose an 'opcode sequence' based analytic system is designed. The system extracts opcode sequences from an application and a classification engine is trained. The samples for training include non-obfuscated malware samples. Later code obfuscation schemes that effect opcode sequences of the android applications are applied on the malware samples. The resultant obfuscated data set is then tested on the designed system and accuracy is measured. The complete experimental setup is depicted in Figure 1.

This section is arranged as follows. First subsection represents the design of the opcode based detection system. Second subsection discusses code obfuscation schemes in detail. Data set preparation and testing on obfuscated data set is also discussed in second sub section.

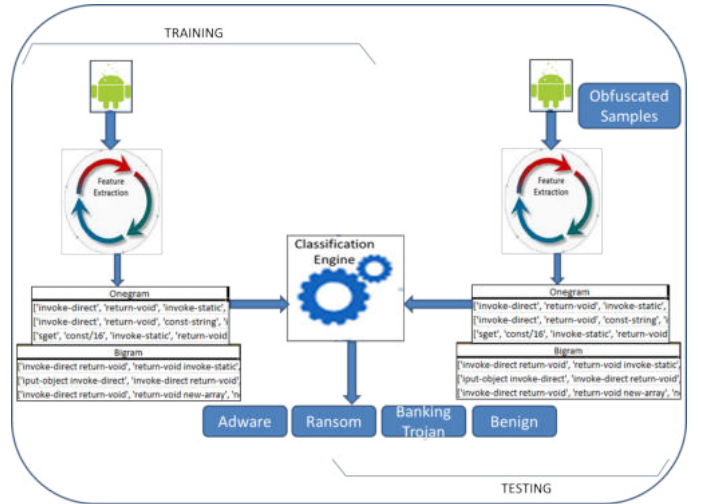


Fig. 1. Evaluation Framework for obfuscated and non-obfuscated applications

### A. Design of Opcode based analytic system

Opcodes represent the instructions used in an application. A sequence of opcodes is a better representative of an application's behavior. Many studies [1] [10] and [4] have created detection systems based on opcode sequences. In this section the design details of the proposed opcode based detection system are discussed. The discussion includes the selected sources of data for non obfuscated malicious applications, the process of feature extraction and the application of classifier.

1) *Data Set*: The malicious and benign applications used for this study are obtained from CICMalDroid2020<sup>4</sup> and Androzoo<sup>5</sup> data sets. The data is divided into groups. The

<sup>4</sup><https://www.unb.ca/cic/datasets/maldroid-2020.html>

<sup>5</sup><https://androzoo.uni.lu/>

TABLE I  
RELATED WORK

Paper Name	Features	Classifier	Obfuscation Tested	Accuracy
Amin et al.[1]	dex bytecode	bi-dir LSTM	No	99.6
Pektas et al. [10]	instruction calls	LSTM, CNN	No	91.6
Chen et al. [4]	3-gram sequences	RandomForest	No	95.3
Ren et al. [13]	dex bytecode	DexCRNN	NO	93.4
McLaughlin et al. [9]	2-gram,3-gram raw bytecode	CNN	No	95

first group contains samples of 4 categories; Adware, Ransom, Banking Trojans and Benign. In the second group data samples are labeled as malicious and benign only. The purpose of creating two groups is to verify the effectiveness of Opcode sequences for classification of 4 class and 2 class problems.

2) *Feature Extraction*: The features used in this experiment are opcode sequences. For extraction of opcodes; the apk file is first disassembled into smali files. After disassembling; the smali files are parsed for selecting the opcodes. A standard list of opcodes is used as a reference. Each extracted line from the smali file is compared with the standard opcode list and standard opcodes used in the apk are extracted.

The process is repeated for each smali file in the apk. It must be noted that opcodes are extracted in two formats; one gram and two grams. Accordingly the experiments are performed against both type of opcode sequences separately. Also note that these csvs are maintained and labeled separately for four class and two class problems.

3) *Application of Classifier*: LSTMs are an extension of RNN networks and have shown promising results in prediction of sequence data. LSTMs are widely used in other domains like speech recognition very successfully. Qui et al. [11] analyzed many studies on android malware detection and found that LSTMs usage with sequential features like opcodes or system calls traces is efficient for malware classification and categorization. Therefore LSTMs have been chosen as the classifier for this study. The input to the LSTM model is the sequence of one gram and two gram opcodes for an apk. The length of each sequence varies as different apks have different number of opcodes.

Keras sequential model is used for the design of classifier. The sequential model consists of 100 LSTM layers, one Dropout layer (for preventing over fitting), one Flatten layer and finally Dense layers. Rectified Linear Unit is used as the activation function in the Dense layers. The model is compiled with adam as optimizer and categorical cross entropy as loss function. The design of classification engine is presented in Figure 2. The compiled model is then trained on the input data. In the training phase number of epochs are set to 100 and a batch size of 128 is used.

In order to make our work helpful for other researchers; we have made the code of our system public <sup>6</sup>. The performance of the system in terms of training and testing accuracy is discussed in section 4.

<sup>6</sup><https://github.com/saneeha-amir/Obfuscated-Android-Malware-Detector>

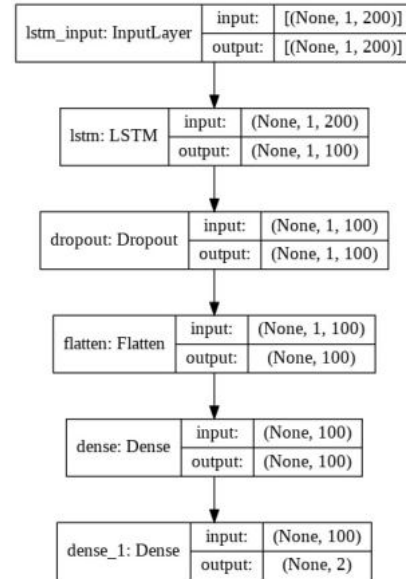


Fig. 2. Design of Classification Engine

## B. Code Obfuscation

Code obfuscation refers to changing the structure of code in order to hide the semantics. For this purpose specific classes and methods need to be accessed and changes are to be made. In order to perform code modifications, the application first needs to be disassembled. Apktool is a famous tool for disassembling of apk file. After disassembling of apk, the class files in the android application are converted into smali files; which contain assembly language instructions. The code changes are then made in the smali file. After updating the smali files the application is repacked and signed.

1) *Obfuscation schemes effecting the regular opcode structure*: Many obfuscation schemes effect the code structure of an application. This study has focused on the obfuscation schemes that particularly effect the opcode sequences of the apk file. Different obfuscation schemes are analyzed and the schemes that particularly effect the opcode sequencing are selected. The selected schemes include:

- Class encryption
- Junk code insertion
- Control flow Obfuscation
- Code re-ordering

a) *Class Encryption*: Class encryption is an obfuscation technique in which the class files of the android application are

encrypted. In this way the code becomes unavailable for static analysis. It is the most powerful obfuscation technique[8]. This technique changes the hash of file and also effects all static code based analytic methods. Class encryption strongly effects the opcode based detection scheme as the extracted opcodes are meaningless due to encryption. DexGuard supports class encryption. It is a paid tool and is not available for educational purposes. However PraGuard data set contains obfuscated samples for class encryption.

b) *Junk code-Insertion - Non-functional methods:* Junk code insertion is one the code obfuscation techniques. In this technique some non-functional code is inserted into the application. Insertion of non-functional methods is one of the techniques for junk code insertion. Non-functional methods may perform some trivial functionality like printing a string. Addition of these methods do not effect the overall working but effects the method table in the Dalvik byte code [17]. As a result the signature of application is also changed. ADAM is the tool that supports this technique for junk code insertion. The tool parses the smali files and inserts non functional methods before the constructor.

c) *Junk code-Insertion - 'No Operation'(Nop) Instructions:* Nop is a valid opcode that does noting. It is a trivial junk code insertion technique. By inserting NOP instructions the hash of the file is changed. The sequence of original opcodes is also changed by adding NOP opcodes in different methods. Obfuscapk[2] is the tool that supports the insertion of NOP in different methods of applications. This tool parses all the smali files of the application and inserts a random number (1 - 5) of nop opcodes after a valid opcode in every method.

d) *Junk code-Insertion - Over Loaded Methods:* Method overloading is a useful feature of Java programming language. In order to add junk code inside application; overloaded method insertion can also be used. The insertion of overloaded methods change the hash of the application and also effects the opcode based static analysis systems. Obfuscapk [2] supports the insertion of overloaded methods. In order to insert an overloaded method; the classes in each smali file are read. The methods in the classes are analyzed and overloaded versions are generated. The overloaded methods have different number of arguments and void return type. The body of these methods is then filled with some arithmetic instructions.

e) *Junk code-Insertion of Arithmetic Branch:* Arithmetic branch is a path based on the result of some arithmetic operation. They can be used as Junk code if the arithmetic operation is never true. Insertion of arithmetic branches changes the hash of code and sequence of opcodes. Obfuscapk [2] supports the insertion of arithmetic branches. The tool parses the smali files and locates the methods which are not abstract. Inside these methods an addition and remainder operation is inserted. 'if' condition is applied and a goto instruction is inserted in the else part which is never taken.

f) *Control flow Obfuscation:* In control flow based obfuscation techniques; the control flow of application is changed by inserting iterative structures, goto statements or code branching instructions. All these techniques change the original op-

code sequence of the application. In order to apply the control flow based obfuscation techniques; application is broken to smali and then changes are applied and later app is rebuilt and resigned. Obfuscapk , DashO , Allotari are some of the tools that support control flow obfuscation.

g) *Code re-ordering:* In code re ordering , the order of the instructions and methods is changed. As the code is re-ordered; the hash of the file changes and the sequence of opcodes also changes. This technique is also applied on the disassembled smali files. Obfuscapk supports code reordering obfuscation.

2) *Data set for Obfuscated Samples:* For the generation of obfuscated samples DashO and Obfuscapk have been used. DashO is a commercial tool and only trial version is available. Obfuscapk is an open source tool and is available on gitHub. Code reordering, Junk Code Insertion and Control Flow modifications are applied on the data set using Obfuscapk and DashO. Samples for class encryption have been obtained from PraGuard<sup>7</sup>data set.

3) *Testing on Classifier:* After the generation of obfuscated samples; two-gram opcode sequences are extracted against each sample. A csv file is maintained against each obfuscation category. Each category of samples is tested against the trained classification engine and it has been noted that different obfuscation schemes have different effect on the performance of the system. The results of testing are presented in section 4.

## V. RESULTS AND DISCUSSION

In this section the accuracy of the designed system on obfuscated and non-obfuscated applications is shown. It has been observed that opcode sequences are meaningful features as the system predicts the 4 class category samples with an accuracy of 92.5 percent and 2 class category is predicted with an accuracy of 97.2 percent. Hence it can be concluded that opcode sequences when used with a powerful sequence based classifier like LSTM forms an efficient malware detection system. The training and validation accuracy of the system is presented in Figures 3, 4, 5 and 6.

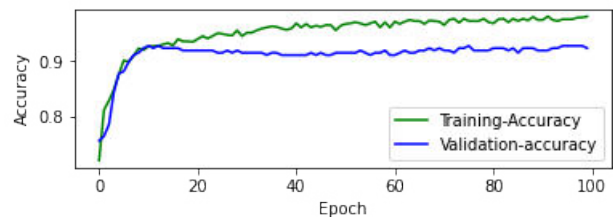


Fig. 3. One-Gram 2 class Model Accuracy

<sup>7</sup><http://pralab.dice.unica.it/en/AndroidPRAGuardDataset>



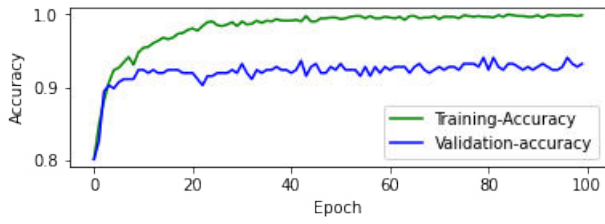


Fig. 4. Two-Gram 2 class Model Accuracy

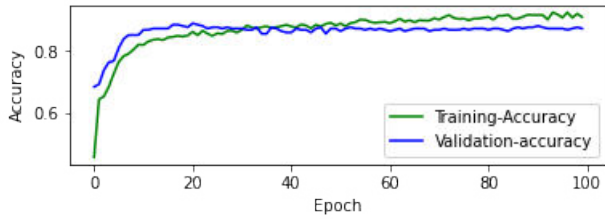


Fig. 5. One-Gram 4 class Model Accuracy

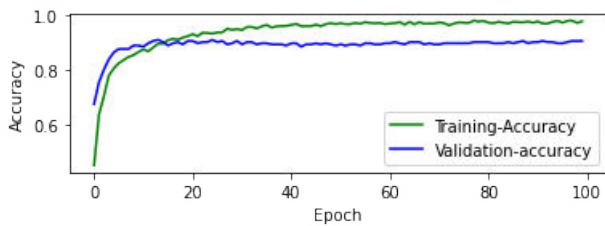


Fig. 6. Two-Gram 4 class Model Accuracy

The performance of system on non-obfuscated test data is shown in table II.

TABLE II  
CLASSIFICATION ACCURACY FOR 1-GRAM AND 2-GRAM OPCODES

Type of Data	Features Type	Precision	Recall	F-Score
Non-Obfuscated 4 class	one-gram	0.88	0.87	0.874
Non-Obfuscated 4 class	two-gram	0.90	0.91	0.904
Non-Obfuscated 2 class	one-gram	0.93	0.935	0.932
Non-Obfuscated 2 class	two-gram	0.94	0.95	0.944

The trained system is then tested on obfuscated samples. As mentioned in previous section that a data set containing obfuscated samples was created for testing the designed system. The data set contains samples for Junk Code insertion techniques, Code re-ordering, Code encryption and class reordering obfuscation techniques. After the generation of data set; feature extraction is run on these samples. 2 gram opcode sequences are extracted from samples against each obfuscation scheme. The resultant feature set is then tested on the trained LSTM network. Following evaluation metrics are used for reporting the performance of designed system on obfuscated samples:

- Accuracy : The percentage of correctly identified both positive and negative samples

TABLE III  
EFFECT OF OBFUSCATION

Obfuscation Technique	FPR (False Positive Rate)	FNR (False Negative Rate)
Class Encryption	0.83	0.80
JNK(Nop Opcode)	0.16	0.05
JNK (De-Functional methods)	0.30	0.20
JNK (Over Loaded methods)	0.28	0.25
JNK (Arithmetic Branches)	0.16	0.15
Control Flow	0.35	0.27
Code Re-ordering	0.20	0.20

- FPR (False Positive Rate): Rate of incorrectly predicting positive class
- FNR (False Negative Rate): Rate of incorrectly predicting negative class

The accuracy obtained with obfuscated samples is shown in Figure 7. The values for False Positive Rates (FPR) and False Negative Rates (FNR) are listed in Table III. It has been observed that the performance of the system is effected when obfuscated applications are tested. Different obfuscation schemes effect the working of the system in different ways.

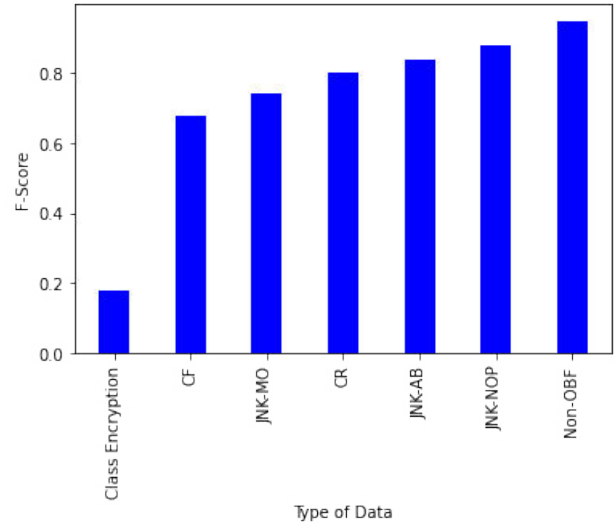


Fig. 7. Accuracy Variations with Obfuscated Samples

The performance of the classification engine is most effected by class encryption. As complete class is encrypted; therefore the semantic of opcodes is no longer preserved. So the efficiency of the system drops by a significant percentage. After class encryption; Control Flow obfuscation effects the working of system most. Here the opcode sequence is effected by adding goto statements, for loops and conditional statements. By adding these structures the opcode sequence is changed and therefore the efficiency of the system drops by a significant percentage.

Detection of samples with Code reordering is also effected by some percentage. Here the data and methods inside the classes are reordered and hence the sequence is altered. Junk code schemes of Inserting overloaded and de-functional methods also effect the schemed by a significant percentage.

Junk code schemes of NOP insertion and Arithmetic branching have lesser impact on the efficiency of the system.

It must be noted that 2-gram opcode sequences are used for testing of obfuscated samples. From these results it can be concluded that 'opcode sequence' based detection schemes are very efficient for the prediction of malicious applications but when obfuscated samples are supplied; their performance gets effected. From the results of this study it can be concluded that static opcode based techniques become less efficient against code obfuscations.

However we suggest that run time extraction of opcodes can help in overcoming this problem as many obfuscations schemes become ineffective when code is analyzed at run time. For example in case of class encryption; decrypted code can be extracted and analyzed dynamically. Similarly the control flow modifications effect the static analysis techniques only as the change of flow is not actually executed at run time. Same is the case with arithmetic branch insertions and over loaded method insertions as these branches and methods are never executed. Run time extraction and analysis of code can help the analytic engine to focus on the actual code of the application and can fade away the effects generated by obfuscation.

## VI. CONCLUSION

In this study an 'opcode sequence' based analytic engine is designed. The system is trained and tested on non obfuscated samples and the results are very promising. But it has been observed that the efficiency of the system drops significantly when samples with different obfuscations are supplied. From this study it can be concluded that opcodes are an efficient feature set specially when used in form of sequence. In order to formulate obfuscation resilient system, it is proposed that code based features like opcodes should be collected dynamically at run time so that the effect of obfuscation is minimized.

## REFERENCES

- [1] Muhammad Amin, Tamleek Ali Tanveer, Mohammad Tehseen, Murad Khan, Fakhri Alam Khan, and Sajid Anwar. Static malware detection and attribution in android byte-code through an end-to-end deep system. *Future Generation Computer Systems*, 102:112–126, 2020.
- [2] Simone Aonzo, Gabriel Claudiu Georgiu, Luca Verderame, and Alessio Merlo. Obfuscapk: An open-source black-box obfuscation tool for android apps. *SoftwareX*, 11:100403, 2020.
- [3] Alessandro Bacci, Alberto Bartoli, Fabio Martinelli, Eric Medvet, Francesco Mercaldo, and Corrado Aaron Visaggio. Impact of code obfuscation on android malware detection based on static and dynamic analysis. In *ICISSP*, pages 379–385, 2018.
- [4] Tieming Chen, Qingyu Mao, Yimin Yang, Mingqi Lv, and Jianming Zhu. Tinydroid: a lightweight and efficient model for android malware detection and classification. *Mobile information systems*, 2018, 2018.
- [5] Melissa Chua and Vivek Balachandran. Effectiveness of android obfuscation on evading anti-malware. In

- Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*, pages 143–145, 2018.
- [6] Abdulbasit Darem, Jemal Abawajy, Aaisha Makkar, Asma Alhashmi, and Sultan Alanazi. Visualization and deep-learning-based malware variant detection using opcode-level features. *Future Generation Computer Systems*, 125:314–323, 2021.
- [7] Mahmoud Hammad, Joshua Garcia, and Sam Malek. A large-scale empirical study on the effects of code obfuscations on android apps and anti-malware products. In *Proceedings of the 40th International Conference on Software Engineering*, pages 421–431, 2018.
- [8] Davide Maiorca, Davide Ariu, Iginio Corona, Marco Aresu, and Giorgio Giacinto. Stealth attacks: An extended insight into the obfuscation effects on android malware. *Computers & Security*, 51:16–31, 2015.
- [9] Niall McLaughlin, Jesus Martinez del Rincon, Boo-Joong Kang, Suleiman Yerima, Paul Miller, Sakir Sezer, Yeganeh Safaei, Erik Trickel, Ziming Zhao, Adam Doupé, et al. Deep android malware detection. In *Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy*, pages 301–308, 2017.
- [10] Abdurrahman Pektaş and Tankut Acarman. Learning to detect android malware via opcode sequences. *Neurocomputing*, 396:599–608, 2020.
- [11] Junyang Qiu, Jun Zhang, Wei Luo, Lei Pan, Surya Nepal, and Yang Xiang. A survey of android malware detection with deep neural models. *ACM Computing Surveys (CSUR)*, 53(6):1–36, 2020.
- [12] Vaibhav Rastogi, Yan Chen, and Xuxian Jiang. Catch me if you can: Evaluating android anti-malware against transformation attacks. *IEEE Transactions on Information Forensics and Security*, 9(1):99–108, 2013.
- [13] Zhongru Ren, Haomin Wu, Qian Ning, Iftikhar Hussain, and Bingcai Chen. End-to-end malware detection for android iot devices using deep learning. *Ad Hoc Networks*, 101:102098, 2020.
- [14] Kimberly Tam, Ali Feizollah, Nor Badrul Anuar, Rosli Salleh, and Lorenzo Cavallaro. The evolution of android malware and android analysis techniques. *ACM Computing Surveys (CSUR)*, 49(4):1–41, 2017.
- [15] Junwei Tang, Ruixuan Li, Yu Jiang, Xiwu Gu, and Yuhua Li. Android malware obfuscation variants detection method based on multi-granularity opcode features. *Future Generation Computer Systems*, 129:141–151, 2022.
- [16] Yinxiang Xue, Guozhu Meng, Yang Liu, Tian Huat Tan, Hongxu Chen, Jun Sun, and Jie Zhang. Auditing anti-malware tools by evolving android malware and dynamic loading technique. *IEEE Transactions on Information Forensics and Security*, 12(7):1529–1544, 2017.
- [17] Min Zheng, Patrick PC Lee, and John CS Lui. Adam: an automatic and extensible platform to stress test android anti-virus systems. In *International conference on detection of intrusions and malware, and vulnerability assessment*, pages 82–101. Springer, 2012.

# Procedural Generation of Game Levels and Maps: A Review

Tianhan Gao  
Software College  
Northeastern University  
Shenyang, China  
gaoth@mail.neu.edu.cn

Jin Zhang  
Software College  
Northeastern University  
Shenyang, China  
2071344@stu.neu.edu.cn

Qingwei Mi  
Software College  
Northeastern University  
Shenyang, China  
2110491@stu.neu.edu.cn

**Abstract**—So far, physical labor has ensured that the quality and quantity of game content match the needs of the game community. However, due to the exponential growth of gaming population and production costs in the past decade, it is facing new scalability challenges. Procedural Content Generation (PCG) can meet these challenges by generating game content in a fully autonomous or hybrid-led manner. Game level and map generation is a sub-field of PCG. In this paper, we summarize the ideas and main processes of the procedural generation methods of game levels and maps for racing games, platform games, and open world games, analyze the current development. Finally, we conclude the cognition and prospects of the procedural generation of game levels and maps.

**Keywords**—procedural content generation; game levels and maps; game artificial intelligence

## I. INTRODUCTION

Procedural content generation (PCG) [1] is a hot sub-field of game AI, which refers to completely autonomous or limited human-controlled methods for generating game content. PCG can help designers create content and enhance the creativity of individual creators, which eliminates the needs of human work. PCG can also create a new type of game that does not end. Combined with player modeling, PCG can create adaptive player games. Through code verification, PCG can help researchers understand the designing process and creativity. Game content is the key to ensuring the player experience, covering levels, maps, rules, maps, plots, items, tasks, music, weapons, vehicles, characters and other detailed branches. Lisapis et al. [2] have identified six creative areas in the game, including levels and maps, auditory effects, visual effects, rules, narratives, and the game itself. This paper will focus on the procedural generation of game levels and maps.

Game levels and maps generation is the most popular area of PCG so far [3] because in the game, levels and maps are equally essential elements as the rules of the game and are the main ways to drive players to interact. Different levels and maps designed under a fixed game mechanism will change the gameplay and player experience. PCG can generate two-dimensional images of Super Mario Bros. simple platform game levels; it can also generate the constrained two-dimensional space in the Candy Crush Saga, the three-dimensional huge urban space in the Assassin's Creed and the Call of Duty, and even the two-dimensional fine structure in the Angry Birds and the open world based on voxel in the Minecraft. At the same time, PCG's commercial applications already exist in the industry, including Rogue and the Dark Destroyer series inspired by Rogue, and more recently Civilization IV and Minecraft.

At present, the research hotspot of PCG of levels and maps is how to generate diversified or adaptive content in a completely autonomous [4] or mixed-dominant manner [5].

Content generation technology and content evaluation are two key points. In terms of generation technology, the main methods currently include technologies based on search, solver, grammar, cellular automata, fractal and deep learning. Recently, some people have also tried to generate game content through reinforcement learning. Due to the complexity and subjectivity of the group structure of game players and the quality of content may be affected by algorithms and their implied randomness, the evaluation of generated content is a challenging process at present.

This paper focuses on the procedural content generation of game levels and maps, aiming to summarize previous studies. At present, in the field of game levels and maps generation, there are few reviews based on the classification of game types. Therefore, this paper introduces the current research methods according to game types, hoping to combine the existing reviews in the current field to have good inspiration for readers.

## II. PROCEDURAL GENERATION METHODS

This paper introduces three classic game types of levels and maps generation methods, including racing games, platform games, and 3D open world games, including a variety of PCG methods. The evaluation methods of generating content are mainly based on representation, agent testing, and player testing.

### A. Racing Game

Racing Game is a type of electronic game, which mainly competes with the speed of the first person or the third person. The reason for choosing to research racing games is that the game type can be directly mapped to the real world, which is of special significance. Racing games are mainly car racing games, as well as some unconventional flying racing games, science fiction racing games, and special racing games. Taking the car racing game as an example, this paper introduces the procedural content generation of tracks. The current PCG research on tracks is relatively few, and most of the methods are based on search, aiming to pursue diversified, personalized high-quality tracks.

A study [6] has proposed a Cascading Elitism algorithm to generate 2D tracks in pursuit of personalized racetrack generation. The algorithm is essentially a variant of an evolutionary algorithm for multivariate problems. The main content of the algorithm is to sort the population individuals according to the fitness standard defined for each generation, leaving a specified number of individuals, while the others are discarded. If there are other fitness standards, the process is repeated according to the importance of fitness, and then the final surviving individuals are used to supplement the population to the initial number through mutation. In the whole process, crossover variation is not used. The track is

encoded as the sequence of fixed-length fragments. The length of each fragment is fixed and limited to three curvatures. Each fragment is mutated from a certain probability to other types of the fragment. The author defines three fitness degrees including progress difference (the difference between the actual completion distance and the target distance of the racing controller), maximum speed and total progress variance (the track with high progress variance is more challenging). The racing controller is trained according to the player's performance, and finally the controller can finish the journey close to the person used for training on the same track and in the same period. Then the performance profile of the controller on the track is used as fitness to screen the track population, and finally the track that meets the player's technical level is obtained. The result is shown in Fig. 1.

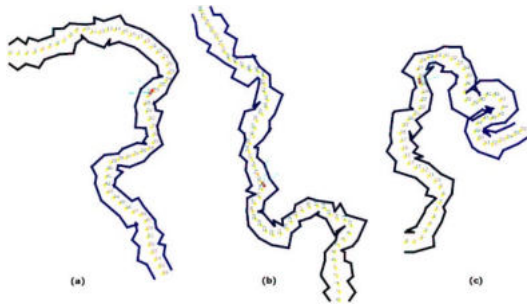


Fig. 1. Three evolved tracks: (a) evolved for a bad player with target progress 1. 1, (b) evolved for a good player with target fitness 1. 5, (c) evolved for a good player with target progress 1. 5 using only progress fitness [6].

Reference [7] has also used the Cascade Elitism algorithm to generate 2D tracks, and further pursued personalization based on reference [6]. Based on the racing controller in Reference [6], the authors improve it to enable it to imitate the driving style of players, such as running in the middle of the straight road but running near the inner wall in the smooth curved road. And the authors change the fitness ranking, followed by progress difference, total progress variance, maximum speed. Each track segment is represented by two control points. The complete track contains 30 track segments. The mutation is realized by perturbation of the position of the control points. The experiment explores the effects of different track coding representations and different mutation operations on the generation of tracks. The first method represents the track as a set of points in the space. The initial population includes the round-angle rectangular track and the track generated by the round-angle rectangular track after the random walk, and the Gauss mutation method is used to disturb the coordinates of points. The second method represents the track as a group of points with radial configuration under polar coordinates, and the initial population is a circular track. The mutation randomly changes the distance between the point and the track center, and the angle position remains unchanged.

Cardamone et al. [8] have used interactive evolution to generate 3D track in Torcs environment. The authors build an online platform in the experiment. Users can participate in the process of interactive evolution to generate the track through the functions provided by the system. There are two modes of track evolution: single-user mode and multi-user mode. In the single-user mode, the user fully controls the evolution process and defines the fitness of the track by scoring 1-5 points for the track or choosing whether to like (like corresponding 5 points, do not like the corresponding 1 points). In multi-user

mode, each user participates in scoring for the track, and the track fitness depends on the scoring average. In the evolutionary process, the racetrack genotype is represented as a set of control points in polar coordinates, and the phenotype is an ordered list of segments in the Torcs environment. The experiment shows that this interactive evolutionary strategy can improve user satisfaction and generate a track that meets the subjective wishes of players.

Loiacono et al. [9] have extended the radial coding method of track and combined information entropy to generate diversified tracks. The authors discuss the relationship between the curvature distribution, velocity distribution and the diversity of the track, and put forward the information entropy of curvature distribution and velocity distribution as two fitness measures of the track. The authors discuss the evolution results of the track by maximizing the curvature distribution entropy and the velocity distribution entropy, respectively, and the evolution of the track by maximizing the two at the same time. The genotype of the track is expressed as a set of control points in polar coordinates, and the phenotype is a second-order Bessel curves sequence. The experiment also verifies the influence of different numbers of control points on the track generation and tests whether the defined fitness is consistent with the player's experience.

A study [10] has proposed technology for real-time adjustment of game experience in games. Combining the gameplay data from the game and the data provided by the sensor, the relevant features are extracted through machine learning technology to construct the player model, and then the next stage of the track is generated according to the player's performance. Building a player model includes the following processes. Feature extraction, that is, extracting features from user input, game output, eye tracker, and head pose of corresponding track segments from raw data provided by game APIs and sensors. Calculating the performance target, that is, the state of the shortest time of the road section is set to the best player state, and the characteristics of the best player state on the track segment are used as the best performance target. Once the player's time on the track segment is shorter, the best performance target will be updated. Building a weight model, that is, calculate the weight of each feature in each track segment. Then combined with the theoretical framework of behavioral analysis, the feature subset and three high-level aspects of the user model are corresponding, namely experience, exploration and physiological attention; Finally, combined with the theory of flow, the player's demand for each section is judged, including maintaining the same, easier or more challenging track. The track is represented as a sequence of track segments, each segment in the algorithm is represented as a nine-order Bessel curve. When a simple segment is needed, calculate the mean path the player has taken through that segment before. This mean path serves as the optimal path to the center of the new segment that is created. If the player needs a more challenging segment, the difficulty of the road is increased by increasing the angle of the curve or the number of turns.

### B. Platform Game

Platform Game is a sub-category of action games. Representative works such as Super Mario Bros. The main game mode is to move and pass-through various obstacles on the suspended platform in various ways on the 2D plane. The game's environment is usually set with uneven terrain. To cross them, players need to manipulate the characters to jump

or climb between these terrains. This article will take Super Mario Bros. as an example to introduce the PCG of the level of the platform game. At present, there are many PCG methods at the Super Mario Bros. level, and this paper summarizes seven methods.

Reference [11] has generated levels based on Multi-dimensional Markov chains (MdmCs) [12]. Generate levels in three ways. (1) Generating and testing, that is, simply regenerate the whole content until the required constraints are met. (2) Discovering and regenerating the part of the illegal constraint. (3) Incremental method, that is, generate and check the level part. The level of Super Mario Bros. is represented as a two-dimensional array, each cell corresponds to a tile, and sentinel elements are added to the left and bottom of the level. The constraints include aesthetic constraints and playability constraints. Aesthetic constraints mainly refer to no damaged pipes. Playability constraints include playability, number of pipes, number of gaps, and the maximum length of gaps. But not all the three generation methods support all constraints. The experiment lists the constraint subsets that each method matches.

Hauck et al. [13] have proposed a levels generation system based on graph grammar. The system can extract data from the input map, process the data, and finally recombine new levels. The level is represented as a graph. Each game tile is associated with the nodes in the graph, and the edges of the nodes are connected to its adjacent tile nodes. As shown in Fig. 2, the algorithm includes three processes: structural identification, structural matching and level generation. There are three inputs in the structure recognition phase, including a series of levels, the minimum number of structures to be identified and the maximum edge length of the structure. The method is to select as far as possible equidistant  $n$  non-air tiles in the level, create nodes for these tiles, indicating that the tile is in  $(x, y)$  coordinates, and then expand outward from these nodes at the same time. The end condition of the expansion is to collide with other structures or reach the maximum edge length. After the end of the expansion, a connector node is created for each structure and is placed in the  $(x, y)$  position of the opposite collision structure node. The structure matching stage evaluates which institutions can be connected through the structure consistency and accessibility, in which the structure consistency is determined by the location of the connector, and the accessibility is verified by the reachability concept [14], forming a list of each structure that can be connected to other structures, and the probability distribution can be set. In the level generation phase, generate new levels based on the syntax obtained in the previous two phases. There are two constraints in this phase. The first is the availability of connector nodes (ensure at least one entry point for substitution at every step of the generation). Second, preventing the overlap of a new joining structure with a structure joined at a previous step of the generation. For the second constraint, the authors discuss whether the overlap of air tiles is regarded as an illegal constraint and analyze the experimental results.

Green et al. [15] have used the FI-2Pop [16] evolutionary algorithm to focus on the generation of playable levels similar to input levels in the pass action sequence. Mario agent will trigger actions when playing games, such as jumping, eating gold coins and so on. The genotype of the level is encoded as the action sequence corresponding to the contained scene. The experiment uses both crossover and mutation for level

evolution. Crossover allows the exchange of any number of scenes between checkpoints. Mutation refers to the operation of adding or deleting the scene and changing the structure of the scene. Playability is guaranteed by the threshold of average completion of the agent after running  $N$  times on the level. Playability depends on the average completion degree of the agent after running  $N$  times on the level. The similarity is measured by comparing the error between the input action sequence and the agent's actual action sequence. Errors include missing action and redundant action. The fitness function is proposed from two aspects of errors, and finally generates playable levels similar to the input level in the sequence of the pass actions.

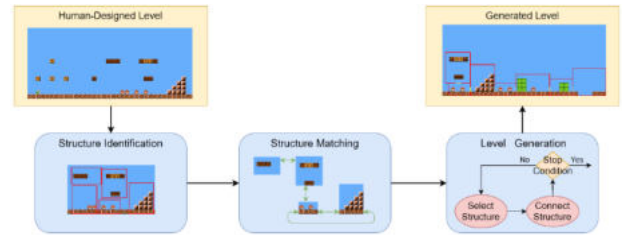


Fig. 2. The three main stages of the proposed system [13].

Summerville et al. [17] used Long Short-Term Memory networks (LSTMs) [18] to generate Mario levels. As shown in Fig. 3, the input layer of the network is composed of one-hot vector encoding, and each kind of tile has a unique encoding. The authors consider three factors in the generation process, including the route to generate the tiles: bottom-to-top or snaking generation, whether the training set contains the player's pass path, and whether column depth information is considered in the training process. The authors train the generator networks for each of the eight scenarios and test their prediction accuracy on the test set. Among them, The Snaking-Path-Depth has the lowest error. At the same time, the experiment also shows that including the player's pass path in the training set can improve the playability of the generated levels.

In Reference [19], the idea of the LVE algorithm [20] was introduced into the generation of levels, and CMA-ES [21] was used to explore the potential vector space in the GAN network [22] to generate the levels with the specified attribute target. As shown in Fig. 4, the method is divided into two stages. Firstly, the GAN network is trained using the existing Mario levels, which are encoded into two-dimensional arrays. After the generation network training is completed, the exploration process of the potential space is placed under the CMA-ES evolutionary control, and the fitness function based on the representation, and the fitness function based on the agent test are used to generate levels with specific properties.

Fontaine et al. [23] have used the most advanced quality diversity algorithm (QD) [24] to explore the potential vector space of GAN (DCGAN) and generated a high-quality set of levels with different eigenvalues of specified feature dimensions. The authors use three quality diversity algorithms include MAP-Elite [25], MAP-Elite (line) [26] and CMA-ME [27], Random and CMA-ES [28] methods to explore the potential vector space. The generation targets are set based on representation, agent test, KL divergence [29], and the generators are evaluated for different targets. The evaluation indexes include the percentage of playable cards, expression range, QD-Score [30], and more. The experimental results show that QD algorithms MAP-Elite, MAP-Elite (line) and

CMA-ME are superior to CMA-ES and random search in finding high-quality scenarios with specified attribute dimensions. In addition, CMA-ME is superior to other test algorithms in the diversity and quality of generated scenes, as is shown in Fig. 5.

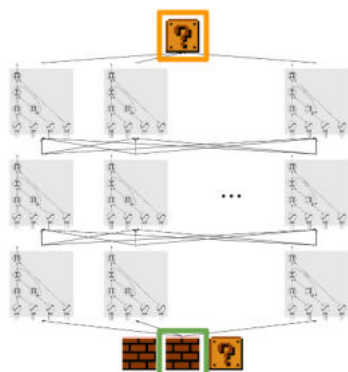


Fig. 3. Graphical depiction of our chosen architecture. The green box at the bottom represents the current tile in the sequence, the tile to the left the preceding tile, and the tile to the right the next tile. The tile in the top orange box represents the maximum likelihood prediction of the system. Each of the three layers consists of 512 LSTM blocks. All layers are fully connected to the next layer [17].

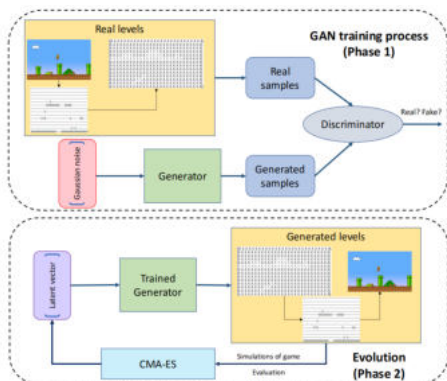


Fig. 4. Overview of the GAN training process and the evolution of latent vectors [19].

Sarkar et al. [31] have used VAE (CVAE) to generate Mario levels. The experiment uses binary vector coding conditional labels to control the game elements and design patterns contained in the generated results. For game elements, the length of the conditional label vector is 5, which corresponds to five game elements: enemy, pipeline, coin, fragile brick and question mark brick, respectively. 0 / 1 represents the absence/existence of game elements. For SMB design patterns, the authors pick 10 such patterns based on the 23 described by Dahlskog and Togelius (2012), such as Enemy Horde (EH): a group of 2 or more enemies, Gap (G): 1 or more gaps in the ground. Each input data involved in training CVAE is associated with a label vector. Then the potential variables of random sampling and the associated conditional labels are input into the trained decoder, which can realize the controllability of generating level fragments. This method can also be used to mix different platform games.

### Open World Game

Open World game, also known as Free Roam, is a kind of game level designed in which players can freely roam in a virtual world and freely choose the time and way to complete

the game task. This section will introduce the application of PCG in open world maps from three aspects: terrain, architecture and cave.

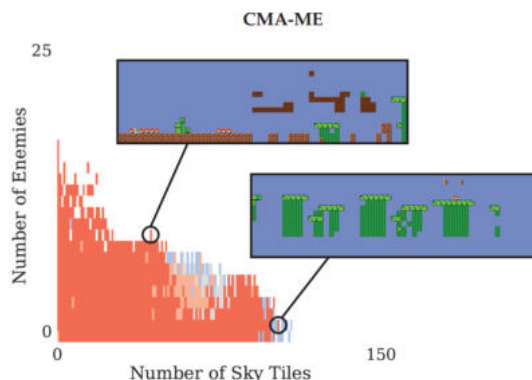


Fig. 5. Mario scenes returned by the CMA-ME quality diversity algorithm, as they cover the designer-specified space of two level mechanics: number of enemies and number of tiles above a given height. The color shows the percentage of the level completed by an A\* agent, with red indicating full completion [23].

Frade et al. [33] have proposed a new technique, Genetic Terrain Programming based on the evolutionary design with GP allow game designers to evolve terrains according to their aesthetic feelings and desired features. The developed application produces Terrains Programs that will always generate different terrains, but consistently with the same features (e. g. valleys, lakes). The terrain is represented as a heightmap. Each GP individual is a tree composed of mathematical functions and heightmaps as terminals. Some terminals depend upon a Random Ephemeral Constant (REC) to define some characteristics, such as inclinations of planes and sizes of the geometric figures terminals. All these terminals depend upon a random number generator, which means that consecutive calls of one TP will always generate different terrains. The population size of evolution and when to stop evolution can be artificially controlled. The initial population is randomly generated, and the designer can perform mutation and crossover operations. The individual leaving completely depends on the designer. The generator can eventually generate aesthetically attractive terrain and terrain with specific features.

Reference [34] has proposed a method of building generation using architectural profiles based on answer set programming. The architectural profile is a kind of semantic specification, which restricts a building solver in a declarative way. The method extends the framework of general building solvers. The general building solver considers tiles, adjacent conditions between building tiles, and validity constraints. The method proposed by the author extends the framework of the general building solver: the products of the general building solver are taken as stage products, and these products are defined as shapes, which can be further combined into complex buildings. Compared with the general building solver, the method in this paper pays more attention to the placement of shapes in the input space. The architectural profiles consist of five parts. (1) Tiles: Each tile has architectural meaning, such as walls, windows and doors. (2) Adjacency Conditions: Semantics is also defined on the adjacencies between tiles, expressing the meaning of how tiles may relate to each other. (3) Shapes: The central concept of an architectural profile is a shape, defined as a connected cluster of tiles. (4) Shape

adjacency conditions: If two tiles in two shapes can be connected, then the two shapes can also be connected. (5) Architectural validity constraints: it can help steer the generation process towards the domain of plausible and feasible architectural models through traversability, gravity, and density constraints. Finally, the architectural profiles are transformed into logical constraints, and the model is generated by the solver.

Freiknecht et al. [35] have presented a program generation method for multistory buildings that contain stairwells and can be traversed. Each building contains a set of rooms connected by doors or stairs and is equipped with windows. The building generation process is divided into object model construction and 3D model generation. Object model building includes the following steps. (1) Design the building shape. (2) Choose the stairwell type. (3) Place the stairwell: Determine the location of stairwells by the optimal fitting algorithm in the vertical shared area of all floors. (4) Place rooms: The algorithm automatically places enough initial rooms with a reasonable layout at each floor shape vertex or subdivision vertex. (5) Expand rooms: Loop to expand each initial room to determine the final size of each room. (6) Find the longest corridor: Find the longest path connecting with the stairwell and connecting all rooms. (7) Then extend the path to a polygon. If the corridor does not need to connect the exterior wall, delete the last edge of the polygon in turn until the deletion will reduce the number of connected rooms and reach the final state. (8) Merge corridors and stairwells. (9) Add Windows and doors: According to the topology of the building, the location, size and adjacency of the room, the building adds the outer door, the inner door, the outer window and the inner window in turn. (10) Calculate the roof area: Calculate the roof area for each floor of the building and specify the type of roof for each roof. The 3D model generation process converts the room information, stair information and roof information contained in the object model into a building model with specified quality. In this process, the author modifies the index group of common grid structures to promote the texture of the grid.

Mark et al. [36] have proposed a modular pipeline to generate underground caves, which includes structural components, cave generator and renderer. Structural components use the L-System to generate a set of overall structural points of the cave. In this process, the authors reduce the self-similarity of the generation structure by longer production rules and fewer rewritings, and introduce the ability of randomly generating production rules to improve the expression ability of the generator. At the same time, the random process is constrained to retain the reliability of the generation process. Then, a certain proportion of dead alleys are linked by curves, and the tree structure is changed into the cave structure. The cave generator uses a twisted metaball to move at the structural point of the cave to build a real cave tunnel. Then, based on the noise value computed for each voxel, placement points are found for stalactites and stalagmites, and objects are grown at these locations by cellular automata. The renderer converts the volume of the cave voxels into 3D geometric data by the Marching Cube algorithm, then calculates the grid normal by sampling the density value of the adjacent voxels, and smoothes the surface, and then applies the texture to the UV-free program grid by using three-sided projection to produce results comparable to those in the real world. At the same time, it can adjust the parameters of the shader and modify the style of the 3D cave greatly.

### III. CONCLUSION

This paper summarizes sixteen methods of generating levels and maps for racing games, platform games and open world games, and describes the ideas and central processes of these methods. Here is a summary of the main content.

#### A. Racing Game

In this paper, the method of generating racing game levels is mainly for the track generation of car racing games. At present, there is not much research in this area. The generation method is mainly based on search, and the core is the evolutionary algorithm. Personalized tracks are generated by combining direct player modeling or indirect player modeling, and diversified tracks are generated by combining information entropy. This paper thinks that more research should be devoted to the generation of the track because car racing games can directly map to real activities. Optimizing players' behavior in the game through a personalized track can also promote real driving behavior. In terms of track generation technology, we think it can be developed from three aspects. First, the track height variable should be considered in the generation process because the track height gap is an important factor for a good experience. Second, there are a lot of tracks in the real world, and we can develop a tool to restore the track according to the track image. Thirdly, more accurate track quality evaluation methods should be found from different perspectives.

#### B. Platform Game

The methods for the generation of platform game checkpoints are aimed at the generation of Super Mario Bros. levels in this paper. The seven methods are based on search, graph grammar, and machine learning methods to generate playable, like the input levels or generate diverse levels. Most of these methods have a certain degree of versatility in the generation of platform game levels. From the trend point of view, the generation method gradually moves closer to deep learning. But deep learning doesn't seem to capture the functional needs and aesthetic attributes of a level very well. Although some studies have considered the constraints of playability and aesthetic attributes, the generated levels still cannot meet the commercial standards, and there are still unplayable parts or wrong tiles in the levels. Therefore, this paper argues that after the generation of levels, it should also go through a repair link to repair the functional and aesthetic attributes of customs and promote its commercialization process. However, there are few studies on this point and need to be studied.

#### C. Open World Game

About the method of open world game maps generation, this paper introduces the terrain, architecture and cave generation to supplement the integrity of the article. The generation method is mainly based on search, fractal, solver and cellular automata. Although it is not detailed in the description of this paper, the generation of open world game maps is also a field with great potential.

Finally, the following is the understanding of the PCG process of game levels and maps. (1) First, we must determine the basic objectives of the content we want to generate, and this paper argues that includes three aspects. The first is the pursuit of high-quality generated content. In this process, the versatility of the method is not considered, and the game

content is devoted to generating extreme aesthetics and meeting functional requirements. The second is to pursue the general method of generating the content, including two aspects. To introduce the method ideas of other fields into PCG, or to propose new methods to generate the same type of game content from the game content, such as generating the Mario levels according to the existing Mario levels. Third, look for relationships between different types of game content and generate different types of game content from the existing game content, such as generating game scenarios based on music. (2) According to our basic goal, we determine the representation method of game levels and maps. In this process, if we refer to the construction method of the mapping object of the game content in the real world, it may benefit to construct a more appropriate representation method. (3) Based on the first two stages, it can be combined with other fields such as player modeling to achieve a fully autonomous or mixed-dominant game content generator for a diversified or personalized generation. Among them, the diversification goals include generating controllable attribute levels, similar levels to input ones, and the interpolation expansion of discretely distributed levels to make them relatively continuous levels space. There are two kinds of personalized performance. One is real-time adjustment according to the player's performance in the game round. The other is to provide users with appropriate levels and maps outside the round based on diversity generation. In the future, we believe that PCG can be widely used in game development in industry, and industry and academia will work together to promote the development of PCG, and create high-quality games in which all aspects of game content are interrelated procedurally. Moreover, PCG can be combined with the Metaverse to shine.

#### ACKNOWLEDGMENT

This paper is supported by the Fundamental Research Funds for the Central Universities under Grant Number: N2017003.

#### REFERENCES

- [1] N. Shaker, J. Togelius, and M. J. Nelson, *Procedural content generation in games*. Cham, Switzerland: Springer International Publishing, 2016.
- [2] A. Liapis, G. N. Yannakakis, and J. Togelius, "Computational game creativity," in *ICCC.*, 2014.
- [3] G. N. Yannakakis and J. Togelius, *Artificial intelligence and games*, vol. 3. New York, NY, USA: Springer, 2018, pp. 184-193.
- [4] G. N. Yannakakis, "Game AI revisited," in *Proc. 9th Conf. Comput. Frontiers.*, 2012, pp. 285-292.
- [5] G. N. Yannakakis and J. Togelius, "Experience-driven procedural content generation," in *ACII.*, 2015, pp. 519-525.
- [6] J. Togelius, R. De Nardi and S. M. Lucas, "Making racing fun through player modeling and track evolution," 2006.
- [7] J. Togelius, R. De Nardi and S. M. Lucas, "Towards automatic personalised content creation for racing games," in *IEEE Symp. Comput. Intell. Games.*, 2007, pp. 252-259.
- [8] L. Cardamone, D. Loiacono and P. L. Lanzi, "Interactive evolution for the procedural generation of tracks in a high-end racing game," in *Proc. 13th Annu. Conf. Genetic. Evol. Comput.*, 2011, pp. 395-402.
- [9] D. Loiacono, L. Cardamone and P. L. Lanzi, "Automatic track generation for high-end racing games using evolutionary computation," *IEEE Trans. Comput. Intell. AI. Games.*, vol. 11, no. 3, pp. 245-259. 2011.
- [10] T. Georgiou and Y. Demiris, "Personalised track design in car racing games," in *IEEE CIG.*, 2016, pp. 1-8.
- [11] S. Snodgrass and S. Ontañón, "Controllable Procedural Content Generation via Constrained Multi-Dimensional Markov Chain Sampling," in *IJCAI.*, 2016, pp. 780-786.
- [12] S. Snodgrass and S. Ontañón, "Experiments in map generation using Markov chains," in *FDG.*, 2014.
- [13] E. Hauck and C. Aranha, "Automatic Generation of Super Mario Levels via Graph Grammars," in *IEEE CoG.*, 2020, pp. 297-304.
- [14] S. Londoño and O. Missura, "Graph Grammars for Super Mario Bros Levels," in *FDG.*, 2015.
- [15] M. C. Green, L. Mugrai, A. Khalifa and J. Togelius, "Mario level generation from mechanics using scene stitching," in *IEEE CoG.*, 2020, pp. 49-56.
- [16] S. O. Kimbrough, G. J. Koehler, M. Lu and D. H. Wood, "On a feasible-infeasible two-population (fi-2pop) genetic algorithm for constrained optimization: Distance tracing and no free lunch," *Eur. J. Oper. Res.*, vol. 190, no. 2, pp. 310-327. 2008.
- [17] A. Summerville and M. Mateas, "Super mario as a string: Platformer level generation via lstms," unpublished.
- [18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.* vol. 9, no. 8, pp. 1735-1780. 1997.
- [19] V. Volz, J. Schrum, J. Liu, S. M. Lucas, A. Smith and S. Risi, "Evolving mario levels in the latent space of a deep convolutional generative adversarial network," in *Proc. Genetic. Evol. Comput. Conf.*, 2018, pp. 221-228.
- [20] P. Bontrager, J. Togelius and N. Memon, "Deepmasterprint: Generating fingerprints for presentation attacks," unpublished.
- [21] N. Hansen, S. D. Müller and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)," *Evol. Comput.*, vol. 11, no. 1, pp.1-18. 2003.
- [22] I. Goodfellow, J. Pouget-Abadie, M. Mirza and others, "Generative Adversarial Nets," in *Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672-2680.
- [23] M. C. Fontaine, R. Liu, J. Togelius and A. K. Hoover and S. Nikolaidis, "Illuminating mario scenes in the latent space of a generative adversarial network," unpublished.
- [24] J. K. Pugh, L. B. Soros and K. O. Stanley, "Quality diversity: A new frontier for evolutionary computation," *Frontiers Robot. AI.*, vol. 3, pp: 40. 2016.
- [25] J. B. Mouret and J. Clune, "Illuminating search spaces by mapping elites," unpublished.
- [26] V. Vassiliades and J. B. Mouret, "Discovering the Elite Hypervolume by Leveraging Interspecies Correlation," in *Proc. Genetic. Evol. Comput. Conf.*, 2018, pp:149-156.
- [27] M. C. Fontaine, J. Togelius, S. Nikolaidis and A. K. Hoover, "Covariance matrix adaptation for the rapid illumination of behavior space," in *Proc. Genet. Evol. Comput. Conf.*, 2020, pp. 94-102.
- [28] N. Hansen, "The CMA evolution strategy: A tutorial," unpublished.
- [29] S. M. Lucas and V. Volz, "Tile pattern kl-divergence for analysing and evolving game levels," in *Proc. Genetic. Evol. Comput. Conf.*, 2019, pp. 170-178.
- [30] J. K. Pugh, L. B. Soros, P. A. Szerlip and K. O. Stanley, "Confronting the challenge of quality diversity," in *Proc. Annu. Conf. Genetic. Evol. Comput.*, 2015, pp. 967-974.
- [31] A. Sarkar, Z. Yang and S. Cooper, "Conditional Level Generation and Game Blending," unpublished.
- [32] K. Sohn, H. Lee and X. Yan, "Learning structured output representation using deep conditional generative models," *Adv. Neural Inf. Proc. Syst.*, vol. 28, pp: 3483-3491. 2015.
- [33] M. Frade, F. F. De Vega and C. Cotta, "Modelling video games' landscapes by means of genetic terrain programming-a new approach for improving users' experience," in *Workshops Appl. Evol. Comput.*, 2008, pp. 485-490.
- [34] L. van Aanholt and R. Bidarra, "Declarative procedural generation of architecture with semantic architectural profiles," in *IEEE CoG.*, 2020, pp. 351-358. IEEE.
- [35] J. Freiknecht and W. Effelsberg, "Procedural Generation of Multistory Buildings with Interior," *IEEE Trans. Games.*, vol. 12, no. 3, pp: 323-336. 2019.
- [36] B. Mark, T. Berechet, T. Mahlmann and J. Togelius, "Procedural Generation of 3D Caves for Games on the GPU," in *FDG.*, 2015.



# Similarity-based Local Feature Extraction for Wafer Bin Map Pattern Recognition

Jieun Kim

Department of Industrial and Management Engineering  
Korea University  
Seoul, South Korea  
techzt@korea.ac.kr

Jun-Geol Baek\*

Department of Industrial and Management Engineering  
Korea University  
Seoul, South Korea  
jungeol@korea.ac.kr

**Abstract**— A wafer bin map consists of a local chip containing key information and a global chip present in all patterns. The defect pattern shows a specific pattern shape on the wafer bin map and is defined based on the existing area information. Global information is not differentiated from local information in classification problems and is recognized as a major characteristic, so it affects the identification of the characteristics of defective patterns. In preparation for this, a method of extracting key local information has been proposed. In this paper, we propose a Skip Connections Denoising Autoencoder-based methodology to extract regional information of defect patterns. Randomly distributed chips are recognized as noise by defining anomaly scores based on the probability of each chip appearing in the wafer bin map. We propose a data transformation and reconstruction methodology for extracting local information based on the anomaly score, which is an uncertainty score index. Through the proposed methodology, it was confirmed that the main information that could not be extracted from the convolutional neural network (CNN) was extracted, and it was confirmed that the method proposed in this paper for WM-811K data is superior to the existing method.

**Keywords**— Semiconductor manufacturing process, Defect pattern recognition, Data augmentation, Anomaly localization, Skip connections denoising autoencoder

## I. INTRODUCTION

A wafer produced through a thin substrate for making an integrated circuit. Thousands of integrated circuit (IC) chips are obtained from wafer, and actual wafers are produced through several steps such as etching and surface polishing [1].

The processed wafer goes through the Electrical Die Sorting (EDS) process, which verifies that the chip reaches desired quality level through electrical property inspection. In semiconductor manufacturing process, chips consist of binary value that indicate each die are classified as defective or normal. When defect dies are concentrated in a specific area and show a certain pattern occur, the label of the defect pattern is defined [2].

Fig. 1 shows eight WBM defect patterns and one normal pattern defined in the last stage of the manufacturing process. Among the defective patterns, the center is caused by surface polishing, and the edge-loc is caused by misalignment between

layers [3]. If the defect pattern is accurately classified, the frequency of occurrence of defects can be reduced by identifying the causal factors that contributed to the occurrence of defects during the process.

To identify the cause of WBM patterns, deep learning based method for extracting features are being attempted. It learns common information and classifies labels based on learned image characteristics. Since the existing deep learning-based defect pattern classification method uses all chip information of WBM, it has a limitation in learning common chip information appearing in multiple defect patterns as main information of each defect patterns.

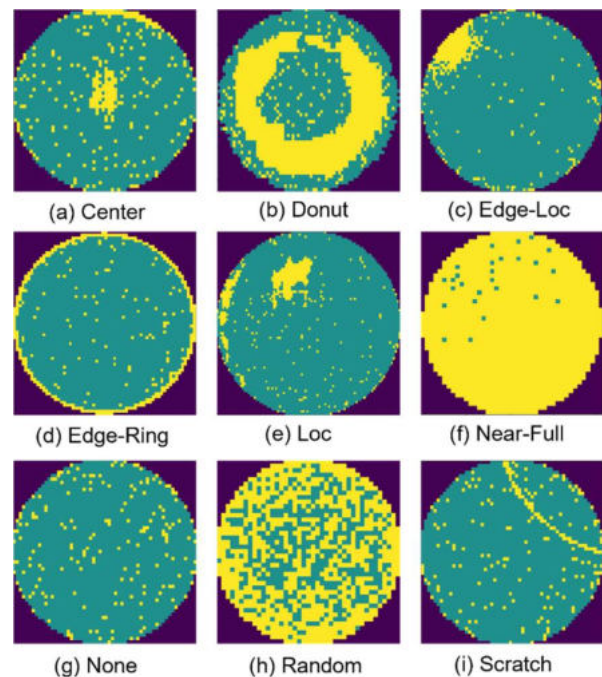


Fig. 1. Wafer bin maps by defect types

In this paper, we propose a regional information extraction methodology for each label of the defect pattern of WBM. As

\* Corresponding author-Tel: +82-2-3290-3396; Fax: +82-2-3290-4550

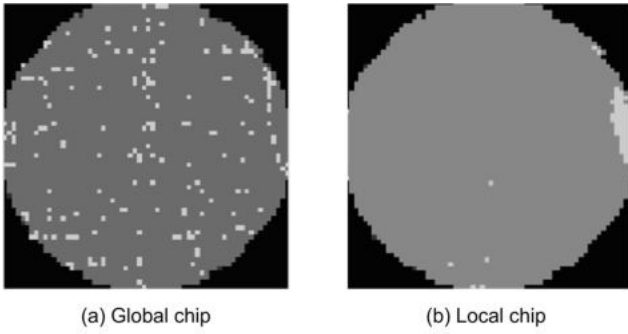


Fig. 2. Significant chip information in Edge-Loc pattern

shown in Fig. 2(a), common global information is removed regardless of the characteristics of the defect pattern.

Our paper is organized as follows. Section II review related studies, and in Section III, describe the structure of method for classifying defect patterns based on local information extraction. In Section IV, the performance of the proposed method is verified by conducting an experiment and describes the conclusion in Section V.

## II. RELATED WORK

The purpose is to classify defect patterns by extracting only regional information representing defective patterns from the WBM. To learn the important characteristics of each pattern, consider main information without high uncertainty information.

This paper assumes that global information is an anomaly element. We introduce existing research on image anomaly detection method, feature extraction-based method and convolutional neural networks (CNN).

### A. Image Anomaly Detection

In order to detect anomaly information in the image, only normal images are trained. Autoencoder (AE) learns by reducing the dimension of the input image through a bottleneck structure and then restoring it. An abnormal image entered for test, restored image only reflect the normal image feature. The reconstructed image is different from the abnormal image used as an input value

Problem of classifying images containing anomalous information, it is determined whether the image is anomaly as shown in Fig. 3 based on the difference from the generated image [4].

Fig. 3 show the framework for anomaly detection, finding difference between the input value  $x$  and the output value  $x'$ , threshold classify normal and abnormal. When the AE is trained using only normal data, a high error value is emitted for abnormal. Since the error value is larger than the set threshold, it can be classified as abnormal.

### B. Feature Extraction-based Method

Denosing Autoencoder (DAE) learns noisy image by removing noise and extracting more features. Recently, a study

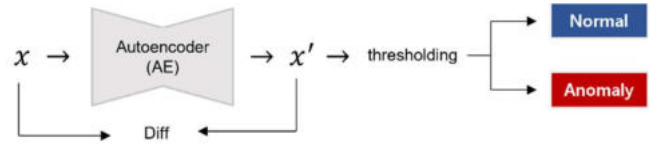


Fig. 3. Image anomaly detection using autoencoder

of classification by applying DAE method to extract features of wafers has been conducted [5]. This paper conducts Stacked Convolutional Sparse Denosing Autoencoder (SCSDAE) to filter noise on the wafer surface. However, the method compresses and classifies the features of the training data through a layered structure, an overfitting problem occurs in the training data. This has a limitation in misclassifying data with noise or deformed data.

To solve this limitation, using skip connection technique was proposed. It does not learn details such as global information and noise in image. It learns without a bottleneck structure that stores dimensionally reduced features and delivers uncompressed information to decoder [6]. AE framework with skip connection structure shows good performance in removing noise from the image and obtaining key information.

## III. METHOD

In this paper, we propose multi step learning model for regional information extraction of WBM. Step 1 is the process of extracting main features from images including noise and global information using Skip Connections Denosing Autoencoder (SCDAE). In step 2, anomaly score is calculated using the distance similarity between pixels of the existing WBM. In the last step, an image is generated based on anomaly score and CNN is trained to classify labels.

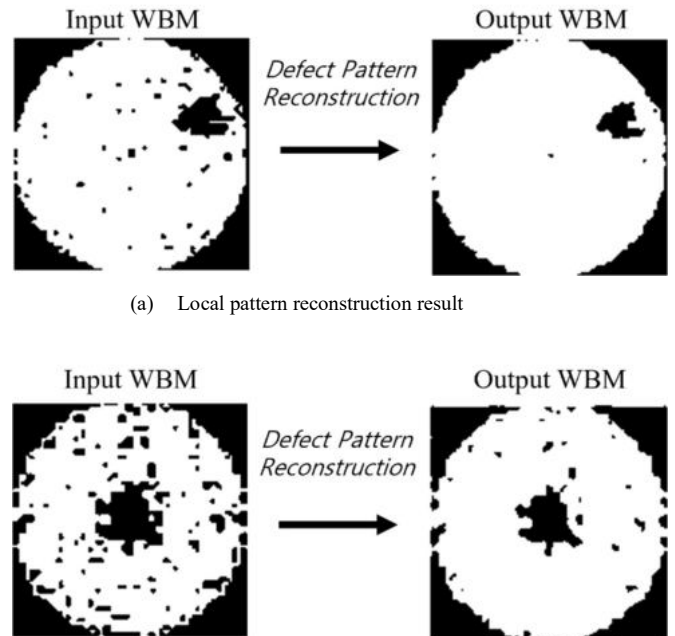


Fig. 4. Reconstruction result using SCDAE

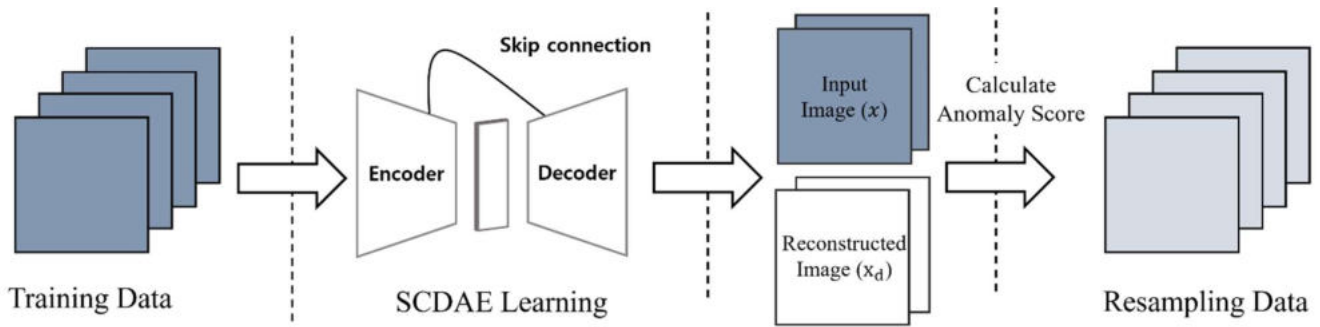


Fig. 5. Pipeline of proposed resampling method using SCDAE

### A. SCDAE based Feature Extraction

WBM include regional and global chip information. Each chip has a binary value by setting a threshold value based on the pixel value of WBM that has completed the EDS process

The existing WBM pattern classification model was trained by reflecting all chip information in the data. This means that global information is also judged as important information and learned in the learning process.

Proposed method extracts an image  $\tilde{x}$  is created by adding Gaussian noise to the input data.  $\tilde{x}$  provides additional random information to remove global information. After that, it learns by comparing  $\tilde{x}$  with the input  $x$  through the SCDAE model.

To extract local feature, add random noise to all data. Each pixel value with consecutive values in the range  $[0, 1]$ . Information from the front part of the encoder is transmitted to the decoder through the skip connection, and only major regional information that can be intuitively identified is learned. Through the learning process, image  $x_d$  including main information can be obtained as shown in Fig. 4. Fig. 4 shows the results of extracting the features of Local and Center patterns by applying SCDAE. Through this, it can be confirmed that the global chip information to the two defective patterns is removed, and local region information that can define the pattern is extracted.

### B. Similarity based Image Generation

The generated image  $x_d$  is the result value of  $p(x|\tilde{x})$  that follows a stochastic distribution for the image  $x$  to which noise is added during the learning process of SCDAE. In the learning process, values other than local information are removed to have a small value. Therefore, an anomaly score for the importance of each chip can be defined as in Equation 1.

$$\text{Anomaly score} = 1 - |x - x_d| \quad (1)$$

Based on the anomaly score defined in Equation 1, it is possible to determine the critical information of each chip in the WBM. Since the anomaly score is based on the difference between  $x$  and  $x_d$ , the chip information removed that has a high

value. This mean that it is insignificant information of the defect pattern.

### A. Stepwise Method for Classification

The proposed method is a stepwise process as shown in Fig. 3.  $x_r$  generated based on the threshold value includes area information of the WBM defect pattern.  $x_r$  not include information on chips with low importance.

The structure for each stage learned from data containing only local information contains only the main information of each defect pattern. Therefore, when new data is entered, the chip corresponding to global information is judged to be insignificant information and has a low probability value.

Last stage CNN consider WBM local location information together. Since the generated image reflects only local information, it learns location information and features for each bad pattern through CNN.

## IV. EXPERIMENTS

### A. Data Description

WM-811K used to classify defect patterns in WBM [7]. WM-811K consists of Center (2.5%), donut (0.3%), Edge-Loc (3.0%), Edge-Ring (5.6%), Loc (2.1%), Random (0.5%), Scratch (0.7%), Near-Full (0.1%) defect pattern.

8 types of defect pattern are used except the None pattern in Fig. 1 to perform verification to extract the main features of the defect pattern. Total of 17,625 data is used, 13,128 are used as training and 4,407 are used as test.

### B. Wafer Map Classification Accuracy

For the performance evaluation of a model with image reconstructed by SCDAE as an input, accuracy is used for image classification model performance evaluation. Each index of the evaluation matrix indicates the following. TP (True Positive) and TN (True Negative) in Equation 2 are cases in which labels are correctly classified. FP (False Positive) and FN (False Negative) are indicators indicating the result of classification differently from the actual label. The combined accuracy represents the percentage of correctly classified labels for the entire data.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (2)$$

Among the 8 types of defective patterns presented in Fig. 1, 8 types of defective patterns excluding normal patterns are classified by applying the proposed model.

TABLE I. CLASSIFICATION ACCURACY

Model	CNN	AE CNN	DAE CNN	Proposed Method
Center	99	100	99	100
Donut	93	77	92	<b>94</b>
Edge-Loc	85	91	88	<b>95</b>
Edge-Ring	97	94	98	<b>98</b>
Loc	83	90	83	83
Near-Full	92	85	98	<b>100</b>
Random	87	4	93	93
Scratch	63	63	73	<b>73</b>

Table I shows the results of classifying 8 types of defect patterns by applying 4 types of method including the proposed model. All CNNs included in the four methodologies have the same structure.

Based on the accuracy of the proposed method, Donut, Edge-Loc, Edge-Ring, Near-Full, and Scratch patterns showed higher performance than the existing methodologies and showed similar performance for three defective patterns. This is a result showing that the proposed method extracts the local information of the wafer bin map defect pattern well and the model shows robust characteristics against new data.

## V. CONCLUSION

The method proposed in this paper is to classify defect patterns by extracting regional information from the WBM. Related research has limitations in applying the deep learning method that reflects all chip information in the WBM. Inconsequential information for each defect pattern was also considered, resulting in confusion among some patterns.

SCDAE is an object localization methodology that used to extract local information of WBM defect patterns. Through this, an anomaly score was defined to generate an image including major features for each defect pattern and to define global information that could be generated from a new input value. Based on the uncertainty, global information was removed and data reflecting only the main information of the defect pattern was generated. Afterwards, it was confirmed that unique regional information for each defect pattern was extracted for new data through the learning process. Through this, it was

found through performance that it is possible to robustly classify the deformed data or the data with added noise.

The main information extraction-based learning method proposed in this study confirms the main characteristics of each defect pattern and contributes to identifying the cause of the defect. Through this, it is possible to define the main characteristics of each defect pattern, and it is possible to easily classify the defect patterns collected in the future process.

## ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (NRF-2019R1A2C2005949). Also, this work was supported by Brain Korea 21 FOUR and Samsung Electronics Co., Ltd.(IO201210- 07929-01).

## REFERENCES

- [1] Uzsoy, R., Lee, C. Y., Martin-Vega, L. A. "A review of production planning and scheduling models in the semiconductor industry part I: system characteristics, performance evaluation and production planning." IIE transactions, vol. 24, no. 4, pp. 47-60, 1992.
- [2] Do, H., Lee, C., and Kim, S. B. "A Hierarchical Spatial-Test Attention Network for Explainable Multiple Wafer Bin Maps Classification." IEEE Trans. Semicond. Manuf., 2021.
- [3] Saqlain, M., Jargalsaikhan, B., and Lee, J. Y. "A voting ensemble classifier for wafer map defect patterns identification in semiconductor manufacturing." IEEE Trans. Semicond. Manuf., vol. 32, no. 2, pp 171-182, 2019.
- [4] Bergmann, P., Löwe, S., Fauser, M., Sattlegger, D., and Steger, C. "Improving unsupervised defect segmentation by applying structural similarity to autoencoders." *arXiv*, 2018.
- [5] Yu, J., Zheng, X., and Liu, J. "Stacked convolutional sparse denoising auto-encoder for identification of defect patterns in semiconductor wafer map." Computers in Industry 109, pp. 121-133, 2019.
- [6] Mao, X., Shen, C., and Yang, Y. B. "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections." Advances in neural information processing systems 29, pp. 2802-2810, 2016.
- [7] Wu, Ming-Ju, Jyh-Shing R. Jang, and Jui-Long Chen. "Wafer map failure pattern recognition and similarity ranking for large-scale data sets." IEEE Trans. Semicond. Manuf., vol. 28, no.1. pp. 1-12, 2014.

# Body Segmentation Using Multi-task Learning

Julijan Jug<sup>1,†</sup>, Ajda Lampe<sup>1,2,†</sup>, Vitomir Štruc<sup>2</sup>, Peter Peer<sup>1</sup>

<sup>1</sup>Faculty of Computer and Information Science, <sup>2</sup>Faculty of Electrical Engineering  
University of Ljubljana, SI-1000 Ljubljana, Slovenia

julijan.jug@gmail.com, {ajda.lampe, vitomir.struc}@fe.uni-lj.si, peter.peer@fri.uni-lj.si

**Abstract**—Body segmentation is an important step in many computer vision problems involving human images and one of the key components that affects the performance of all downstream tasks. Several prior works have approached this problem using a multi-task model that exploits correlations between different tasks to improve segmentation performance. Based on the success of such solutions, we present in this paper a novel multi-task model for human segmentation/parsing that involves three tasks, i.e., (i) keypoint-based skeleton estimation, (ii) dense pose prediction, and (iii) human-body segmentation. The main idea behind the proposed Segmentation–Pose–DensePose model (or SPD for short) is to learn a better segmentation model by sharing knowledge across different, yet related tasks. SPD is based on a shared deep neural network backbone that branches off into three task-specific model heads and is learned using a multi-task optimization objective. The performance of the model is analysed through rigorous experiments on the LIP and ATR datasets and in comparison to a recent (state-of-the-art) multi-task body-segmentation model. Comprehensive ablation studies are also presented. Our experimental results show that the proposed multi-task (segmentation) model is highly competitive and that the introduction of additional tasks contributes towards a higher overall segmentation performance.

**Index Terms**—computer vision, segmentation, human body parsing, multi-task learning

## I. INTRODUCTION

In recent years, great progress has been made in the field of computer vision. Modern generative models, such as GANs, have made it possible to generate photorealistic images with convincing visual quality. Much research is also focused on the application of such models. One such challenge is the generation of photorealistic images of people in desired clothing or the problem of virtual try-on [1], [2]. Such applications have great potential for use in online clothing stores and enhance the user experience of online shopping. With the development of deep neural networks, there has also been a great leap in the field of semantic segmentation [3], [4]. However, there is still much room for improvement in certain areas, such as human body segmentation. Currently, the best segmentation models still do not perform as well as they should for applications such as virtual clothing try-on. Most of the problems with current models are caused by images taken under less than ideal conditions and partially obscured views of the subject.

A significant amount of research has been conducted recently to improve such models by using additional information

<sup>†</sup> First authors with equal contributions.



Fig. 1. This example shows that the pose and dense pose subtasks provide helpful contextual and structural information about the human body. The second image shows the segmentation mask produced by our multi-task model containing segmentation and pose estimation tasks. The third image shows the segmentation mask created by our multi-task model with the additional task of dense pose estimation. We can see that the additional task of dense pose estimation significantly improves the segmentation performance.

to drive and support segmentation models. By providing additional contextual information, it is assumed that the model may obtain a better understanding of the image content and human anatomy. Existing work is, therefore, looking at combining segmentation models with other related tasks in so-called multi-task architectures. Most commonly existing models include pose estimation as a supporting task, e.g., [5]. Previous research has also shown that using a multi-task learning contributes to the quality of human segmentation. Based on this insight, we explore in this paper additional possibilities for extending this type of models with additional tasks that could further aid the segmentation process.

While most existing work includes pose estimation as a supporting task, our work focuses on improving the quality of segmentation results by utilizing an additional task. To this end, we propose a new architecture of a multi-task model that includes the task of inference of skeletal position or posture and dense pose in addition to the task of body segmentation. To this end, we propose a novel multi-task segmentation model called SPD, which considers all three tasks. The letters in the name represent each task: S – segmentation, P – pose, and D – dense pose. We propose a multi-task architecture based on a shared backbone neural network using three specialized branches on top, one for each of the selected tasks. The purpose of such an approach is to improve the segmentation task. We evaluate the proposed model on the LIP and ATR dataset and report highly encouraging results. We also perform extensive ablation studies to support our hypothesis that adding tasks improves the overall performance of the model.

The main contributions of this paper are:

- We present SPD, a novel multi-task model for human body segmentation that includes pose estimation and

dense pose prediction tasks.

- We show that adding additional tasks improves performance for the primary task.

## II. RELATED WORK

One of the more specialized application domains of semantic segmentation is the segmentation of the human body and clothing. The need for such segmentation algorithms arises from the requirements of various vision systems related to human image analysis, such as virtual clothing try-on [1], [2] or re-identification [6]. Recently, much research has been done on human segmentation [7]–[9] using deep convolutional neural networks. The disadvantage of these models is that they do not take into account the structure or anatomy of the human body, so the segmentations often contain errors that are unreasonable from a human perspective. A considerable amount of research has, therefore, focused on solving this problem by incorporating additional information into the segmentation procedure related to body posture and anatomy.

One way to introduce supporting information to the model is the multi-task learning approach, where the model is simultaneously trained to solve multiple tasks. Due to the good results in recent years, multi-purpose learning has been widely used in various natural language processing and computer vision applications [10]–[12]. Gong *et al.* [5], for example, proposed a model that predicts semantic segmentation masks and estimates body joint positions based on the generated segmentation map. The model is optimized based on the quality of both the segmentation map and the joint locations to ensure it learns a semantically consistent representation of the human body. Liang *et al.* [13] build on this approach by using a common base network, followed by two smaller modules, specialized for joints estimation and semantic segmentation. The modules are built in a two-stage coarse-to-fine manner and share the intermediate coarse results. The proposed model, called JPPNet, achieves impressive results and outperforms previous work in a convincing manner. However, there is still room for improvement in the model’s body representation, as some structural flaws persist. Given the promising results, we explore the potential of introducing an additional task in improving the final semantic segmentation result.

## III. METHODOLOGY

We propose a multi-task model, called SPD, for human body parsing that is learned based on three distinct tasks: segmentation mask generation, keypoint-based pose estimation, and dense pose prediction [14]. The model is inspired by the success of existing multi-task models, such as JPPNet [13], that have been shown to ensure competitive performance, while also exhibiting desirable architectural features.

### A. Model Overview

Fig. 2 shows the basic architecture of our model, which consists of a backbone feature extractor and three distinct branches: (i) one for human body segmentation, (ii) one for key-point based pose estimation, and (iii) one dense pose

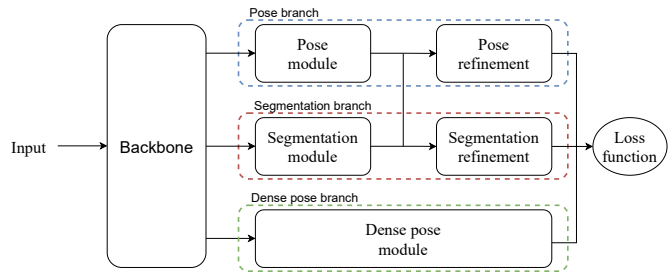


Fig. 2. High-level architectural diagram of the proposed SPD model. The common ResNet backbone of the SPD model is shared between three specialized model branches designed specifically for human body segmentation, skeleton/pose prediction, and dense pose estimation.

prediction. The main goal of the model is to ensure efficient body part segmentation, so the segmentation branch is treated as the main component of the model, whereas the remaining two branches perform auxiliary tasks. The main backbone model common to all tasks is a ResNet-101 [15] deep residual network, which consists of 101 convolutional layers arranged across 5 residual blocks. In the SPD model, part of this backbone is shared between the three branches, which acts as a link between the three considered tasks.

The three branches allow for the definition of three separate learning objectives, i.e., one per tasks, that are then used jointly to learn the model. Specifically, the overall loss function used with SPD is calculated as a weighted sum of the three task-specific losses, i.e.:

$$\mathcal{L} = \lambda_s \mathcal{L}_s + \lambda_p \mathcal{L}_p + \lambda_d \mathcal{L}_d, \quad (1)$$

where  $\lambda_s$ ,  $\lambda_p$  and  $\lambda_d$  are balancing weights corresponding to the segmentation loss  $\mathcal{L}_s$ , the keypoint-based pose loss  $\mathcal{L}_p$ , and the dense-pose loss  $\mathcal{L}_d$ , respectively. Empirically, we chose a higher weight for the segmentation part of the loss function and lower values for the other two tasks to ensure that the segmentation task is given preference in the optimization procedure. We set  $\lambda_s = 1$ ,  $\lambda_p = 0.8$  and  $\lambda_d = 0.6$  based on preliminary experiments to provide a good trade-off between the three tasks. The individual loss functions are presented in the following subsections.

### B. Segmentation Branch

Usually, only information from the ground truth segmentation masks is used to learn the task of segmenting individuals. In our approach, we also incorporate contextual information of the skeleton directly into the segmentation network. Fig. 3 shows a high-level overview of the components in the segmentation branch. As can be seen, the output of the fifth residual block is used as the initial representation for the segmentation branch. To generate the an initial estimate of the segmentation mask, an additional layer of Atrous Spatial Pyramid Pooling (ASPP) is used on top of the extracted ResNet features. ASPP performs multiple convolutions over the input data at different sampling rates and mask sizes, capturing objects and contextual information at different scales. In parallel to the ASPP component, we create what we call *segmentation context*

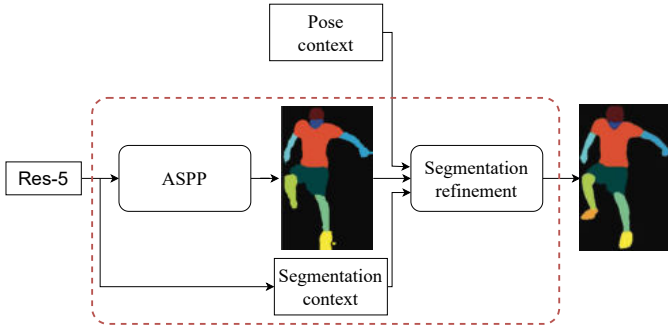


Fig. 3. Overview of the segmentation branch of the SPD model. The branch consists of two parts, where the first generates an initial segmentation result based on features produced by the backbone model, whereas the second refines this initial estimate using different types of input information - also from other branches.

by processing the output of the fifth residual layer through two additional convolutional layers. This context is later used in the second stage of the segmentation branch along with other sources of information to further refine the segmentation results.

The refinement network in the second part of the segmentation branch takes the segmentation context as well as the initial (rough) estimate of the segmentation masks as input and combines these inputs with what we call *pose context* - a representation generated by the pose estimation branch of the model. This is followed by four levels of convolutions with the purpose of capturing the local context and learning the key connections between the pose and segmentation contexts. The result of these convolutional layers is reshaped and passed through another ASPP component. This last ASPP component, thus, generates the final segmentation masks. The segmentation loss defined on top of this branch is expressed in terms of pixel-wise softmax cross entropy, i.e.:

$$\mathcal{L}_s = \frac{1}{M} \sum_{k=1}^K \sum_{m=1}^M y_m^k \times \log(h_\theta(x_m, k)), \quad (2)$$

where  $M$  is the number of samples,  $K$  is the number of segmentation classes,  $y_m^k$  is the target classification for a sample  $m$  and a class  $k$ . The input sample is denoted by  $x$  and the prediction model by  $h$ .

### C. Pose Estimation Branch

Fig. 4 shows a high-level overview of the components involved in creating pose representations, i.e., keypoint of the human skeleton. Unlike in the segmentation branch, the input to the pose branch is the output of the fourth residual block, following the suggestions from [13]. The initial pose module in this branch consists of 8 convolutional layers, the initial six obtain skeleton features, and the two on top produce the first version of the skeleton representation in the form of a tensor with sixteen coordinates of skeleton joints. Similarly as in the segmentation branch, a refinement step is used in the second stage of this branch that take the initial pose predictions, pose context and the segmentation context, produced by the

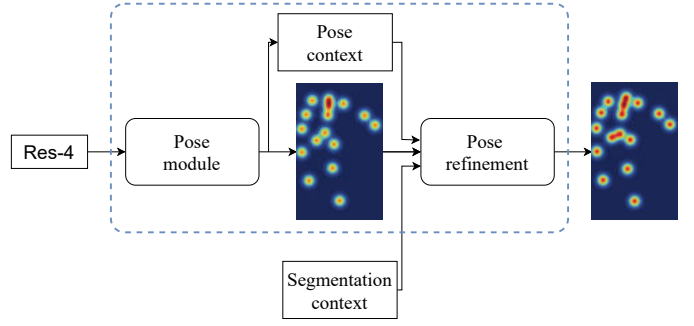


Fig. 4. Overview of the pose estimation branch of the SPD model. The branch consists of two parts, where the first generates an initial keypoint prediction based on features produced by the backbone model, whereas the second refines this initial estimate using different types of input information - also from other branches.

segmentation branch as input, and then applies 4 levels of convolutions over these input to capture the local context at different scales. Finally, two additional convolutional layers are utilized to generate the refined skeleton keypoints. An L2 loss is defined on top of the branch to facilitate training, i.e.:

$$\mathcal{L}_p = \frac{1}{2N} \sum_{i=1}^N \|p_i - p_i'\|^2, \quad (3)$$

where  $N$  represents the number of defined joints in the skeleton,  $p_i'$  the predicted coordinates of the joint, and  $p_i$  the annotated coordinates of the joint.

### D. Dense Pose Branch

Fig. 5 presents the architecture of the dense pose branch. Similarly as in the pose branch, we use the first the output of the fourth residual block of the backbone model for the initial encoding. Following the ResNet network is a module for sampling regions of interest (ROIs), which is used to (cascadely) capture local contexts at various scales. Attached to the ROI pooling module is a *dense pose* head, composed of two dedicated CNN heads, a classification head and a regression head. The first head is used to assign the image elements to the corresponding body segment, i.e., the classification of component  $I$ . The second head determines the position of the image elements within the corresponding segments, i.e., it is used to determine the components  $U$  and  $V$ .

The loss function for the dense pose branch consists of two parts. The first part relates to the component  $I$  and is computed in the same way as in the main segmentation task, i.e., using cross entropy. The second part, which refers to the coordinates  $U$  and  $V$ , is computed using the Huber loss function:

$$\mathcal{L}_d = \sum_{m=1}^M CSE(x_I) \cdot L_1(x_U, x_V), \quad (4)$$

where  $x_I, x_U, x_V$  are the components of the depth representation,  $CSE$  is the transverse entropy function for the segmentation part and  $L_1$  is the Huber loss function for the position part.

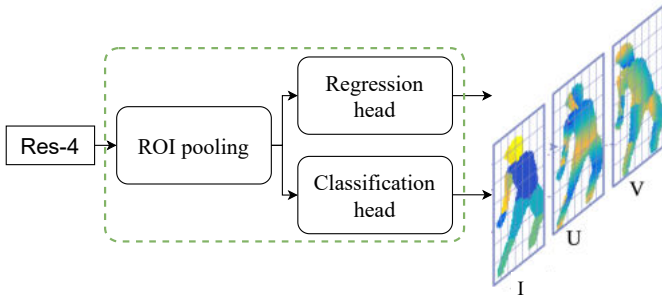


Fig. 5. The figure shows a high-level diagram of the architecture of the DensePose model. Dense pose components visualization is taken from a DensePose article [14].

### E. Training Details

The models are trained on an Nvidia 1080Ti GPU with 11 GB of memory. We empirically determined the weights  $\lambda_s = 1$ ,  $\lambda_p = 0.8$ ,  $\lambda_d = 0.6$  for the objective in (1). The full model was trained for 400.000 iterations.

## IV. EXPERIMENTS AND RESULTS

In this section, we present the datasets selected for the experiments. We describe the protocol used to evaluate the proposed SPD model and discuss the performance measures utilized for performance reporting for all three model tasks. We then comment on and analyze the results of the model. We also perform an ablation analysis to demonstrate the contribution of each task to the final accuracy of the proposed model. Finally, we present examples of the generated segmentation masks and analyze them qualitatively.

### A. Datasets

Dataset selection plays an important part in the training of the proposed SPD model. For our purposes, we used several datasets containing images of people in different clothing, situations, contexts, and body positions. A particular challenge of our multi-task modeling approach is the need for a database that contains several different types of annotations.

For learning a multi-task model that includes the generation of segmentations, body poses, and dense pose representations, we need a dataset that contains all three types of annotations. To this end, we selected the LIP dataset [5] that contains segmentation and skeleton annotations for over 50.000 images. An example image with the segmentation masks and pose annotations from this dataset is presented in Fig. 6. For dense pose annotations, we used the COCO [16] database, which is a superset of LIP. We merged the annotations from both dataset to generate the reference data needed to train the SPD model. In the final setup we have dense-pose annotations for all input images, a 19-class markup for the segmentation task, and a 16-point markup for the skeleton keypoints. To evaluate the performance of all tasks of SPD, we use a hold-out set from LIP, as well as images from the ATR [17] dataset.



Fig. 6. Example image from the LIP dataset and the corresponding segmentation masks and 16-point skeleton markup.

### B. Performance Measures

Following standard evaluation methodology, we use four performance measures to report performance for the segmentation task, i.e., the Jaccard index  $IoU$ , precision, recall, and the  $F1$  score [18], [19]. The first measure is the Jaccard index or the weighted average of the intersection over union. The measure  $IoU$  (intersection over union) is defined as:

$$IoU = \sum_{i=1}^K \frac{S'_i \cap S_i}{S'_i \cup S_i}, \quad (5)$$

where  $S'$  represents the predicted area and  $S$  represents the annotated area of the  $i$ -th instance class and  $K$  is the number of annotated reference classes. The maximum value of  $IoU = 1$  indicates ideal performance. When looking at semantic segmentation as a pixel-level classification problem, precision (6) is defined as the ratio of correctly classified pixels among all pixels classified to a class, whereas recall (7) is the fraction of correctly classified pixels among all pixels belonging to a class, i.e.:

$$Precision = \frac{TP}{TP + FP}, \quad (6)$$

$$Recall = \frac{TP}{TP + FN}, \quad (7)$$

where  $TP$ ,  $FP$ ,  $TN$  and  $FN$  denote true positives, false positives, true negatives and false negatives, respectively. The  $F1$  score is the harmonic mean between precision and recall:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}. \quad (8)$$

For the pose estimation task we report the mean Euclidean distance (mED) between the predicted  $p_i'$  and reference pose key points  $p_i$ . The measure is defined as:

$$mED = \sum_{i=1}^N d_{L_2}(p_i, p_i'), \quad (9)$$

where  $d_{L_2}(\cdot)$  is the Euclidean distance function, and  $N = 16$  is the overall number of annotated key points.

For the dense pose prediction task, we use a measure of geodesic point similarity between the generated and reference



TABLE I

SEGMENTATION AND ABLATION RESULTS ON THE HOLD-OUT SET OF LIP. THE ARROWS INDICATE WHETHER HIGHER OR LOWER SCORES CORRESPOND TO BETTER PERFORMANCE.

Experiment	Model	Performance measures			
		$IoU \uparrow$	$Pr \uparrow$	$Rec \uparrow$	$F1 \uparrow$
Comparison	JPPNet [13]	0.538	0.68	0.66	0.66
	SPD (ours)	<b>0.547</b>	<b>0.76</b>	<b>0.68</b>	<b>0.71</b>
Ablation study	SP	0.535	0.74	0.52	0.63
	SD	0.478	0.67	0.50	0.57
	S	0.483	0.62	0.49	0.54

dense-pose body representations, as defined in [14]. The measure is defined as follows:

$$GPS = \frac{1}{|P|} \sum_{p_i \in P} \exp\left(\frac{-d(\hat{p}_i, p_i)^2}{2k(p_i)^2}\right). \quad (10)$$

In the above definition,  $P$  represents the set of annotated surface points,  $|\cdot|$  is the set cardinality,  $\hat{p}_i$  denotes the  $i$ -th predicted point on the surface, and  $p_i$  the corresponding annotated point on the person’s surface. The function  $d$  represents the geodesic distance between the points and  $k$  the normalization factor specific to each body part. The values of the normalization factors are taken from [14].

### C. Segmentation Results and Ablations

**Comparison with the State-Of-The-Art.** With the proposed SPD model we aim to improve on the results of existing body segmentation models. Specifically, we build on the recent JPPNet approach from [13] and, therefore, include this approach for baseline comparisons in the experiments. Table I shows the results for the segmentation task on the hold-out set of LIP. As can be seen, on the LIP dataset, the SPD model achieves an  $IoU$  result of 0.547, compared to the JPPNet model, which results in a score of 0.538. In terms of the  $F1$  score, the proposed model outperforms JPPNet by approximately 5%. Similar performance improvements are also observed for precision and recall. To further verify the performance of SPD on an independent dataset with characteristics different from the training data, we also evaluate our model on the ATR dataset. The segmentation results in Table II again show that SPD outperforms JPPNet in terms of all reported performance metrics. We attribute the observed performance gains to the interaction of the three different tasks considered during training, which allow our model to better learn how to efficiently parse images of humans and generate reliable segmentation masks across a diverse set of image characteristics.

**Ablation Study.** To demonstrate the importance of all tasks in the multi-task design of SPD, we perform an ablation study, where different tasks are removed from the overall model. Three additional models are implemented and trained for this experiment, i.e.: (i) the SPD model without the dense-pose prediction task (SP hereafter), (ii) the SPD model without the keypoint-based pose prediction task (SD hereafter), and (iii) the SPD model without both pose-related tasks (S hereafter).

TABLE II

SEGMENTATION AND ABLATION RESULTS ON THE ATR DATASET. THE ARROWS INDICATE WHETHER HIGHER OR LOWER SCORES CORRESPOND TO BETTER PERFORMANCE.

Experiment	Model	Performance measures			
		$IoU \uparrow$	$Pr \uparrow$	$Rec \uparrow$	$F1 \uparrow$
Comparison	JPPNet [13]	0,464	0,66	0,67	0,66
	SPD (ours)	<b>0,472</b>	0,67	<b>0,70</b>	<b>0,68</b>
Ablation study	SP	0,423	<b>0,69</b>	0,53	0,60
	SD	0,340	0,59	0,44	0,50
	S	0,291	0,50	0,56	0,52

The results of this experiment are presented in Tables I and II for the LIP and ATR datasets, respectively. It can be seen, that each added task provides the model with new useful information to improve the segmentation results. Removing the dense pose estimation task results in a drop of the segmentation performance across all performance scores. The removal of the keypoint-based pose estimation task has an even bigger adverse effect on performance. If both task are ablated, we observe the most significant performance degradation suggesting that both pose-related tasks provide important information for further improving segmentation results. Interestingly, we see larger performance drops on the ATR dataset than on LIP. This is likely a result of the fact that the model was trained on part of the data in LIP, so auxiliary tasks are more critical when the characteristics of the data change. In the cross-dataset experiment on the ATR dataset, the added information from the dense-pose and keypoint-based pose estimation tasks is needed to produce competitive segmentation performance with SPD.

### D. Results of Auxiliary Tasks

Because SPD is trained in a multi-task manner, it also produces predictions of skeleton/pose keypoints and dense-pose representations of the input images. To better understand the behavior of the model, we report here results for the keypoint prediction and dense-pose estimation tasks on the test part of the LIP dataset.

**Keypoint Prediction.** For the first experiment we evaluate three models, the proposed SPD, the reference JPPNet and SPD model without the dense-pose prediction task, i.e., SP. On the LIP test data, the JPPNet model results in the lowest  $mED$  value of 51.2 pixels, followed by the SPD model with a value of 55.01 pixels. The weakest model in this experiment is the SP model with a  $mED$  value of 56.82 pixels. These results suggest that the addition of the dense-pose estimation task clearly improves performance for the keypoint prediction task. However, the final results are inferior to JPPNet, due to the fact that the segmentation tasks was given higher priority in the balancing of the loss term in Eq. (1).

**Dense-pose Prediction.** The third task performed within the SPD model is the prediction of the dense depth representation of the body. Because JPPNet does not generate dense-pose predictions, we only report results for the complete SPD model and the model without the keypoint-based pose prediction

task, i.e., SD. On the LIP test data, the SPD model achieves a *GPS* score of 48.2% and the SD model with a score of 50.1%. The results show that adding the keypoint-based pose prediction task does not help to improve dense-pose estimation performance. Both models result is very similar *GPS* scores. This observation suggests that even though segmentation can benefit from the additional tasks, the balancing weights used in our optimization objective do not ensure consistent performance improvements across all considered tasks. Nevertheless, if dense-pose prediction is treated as the primary optimization target, improved results can also be expected for this task due to the multi-task learning.

### E. Qualitative analysis

In this section, we present and analyze qualitative results generated by the segmentation branch of the SPD model. Fig. 7 shows a comparison of the segmentation results generated by SPD and JPPNet together with the original input images the ground truth segmentation masks for two selected images from the LIP dataset.



Fig. 7. Comparison of the segmentation results generated by the proposed SPD model and the competing JPPNet on selected images from the LIP dataset. The first row showed the selected input images, the second row shows the ground truth annotations, whereas the third and fourth row show results for JPPNet and SPD, respectively.

The first image shows a tennis player and a person in the background, who is out of focus and partially obscured. We can see that the SPD model is the only one that correctly detected only the player in the foreground. The competing model has problems with the person in the background, as it is very close to the tennis player in the foreground. The difference in segmentation quality is also visible in the definition of the fingers on the right hand, where the SPD model recognized individual fingers much better than the JPPNet modes. The second example image shows a woman partially hidden behind a chair. In this case, the JPPNet model omits the entire leg area, although it is still partially visible behind the chair. Despite the overlap, the SPD model recognizes the position of the leg and marks it correctly. Another unique feature of this image is the classification of the upper part of the garment. The upper part of the woman’s body is annotated as an upper clothing class, JPPNet model falsely classifies it as a coat, while the SPD model correctly classifies the area as an upper clothing class, as a result of the contextual information provided by the other two tasks. In the third image, we see a man surfing on water. In this case, the JPPNet model results in the best segmentation according to the annotations, as it appropriately marks the upper part of the garment and separates that from the pants. Our model classifies the entire area as a one-piece jumpsuit, which is a reasonable classification given the appearance of the image from a human perspective.

### V. CONCLUSION

In this work, we presented a multi-task segmentation model called SPD. In addition to the primary task of body segmentation, the model also includes the task of keypoint-based pose estimation and dense pose prediction. The segmentation part of the model was evaluated on the LIP and ATR datasets, and for both datasets SPD achieved better results than the reference model JPPNet. Furthermore, through rigorous ablation studies it was shown that models that considered a lesser number of tasks resulted in worse performance. In the ablation analysis, we presented the contribution of each of the tasks and found that using the skeleton and depth task together adds more value than using either one of them on its own. To further improve results, we plan to explore additional tasks in the learning procedure that could provide additional cues for the segmentation procedure.

### ACKNOWLEDGEMENTS

This research was supported in parts by the ARRS Project J2-2501 “Deep Generative Models for Beauty and Fashion (DeepBeauty)”, the ARRS Research Programme P2-0250(B) “Metrology and Biometric Systems” and the ARRS Research Programme P2-0214 “Computer Vision”.

### REFERENCES

- [1] X. Han, Z. Wu, Z. Wu, R. Yu, and L. S. Davis, “Viton: An image-based virtual try-on network,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7543–7552.
- [2] B. Fele, A. Lampe, P. Peer, and V. Struc, “C-vton: Context-driven image-based virtual try-on network,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2022.

- [3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," in *Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4. IEEE, Apr. 2018, pp. 834–848. [Online]. Available: <https://doi.org/10.1109/tpami.2017.2699184>
- [4] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao, "Deep high-resolution representation learning for visual recognition," vol. 43, no. 10. IEEE, Oct. 2021, pp. 3349–3364. [Online]. Available: <https://doi.org/10.1109/tpami.2020.2983686>
- [5] K. Gong, X. Liang, D. Zhang, X. Shen, and L. Lin, "Look into person: Self-supervised structure-sensitive learning and a new benchmark for human parsing," in *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jul. 2017, pp. 556–567. [Online]. Available: <https://doi.org/10.1109/cvpr.2017.715>
- [6] R. Zhao, W. Ouyang, and X. Wang, "Unsupervised saliency learning for person re-identification," in *Conference on Computer Vision and Pattern Recognition*. IEEE, Jun. 2013, pp. 500–518. [Online]. Available: <https://doi.org/10.1109/cvpr.2013.460>
- [7] X. Liang, S. Liu, X. Shen, J. Yang, L. Liu, J. Dong, L. Lin, and S. Yan, "Deep human parsing with active template regression," in *Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 12. IEEE, Dec. 2015, pp. 2402–2414. [Online]. Available: <https://doi.org/10.1109/tpami.2015.2408360>
- [8] X. Liang, X. Shen, D. Xiang, J. Feng, L. Lin, and S. Yan, "Semantic object parsing with local-global long short-term memory," in *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jun. 2016, pp. 710–724. [Online]. Available: <https://doi.org/10.1109/cvpr.2016.347>
- [9] X. Liang, C. Xu, X. Shen, J. Yang, S. Liu, J. Tang, L. Lin, and S. Yan, "Human parsing with contextualized convolutional neural network," in *International Conference on Computer Vision (ICCV)*. IEEE, Dec. 2015, pp. 1150–1168. [Online]. Available: <https://doi.org/10.1109/iccv.2015.163>
- [10] I. Kokkinos, "UberNet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory," in *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jul. 2017, pp. 1380–1410. [Online]. Available: <https://doi.org/10.1109/cvpr.2017.579>
- [11] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *International Conference on Computer Vision (ICCV)*. IEEE, Dec. 2015, pp. 800–820. [Online]. Available: <https://doi.org/10.1109/iccv.2015.304>
- [12] B. Bischke, P. Helber, J. Folz, D. Borth, and A. Dengel, "Multi-task learning for segmentation of building footprints with deep neural networks," in *International Conference on Image Processing (ICIP)*. IEEE, Sep. 2019, pp. 630–647. [Online]. Available: <https://doi.org/10.1109/icip.2019.8803050>
- [13] X. Liang, K. Gong, X. Shen, and L. Lin, "Look into person: Joint body parsing & pose estimation network and a new benchmark," in *Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 4. IEEE, Apr. 2019, pp. 871–885. [Online]. Available: <https://doi.org/10.1109/tpami.2018.2820063>
- [14] R. A. Guler, N. Neverova, and I. Kokkinos, "DensePose: Dense human pose estimation in the wild," in *CVF Conference on Computer Vision and Pattern Recognition*. IEEE, Jun. 2018, pp. 1120–1135. [Online]. Available: <https://doi.org/10.1109/cvpr.2018.00762>
- [15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
- [16] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Computer Vision – ECCV*. Springer International Publishing, 2014, pp. 740–755. [Online]. Available: [https://doi.org/10.1007/978-3-319-10602-1\\_48](https://doi.org/10.1007/978-3-319-10602-1_48)
- [17] X. Liang, S. Liu, X. Shen, J. Yang, L. Liu, J. Dong, L. Lin, and S. Yan, "Deep human parsing with active template regression," vol. 37, no. 12. IEEE, Dec. 2015, pp. 2402–2414. [Online]. Available: <https://doi.org/10.1109/tpami.2015.2408360>
- [18] P. Rot, M. Vitek, K. Grm, Ž. Emeršič, P. Peer, and V. Štruc, "Deep sclera segmentation and recognition," in *Handbook of vascular biometrics*. Springer, Cham, 2020, pp. 395–432.
- [19] Ž. Emeršič, D. Sušan, B. Meden, P. Peer, and V. Štruc, "Contextednet: Context-aware ear detection in unconstrained settings," *IEEE Access*, vol. 9, pp. 145 175–145 190, 2021.

## APPENDIX

In the main part of the paper, we mostly focused on the presentation of segmentation results, since it is the main task that our model is addressing. Here, we present some additional visual results for segmentation, as well as for the supporting tasks.

### A. Segmentation results on ATR

Fig. 8 shows qualitative segmentation results for the ATR dataset. Both JPPNet and SPD achieve similar performance on the first and third example image. It appears that both models have difficulty distinguishing between various types of upper clothing. The second image shows a non-typical example that is composed of two separate images. Our SPD model achieves far superior performance on the upper part of the image, whereas both models struggle to handle the lower part. This is most likely due to the models being trained to segment images of a single person.

### B. Pose results

Figs. 9 and 10 show visual results of our pose module compared to the JPPNet on LIP and MPII datasets, respectively. In Fig. 9, the first column shows a simple example - a full image of a person where all limbs are well visible. The other two columns show more challenging examples, where the target is not fully visible in the image. Both models still provide good estimation of the upper body and limbs, but struggle with the legs due to occlusion or not being fully included in the image. In the last image, our model provides a better estimate of the occluded right leg than JPPNet. Both models achieve comparable results on the images from the MPII dataset which was not used for training any of the models. It again shows that both of the models have difficulty handling lower limb occlusions.

### C. Dense Pose Results

Due to the dense pose task not being included in the JPPNet model, we compare our model's performance to the original DensePose model [14]. Visual results are shown in Figure 11. It is difficult for a human to visually evaluate the dense pose performance in detail, due to the nature of representation, however we observe that our model occasionally fails to cover all of the body area. Despite that, as pointed out in the main paper, the imperfect dense pose branch still directs the model learning enough to improve the overall segmentation predictions.

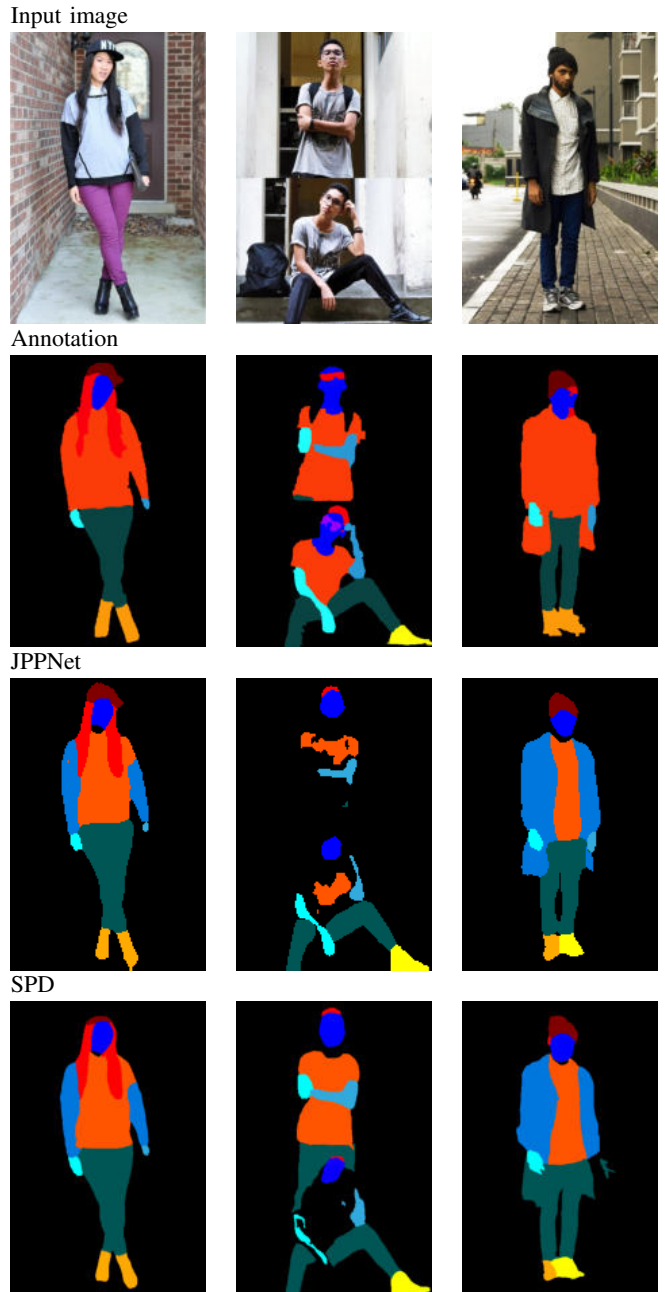
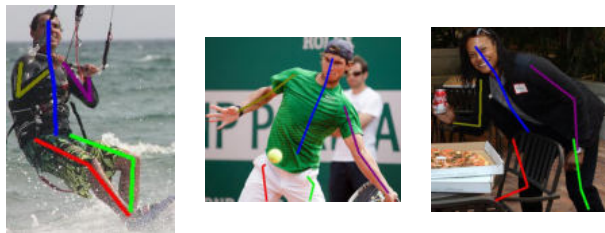


Fig. 8. Comparison of segmentation masks for selected images from the database ATR.

Annotation



JPPNet



SPD

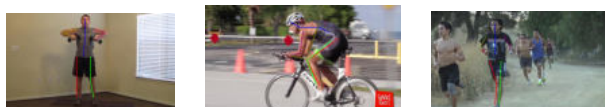


Fig. 9. Comparison of pose estimation results for selected images from the database LIP.

Annotation



JPPNet



SPD



Fig. 10. Comparison of pose estimation results for selected images from the database MPII.

DensePose



SPD

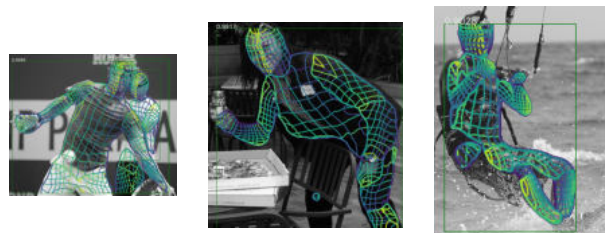


Fig. 11. Comparison of dense pose estimation results for selected images from the database COCO.

# Aerial Supervision of Drones and Other Flying Objects Using Convolutional Neural Networks

Vivian Ukamaka Ihekoronye, Simeon Okechukwu Ajakwe, Dong-Seong Kim, Jae Min Lee  
Department of IT Convergence Engineering, Kumoh National Institute of Technology, Gumi, South Korea  
(ihekoronyevivian, simeonajlove)@gmail.com,(dskim, ljmpaul)@kumoh.ac.kr

**Abstract**—Accurate detection of distant drones in clustered environment amidst other flying objects such as birds is of critical importance in anti-drone system design. This study proposed a novel object detection model that efficiently detect and differentiate drones from other flying objects under different weather conditions. The custom dataset consists of manually generated drone images and bird samples under sunny, cloudy and evening conditions. The simulation result shows that KITYOLO outperformed YOLOv5 both in precision (sunny 96.2% vs 85%; cloudy 73.7% vs 26.3%; evening 58.5% vs 26.1%) and recall (evening 42.4% vs 15%) in all aspects with an overall F1-score of 98% as against 91.9% while maintaining timeliness and memory usage.

**Index Terms**—Aerial supervision, CNN, Drone, Detection, YOLO.

## I. INTRODUCTION

Drone companies are currently experiencing tremendous increase in sales due to the varying application of drones in different sectors. Gone are the days when drones are mostly deployed in the military sector for supervision and most times as a means of penetrating adverse terrain. Currently, drones are widely used by individuals and in the entertainment industries as hobbyist drones for aerial photography, agriculture application, object detection and logistics as seen by the Amazon groups. The high applications of drones is majorly due to the reduced cost and the miniature size of drones, experienced in its very swift maneuverability. The increase in the mis-usage of drones internationally is devastating, leading to economic loss of lives and properties. In early 2019, different airports in the UAE, USA and UK experienced mishaps as a result of drone operations with the recent violations in Saudi's Oilfield Aramco and Abha airport, where drones were illegally flown [1]. These circumstances require austere security measures because most drones have cameras mounted on them, making them capable of spying and retrieving confidential information in restricted areas. Also, transportation of explosives can easily be achieved with drones, making them very dangerous when used by attackers or terrorists. Thus, detecting and preventing such malicious practice implemented through the deployment of drones is crucial for the security of the society.

Drone detection, also known as anti-drone technology, is an act of detecting and/or tracking unwanted drones in any given restricted area or territory [2]. However, the similarity of drones and other flying objects in aerial view is the major challenge of detecting and restricting drones. Different techniques have been adopted for the detection of drones, ranging from

acoustic [3] method that uses sensors to determine the sound emitted by the drone; radar approach [4] that implements radio waves to determine the distance, angle and velocity of the target object, infrared sensor; which uses the heat signature of the drone for detection [5], to the most paving technique which is computer vision (CV) technology, a field of Artificial Intelligence (AI) that enables computer to retrieve information from digital images, videos and other visual inputs, while reporting to the ground control stations.

Currently, computer vision is being used in solving object detection problems which is a peculiar AI problem, by deploying deep learning algorithms. Innovations in the different deep learning models have displayed consequential usage for object detection in ground based applications [6]. The extraction of meaningful information from images and videos can be achieved through detecting the image and also classifying it. To achieve the detection and classification of objects, Convolution Neural Networks (CNN), a deep learning algorithm is used for this purpose. CNN is responsible for the deep extraction of image features at different layers [7]. This paper seeks to address the challenge of aerial supervision of drones as target whilst accurately recognizing and predicting it from other objects such as birds in any weather condition. A state of the art CNN model was designed to solve this challenge, and also the proposed model was further compared with a very fast object detector based on computational complexity, accuracy and timeliness while achieving the following specific objectives:

- 1) Deploying computer vision for image capturing and processing of targets(drones and birds);
- 2) Gathering and labelling different datasets of 2 different drones and birds on flight;
- 3) Designing a state-of-the-art model that can optimally detect, predict and classify drones as well as birds based on computer vision and CNN;
- 4) Evaluation of the feasibility of the proposed model with the state-of-the-art model based on accuracy, sensitivity and computational complexity.

The remaining sections of this paper are categorized as: Section II, captures Related Works; the System Design is extensively discussed in Section III, while the Result Discussion, Evaluation Performance and Conclusion are captured in Sections IV and V respectively.

## II. RELATED WORKS

Computer vision is one of the most essential domain of AI, having different sectors such as object detection, image recognition and surveillance [8], with object detection being the most blooming sector as a result of its enormous applications. Object detection is the ability of computer (i.e anti-drone system in this research work) and software systems (i.e the proposed system) to locate objects in an image/ video and accurately distinguish each object. The rapid adoption of deep learning in computer vision has brought breakthrough to highly accurate object detection algorithms such as You Only Look Once (YOLO) [9], Single Shot Detectors(SSD), Fast-Regional Convolution Network (FRCNN) and Faster RCNN [9]. Nowadays, object detection is being deployed for surveillance operations, face recognition and security systems. Also, UAVs application is on the rise due to their high mobility, suitable incorporation in object detector models, easy deployment and their capacity to capture images at any altitude in respect to views, angles and scalar differences [10]. This in turn has posed challenges such as densely distributions of target objects, scale variance of aerial objects, and differentiation of UAVs and other flying objects in different weather conditions in airborne.

To contribute towards solving these recurring challenges, researchers have resorted to CNN for optimal solutions [11]. CNN also known as a deep learning algorithm, receives images as inputs, delegates learnable weights and biases to the objects in the image, then carry out prediction tasks on the objects based on their various classes. Several layers exist and are interconnected in the architecture of the CNN which is analogous to the connective patterns of neurons of the human brain, making it to be trained on any particular tasks based on the given parameters. With a focus on visual capturing and detection, most anti-drone detecting systems [5] are equipped with cameras that aids in the panning, tilting and zooming of target objects. The automatic techniques employed by CNN in image processing has also resulted to the wide interest of researchers, owing to the fact that CNN displays excellent performance in object detection and classification .

Classification of birds, drones and backgrounds were carried out by [5], evaluating several CNN models such as Resnet-50, Resnet-18, VGG16, Gogglenet, AlexNet and SqueezeNet to ascertain the best classifier. Although, these classifiers have already been validated in the ImageNet Large Scale Visual Recognition Competition (ILSVRC) with 1000 labels classification, their experiment however displays that for the classification of small number of labels, a simplified CNN model results to better performance. As Alexnet, Restnet18 and Squeezenet performed optimally than the other models when classifying just 3 labels, which was contrary to the result gotten from the 1000 label classification of ILSVRC. While, author [12] compared two variants object detectors, that is YOLO versions 2 and 3, for the detection and classification of drones from no drones with a total of 149 images for the training and validation of the model, having a higher accuracy

of 95.20% for YoloV3.

To solve the problem of scale variations and densely distributions of objects, researchers [13] designed SPB-YOLO model, an end-to-end detector that has the strip bottleneck (SPB) module which used the attention mechanism approach to solve the dependency of scalar variations of UAV images. Also, by the upsampling of the detection head of YOLOv5 in the addition of a detection head based on Path Aggregation Network, the challenge of dense object distribution was mitigated. However, the disparity experienced in the detection and classification of aerial targets in different weather conditions is still a research gap. For anti-drones to be efficacious in drone detection and prediction of similar targets even during weather obscurity, an efficient state-of-the-art model needs to be embedded in it for optimal accuracy and speed.

YOLO architecture is a plausible innovation of Artificial intelligence for computer vision. YOLO is a single-shot object detector that is extremely fast when compared with its counterpart; multiple-shot detectors such as Fast-RCNN and Faster-RCNN. This is as a result of the YOLO technique, that uses the features extracted from the entire captured image while predicting the classes of the images simultaneously from bounding boxes. The architecture and mode of detecting and predicting drone images from other images by the YOLO model which is incorporated in the anti-drone system will be explained in the subsequent sections.

## III. SYSTEM DESIGN

The operational processes of the proposed model is captured in Fig. 1. This model adopts YOLO architecture for the detection and classification of drones and other objects in aerial perspective. During surveillance, the anti-drone system captures all aerial images within its peripheral and central vision; drones and birds alike. The input to the proposed model are images extracted from the anti-drone system, which are subjected to further processing deploying the model's architecture.

The main functionality of this model is to detect and distinguish drones from birds, without being deterred by obscure weather conditions nor altitude of the object as it was trained under a sunny, cloudy and gloomy (evening) scenarios and at various heights, so as check the robustness of the model to accurately detect and predict the movement of drones in restricted places. Therefore, the inputs to the system is either drones or birds, relying on the architectural framework of the system; it processes the input and generate outputs based on the classifications of the object.

### A. Custom Drone Detection Strategy

The standard YOLO architecture detector is designed on three distinct modules. The Backbone Module that adopts the Cross Stage Partial Network (CSPNet) [14] responsible for drone/bird feature extractions. Next, is the Neck built on the Feature Pyramid Network, for features aggregation. Lastly, the Head Module that aids the model to handle varying sizes of objects and capable of generating multi-scale predictions.

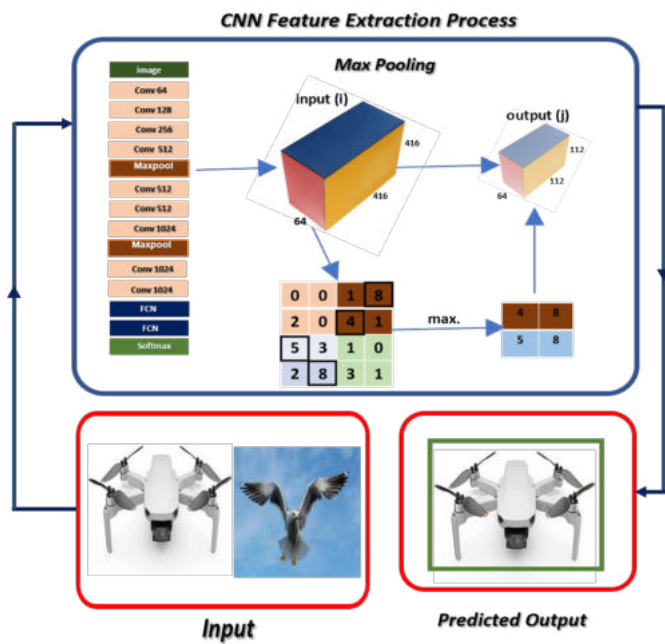


Fig. 1. Overview of System Design

Due to the dynamic and flexible nature of the architecture of YOLOv5, the model can be augmented considering the relative challenge to be solved, hence, the addition of Path Aggregation Network (PANet) to the proposed model which is an integral part, so as to mitigate the sparsely and densely distribution of the nature of the target and accurately classify it from other objects in airborne.

As earlier stated, the proposed model, known as KITYOLO, is designed deploying the framework of YOLOv5; being the latest version of the YOLO series (v1, v2, v3 and v4). Though YOLOv5 is a very fast object detector that is capable of extracting 140 frames per second (fps) in real time, its major challenge is extracting features from densely distributed objects with optimal accuracy. Therefore, KITYOLO is designed to solve this inherent problem peculiar to YOLOv5, which is a vital issue in detecting and preventing drones in restricted areas. The disparity, similarity, and tininess of the targets (drones and birds) created the need for instance segmentation. That is, the need to explicitly detect, classify and localize various object instances in an image. Hence, the addition of Path Aggregation Network (PANet) to KITYOLO (which is missing in the standard YOLOv5), so as to enhance the propagation of low-level features, captured in Fig. 2; depicting an improved and better architecture.

As feature extraction takes place in the network, from high level to low level layers, the complexity of consecutive layers increases, leading to a corresponding decrease in spatial resolution. Fig. 3 explains the PANet adopted in KITYOLO and Feature Pyramid Network (FPN) used in the architectural design of YOLOv5. The FPN deployed in YOLOv5 follows a top-down path (Fig. 3(a)) integrating rich features from high level layers with accurate localization from lower level

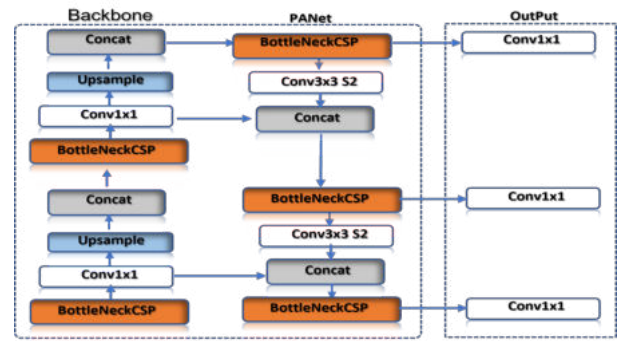


Fig. 2. Custom KITYOLO Drone Detector Model

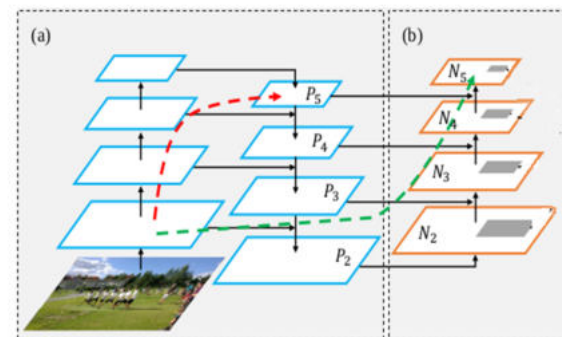


Fig. 3. Custom backbone PANet (a) FPN backbone used in YOLOv5 (b) Bottom-up Path Supplement

layers, by upsampling the layers. This approach follows a longer path which increases the network complexity as well as the latency, thereby reducing the model's accuracy when detecting very tiny objects. Unlike FPN, PANet deployed in the proposed model follows a bottom-up path approach (Fig. 3(b)), which reduces the number of paths as well as the network's complexity, having a resultant positive effect in the accurate detection of very tiny objects. The Neck compartment generally is responsible for feature aggregation and to improve the accurate localization of features in lower layers, leading to the overall object location accuracy.

The difference between drones and birds seems to fade off once the detection distance reaches or supersedes 100 meters as in the case of this research, making both objects appear similar during detection when at flight. To detect and classify drones from birds, KITYOLO captures the entire image during run time using a single convolution network, making it capable of predicting objects of different classes based on confidence at a faster rate. The input image in the YOLO architecture is splitted into  $S \times S$  number of grids, with each grid having  $B$  bounding boxes along side their confidence scores as displayed in Fig. 4. Also, each bounding box is made up of 5 predictions ( $x, y, w, h$  and  $c$ ); where  $x, y$  depicts the coordinates representing the center of the box of the grid cell,  $w, h$  representing the width and height of the grid and  $c$  the confidence prediction, representing the Intersection Of Union (IOU) between the



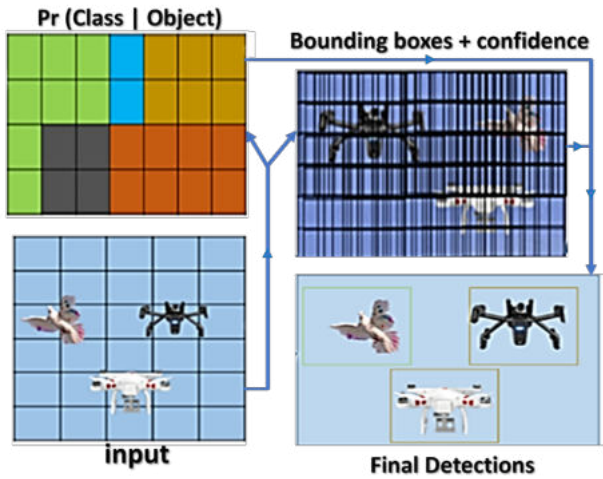


Fig. 4. Processing of Capturing Image value using bounding boxes

predicted box and ground truth box as shown in equation (1):

$$Box_{(cs)} = Pr_{(object)} \times IOU_{(b, object)}, \quad (1)$$

where  $Box_{(cs)}$  is the box confidence score,  $Pr_{(object)}$  is the probability of an object in the grid, and  $IOU_{(b, object)}$  is Intersection of Union express as area of union of two boxes. For non-linearity in the network, YOLOv5 uses Softmax activation function to classify its multi-classes output. Softmax function returns the probability of each class using the given equation (2):

$$\sigma(Z^{\rightarrow})_i = \frac{e^{Z_i}}{\sum_{j=i}^k e^{Z_j}} \quad (2)$$

where  $\sigma$  is softmax,  $(Z^{\rightarrow})_i$  is input vector,  $e^{Z_i}$  is the standard exponential function for input vector,  $k$  is the number of classes in the multi-class classifier,  $e^{Z_j}$  is the standard exponential function for output vector.

### B. Dataset Capturing and Description

The dataset used for this research comprises drones and birds images. Two different drones; Mavic-Air and Mavic-Enterprise were separately flown in three different scenarios of *sunny*, *evening(gloomy)* and *cloudy* weather conditions. The videos of flown drones were captured at different time of the day to reflect their distinct scenario characteristics. Image frames were extracted from the video sequence of 1190 data frames from both drones, and labelled using Makesense software to generate ground truth values from the initial background values viz bounding boxes. The bird datasets of 950 images were gotten from Kaggle, a free online resource for datasets, but the images were manually labelled.

### C. Simulation and Experimental Setup

The datasets of birds and drones were combined and distributed in a ratio of, 70% for training of the model. 20% for model testing and 10% for the validation of the model, so as to avoid overfitting and to achieve optimal performance. In

addition, the simulation was done in a Python environment on a system configuration of Intel(R) Core(TM) i5-7400 CPU @ 3.00GHz, 8GB RAM, GPU Tesla K80. Several hyper-parameters such as best weights for weights initiation, input size of 416, batch size of 16 and learning rate with an epoch of 100 were used.

## IV. RESULT DISCUSSION AND PERFORMANCE EVALUATION

To test the model's effectiveness in detecting and classifying drones from birds under different climatic conditions, metrics such as F1-score, precision, recall, number of frame per second (fps), and memory usage (GLOPS) were used to compare the proposed model with YOLOv5 model, as both models were trained and tested with the same dataset .

The result in Table I highlights the detection performance of KITYOLO and YOLOv5 models under different weather conditions and heights using a uniform dataset to prevent every form of bias.

TABLE I  
DETECTION RESULTS OF DRONE-BIRD

Scenario	Drone-Bird Detection			
	KITYOLO (%)		YOLOv5 (%)	
	Precision	Recall	Precision	Recall
Mavic_Enter_Cloudy	69.9	73.7	27.4	26.3
Mavic_Enter_Evening	57.5	60.0	47.0	90.0
Mavic_Enter_Sunny	92.9	90.0	90.5	95.5
Mavic_Air_Cloudy	96.1	1.00	95.6	1.00
Mavic_Air_Evening	58.5	42.4	26.1	15.0
Mavic_Air_Sunny	96.2	1.00	85.1	1.00
Bird	79.8	91.6	66.8	94.7

For drone-bird detection, the result from Table I indicates that KITYOLO has a superior precision value of 79.8% than YOLOv5 which is 66.8%. Across weather conditions, KITYOLO had a higher recall of 73.7% as against 26.3% of YOLOv5 in a cloudy weather. Also for evening, KITYOLO had a better precision and recall values of 58.5% and 42.4% than 26.1% and 15.0% of YOLOv5. Lastly, in sunny condition, a 96.2% precision value by KITYOLO affirms its detection capability than the 85.1% value of YOLOv5.

### A. Performance Evaluation

The result in Table II highlights the comparison of KITYOLO with YOLOv5 in terms of speed, rationality of detection (F1-score), and memory usage (GFLOPS). F1-score is a test

TABLE II  
MODELS PERFORMANCE EVALUATION

Models	Performance of Models for Weapon Detection		
	F1Score (%)	Time (FPS)	GFLOPS
<b>KITYOLO</b>	<b>98.0</b>	<b>0.022s</b>	<b>16.4</b>
YOLOv5s	91.9	0.022s	16.4

of the behaviour of model with changes in its precision and recall expressed as:

$$\hookrightarrow F1score = \frac{2(Precision \times Recall)}{Precision + Recall}, \quad (3)$$

From Table II and Fig. 5, it can be clearly seen that KITYOLO achieved a superior detection performance of 98% than YOLOv5 of 91.9%; which is a significant 6.1% increase in detection rationality despite that the two models had the same time of detecting each object per second (0.022s) and 16.4 GFLOPS.

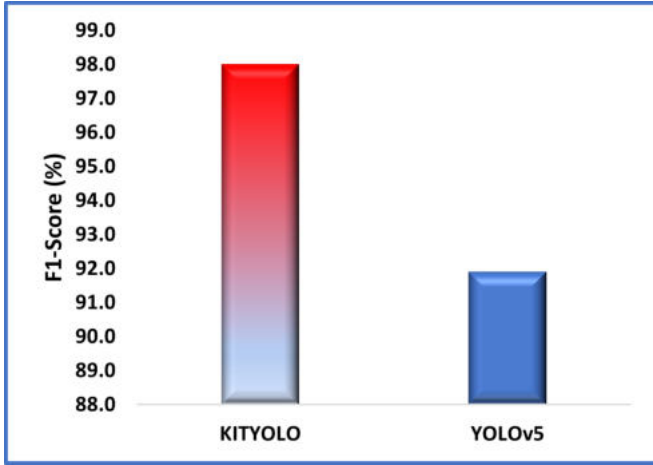


Fig. 5. F1-score Performance Comparison of KITYOLO and YOLOv5s

The confusion matrix in Fig. 6 indicates that KITYOLO can not only accurately detect different types of drones under different weather conditions, but also differentiate it from all kinds of birds in a timely manner and with less computational complexity and minimal false alarm rate. The images on Fig. 7 are samples of drones and birds detection by KITYOLO under different weather conditions and heights. The displayed results are detection and classification tasks carried out concurrently by the proposed model showing degree of accuracy and sensitivity.

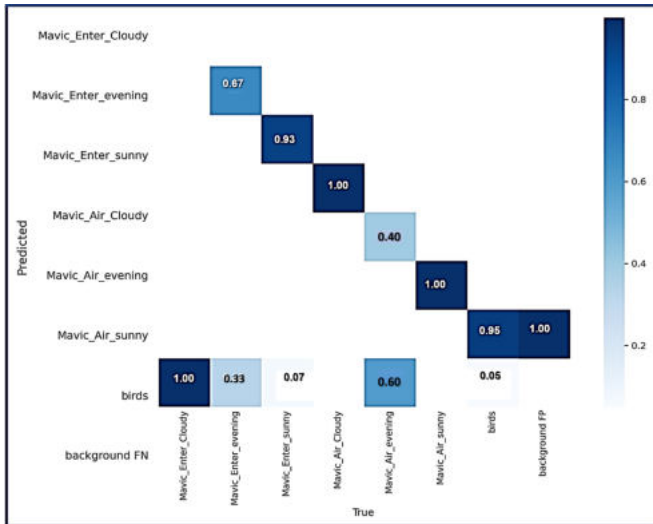


Fig. 6. Confusion Matrix of KITYOLO

These results show a high detection improvement by the proposed model in comparison with YOLOv5 in detecting tiny objects under different weather conditions in a timely

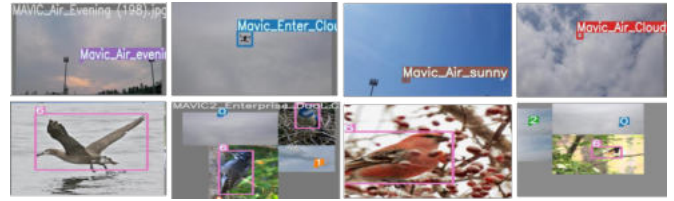


Fig. 7. Samples of drone-bird detection by KITYOLO

manner and less resource usage. However, a closer look at the results in Table I indicates a drop in the detection performance during evening/gloomy conditions which is an ongoing research challenge in computer vision.

## V. CONCLUSION

This work presents a novel drone detection model; KITYOLO that improved the accuracy and precision of tiny objects in different weather condition while maintaining time-liness and computational complexity. In the future, we hope to increase our dataset and improve the model for robust performance.

## ACKNOWLEDGMENT

This research work was supported by Priority Research Centers Program through NRF funded by MEST (2018R1A6A1A03024003) and the Grand Information Technology Research Center support program (IITP-2021-2020-0-01612) supervised by the IITP by MSIT, Korea.

## REFERENCES

- [1] S. Al-Emadi and F. Al-Senaid, "Drone Detection Approach Based on Radio-Frequency Using Convolutional Neural Network," in *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*, 2020, pp. 29–34.
- [2] S. Ajakwe, R. Arkter, D. Kim, D. Kim, and J.-M. Lee, "Lightweight cnn model for detection of unauthorized uav in military reconnaissance operations," in *2021 Korean Institute of Communication and Sciences Fall Conference. (KICS)*, 11 2021.
- [3] M. Z. Anwar, Z. Kaleem, and A. Jamalipour, "Machine Learning Inspired Sound-Based Amateur Drone Detection for Public Safety Applications," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2526–2534, 2019.
- [4] R. H. Geschke, A. Shoykhetbrod, R. Brauns, C. Schwäbig, S. Wickmann, S. Leuchs, C. Krebs, A. Küter, and D. Nüssler, "Post-integration Antenna Characterisation for a V-band Drone-detection Radar," in *2021 15th European Conference on Antennas and Propagation (EuCAP)*, 2021, pp. 1–4.
- [5] H. M. Oh, H. Lee, and M. Y. Kim, "Comparing Convolutional Neural Network(CNN) models for machine learning-based drone and bird classification of anti-drone system," in *2019 19th International Conference on Control, Automation and Systems (ICCAS)*, 2019, pp. 87–90.
- [6] P. Gajalakshmi, J. V. Satyanarayana, G. V. Reddy, and S. Dhavale, "Detection of Strategic Targets of Interest in Satellite Images using YOLO," in *2020 4th International Conference on Computer, Communication and Signal Processing (ICCCSP)*, 2020, pp. 1–5.
- [7] W. Budiharto, A. A. S. Gunawan, J. S. Suroso, A. Chowanda, A. Patrik, and G. Utama, "Fast object detection for quadcopter drone using deep learning," in *2018 3rd International Conference on Computer and Communication Systems (ICCCS)*, 2018, pp. 192–195.
- [8] J. Harikrishnan, A. Sudarsan, A. Sadashiv, and R. A. Ajai, "Vision-face recognition attendance monitoring system for surveillance using deep learning technology and computer vision," in *2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (VITECoN)*, 2019, pp. 1–5.

- [9] D. K. Behera and A. Bazil Raj, "Drone Detection and Classification using Deep Learning," in *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2020, pp. 1012–1016.
- [10] S. Ajakwe, R. Arkter, D. Kim, G. Mohatsin, D. Kim, and J.-M. Lee, "Anti-drone systems design: Safeguarding airspace through real-time trustworthy ai paradigm," in *The 2nd Korea Artificial Intelligence Conference. (KAIC)*, 09 2021.
- [11] D. T. Wei Xun, Y. L. Lim, and S. Srigrarom, "Drone detection using YOLOv3 with transfer learning on NVIDIA Jetson TX2," in *2021 Second International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP)*, 2021, pp. 1–6.
- [12] S. A. Hassan, T. Rahim, and S. Y. Shin, "Real-time UAV Detection based on Deep Learning Network," in *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, 2019, pp. 630–632.
- [13] X. Wang, W. Li, W. Guo, and K. Cao, "SPB-YOLO: An Efficient Real-Time Detector For Unmanned Aerial Vehicle Images," in *2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC)*, 2021, pp. 099–104.
- [14] K. Luo, R. Luo, and Y. Zhou, "Uav detection based on rainy environment," in *2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, vol. 4, 2021, pp. 1207–1210.

# Performance Analysis of UAV-based Array Antenna Arrangement for Target Detection

Ji-Hyeon Kim

Dept. of Electronics Engineering  
Pusan National University  
Busan, Republic of Korea  
kjihyeon@pusan.ac.kr

Soon-Young Kwon

Dept. of Electronics Engineering  
Pusan National University  
Busan, Republic of Korea  
ysk1680@pusan.ac.kr

Hyoungh-Nam Kim

Dept. of Electronics Engineering  
Pusan National University  
Busan, Republic of Korea  
hkim@pusan.ac.kr

**Abstract**—UAV-based array antennas can autonomously make the shapes of UAV array to provide an optimized placement to achieve a specific goal. In this paper, three representative antenna arrays are considered as a candidate for the optimized placement and their detection performance is analyzed in terms of target detection performance. Through the simulation results, we present the most suitable antenna arrangement for the given environment.

**Keywords**—UAV, array antenna, target detection

## I. INTRODUCTION

As the use of unmanned aerial vehicles (UAVs), popularly known as drones, is growing rapidly, research on the operation and application of UAV has been actively conducted in various fields. In the military field, UAVs have been used for reconnaissance and surveillance, electronic warfare, attack missions using UAVs. Private sector UAVs are used in many domains such as performances and home delivery [1-5]. In particular, UAVs can play a key role in enabling wireless connectivity in various scenarios such as public safety and Internet of Things (IoT) scenarios [1-3]. Effective use of UAVs in such scenarios requires array signal processing technology that provides optimal UAV antenna arrangement.

Whereas conventional array antennas have a specific type of fixed antenna structure to achieve their purpose, UAV-based array antennas can autonomously transform the shapes of UAV array to provide an optimized placement to achieve a specific goal. For example, if it is necessary to acquire a low-frequency signal in the battlefield area, it may be changed to a linear array antenna having an appropriate interval for a search frequency. It can be also changed to an array with high directivity to improve search and jamming performance for specific areas [4-7].

In this paper, we try to find an optimal antenna array when implementing a target detection system using a UAV array antenna. For the study of UAV array antenna arrangement for target detection, the detection performance of three representative antenna arrays is analyzed. The types of arrays use linear, circular, and rectangular arrays, and an antenna array suitable for a fixed target is obtained through comparison of beamforming gains of each array antenna.

This paper organized as follows: Section II briefly describes the UAV system model and antenna arrays. Through simulation, the target detection performance of the UAV array antenna is analyzed in Section III. Finally, we conclude this paper in Section IV.

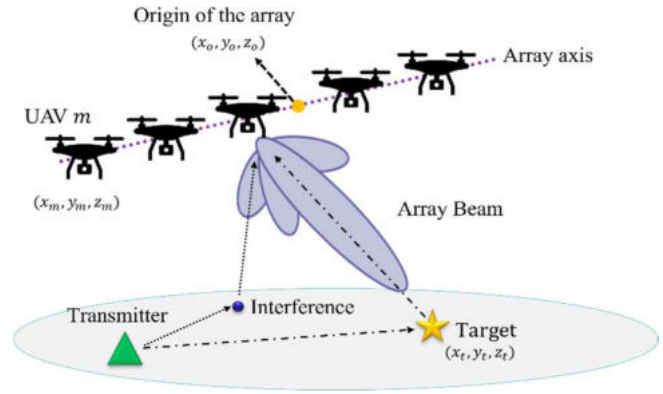


Fig. 1. UAV-based antenna array.

## II. SYSTEM MODEL AND ANTENNA ARRAY ARRANGEMENT

### A. System Model

Consider a target located within a given geographic area. In this area, a set  $\mathcal{M}$  of  $M$  UAVs are used as flying objects to detect targets on the ground. The  $M$  UAVs forms an antenna array where each element is a single antenna UAV, as shown in Fig. 1. For tractability, we consider a linear antenna array whose elements are symmetrically located to the origin of the array. The three-dimensional (3D) locations of UAV  $m \in \mathcal{M}$  and target are given by  $(x_m, y_m, z_m)$  and  $(x_t, y_t, z_t)$ . To avoid collisions between UAVs, we assume that adjacent UAVs in the array are separated by at least  $D_{min}$  [7].

For this UAV array system, a transmitter is employed to send narrowband signals, and the echo signals reflected from far-field targets are then received by the each UAV. There are also interference signal. Assume that there are  $K$  narrowband signals  $s_{m,k}(t)$ ,  $k = 1, 2, \dots, K$ , observed at the  $m$ -th array element. And we use  $\mathbf{x}_m(t)$  to represent the observed signal vector, and the narrowband array output model is given by

$$\mathbf{x}_m(t) = \mathbf{A}(\boldsymbol{\theta}_m, t)\mathbf{s}_m(t) + \bar{\mathbf{n}}_m(t) \quad (1)$$

where  $\mathbf{s}_{m,k}(t) = [s_{m,1}(t), s_{m,2}(t), \dots, s_{m,K}(t)]^T$  is the signal vector consisting of all reflected signals, and  $\{\cdot\}^T$  denotes the transpose operation.  $\bar{\mathbf{n}}_m(t)$  represents the noise vector of the  $m$ -th array UAV. And the steering matrix  $\mathbf{A}(\boldsymbol{\theta}_m, t) = [\mathbf{a}(\theta_{m,1}, t), \dots, \mathbf{a}(\theta_{m,K}, t)]$ , with its  $k$ -th column vector  $\mathbf{a}(\theta_{m,k}, t)$  being the steering vector corresponding to the  $k$ -th source signal.

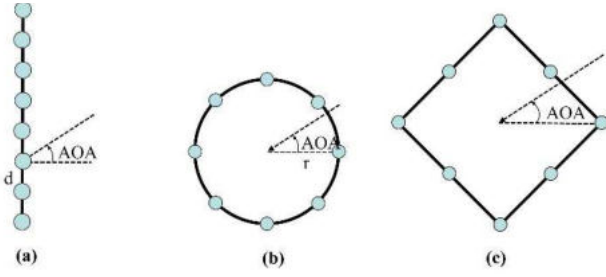


Fig. 2. Antenna array configurations: (a) ULA, (b) UCA, (c) URA.

### B. Antenna Array Arrangement

In this paper, three representative antenna types including ULA, UCA, and URA are chosen to analyze target detection performance. The spatial structure of these three antenna arrays are illustrated in Fig. 2.

#### 1) Uniform Linear Array (ULA)

Uniform linear array (ULA) is a collection of sensor elements equally spaced along a straight line. The property means that the array accepts a signal from a particular direction and rejects the signal from another direction [10]. The expression of the steering vector of ULA at each angle of arrival (AoA) is defined as follows:

$$\mathbf{a}(\phi) = [1, \dots, e^{jkd \sin \phi}]^T \quad (2)$$

with  $d$  is distance between the antennas, and  $k = 2\pi/\lambda$  is the wave number.

#### 2) Uniform Circular Array (UCA)

Uniform circular array (UCA) is formed from identical sensor elements equally spaced around a circle. As a type of planar array, it provides a more symmetrical pattern with lower side lobes and much higher directivity [10]. The steering vector of UCA at each AoA can be obtained as follows:

$$\mathbf{a}(\phi) = [e^{jkr \cos(\phi - \phi_1)}, \dots, e^{jkr \cos(\phi - \phi_M)}]^T \quad (3)$$

with  $r$  being the radius of the circular array, and  $\phi_M$  the angle of the  $m$ -th array element with respect to horizontal axis.

#### 3) Uniform Rectangular Array (URA)

Uniform rectangular array (URA) refers to an antenna in which sensors are arranged at equal intervals on a square plane.

TABLE I. SIMULATION PARAMETERS

Parameters	Values or variables	
	Signal 1 (SOI)	Signal 2 (interference)
Carrier frequency $f_c$	150 MHz	
Angle of arrival (AoA) (azimuth, elevation)	$(-37^\circ, 10^\circ)$	$(17^\circ, 10^\circ)$
# of sensor elements $M$	16	
Signal to noise ratio (SNR)	-20 dB	

It has the characteristics of a planar array like UCA and can be used to scan the main beam towards any point in space. The steering vector of URA at each AoA is expressed as follows:

$$(\phi) = \begin{bmatrix} 1 \\ e^{-j\chi} \\ \vdots \\ e^{-j(Q-1)\chi} \\ e^{-j\gamma} \\ e^{j(\chi+\gamma)} \\ \vdots \\ e^{-j((Q-1)\chi+(P-1)\gamma)} \end{bmatrix} \quad (4)$$

$$\chi \triangleq 2\pi \left(\frac{d}{\lambda}\right) \sin \theta \cos \phi, \quad \gamma \triangleq 2\pi \left(\frac{d}{\lambda}\right) \sin \theta \sin \phi \quad (5)$$

In Equations (4) and (5),  $\theta$  and  $\phi$  represent the elevation angle and azimuth angle of the signal,  $Q$  and  $P$  are the rows and columns of the array [9].

### III. SIMULATION RESULTS

In this section, simulations are performed to compare the target detection performance according to the arrangement of the UAV array antenna. The simulation results show the beamforming gain of the signal of interest (SOI) and interference that obtained according to the three antenna arrays. Specific simulation parameters are summarized in Table 1. The number of sensors is 16, in the case of URA, it has a  $4 \times 4$  arrangement. For convenience, the elevation angle of the SOI and the interference signal were set to be the same. A narrowband minimum-variance distortionless-response (MVDR) was used as the beamforming algorithm [11].

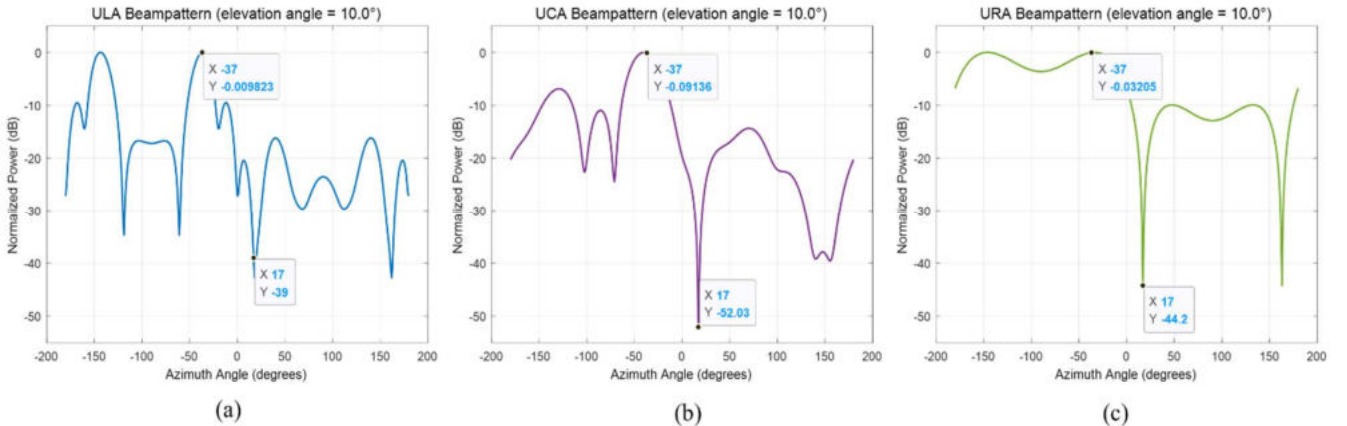


Fig. 3. Beampattern obtained with different antenna arrays: (a) ULA, (b) UCA, (c) URA.

Fig. 3 shows the beamforming gain obtained with three antenna arrays. All array antennas have the largest beam gain at  $-37^\circ$ , which is the direction of the SOI, and it can be confirmed that a deep null is formed at direction of the interference,  $17^\circ$ . Comparing the target detection performance of these arrays, UCA has the largest gain difference between the direction of the SOI and the interference signal by about 52 dB.

#### IV. CONCLUSIONS

For the study of UAV array antenna arrangement to detect the target, we analyzed detection performance using three representative antenna arrays such as ULA, UCA, and URA. Simulation results show that all three antenna arrays can extract the target signal and remove the interference signal. In particular, UCA has better performance than the other two arrays. A study to find an optimal antenna arrangement for various signals and operating environments will be conducted in the future works.

#### ACKNOWLEDGMENT

This research was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2021R1F1A1060025).

#### REFERENCES

- [1] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs," *IEEE Transactions on Wireless Communications*, vol. 15, no. 6, pp. 3949–3963, June 2016.
- [2] M. Alzenad, A. El-Keyi, and H. Yanikomeroglu, "3D placement of an unmanned aerial vehicle base station for maximum coverage of users with different QoS requirements," *IEEE Wireless Communications Letters*, Early access, 2017.
- [3] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for UAV-enabled multiple access," *IEEE Transactions on Wireless Communications*, Early access, 2017.
- [4] Ho Young Jeong, Byung Duk Song, Seokcheon Lee, "The Flying Warehouse Delivery System: A Quantitative Approach for the Optimal Operation Policy of Airborne Fulfillment Center," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1-10, July 2020.
- [5] Y. Zhou, C. Pan, P. L. Yeoh, K. Wang, M. ElKashlan, B. Vucetic, Y. Li, "Communication-and-Computing Latency Minimization for UAV-Enabled Virtual Reality Delivery Systems," *IEEE Transactions on Communications*, Nov. 2020.
- [6] P. S. Bithas, V. Nikolaidis, A. G. Kanatas, G. K. Karagiannidis, "UAV-to-Ground Communications: Channel Modeling and UAV Selection," *IEEE Transactions on Communications*, Vol. 68, Issue. 8, Aug. 2020.
- [7] M. Mozaffari, W. Saad, M. Bennis and M. Debbah, "Drone-Based Antenna Array for Service Time Minimization in Wireless Networks," *2018 IEEE International Conference on Communications (ICC)*, pp. 1-6, May 2018.
- [8] Garza Jesus, Panduro Marco A., Reyna Alberto, Romero Gerardo, and del Rio Carlos. "Design of UAVs-Based 3D Antenna Arrays for a Maximum Performance in Terms of Directivity and SLL," *International Journal of Antennas and Propagation*, Aug. 2016.
- [9] Ji-Youn Mun and Suk-Seung Hwang, "Performance Analysis of Adaptive Beamforming System Based on Planar Array Antenna," *Journal of the KIECS*. pp. 1207-1212, vol. 13, no. 6, Dec. 31 2018.
- [10] Antonio Forenza, David J. Love, and Robert W. Heath Jr, "Simplified Spatial Correlation Models for Clustered MIMO Channels With Different Array Configurations," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 4, Aug. 2007.
- [11] Van Trees, H. "Optimum Array Processing," New York: Wiley-Interscience, 2002.

# Image Prediction for Lane Following Assist using Convolutional Neural Network-based U-Net

Byung Chan Choi<sup>1,2</sup>, Jaeroek Kwon<sup>3</sup>, and Haewoon Nam<sup>1</sup>

<sup>1</sup>Division of Electrical Engineering, Hanyang University, Ansan, Korea

<sup>2</sup>RF Seeker R&D, LIG Nex1, Yongin, Korea

<sup>3</sup>College of Engineering and Computer Science, University of Michigan-Dearborn, Dearborn, MI, USA

**Abstract**—Current autonomous driving systems compute steering and throttle control commands by running perception-decision-action pipeline at high frequency. Although human drivers cannot react or control the vehicles as quickly as the autonomous driving softwares, most drivers control their vehicles to stay in lane unless they intend to break away from the lane. According to forward internal model theory, human can choose an optimal action for the best outcome by internally simulating all the possible consequences of various actions. This means that humans drivers choose the optimal motor commands for lane following based on their internal simulation of near-future lane changes. This paper proposes a convolutional neural network-based U-Net as a state estimator for forward internal model-based lane following assist. This state estimator can predict the lane image of near-future based on current lane image and driving status data, such as speed and steering angle. This paper also explains how time difference between current lane image and the next one to be predicted will affect the training and prediction output of the estimator.

**Index Terms**—Lane Following Assist, Deep Learning, Convolutional Neural Network, Internal Model

## I. INTRODUCTION

Lane Following Assist (LFA) is one of the basic functions that make the autonomous vehicle detect and follow the lanes on the road. Performance of LFA is determined by the vehicle's reaction time for wheel control. An autonomous vehicle needs time to process its sensor data and run lane detection algorithms before applying wheel controls. As a result, many automotive companies try to improve the performance of LFA by minimizing processing time for lane detection. However, this approach cannot make processing time for lane detection into zero. There will always be a delay between lane detection and wheel control.

Similar to LFA, human drivers also suffer the delay between lane detection and wheel control. However, although human drivers cannot control wheels as quickly or precisely as the autonomous vehicles, they can follow the lanes without any troubles. This is because human drivers use a different approach for lane following. Human drivers internally simulate how the lane will change in the future based on current driving status and choose the optimal action to achieve the best outcome for following the lanes on the road. One of the theoretical frameworks for a human to choose actions based on *internal simulation* is *forward internal model principle* of the cerebellum [1] [2] [3]. One key function of the cerebellum is to predict the sensory consequences of the motor outputs

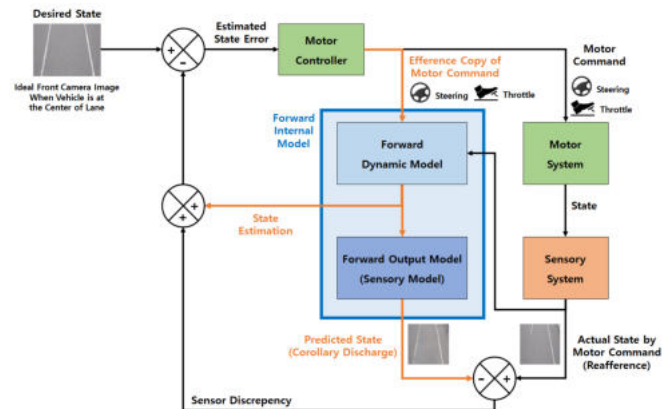


Fig. 1. Smith Predictor for Forward Internal Model-based LFA

by comparing its prediction to the sensory feedback and minimizing its error [4].

Human driver's forward internal model compensates for the latency between lane detection and wheel control. This internal model-based control can be implemented as Smith predictor, a feedback control system with time-delay compensation scheme [5] [6]. Fig 1 is the diagram of vision-based LFA system with forward internal model and Smith predictor design. The system receives the desired state for its task. For LFA, the desired state is the ideal front camera image when the vehicle is driving at the center of lane. When a driver applies throttle and steering controls, the system feeds these control commands to forward internal model and motor system. Motor system changes the vehicle's speed and orientation based on the driver's motor commands. Sensory system perceives the changes in state by motor commands and produces the front camera image output of changed state. Forward internal model uses an efference copy of motor commands and produces the prediction of next lane image state. Sensor discrepancy between the next lane image prediction and the front camera image of changed state will be added with state estimation results from forward internal model in order to adjust the estimation. The actions that produce the lowest error between the desired state and the state estimation by current motor commands will be selected as next motor commands.

Inspired by this idea, this paper proposes a deep neural network-based state estimator for forward internal model-

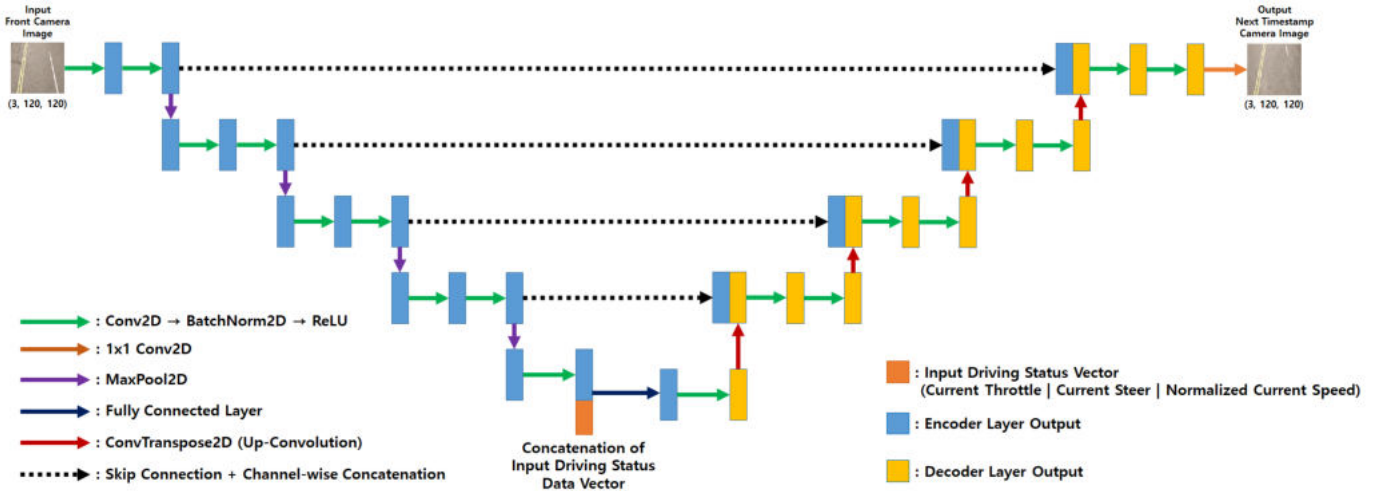


Fig. 2. CNN-based U-Net for State Estimation of Internal Model

based LFA. It utilizes Convolutional Neural Network (CNN)-based U-Net from [7] to predict next lane image from current lane image and driving status data. U-Net’s skip architecture provides feature reusability that creates thorough gradient update flow and prevent gradient vanishing during the training. This approach can be later implemented in internal simulation of forward internal model-based LFA in order to compensate for the latency between lane detection and wheel control.

## II. BACKGROUNDS

### A. Internal Model

In neuroscience, the internal model is a cognitive interpretation of organism motion planning. It claims that an organic agent, such as human, can anticipate the consequences of its actions without actually committing them [8] [9]. According to Wolpert et al. [10], there are two types of motion planning models : forward model and inverse model. In forward model, the agent predicts the sensory outcome of an action based on current state information and motion commands. With the proper forward model, it internally simulates actions and matches the outcome with the closest target outcome. In inverse model, the agent guesses which action has led to the current state.

The forward internal model is an intuitive interpretation of how an organic agent chooses its actions for its task. McNamee and Wolpert show that forward model consists of four stages [8]. First, in perception stage, an agent receives sensor input by monitoring the environment and its current action status. Sensory input will contain noise, because the agent cannot observe non-visible environment parameters, such as speed and spin. Also, there is noise in the agent’s sensory system. Second, in simulation stage, the agent predicts how the environment state will change in near future. Third, in motion planning stage, the agent produces all the possible outcomes by its given action options. Among all the state and action predictions, the agent chooses the action that can lead

to the outcome closest to its target outcome. Fourth, in optimal feedback control stage, the agent applies motor commands for its chosen action. The agent uses optimal feedback controller when applying actions in order to adjust the motor commands according to current sensory feedback and environment state.

### B. Forward Internal Model for Autonomous Driving

According to Plebe et al, current autonomous driving algorithm loop is strictly divided into perception-decision-action [11]. Deep neural network is often applied in perception stage as a single module, because it is well suited for its generalization in object detection and classification task. However, Plebe et al. suggest that if this rigid division between perception, decision, and action can be collapsed, deep neural network itself can be implemented as the complete perception-decision-action loop [11]. Inspired by neuroscience, the entire autonomous driving algorithm loop can be re-defined with three pathways. First pathway, dorsal stream, is the sensor data tensor flow through the entire deep neural network. It is based on the visual pathway of the primate brain [12]. Second pathway, cerebellum loop, also known internal simulation, represents the network’s capability to internally predict the consequences of various actions. Third pathway, action selection loop, is the network’s capability to select the optimal action decision based on its internal simulation results. In order to implement internal model for autonomous driving, the system needs deep neural networks for internal simulation and action decision selection.

## III. PROPOSED METHOD

Forward internal model requires an internal simulation mechanism that can predict near-future state based on current sensory state input and system action output. Therefore, in order to integrate forward internal model into vision-based LFA system, it requires a state estimator that can predict the lane image of near-future based on current lane image and



driving status data. This paper utilizes CNN-based U-Net as a state estimator for forward internal model-based LFA.

#### A. CNN-based U-Net for State Estimation of Internal Model

U-Net is a convolutional networks for biomedical image segmentation [7]. In U-Net, Ronneberger et al. implement skip connections between matching encoding layers and decoding layers [7]. These skip connections can prevent gradient vanishing by allowing gradient information to be maintained all the through the layers. In next lane image prediction, it is important that the network learns to use the features from current lane image and driving status data to produce next lane image. U-Net’s skip connections can achieve this by creating a thorough gradient information flow among the layers.

CNN-based U-Net is trained to produce a lane image of next timestamp from a current lane image and driving status data vector. Current lane image and driving status data are used as input to the network. Current lane image is processed into latent feature vectors by CNN-based encoders. Driving status data is appended in the encoder’s latent feature vector. In order to produce the output image with the same shape as input image, the appended latent feature vector is reshaped into 1x1000 shape by fully connected layer. The output of fully connected layer is then processed into an image through CNN-based decoder. During the decoding process, output vectors of matching encoders will be appended into the input of decoders in order to establish feature reusability and maintain gradient flow through skip connections. Along with vision-based steering control, this deep learning-based next lane image prediction can play as a state estimation pathway for forward internal model-based LFA.

#### B. Effect of Time Difference for Lane Prediction

Longer the time difference between current and next timestamps, there will be greater displacement between current and next lane images. In order to determine the capability of next lane image prediction, it is necessary to figure out how much displacement the neural network can be trained to handle. In this paper, we tested the network under four timestamp differences. We observed how the length of timestamp difference affects the neural network’s training for next lane image prediction.

### IV. EXPERIMENT

#### A. Dataset Collection and Preparation using CARLA

This paper uses CARLA, open-source autonomous driving simulator, in order to collect an extensive amount of lane image and driving status data [13]. We added a dataset recording function on top of the autonomous driving example provided by CARLA. This function records lane images from the vehicle’s front camera. It also collects driving status data, wheel steering, throttle, and speed, with the matching simulation timestamp. Dataset was collected from four maps, Default Town, Town01, Town 06, and Town 04, with different weather settings. Dataset was collected in the driving environment and weather conditions, where the lanes are clearly visible.

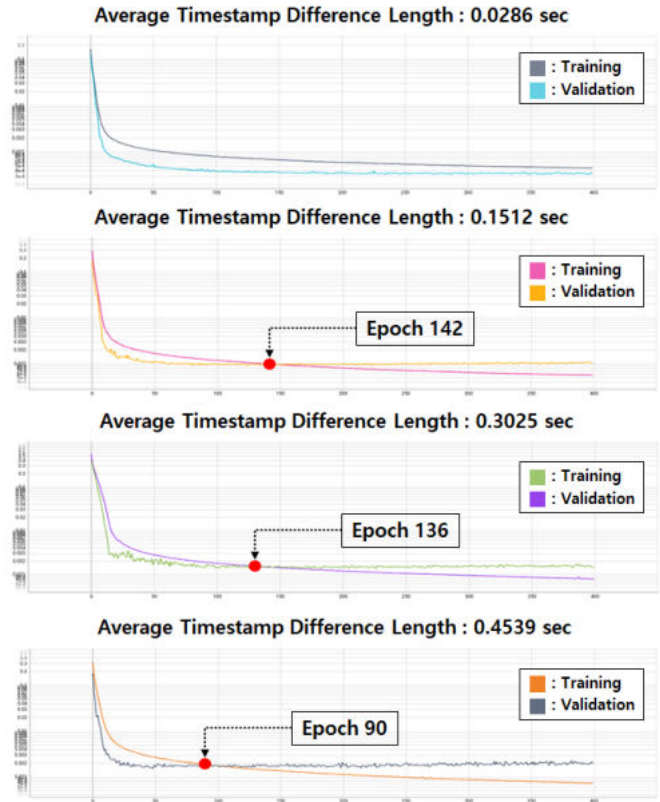


Fig. 3. Loss Graph Comparison under Different Timestamp Differences

For stable training, it is imperative that both input data and output data are normalized with same range. Original lane images are scaled between 0 and 1 by being multiplied by 1/255. Original wheel steering and throttle are already normalized between 0 and 1 from CARLA. Original speed data is scaled between 0 and 1 by being multiplied by 1/100. This assumes that the maximum driving speed is 100km/h. Normalizing input lane image data with the same range as driving status data can prevent unstable gradient backpropagation during the training process.

#### B. Training and Experiment Setup

CNN-based U-Net proposed in Fig 2 is implemented using PyTorch 1.9.0. It is trained for 400 epochs on Nvidia RTX 3090. It is trained by Adam optimizer with learning rate of 1e-5. Mean Squared Error (MSE) is used as the loss function between target next lane image and the network’s prediction output image.

#### C. Training Result Comparison

Fig 3 is the compilation of training and validation loss function graphs of CNN-based U-Net trained with four different timestamp difference lengths. It is log-scaled on y-axis in order to clearly show how a loss function graphs changes in different timestamp length. Fig 3 shows that the network trained to predict the image with longer timestamp difference suffers faster overfitting. This is because under longer timestamp

Average Timestamp Difference Length	[Training / Epoch 330]		[Validation / Epoch 330]	
	Network Prediction Output	Groundtruth Next Lane Image	Network Prediction Output	Groundtruth Next Lane Image
0.0286sec				
0.1512sec				
0.3025sec				
0.4539sec				

Fig. 4. Prediction Output Image Comparison

difference, there will be greater displacement between between current and next lane images. As a result, a larger portion of input lane image will be considered unrelated to next lane image groundtruth.

Fig 4 shows the prediction output image of CNN-based U-Net with different timestamp differences. In Fig 4, the network trained with longer timestamp difference produces more blurry prediction output image. Based on the analysis from Fig 3 that a larger part of current lane image input is considered unrelated to next lane image prediction in longer timestamp difference condition, the level of feature reuse will decrease. This results in more blurriness between current lane image input and next lane prediction output. However, even in longer timestamp difference, CNN-based U-Net's output image still contains lanes that can be used for vision-based steering. This result shows that CNN-based U-Net can be trained to estimate next lane image and implemented as a state estimator for forward internal model-based LFA.

## V. CONCLUSION

This paper presents CNN-based U-Net as a state estimator for forward internal model-based LFA system. It shows how timestamp difference length between current and next lane image affects training characteristics and output prediction quality. Although longer timestamp difference length results in overfitting by increasing the displacement between current and next lane image, the lanes in prediction output image produced by the network trained with longer timestamp difference are visible enough for vision-based LFA.

CNN-based U-Net can be later integrated into the forward internal model-based LFA system from Fig 1 as a state estimator in forward internal model. This implementation can provide

a deep learning-based LFA pipeline that can mimic human driver behaviors. Training the network with additional dataset with more diverse weather conditions, illumination changes, and traffic elements can further improve its performance as a state estimator.

## ACKNOWLEDGMENT

This research was supported by the MSIT(Ministry of Science, ICT), Korea, under the High-Potential Individuals Global Training Program(2020-0-01513) supervised by the IITP(Institute for Information and Communications Technology Planning & Evaluation) and in part by the National Research Foundation of Korea (NRF) grant funded by the Korean Government (MSIT) (No. 2019R1A2C109009612).

## REFERENCES

- [1] M. Kawato, "Internal models for motor control and trajectory planning," *Current Opinion in Neurobiology*, vol. 9, no. 6, pp. 718–727, Dec. 1999.
- [2] J. Stein, "Cerebellar forward models to control movement," *The Journal of Physiology*, vol. 587, no. Pt 2, p. 299, Jan. 2009. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2670044/>
- [3] Q. Welniarz, Y. Worbe, and C. Gallea, "The forward model: A unifying theory for the role of the cerebellum in motor control and sense of agency," *Frontiers in Systems Neuroscience*, vol. 15, p. 22, 2021. [Online]. Available: <https://www.frontiersin.org/article/10.3389/fnsys.2021.644059>
- [4] L. S. Popa and T. J. Ebner, "Cerebellum, Predictions and Errors," *Frontiers in Cellular Neuroscience*, vol. 12, p. 524, 2019. [Online]. Available: <https://www.frontiersin.org/article/10.3389/fncel.2018.00524>
- [5] O. J. M. Smith, "A controller to overcome dead time," in *ISA Journal*, vol. 6, 1959, pp. 28–33.
- [6] N. Abe and K. Yamanaka, "Smith predictor control and internal model control - a tutorial," in *SICE 2003 Annual Conference (IEEE Cat. No.03TH8734)*, vol. 2, 2003, pp. 1383–1387 Vol.2.
- [7] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, ser. LNCS, vol. 9351. Springer, 2015, pp. 234–241, (available on arXiv:1505.04597 [cs.CV]). [Online]. Available: <http://lmb.informatik.uni-freiburg.de/Publications/2015/RFB15a>
- [8] D. McNamee and D. M. Wolpert, "Internal models in biological control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, no. 1, pp. 339–364, 2019. [Online]. Available: <https://doi.org/10.1146/annurev-control-060117-105206>
- [9] S. J. Blakemore, S. J. Goodbody, and D. M. Wolpert, "Predicting the consequences of our own actions: The role of sensorimotor context estimation," *Journal of Neuroscience*, vol. 18, no. 18, pp. 7511–7518, 1998. [Online]. Available: <https://www.jneurosci.org/content/18/18/7511>
- [10] D. M. Wolpert, Z. Ghahramani, and M. I. Jordan, "An internal model for sensorimotor integration," *Science*, vol. 269, no. 5232, pp. 1880–1882, 1995. [Online]. Available: <https://www.science.org/doi/abs/10.1126/science.7569931>
- [11] A. Plebe, M. Da Lio, and D. Bortoluzzi, "On reliable neural network sensorimotor control in autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 2, pp. 711–722, 2020.
- [12] B. R. Sheth and R. Young, "Two Visual Pathways in Primates Based on Sampling of Space: Exploitation and Exploration of Visual Information," *Frontiers in Integrative Neuroscience*, vol. 10, p. 37, 2016. [Online]. Available: <https://www.frontiersin.org/article/10.3389/fnint.2016.00037>
- [13] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

# Forward and Backward Warping for Optical Flow-Based Frame Interpolation

Joi Shimizu<sup>†</sup>, Heming Sun<sup>‡\*</sup>, Jiro Katto<sup>†</sup>

<sup>†</sup>Dept. of Computer Science and Communications Engineering, Waseda University, Tokyo, Japan

<sup>‡</sup>Waseda Research Institute for Science and Engineering, Waseda University, Tokyo, Japan

<sup>\*</sup>JST, PRESTO, Kawaguchi, Saitama, Japan

joey.shimizu@toki.waseda.jp, hemingsun@aoni.waseda.jp, katto@waseda.jp

**Abstract**—Frame interpolation methods generate intermediate frames by taking consecutive frames as inputs. This enables the generation of high frame rate videos from low frame rate videos. Recently, many deep learning-based frame interpolation methods have been proposed. One way of frame interpolation is by using the bi-directional optical flow. In many cases, these methods use backward warping to warp the input images to the desired frame. However, forward warping can also be used to warp the input frames. In this paper, we propose a frame interpolation method that utilizes both forward warping and backward warping. Experimental results show that utilizing both warping methods can enhance the performance compared to only using backward warping.

**Keywords**—frame interpolation, deep learning, optical flow

## I. INTRODUCTION

Frame interpolation allows us to generate intermediate frames of consecutive frames. With this technology, high frame rate videos can be generated from lower frame rate videos. For example, slow motion videos can be obtained from ordinary videos without using high speed cameras. Also, frame interpolation is applied in some video compression models for inter prediction. Video compression models with deep learning [1, 2] can replace the block-based flow estimation, that is a process in video compression standards (such as AVC [3], HEVC [4], and VVC [5]), with frame interpolation.

One approach for frame interpolation is by using bi-directional optical flow and using them to warp the input frames to the desired frame. Super SloMo [6] is one of those methods. In Super SloMo, the bi-directional flow is estimated, and the flows are used for backward warping. The challenge in backward warping, however, is that optical flow used for backward warping is not stable, often resulting in inaccurate predictions. We believe that forward warped images can also provide meaningful information for interpolation, as they are able to use more accurate flows for warping. Therefore, we utilize both forward and backward warping in Super SloMo. Experimental results show the effectiveness of using both warping methods.

## II. RELATED WORK

### A. Frame Interpolation

Many deep learning-based frame interpolation have been proposed. There are several types of approaches. One approach is the flow-based approach [6, 7, 8]. Liu *et al.* [8] proposed a network that learns to synthesize video frames by flowing pixel values from existing ones, which they call deep voxel flow. [6, 7] calculate the bi-directional flow and

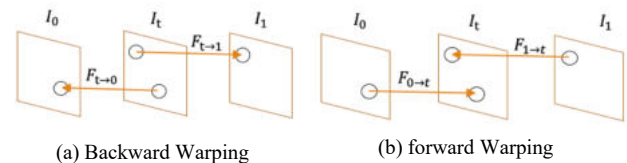


Fig. 1. Backward warping and forward warping

use backward warping. Backward warping is a popular warping method for flow-based approaches.

However, recently, forward warping-based approach is proposed [9]. In Fig. 1, we explain the differences between the two warping methods. In backward warping, each pixel of the warped frame is mapped from the reference frame, thus creates less occlusion. On the other hand, forward warping maps each pixel of reference frame to the warped frame. Different pixels in the reference frame may be mapped to the same pixel on the warped frame. Also, no pixels may be mapped to a pixel on the warped frame, creating occlusions. Examples of occlusions caused by forward warping can be seen in Fig. 3 (a). For these reasons, backward warping is a popular method for frame interpolation. However, [9] proposed a new way to handle the cases where multiple source pixels are mapped to the same target location.

While methods like [6, 7] require computationally expensive calculations to get the optical flows from the interpolated image to the input images, Huang *et al.* [10] proposed estimating  $F_{t-0}$  and  $F_{t-1}$  directly. Another approach is the kernel-based approach [11]. Combination of flow-based and kernel-based approaches using a network inspired by deformable convolution are also proposed [12]. Meyer *et al.* [13] regards video frames as linear combinations of wavelets and propose a phase-based approach.

### B. Optical Flow Estimation

Convolutional Neural Networks (CNNs) are used in many computer vision tasks, including optical flow estimation. Optical flow plays an important role for flow-based interpolation methods. FlowNet [14], an encoder-decoder-based model, was the first work that implemented optical flow estimation with end-to-end training. Later research further enhanced the precision by developing better end-to-end architectures, such as coarse-to-fine flow prediction model using a pyramid architecture [15, 16]. Long *et al.* [17] use a CNN to predict optical flow by synthesizing interpolated frames, and then inverting the

CNN. Another work, proposed a network which extracts per-pixel features of the input frames, create 4D correlation volumes from those features, and iteratively update the flow field [18].

### C. Video Compression

Recent trend in video compression research is using deep learning, partially or end-to-end. These methods aim to outperform the widely used video compression standards, such as AVC [3], HEVC [4], and VVC [5]. Frame interpolation models are often used in deep learning-based video compression models. [1] uses a Variational Autoencoder (VAE)-based model for intra prediction and replace the block-based optical flow estimation with frame interpolation. [2] combines the current video compression standards with state-of-the-arts frame interpolation methods, proving that similar performances compared with the compression standards can be achieved with deep learning-based interpolation models.

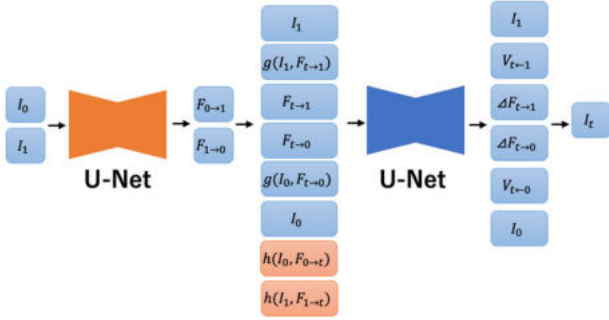


Fig. 2. Proposed approach

## III. PROPOSED APPROACH

### A. Utilization of Forward Warped Images

We incorporate forward warping into Super SloMo. Fig. 2 shows our proposed method. First, the two input frames,  $I_0$  and  $I_1$ , are fed into the first U-Net and the bi-directional optical flow,  $F_{0 \rightarrow 1}$  and  $F_{1 \rightarrow 0}$  is computed. Next, in order to perform backward warping,  $F_{t \rightarrow 0}$  and  $F_{t \rightarrow 1}$  are calculated with the equation below:

$$F_{t \rightarrow 0} = -(1-t)tF_{0 \rightarrow 1} + t^2F_{1 \rightarrow 0} \quad (1)$$

$$F_{t \rightarrow 1} = (1-t)^2F_{0 \rightarrow 1} - t(1-t)F_{1 \rightarrow 0} \quad (2)$$

By using  $I_0, I_1, F_{t \rightarrow 0}$  and  $F_{t \rightarrow 1}$ , backward warped images  $g(I_0, F_{t \rightarrow 0})$  and  $g(I_1, F_{t \rightarrow 1})$  are calculated. The original Super SloMo inputs these features to the second U-Net, but in our model, we also use the forward warped images  $h(I_0, F_{0 \rightarrow t})$  and  $h(I_1, F_{1 \rightarrow t})$  as inputs. For forward warping, we use Softmax Splatting, which was proposed in [9]. Optical flows  $F_{0 \rightarrow t}$  and  $F_{1 \rightarrow t}$ , which are needed for forward warping, are calculated with the equation below:

$$F_{0 \rightarrow t} = t * F_{0 \rightarrow 1} \quad (3)$$

$$F_{1 \rightarrow t} = (1-t) * F_{1 \rightarrow 0} \quad (4)$$

The outputs of the second U-Net are optical flow residuals  $\Delta F_{t \rightarrow 0}$  and  $\Delta F_{t \rightarrow 1}$ , and the Visibility Map  $V_{t \rightarrow 0}$  and  $V_{t \rightarrow 1}$ . The Visibility Map satisfy the following constraint:

$$V_{t \rightarrow 0} = 1 - V_{t \rightarrow 1} \quad (5)$$

Finally, with the warped frames and the visibility map, the final interpolated frame is calculated.

### B. Utilization of RAFT Optical Flows for Forward Warping

Our method explained in section A uses the optical flow from the first U-Net for forward warping. However, the flows  $F_{0 \rightarrow 1}$  and  $F_{1 \rightarrow 0}$  from the first U-Net are optimized for backward warping, meaning they are not as precise as other optical flow estimation models (since optical flow models are normally optimized for forward warping, rather than backward warping). Therefore, we utilize an off-the-shelf optical flow estimator, RAFT [18], for flow estimations needed for forward warping. This results in a much more precise forward warped image, but bigger occlusions can be spotted, which leads to worse results. To overcome this issue, we fill the occlusions with the already calculated backward warped images. The occlusions on  $h(I_0, F_{0 \rightarrow t})$  are filled with pixel values from  $g(I_1, F_{t \rightarrow 1})$ . Similarly, the occlusions on  $h(I_1, F_{1 \rightarrow t})$  are filled with pixel values from  $g(I_0, F_{t \rightarrow 0})$ .

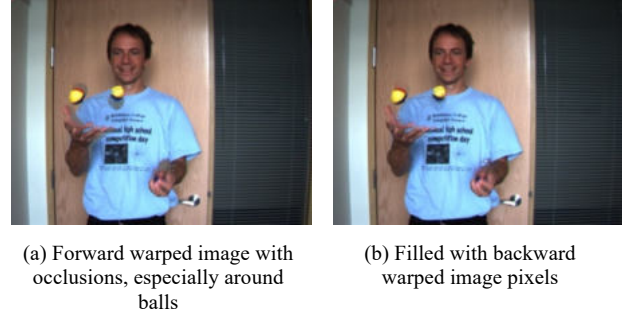


Fig. 3. Forward warped image with RAFT

## IV. EXPERIMENTS

### A. Dataset

For training, a combination of different 240 fps videos and datasets were used. First, just like the original Super SloMo reported in [6], the Adobe 240-fps dataset from [19] was used. Also, the GOPRO dataset [20] was used. In addition, we collect 190 sequences from YouTube. Finally, we collect our original 240 fps videos using iPhone. Figs. 4 and 5 show a snapshot of randomly selected video frames of the YouTube and iPhone videos. In the dataset, there are a great variety of scenes such as sports, animals, moving vehicles, etc. Table I shows the number of video clips, the number of frames, and the resolution of each video set.

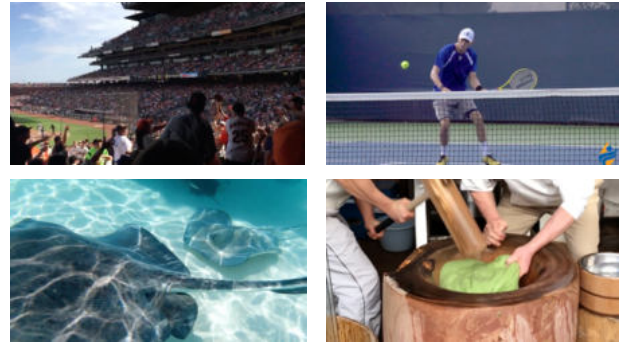


Fig. 4. Snapshot of YouTube videos



Fig. 5. Snapshot of iPhone videos

TABLE I. DATASET INFORMATION

	Adobe240	GOPRO	YouTube	iPhone
# video clips	133	33	190	83
# video frames	124,841	34,874	93,161	117,320
resolution	720p	720p	Various resolutions	1080p

### B. Training Settings

Training of the models is conducted by comparing the output  $\hat{I}_t$  and the actual intermediate frame  $I_t$ . The loss function is a linear combination of four terms:

$$l = \lambda_r l_r + \lambda_p l_p + \lambda_w l_w + \lambda_s l_s \quad (6)$$

*Reconstruction loss*  $l_r$  determines how well the reconstruction of the interpolated frame is. It is calculated with the following equation:

$$l_r = \frac{1}{N} \sum_{i=1}^N \| \hat{I}_t - I_t \|_1 \quad (7)$$

*Perceptual loss*  $l_p$  is also used to reduce blur and make interpolated frames sharper. It is calculated by using VGG16 model [21]  $\phi$ :

$$l_p = \frac{1}{N} \sum_{i=1}^N \| \phi(\hat{I}_t) - \phi(I_t) \|_2 \quad (8)$$

*Warping loss*  $l_w$  is calculated for optimization of optical flow predictions. The warping loss includes errors of warped frames using  $F_{0 \rightarrow 1}, F_{1 \rightarrow 0}, F_{t \rightarrow 0}$ , and  $F_{t \rightarrow 1}$ :

$$l_w = \| I_0 - g(I_1, F_{0 \rightarrow 1}) \|_1 + \| I_1 - g(I_0, F_{1 \rightarrow 0}) \|_1 + \frac{1}{N} \sum_{i=1}^N \| I_t - g(I_0, F_{t \rightarrow 0}) \|_1 + \frac{1}{N} \sum_{i=1}^N \| I_t - g(I_1, F_{t \rightarrow 1}) \|_1 \quad (9)$$

*Smoothness loss*  $l_s$  is also added to encourage neighboring pixels to have similar values.  $\nabla$  represents total variation regularization which was also used for training of DVF [8]. Smoothness loss is calculated with the following equation:

$$l_s = \| \nabla F_{0 \rightarrow 1} \|_1 + \| \nabla F_{1 \rightarrow 0} \|_1 \quad (10)$$

The weights are kept the same as [6].

$$\lambda_r = 0.8 \quad (11)$$

$$\lambda_p = 0.005 \quad (12)$$

$$\lambda_w = 0.4 \quad (13)$$

$$\lambda_s = 1 \quad (14)$$

The models are trained for 250 epochs. The learning rate is set to 0.0001 and decreases by a factor of 10 every 100 epochs.

All the videos are divided into groups of 12 consecutive frames. During training, 9 consecutive frames are randomly chosen out of the 12. The first frame and the ninth frame are used as inputs, and the target frame for interpolation is randomly chosen.

### C. Evaluation Results

Two datasets, Middlebury and DAVIS, are used for evaluation. The evaluation metric is Peak Signal-to-Noise Ratio (PSNR). For the DAVIS dataset, the 10th frame and the 12th frame were used as inputs to interpolate the 11th frame. Results are shown on Table II. ‘‘Ours w/o RAFT’’ indicates our model which uses the optical flow from the first U-Net for forward warping. ‘‘Ours w/ RAFT’’ indicates our model which uses the optical flow from RAFT for forward warping. The red numbers indicate the best performance and the blue numbers indicate the second best performance.

TABLE II. EVALUATION RESULTS WITH MIDDLEBURY AND DAVIS DATASETS

	Middlebury	DAVIS
Overlapping	27.97	-
Phase-Based [13]	31.12	-
MIND [17]	31.35	-
DVF [8]	34.34	-
Super SloMo	34.24	27.00
Ours w/o RAFT	<b>34.43</b>	<b>27.04</b>
Ours w/ RAFT	<b>34.51</b>	<b>27.13</b>

The results for Overlapping, Phase-Based, MIND and DVF are directly taken from [12]. Also, the results of Super SloMo for the Middlebury dataset are almost identical to the results reported in [12]. We can see that by using the optical flow calculated by U-Net for forward warp, the interpolation accuracy enhances for both datasets. This model (Ours w/o RAFT) performed 0.19dB better in Middlebury and 0.04dB better in DAVIS dataset. For the Middlebury dataset, the original Super SloMo performs worse than DVF, but with our new model, it exceeds DVF’s performance.

The accuracy becomes even better when better optical flow is used for forward warping. When compared with the original Super SloMo, our final model performed 0.27dB better in Middlebury and 0.13dB better in DAVIS dataset.

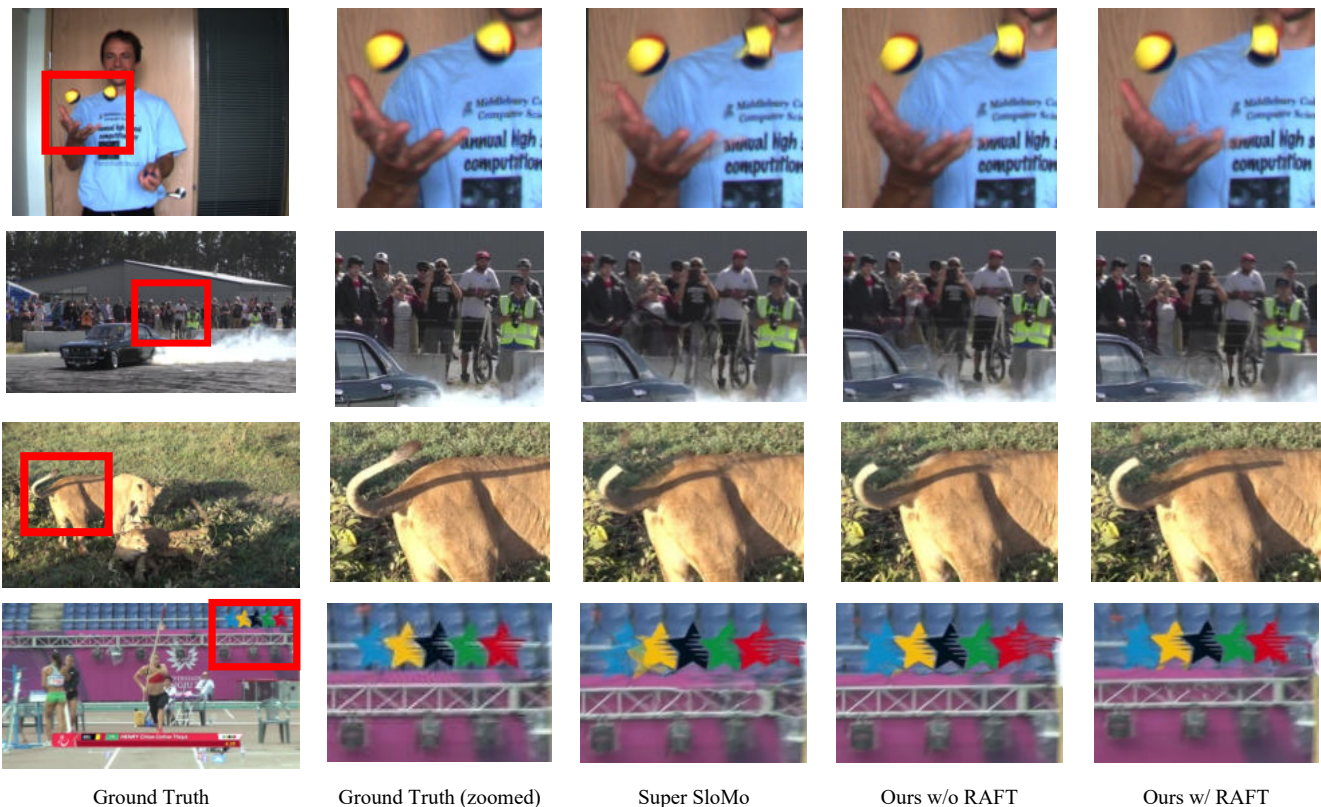


Fig. 6. Visual results for Super SloMo and our proposed approach. The sequences are Beanbags from the Middlebury dataset and burnout, lions, pole-vault from the DAVIS dataset.

Visual results of Super SloMo and our proposed approach are shown on Fig. 6. We can see from Fig. 6 that our proposed approach, especially the one using RAFT, outputs images that are visually closer to the ground truth image. Although, we still see distorted regions in images. One reason for this is the large motions. Large motions are still hard to predict in our model, resulting in blurry or distorted images. Another reason is the non-linear motions. For equations (1), (2), (3) and (4), linear calculations are used to obtain the desired optical flow. Although, in real life, movements are not linear (ex. the movements of the fingers in the Beanbags sequence).

## V. CONCLUSIONS

We have proposed a frame interpolation method that utilizes both forward warping and backward warping. We have learned that by adding forward warping to a backward warping-based model, Super SloMo, our method can enhance the performance. Also, we found that by using a better optical flow method for forward warping, even greater performance can be achieved. As future work, we would like to conduct similar experiments with other models that only uses one warping method. Also, we would like to use non-linear calculations to better understand the movements in the sequences.

## VI. ACKNOWLEDGEMENT

This work was supported in part by NICT, Grant Number 03801, Japan and JST, PRESTO Grant Number JPMJPR19M5, Japan. Also, we would like to thank Yuya Ishii, Masaaki Kitamoto, Tatsuhiko Furusawa, Alaric-Yohei Kawai and Ryoichi Sakamoto for helping us create the iPhone 240fps dataset.

## REFERENCES

- [1] Z. Cheng, H. Sun, M. Takeuchi, J. Katto, "Learning Image and Video Compression through Spatial-Temporal Energy Compaction", IEEE CVPR, pp. 10063-10072, 2019.
- [2] J. Shimizu, Z. Cheng, H. Sun, M. Takeuchi, J. Katto, "HEVC Video Coding with Deep Learning Based Frame Interpolation", IEEE GCCE, pp. 433-434, 2020.
- [3] T. Wiegand, G. J. Sullivan, G. Bjontegaard, A. Luthra, "Overview of the H.264/AVC Video Coding Standard", IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no.7, pp. 560-576, 2003.
- [4] G. J. Sullivan, J. Ohm, W. Han and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard", IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1649-1668, 2012.
- [5] G. J. Sullivan, J. R. Ohm, "Versatile Video Coding Towards the Next Generation of Video Compression", Picture Coding Symposium, 2018.
- [6] H. Jiang, D. Sun, V. Jampani, M. Yang, E. Miller, J. Kautz, "Super SloMo: High Quality Estimation of Multiple Intermediate Frames for Video Interpolation", IEEE CVPR, pp. 9000-9008, 2018.
- [7] W. Bao, W. Lai, C. Ma, X. Zhang, Z. Gao, M. Yang, "Depth Aware Video Frame Interpolation", IEEE CVPR, pp. 3698-3707, 2019.
- [8] Z. Liu, R. A. Yeh, X. Tang, Y. Liu, A. Agarwala, "Video Frame Synthesis using Deep Voxel Flow", IEEE ICCV, pp. 4473-4481, 2017.
- [9] S. Niklaus, F. Liu, "Softmax Splatting for Video Frame Interpolation", IEEE CVPR, pp. 5436-5445, 2020.
- [10] Z. Huang, T. Zhang, W. Heng, B. Shi, S. Zhou, "RIFE: Real-Time Intermediate Flow Estimation for Video Frame Interpolation", arXiv:2011.06294, 2020.
- [11] S. Niklaus, L. Mai, F. Liu, "Video Frame Interpolation via Adaptive Separable Convolution", IEEE ICCV, pp. 261-270, 2017.
- [12] H. Lee, T. Kim, T. Chung, D. Pak, Y. Ban, S. Lee, "AdaCoF: Adaptive Collaboration of Flows for Video Frame Interpolation", IEEE CVPR, pp. 5315-5324, 2020.

- [13] S. Meyer, O. Wang, H. Zimmer, M. Grosse, and A. S. Hornung, "Phase-Based Frame Interpolation for Video", IEEE CVPR, pp. 1410-1418, 2015.
- [14] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. Smagt, D. Cremers, T. Brox, "FlowNet: Learning Optical Flow with Convolutional Networks", IEEE ICCV, pp. 2758-2766, 2015.
- [15] A. Ranjan, M.J. Black, "Optical Flow Estimation Using a Spatial Pyramid Network", IEEE CVPR, pp. 2720-2729, 2017.
- [16] D. Sun, X. Yang, M. Liu, J. Kautz, "PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume", IEEE CVPR, pp.8934-8943, 2018.
- [17] Gucan Long, Laurent Kneip, Jose M Alvarez, Hongdong Li, Xiaohu Zhang, and Qifeng Yu. "Learning Image Matching by Simply Watching Video", Springer ECCV, 2016.
- [18] Z. Teed, J. Deng, "RAFT: Recurrent All-Pairs Field Transforms for Optical Flow", Springer ECCV, 2020.
- [19] S. Su, M. Delbracio, J. Wang, G. Sapiro, W. Heidrich, O. Wang. "Deep Video Deblurring for Hand-held Cameras", IEEE CVPR, pp. 237-246, 2017.
- [20] S. Nah, T. H. Kim, K. M. Lee, "Deep Multi-scale Convolutional Neural Network for Dynamic Scene Deblurring", IEEE CVPR, pp.257-265, 2017.
- [21] K. Simonyan, A. Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition", *CoRR*, abs/1409.1556, 2014.

# Performance Improvement Method of the Video Visual Relation Detection with Multi-modal Feature Fusion

Kwang-Ju Kim

*Electronics and Telecommunications Research Institute  
1 Techno Sunhwan-ro 10-gil, Yuga-eup, Dalseong-gun  
Daegu, Korea 42994  
kwangju@etri.re.kr*

Pyong-Kun Kim

*Electronics and Telecommunications Research Institute  
1 Techno Sunhwan-ro 10-gil, Yuga-eup, Dalseong-gun  
Daegu, Korea 42994  
iros@etri.re.kr*

Kil-Taek Lim

*Electronics and Telecommunications Research Institute  
1 Techno Sunhwan-ro 10-gil, Yuga-eup, Dalseong-gun  
Daegu, Korea 42994  
ktl@etri.re.kr*

Jong Taek Lee

*Electronics and Telecommunications Research Institute  
1 Techno Sunhwan-ro 10-gil, Yuga-eup, Dalseong-gun  
Daegu, Korea 42994  
jongtaeklee@etri.re.kr*

**Abstract**—Video visual relation detection is a novel research problem that aims to detect instances of visual relations of interest in a video. In this paper, we propose a performance improvement method of the video visual relation detection with multi-modal feature fusion. First, we introduce a spatial feature extraction method that is designed to include the relative positions of objects itself and between objects in the image. Next, we suggest a relationship classifier that is designed to accommodate the complexity of the input features. Our proposed method achieves 6.65 mAP, and ranked the 2nd place in the visual relation detection task of Video Relation Understanding Challenge (VRU), the ACM Multimedia 2020.

**Index Terms**—component, formatting, style, styling, insert

## I. INTRODUCTION

In recent years, deep learning technologies have achieved great success in computer vision tasks, such as object classification, detection, attribute detection, and segmentation [1]–[5]. These computer vision researches have improved the performance in various tasks of image understanding. However, high-level image understanding tasks such as image captioning, scene graph, visual question answering, image retrieval, and other related works remain open-challenging tasks [6]–[9]. As a mid-level learning task, visual relationship detection (VRD) can provide rich information for high-level image understanding tasks. A VRD is generally defined as a pair of objects localized by bounding-boxes together with a predicate to connect them. It aims to construct a holistic representation by identifying triplets in the form (subject, predicate, object). Comparing with VRD on static image, video visual relation detection (VidVRD) is much more practical and challenging than VRD. Firstly, dynamic interactions between objects can only be observed in videos. Secondly, there is a high variability of interactions between two specific objects in a video which causes another challenging problem. Therefore,

if VRD methods used in the static image are applied to video, it is difficult to achieve high performance.

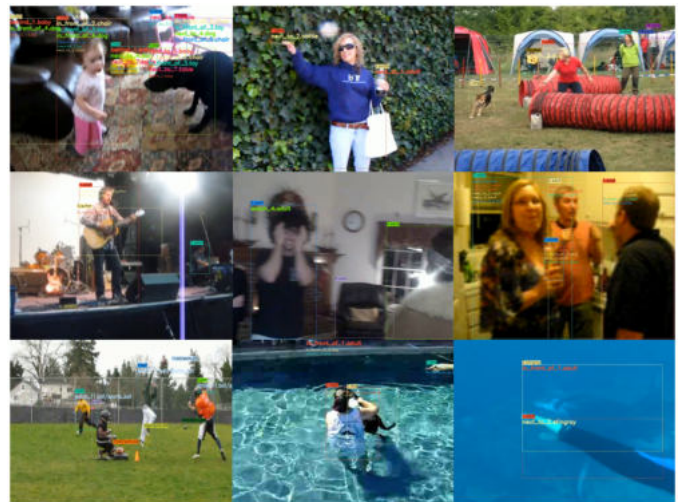


Fig. 1. Several examples of the VidOR dataset. Each object is spatio-temporally annotated, and the relation instances between each pair of objects are annotated in the videos.

To tackle these problems, several researches have been recently proposed. Their natural VidVRD’s method is to generate features of dynamic and time-varying relationships between entities. On the other side, new challenges such as The ACM Multimedia 2019 VRU Challenge with Video Object Relation (VidOR) dataset [10], is designed to encourage this research. Figure 1 shows several examples of VidOR dataset, which contain 80 categories of objects annotated with a bounding-box trajectory to indicate their spatio-temporal location in the videos and 50 categories of relation predicates annotated



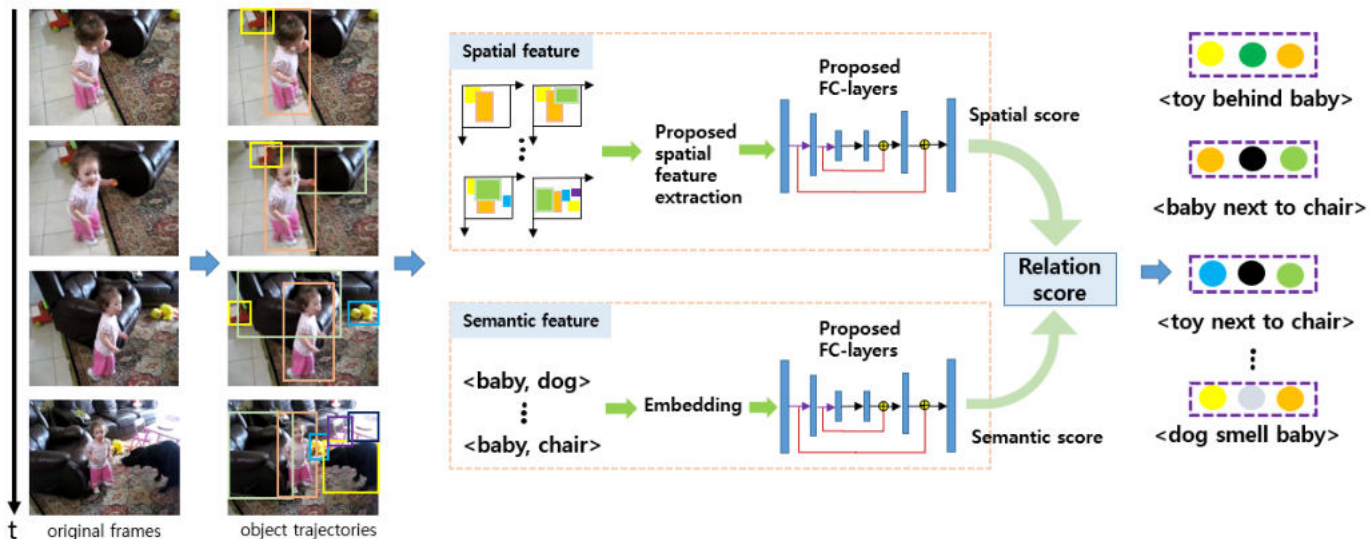


Fig. 2. The overview of our proposed model. The circles with different colors represent different predicates in the relation prediction results.

among all pairs of annotated objects with starting and ending frame index. The ACM Multimedia 2019 VRU Challenge winner proposed a multi-modal feature fusion method for VidVRD [11]. However, the winner’s method still leaves room for performance improvements by modifying multi-modal feature extraction and relations prediction classification. In this paper, we propose a performance improvement method of video visual relation detection via multi-modal feature fusion.

## II. RELATED WORK

In recent years, many previous works have been studied the problem of the visual relationship prediction. We present the related work of video visual relation detection (VidVRD) as well as video object detection (VID).

### A. Video Object Detection

VID is the task of detecting objects from a video as opposed to images. When image-based object detection methods applied to the video data, they can cause more miss detections because the appearance of objects becomes often blurred or even occluded in frames. After introducing the ImageNet video object detection challenge (ImageNet VID) [12], many object detection research efforts have been extended to video object detection. Many works utilized the idea of feature aggregation to enhance per-frame features by aggregating nearby frames’ features. Specifically, Flow-Guided Feature Aggregation (FGFA) [13] utilizes an optical flow network from FlowNet [14], [15] for estimating the pixel-level motions on feature maps of adjacent frames for feature aggregation. Another solution to video object detection is to explore mapping strategies to link the static image detection results of the same object identity into a bounding-box trajectory. Seq-NMS [16] proposes a post-processing heuristic method consisting of three steps: sequence selection, re-scoring, and suppression. Through this method, the overall score was improved by

correcting the score of weaker detection. Detect and Track (D&T) [17] generates a tracking formulation given two (or more) frames as input into R-FCN [18] to perform object detection and across-frame track regression. There is also proposed a method [19] to calibrate object feature at the box level to improve video object detection with an extended version of FGFA.

### B. Visual Relation Detection

VRD aims to identify groups of objects and their relationships in an images in the form of (subject, predicate, object). Specifically, this task is to detect all objects presented in the image and predict all possible visual relationships between two of the detected objects. In the past few years, several approaches have been proposed to recognize the relationship from the static images. These approaches have been also applied to VidVRD without substantial modification. However, comparing with VRD in the static image, VidVRD is not only practical but also challenging than VRD, as mentioned in the introduction section. Several well-designed models have been proposed to solve this problem. Shang et al. [20] proposed the first VidVRD framework to temporarily localize and recognize dynamic relationships. they also contributed the first VidVRD dataset which contains rich labeled relations. Tsai et al. [21] proposed a fully-connected spatial-temporal graph constructed for each video and graph convolutional network formulated feature interaction. They proposed constructing a graph similar to the above in a subsequent study but using conditional random fields to take advantage of the statistical dependencies between objects. Sun et al. [11] proposed a video relation model with multi-modal feature fusion and achieved state-of-the-art performance on VidOR dataset in ACM Multimedia 2019 VRU Challenge.

TABLE I  
PROPOSED SPATIAL FEATURE CALCULATION

Index	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$
Feature	$\frac{x_{min}+x_{max}}{2}$	$\frac{y_{min}+y_{max}}{2}$	$x_{max} - x_{min}$	$y_{max} - y_{min}$	$\frac{(x_{min}+x_{max})*img_w}{2}$	$\frac{(y_{min}+y_{max})*img_h}{2}$
Index	$f_7$	$f_8$	$f_9$	$f_{10}$	$f_{11}$	$f_{12}$
Feature	$(x_{max} - x_{min}) * img_w$	$(y_{max} - y_{min}) * img_h$	$\frac{x'_{min}+x'_{max}}{2}$	$\frac{y'_{min}+y'_{max}}{2}$	$x'_{max} - x'_{min}$	$y'_{max} - y'_{min}$
Index	$f_{13}$	$f_{14}$	$f_{15}$	$f_{16}$	$f_{17}$	$f_{18}$
Feature	$\frac{(x'_{min}+x'_{max})*img_w}{2}$	$\frac{(y'_{min}+y'_{max})*img_h}{2}$	$(x'_{max} - x'_{min}) * img_w$	$(y'_{max} - y'_{min}) * img_h$	$\log \frac{h}{h'}$	$\log \frac{h*w}{h'*w'}$

### III. THE PROPOSED APPROACH

In the following section, we describe our strategy which is to modify spatial feature extraction and insert skip-connection into FC-layers to improve accuracy for the relation prediction. At first, we describe the proposed spatial feature extraction method in section 3.1. Then, we present the insertion of the proposed skip-connection embedded FC-Layers in section 3.2.

#### A. Relation Instance Generation

The proposed method is based on a framework which is described in Sun et al. [11] and Shang et al. [20]. It consists of three steps: decomposing one video into segments, predicate recognition on segments and merging relationship predictions in neighboring segments through a greedy association algorithm. We also used pre-computed bounding box trajectories that provided by VRU challenge organizers. We proposed the spatial-temporal feature extraction method that extracts relative location feature and motion feature. We defined the object relative location feature as  $f_{RI} = [f_1, f_2, \dots, f_{18}]$ . It is calculated as shown in table 1, where  $(x,y,w,h)$  and  $(x',y',w',h')$  are the bounding box coordinates of subject and object, respectively.  $(img_w, img_h)$  is the height and width of the input image. Motion features are defined as follows:

$$f_{Mot} = f_{RI}^e - f_{RI}^s \quad (1)$$

This feature extracts various locations over time between the subject and the object, where  $f_{RI}^e$  and  $f_{RI}^s$  are our proposed spatial features extracted from the end and start frames of the candidated segment, respectively. Finally, spatial-temporal features ( $f_{ST}$ ) are generated by concatenating the features computed above  $f_{RI}^e$ ,  $f_{RI}^s$ , and  $f_{Mot}$ . We use a pre-trained word2vec model [22], [23] to extract feature  $f_{Lan}$  for encoding subject/object categories. It was trained on GoogleNews dataset.

#### B. Relationship Classification Model

After we generated  $f_{ST}$  and  $f_{Lan}$ , the features are fed into our two independent classification models which are trained separately. Our model is designed to increase the complexity of the Multi-Layer-Perceptron(MLP) because it is difficult to accommodate the complexity of the input features with a simple MLP model. To this end, we adapted the number of nodes in the MLP and introduced a skip-connection method.

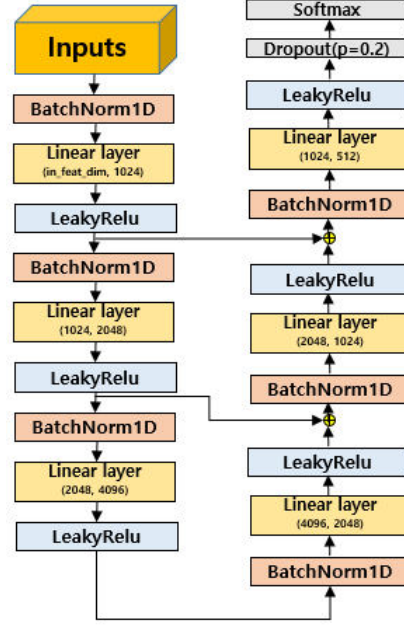


Fig. 3. Proposed relation prediction classifier.

### IV. EXPERIMENTS

#### A. Dataset and Training Details

The VidOR dataset consists of 7,000 videos for train, 835 videos for validation and 2,165 videos for test. 80 categories of objects are annotated with bounding-box trajectory to indicate their spatio-temporal location in the videos; and 50 categories of relation predicates are annotated among all pairs of annotated objects with starting and ending frame index. Our proposed model was trained with Stochastic Gradient Descent (SGD) optimizer, where batch size is 32, momentum is 0.9, weight decay is 0.1, and on NVIDIA GeForce GTX TITAN XP GPU with 12GB memory. The learning rate is set to 0.01, and reduces from 0.01 to 0.0001 for each 10 epochs. The experiments were done with cuDNN v7.5 and CUDA 10.1. for the test, we linearly combine the two prediction confidences of classifiers as follows:

$$P(c_p | f_{ST}, f_{Lan}) = \lambda P(c_p | f_{ST}) + (1-\lambda) P(c_p | f_{Lan}) \quad (2)$$

where  $c_p$  denotes predicate category,  $\lambda$  is set to 0.3.

#### B. Evaluation Metrics

VRU Challenge adopts Average Precision (AP) to evaluate the detection performance per video and finally calculate

TABLE II  
COMPARISON BETWEEN OUR PROPOSED METHOD AND THE TOP-PERFORMING OF THE VRU'19 CHALLENGE ON VIDOR VALIDATION-SET

Method	Tagging precision@1	Tagging precision@5	Tagging precision@10	Recall@50	Recall@100	mAP
Re-produced top-1 solution in VRU'19 challenge	50.48	39.91	32.44	6.69	8.71	6.02
Pre-computed feature + ours model	52.16	<b>40.35</b>	<b>33.03</b>	6.97	9.08	6.50
Ours feature + model w.o skip-connection	<b>52.52</b>	40.18	32.95	6.99	9.12	6.56
<b>Ours</b>	51.68	40.04	33.01	<b>7.01</b>	<b>9.14</b>	<b>6.60</b>

TABLE III  
FINAL RESULTS ON VIDOR TESTSET

Method	Tagging precision@1	Tagging precision@5	Recall@50	Recall@100	mAP
Ours	52.69	42.19	7.16	9.36	6.65

the mean AP (mAP) over all testing videos as the ranking score. To match a predicted relation instance ( $\langle s, p, o \rangle^p, (\tau_s^p, \tau_o^p)$ ) to a ground truth ( $\langle s, p, o \rangle^g, (\tau_s^g, \tau_o^g)$ ), the requirements should be satisfied as follows: (1) their relation triplets are exactly same, i.e.  $\langle s, p, o \rangle^p = \langle s, p, o \rangle^g$ . (2)  $\mathbf{vIoU}(\tau_s^p, \tau_s^g) \geq 0.5$  and  $\mathbf{vIoU}(\tau_o^p, \tau_o^g) \geq 0.5$ , where  $\mathbf{vIoU}$  refers to the volume intersection over union [24]. (3) the minimum overlap of the subject trajectory pair and the object trajectory pair  $\mathbf{ov}_{\text{pg}} = \min(\mathbf{vIoU}(\tau_s^p, \tau_s^g), \mathbf{vIoU}(\tau_o^p, \tau_o^g))$  is the maximum among those paired with the other unmatched ground truths  $\mathcal{G}$ , i.e.  $\mathbf{ov}_{\text{pg}} \geq \mathbf{ov}_{\text{pg}'} (\mathbf{g}' \in \mathcal{G})$ .

### C. Results Analysis

Table 3 shows the final results on VidOR test-set. Our proposed method achieves mAP of 6.65%, which is 0.34% higher than the method of the winner in the VRU'19 challenge. We also compared the performances using the VidOR validation set as shown in Table 2 before submitting our final results. The performance is improved when our proposed relationship classification model is connected to the pre-computed features of the VRU'19 challenge winner. Also, the performance of the skip-connection method is slightly enhanced compared to the case of no skip-connection. When both the proposed spatial feature and relationship classification model were applied, there was a better performance improvement than the last year's winning model in most evaluation metrics.

## V. CONCLUSION

In this paper, we have proposed a spatial feature extraction and relationship classifier for video visual relation detection in the VidOR dataset. Specifically, the proposed spatial feature extraction method is designed to include the relative position of objects in the image and the relative position between objects. In addition, the relationship classifier is designed to accommodate the complexity of the input features. The experiment results indicate that the proposed model outperforms the last year's winning model in the visual relation detection task of VRU challenge. Our team (*ETRI\_DGRC*) ranked in the 2nd place of the visual relation detection task in the VRU'20 Challenge.

## ACKNOWLEDGMENT

This work was supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government [22ZD1120, Development of ICT Convergence Technology for Daegu-Gyeongbuk Regional Industry]

## REFERENCES

- [1] X. Wu, D. Sahoo, and S. C. Hoi, "Recent advances in deep learning for object detection," *Neurocomputing*, 2020.
- [2] G. Ciarrone, F. L. Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, and F. Herrera, "Deep learning in video multi-object tracking: A survey," *Neurocomputing*, vol. 381, pp. 61–88, 2020.
- [3] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, and R. Qu, "A survey of deep learning-based object detection," *IEEE Access*, vol. 7, pp. 128 837–128 868, 2019.
- [4] W. Wang, H. Song, S. Zhao, J. Shen, S. Zhao, S. C. Hoi, and H. Ling, "Learning unsupervised video object segmentation through visual attention," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 3064–3074.
- [5] Z.-Q. Zhao, S.-T. Xu, D. Liu, W.-D. Tian, and Z.-D. Jiang, "A review of image set classification," *Neurocomputing*, vol. 335, pp. 251–260, 2019.
- [6] Z.-J. Zha, D. Liu, H. Zhang, Y. Zhang, and F. Wu, "Context-aware visual policy network for fine-grained image captioning," *IEEE transactions on pattern analysis and machine intelligence*, 2019.
- [7] H. Xu, B. Li, V. Ramanishka, L. Sigal, and K. Saenko, "Joint event detection and description in continuous video streams," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2019, pp. 396–405.
- [8] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, "Video summarization with long short-term memory," in *European conference on computer vision*. Springer, 2016, pp. 766–782.
- [9] N. Passalis and A. Tefas, "Learning neural bag-of-features for large-scale image retrieval," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 10, pp. 2641–2652, 2017.
- [10] X. Shang, D. Di, J. Xiao, Y. Cao, X. Yang, and T.-S. Chua, "Annotating objects and relations in user-generated videos," in *Proceedings of the 2019 International Conference on Multimedia Retrieval*, 2019, pp. 279–287.
- [11] X. Sun, T. Ren, Y. Zi, and G. Wu, "Video visual relation detection via multi-modal feature fusion," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 2657–2661.
- [12] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.
- [13] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei, "Flow-guided feature aggregation for video object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 408–417.

- [14] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "Deepflow: Large displacement optical flow with deep matching," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 1385–1392.
- [15] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2758–2766.
- [16] W. Han, P. Khorrani, T. L. Paine, P. Ramachandran, M. Babaeizadeh, H. Shi, J. Li, S. Yan, and T. S. Huang, "Seq-nms for video object detection," *arXiv preprint arXiv:1602.08465*, 2016.
- [17] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Detect to track and track to detect," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3038–3046.
- [18] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in neural information processing systems*, 2016, pp. 379–387.
- [19] P. Tang, C. Wang, X. Wang, W. Liu, W. Zeng, and J. Wang, "Object detection in videos by high quality object linking," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 5, pp. 1272–1278, 2019.
- [20] X. Shang, T. Ren, J. Guo, H. Zhang, and T.-S. Chua, "Video visual relation detection," in *ACM International Conference on Multimedia*, Mountain View, CA USA, October 2017.
- [21] Y.-H. H. Tsai, S. Divvala, L.-P. Morency, R. Salakhutdinov, and A. Farhadi, "Video relationship reasoning using gated spatio-temporal energy graph," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10424–10433.
- [22] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.
- [23] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei, "Visual relationship detection with language priors," in *European conference on computer vision*. Springer, 2016, pp. 852–869.
- [24] X. Shang, T. Ren, H. Zhang, G. Wu, and T.-S. Chua, "Object trajectory proposal," in *2017 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2017, pp. 331–336.

# A high-speed driver behavior detection deep learning system using the amount of change in contrast between frames

Min Woo Yoo, Jihun Kim and Dae Woong Cha and Woo Sung Son and Donggyu Lee and Dong Seog Han\*  
School of Electronic and Electrical Engineering  
Kyungpook National University  
Daegu, Republic of Korea  
dshan@knu.ac.kr\*

**Abstract**—This paper proposes a deep learning system that detects driver behavior for safe driving. A method of detecting the dangerous behavior of an existing driver uses a method of deep learning object detection that detects a class and a location of an object in an image. However, the deep learning object detection algorithm uses many computational resources, so it cannot be used in vehicle embedded environments with limited computational resources. In the case of an object classification algorithm that classifies a single object in an image, fewer computational resources are used than that of a deep learning object detection algorithm. However, it cannot be applied because various objects in the camera image cannot be classified as a single object. In the paper, We propose an algorithm that infers the driver's behavioral area using the driver's static movement in a vehicle and then applies deep learning objects to the inferred area. The proposed algorithm may be applied to a vehicle embedded environment because the calculation time is faster and more accurate than the deep learning object detection algorithm.

**Keywords**—deep learning, object detection, classification

## I. INTRODUCTION

The driver monitoring system monitors the driver's condition and alerts the driver to prevent accidents. Recently, deep learning technology has been widely applied to driver monitoring systems. When detecting the driver's face [1] and mobile phone [2] use and smoking [3], a deep learning object detection algorithm is used. Representative deep learning object detection algorithms include SSD [4], YOLO [5], and RCNN [6]. SSD and YOLO are very fast to detect, but RCNN is very slow. A deep learning object classification algorithm is used when classifying the driver's emotions [7] and the driver's gaze [8]. Representative lightweight object classification algorithms include MobileNet [9] and SqueezeNet [10]. The object detection model generally uses more operator resources than the object classification model. In a limited vehicle embedded environment, a deep learning object detection algorithm that uses many computational resources causes system instability. Therefore, it is inappropriate to detect mobile phones and cigarettes using a deep learning object detection model.

The object classification algorithm's existing driver behavior detection algorithm consists of two steps. The first step detects a face area, and the second step is to enter

the area of the ear or mouth into the object classification model [11]. However, this method has a problem in that behavior cannot be detected when there are no objects around the mouth and ears. In addition, additional computational resources are consumed because the driver's face must be detected to detect behavior.

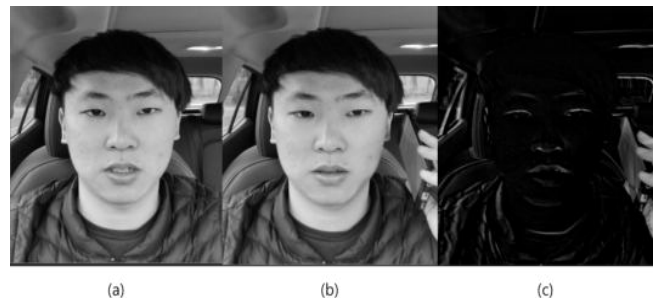


Figure 1. Changes in contrast when a driver uses a mobile phone: (a) Previous frame, (b) current frame, (c) contrast change image

The vehicle interior environment is very different from the general environment. The general environment is very dynamic. On the other hand, the environment observed by the vehicle interior camera is very static. Figure 1 shows the previous and current frames when driving after setting the camera frame to 5 FPS. Figure 1(a) is the previous frame, and figure 1(b) is the current frame using a mobile phone. Figure 1 (c) can be obtained by subtracting Figure 1 (a) and Figure 2(b). Through Figure 1, we confirmed a significant change in the contrast of the behavioral area when the driver acted in a static vehicle environment. On the other hand, there is no movement in the non-behavior area, so the change in contrast is negligible.

In this paper, we propose an algorithm to detect drivers' dangerous behaviors by using the change in contrast in the behavioral area. The proposed algorithm detects the final driver behavior through two steps. The first step is to detect the driver's behavior area. The second step is to classify the final driver behavior using a deep learning classification algorithm

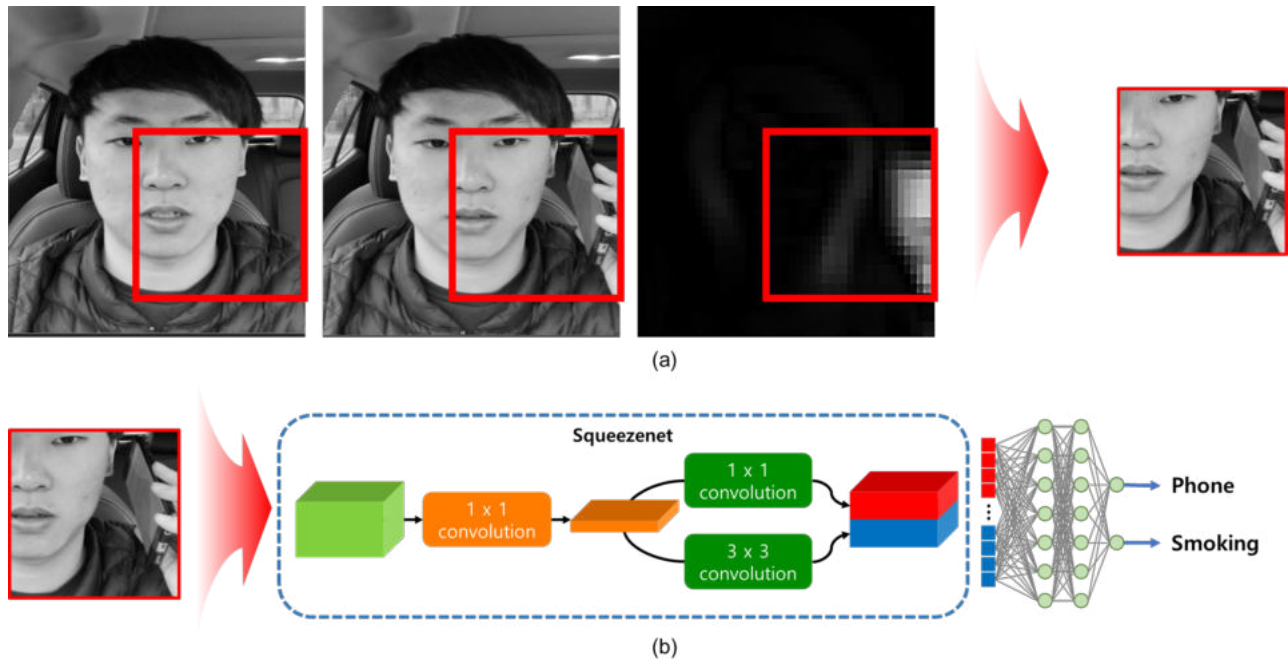


Figure 2. The structure of the driver behavior detection algorithm: (a) inferring the behavior area, (b) classifying the behavior area

in the behavioral area. Through this process, the driver's behavior can be detected using only deep learning object classification without using a deep learning object detection algorithm.

The rest of the paper consists of the following. The proposed driver behavior detection algorithm is described in Section II. The experimental results and analysis are described in Section III, and the conclusions are summarized in Section IV.

## II. BEHAVIOR DETECTION ALGORITHM

Figure 2 is the structure of the proposed algorithm. The proposed algorithm consists of two steps behavioral area inference and classification. The first step is to compare the contrast between the previous and current frames, as shown in Figure 2(a), to find the active area with the most considerable contrast. The second step classifies the driver behavior by inputting the active region of Figure 2(a) into the deep learning classification algorithm, as shown in Figure 2(b).

### A. infer the behavior area

The process of inferring the behavioral area consists of four steps. In the first step, the previous frame and the current frame are downsampled to a size of  $48 \times 36$ . The second step is to create an active image, a contrast change image, as shown in Figure 3. Figures 3(a) and 3(b) are the results of applying a  $5 \times 5$  smoothing filter to the frame. The active image in Figure 3(c) is generated by the difference in contrast values between Figure 3(a) and Figure 3(b), to which smoothing is applied. The third step is to downsample the active image to a size of  $8 \times 6$ , as shown in Figure 4. The brightest area in Figure 4 is the area where the driver acted. If the average of the total contrast values of the active image exceeds 60, the light

change is excessive. The typical environment where there are many light changes is the tunnel's entrance and exit. In such an environment, it is impossible to infer the behavior area. Therefore, only when the average of the total contrast values of the active image is less than 50 does it move on to the area inference step. The final step is to store each sliding value by sliding window size of  $3 \times 3$  in the downsampled active image. An area with the most considerable value in the sliding window becomes an active area. When the sum of pixels in the extracted active area exceeds 300, it is input to the following behavioral area classification process.

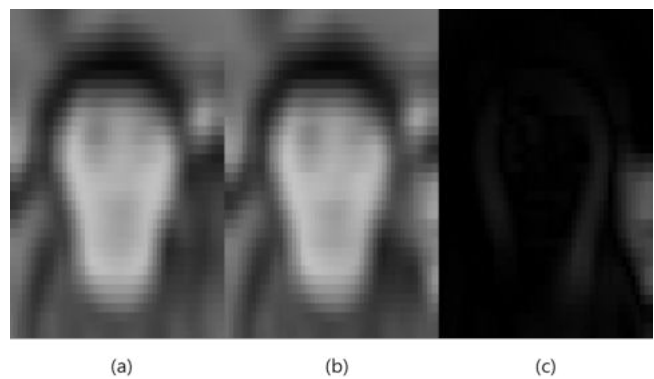


Figure 3. Generating an active image through a difference between a previous frame and a current frame: (a) previous frame, (b) current frame, (c) active image

In the process of inference of behavioral areas, downsampling uses area interpolation. Two benefits can be obtained by downsampling by area interpolation. First, as the image becomes smaller, the following operation's calculation time

can be reduced. Second, to find the behavioral area, the change in some areas is more important than the change in a specific pixel.

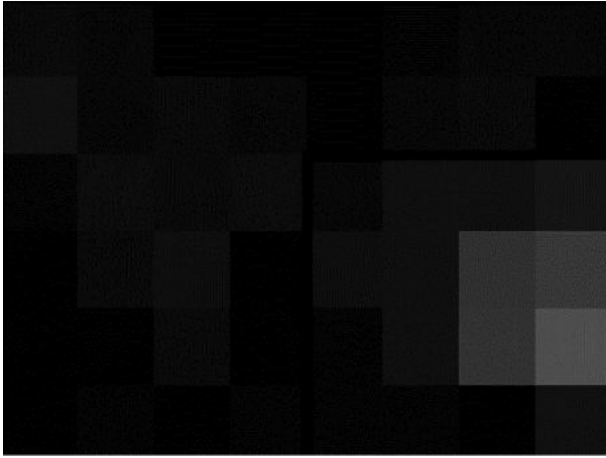


Figure 4. The result of downsampling the active image to  $8 \times 6$

### B. Classification of behavioral areas

To classify behavior, this paper used SqueezeNet and a deep learning classification algorithm [10]. Existing deep learning classification models focus on increasing accuracy. However, SqueezeNet is a small classification network with a small number of parameters. Models with few parameters are suitable for hardware with limited computational resources, such as vehicle embedded environments. Learning data were collected to learn the behavior classification model. Learning data consisting of smoking and cigarettes were collected 1,000 sheets each. 800 data were used for learning, and the remaining 200 were used for algorithmic performance tests. Data amplification used rotation, flip, and movement. Rotational amplification was applied at  $\pm 20^\circ$  considering the rotation of the driver's face. Flip amplification was applied only to the left and right. Movement applied 20% of the image size. Figure 5 is the result of classifying the behavioral area through the classification algorithm. <https://ko.overleaf.com/project/61a713fa1d603bc1333327dc>

### III. EXPERIMENT RESULTS AND ANALYSIS

To analyze the performance of the behavior position detector, 100 pairs of general state frames were collected. We also collected 100 pairs of data on driver behavior. We confirmed that the behavioral location detector has an accuracy of 90%. For the performance analysis of the behavior classifier, 200 behavior data were collected. We confirmed that the behavior classifier has an accuracy of 95%. Finally, We confirmed that our final overall system has an accuracy of 85%.

To compare the operation speed, the SSD detector and the proposed algorithm were compared [4]. Computer equipment and environment are CPU i7-8086, RAM 16GB, Windows, Keras. The operation speed of SSD is 8FPS, and the proposed algorithm is 25FPS.



Figure 5. The result of detecting a driver using a mobile phone

The proposed algorithm was more accurate in the environment where there was no change in the interior contrast caused by lighting and sunlight. However, the accuracy was low in an environment where there was much change in vehicle interior contrast due to lighting and sunlight. The cause of the performance degradation is the behavioral position detector. The behavior region detector uses the contrast between the previous frame and the current frame. // If there is a difference in contrast due to lighting and sunlight, not the difference in contrast due to behavior, the proposed algorithm incorrectly detects the location.

### IV. CONCLUSION

In this paper, we propose a behavior detection algorithm that reduces the computation time by 70% compared to the deep learning object detection model by using the static movement of the driver. The proposed algorithm consists of two steps. The first step is finding the behavior position in the previous and current frames. The second step is to input the behavioral location into the classification network to detect the final driver's behavior. The accuracy of the proposed behavior detector is 85%.

### ACKNOWLEDGMENT

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2021-2020-0-01808) supervised by the IITP (Institute of Information Communications Technology Planning Evaluation); and the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2021R1A6A1A03043144).

### REFERENCES

- [1] X. Sun, P. Wu, and S. C. Hoi, "Face detection using deep learning: An improved faster rcnn approach," *Neurocomputing*, vol. 299, pp. 42–50, 2018.

- [2] T. Hoang Ngan Le, Y. Zheng, C. Zhu, K. Luu, and M. Savvides, "Multiple scale faster-rcnn approach to driver's cell-phone usage and hands on steering wheel detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2016, pp. 46–53.
- [3] T.-C. Chien, C.-C. Lin, and C.-P. Fan, "Deep learning based driver smoking behavior detection for driving safety," *Journal of Image and Graphics*, vol. 8, no. 1, 2020.
- [4] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.
- [5] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.
- [6] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.
- [7] S. A. Hussain and A. S. A. Al Balushi, "A real time face emotion classification and recognition using deep learning model," in *Journal of Physics: Conference Series*, vol. 1432, no. 1. IOP Publishing, 2020, p. 012087.
- [8] R. A. Naqvi, M. Arsalan, G. Batchuluun, H. S. Yoon, and K. R. Park, "Deep learning-based gaze detection system for automobile drivers using a nir camera sensor," *Sensors*, vol. 18, no. 2, p. 456, 2018.
- [9] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.
- [10] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and < 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.
- [11] K. Seshadri, F. Juefei-Xu, D. K. Pal, M. Savvides, and C. P. Thor, "Driver cell phone usage detection on strategic highway research program (shrp2) face view videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 35–43.



# Intelligent Receiver for Optical Camera Communication

Ida Bagus Krishna Yoga Utama, Md. Habibur Rahman, ByungDeok Chung\*, and Yeong Min Jang  
Department of Electronics Engineering, Kookmin University, Seoul, South Korea

\*ENS. Co. Ltd, Ansan 15655, Korea

Email: idabaguskrishnayogautama@gmail.com; rahman.habibur@ieee.org; \*bdchung@ens-km.co.kr; yjang@kookmin.ac.kr

**Abstract**—Optical camera communication (OCC) is considered as a potential option for wireless communication due to the recent surge adoption of LED technology. Massive adoption of LED technology creates a problem in OCC due to the interference caused by other LEDs that do not transmit data. Hence, in this research work, we propose an intelligent receiver for the OCC system which can differentiate LED, whether the LED is transmitting data or not. We differentiate the LED by using an AI-based object detection algorithm and we compare the performance of four object detection algorithms, such as Faster RCNN, MobileNet SSD, YOLO V4, and YOLO V4 Tiny, to detect the data transmitting LEDs or non-data-transmitting LEDs. Experimental results show that the four algorithms can achieve above 0.95 mAP when detecting and classifying the LEDs. Thus, the object detection algorithm can be implemented in the OCC system and can be used to improve the OCC performance and robustness.

**Index Terms**—optical camera communication, object detection, intelligent receiver.

## I. INTRODUCTION

The wireless communication system has a lot of advantages compared to the wired communication system. The wireless communication system is easier to deploy due to the absence of wire. By using wireless communication, the data communication is conducted wirelessly as long as the receiver is still in the coverage area of the transmitter. Wireless communication nowadays is utilized in various application fields such as mobile networks, Internet of Things (IoT), and satellite communication systems. Radiofrequency (RF) technology is one of the popular wireless communication technology. Currently, the RF technology is at the center of attraction although the exponential growth of users causes the traffic to become congested [1] and the interference due to too many RF devices in a region is forcing them to search for complementary spectrum [2][3]. However, due to the nature of RF technology which uses radio to transmit the data, increasing the data rate using RF technology can be done by increasing the frequency band. However, that technique of increasing the frequency band has a negative effect on human health [4].

Optical wireless communication (OWC) is a potential alternative to existing RF technology communication systems [5]. Basically, OWC is divided into three techniques: light fidelity (Li-Fi), visible light communication (VLC), and optical camera communication (OCC) [6]. All three techniques have similarities which using LEDs as the transmitter because of

many benefits of LED such as high energy efficiency and low power consumption. Meanwhile, for the receiver, the VLC and Li-Fi are using photodiodes to receive the signal transmitted from the LED transmitter that propagates through the optical channel [7]. For OCC, it uses an image sensor as the receiver which the image sensor can be divided into two types, global shutter camera and rolling shutter camera. Compared with other OWC techniques, OCC presents superiority such as low cost, high reliability, and high resistance against interference [1].

In OCC, the camera received the optical signal with non-interference communication because the camera's image sensor has the capability to do spatial separation [8]. The image received by the camera is processed frame-wise which limits the OCC system data rate. To solve that issue, we can use a high-speed camera to increase the number of frames processed at a time [9]. However, using a high-speed camera will need a high-performance computer because it also needs a fast frame processing time. Another disadvantage is the price of the high-speed camera is expensive. Hence, MIMO techniques are proposed to increase the data rate of OCC systems [10]. Recently, several researchers also propose hybrid modulation techniques based on intensity, spatial, color, frequency, and phase to increase the data rate in the OCC system [11].

Due to the increasing number of LEDs adopted in daily life, those LEDs cause interference with the OCC systems. The OCC systems cannot focus only on the LED that actually transmitting the data because the other LED that is located near with the transmitting LED also blinking but doesn't transmit any data. In this paper, we designed a system that enables the OCC system to locate the location of LED that transmits modulated data. We use an object detection algorithm to detect the location of transmitting LED and non-transmitting LED. Then, we also compare the performance of four object detection algorithm to detect the LEDs.

## II. METHODOLOGY

### A. Experiment Scenario

For conducting the experiment, we build an OCC system that consists of a transmitter and receiver. For the transmitter part, we use one LED to transmit the modulated data. The modulation technique used is FSOOK modulation that modulates the data at a frequency between 2-4kHz. Then, we also use another LED that only turned on without transmitting

any data. Those LEDs are separated 8cm horizontally and both are controlled by an Arduino microcontroller board. The receiver, which is a camera with a frame rate of 60 fps, is located in front of the LEDs and establishes a line of sight (LOS) connection. For this experiment, we variate the distance between the LED transmitter and receiver with a distance of 10cm, 20cm, and 30cm. Then, we use four types of object detection algorithms: Faster RCNN [12], MobileNet SSD [13][14], YOLO V4 [15], and YOLO V4 Tiny. From those four object detection algorithms, we will compare their performance in detecting and classifying the LEDs from a distance of 10cm, 20cm, and 30cm. Also, we will calculate the data rate of the OCC system from various distances of 10cm, 20cm, and 30cm. Figure 1 shows the OCC architecture used in this experiment.

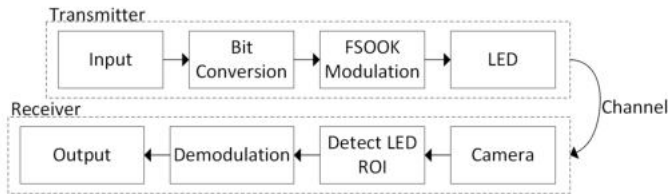


Fig. 1. OCC Architecture.

### B. Dataset

To develop the AI object detection, we need an image dataset to train the AI algorithm for detecting the LEDs. We create the dataset by doing the scenario as described in the previous section. The difference is we don't detect the ROI and demodulate the data, we only record the LEDs from a distance of 10cm, 20cm, and 30cm. At each distance, we record a video for 60 seconds long. Because our camera operates at 60fps, hence, for 60 seconds, we will get 3600 individual frames. According to the scenario, we will get three videos and total images of 10,800 images. From those images, that contain the transmitting LED and non-transmitting LED, we manually label each image by applying a bounding box to each LED location and the class of that object. The labeling is done by using labelImg software.

From 10,800 images in the dataset, we divide it into three parts, first, 70% of the data will be used for the training process. Then, the other 20% is used for the testing process and the last 10% will be used for validation. After dividing the data, we also apply preprocessing to the data such as resizing the images, rotating, and mirroring the images randomly.

### III. DEVELOPMENT OF THE OBJECT DETECTION AI MODELS

After creating the database, we have data that can be used for the AI model to learn. In this paper, we decide to use four types of AI models: Faster RCNN, MobileNet SSD, YOLO V4, and YOLO V4 Tiny. We choose those AI models because those models are notorious for their detection performance in detection accuracy and detection speed.

To develop the AI models, we use two different environments. For YOLO V4 and YOLO V4 Tiny, we use darknet to train the models using our dataset. For the training process in the darknet, we use hyperparameters such as a learning rate of 0.001, a momentum of 0.9, and a decay rate of 0.0005. Both AI models is using input images with a size of 416x416 pixels, and each image has 3 channels. Then, we train the models for 2000 epochs while calculating the training mAP every 100 epoch. After the training process is finished, we test our model using the validation dataset and calculate the mAP. Figure 2 shows the training loss for YOLO V4 Tiny and YOLO V4. The loss is already very small which indicates that the models have already learned the image dataset. Also, the mAP, which shown in Figure 3, shows that the model is able to learn well about the target.

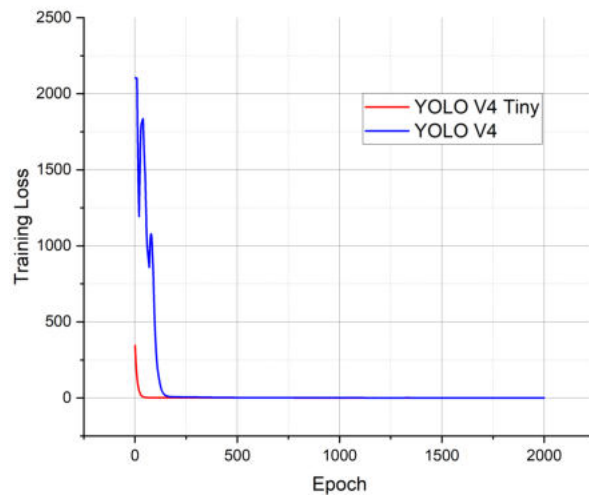


Fig. 2. Training Loss for YOLO V4 and YOLO V4 Tiny

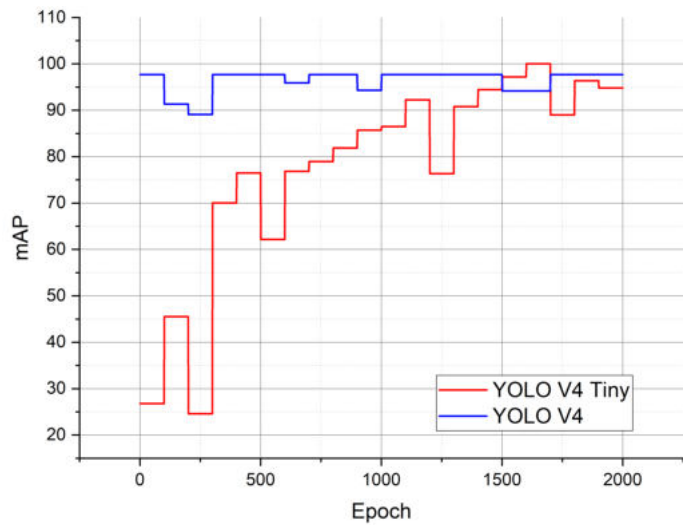


Fig. 3. mAP during training for YOLO V4 and YOLO V4 Tiny

Then, for the Faster RCNN and MobileNet SSD, we use

Tensorflow development environment. The training method is similar to the darknet. Here, we use hyperparameters such as a learning rate of 0.01 and momentum of 0.9. For this part, we use an image size of 640x640 pixels as input. Then, we train the models for 10,000 epochs. After the training is finished, we calculate the mAP of the model by testing it using the validation dataset. The training result is shown in Figure 4 and Figure 5 which shows the training loss for 10,000 epochs and mAP during training process of Faster RCNN and MobileNet SSD.

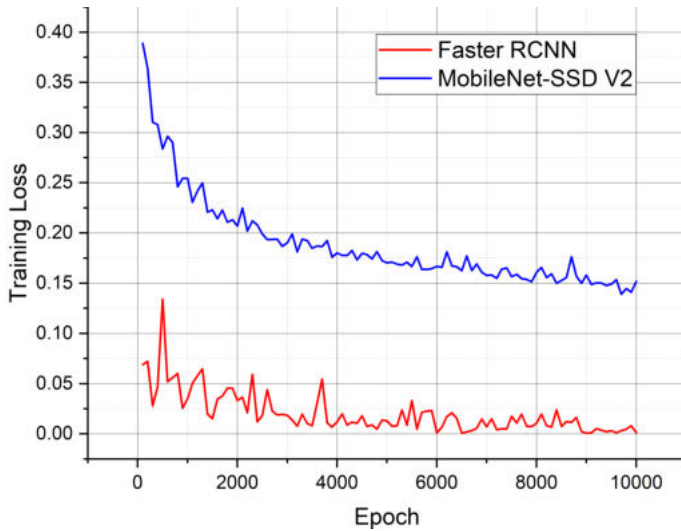


Fig. 4. Training Loss for Faster RCNN and MobileNet SSD

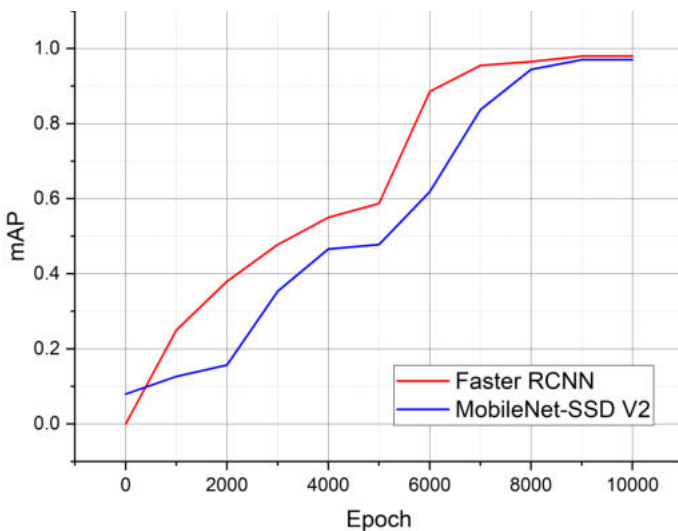


Fig. 5. mAP during training for Faster RCNN and MobileNet SSD

After the training process is finished, we will use the weight of the model for the testing process. We conduct the testing process by implementing it in the OCC system so that we can measure the data rate and mAP of each model. Different from the training process, we conduct the testing process using embedded computer NVIDIA Jetson Xavier NX. The

embedded computer will be installed by the AI models and OCC receiver software to detect the transmitting LED and non-transmitting LED.

#### IV. EXPERIMENT RESULT

The experiment is done by implementing an AI object detection model in the receiver part to intelligently differentiate the transmitting LED and non-transmitting LED. The camera continuously captures the image of the LED at 60fps and the receiver system will process that frame. The AI model is utilized to detect the location of the actual transmitting LED and LED that is only turned on. Then, the AI model will output the predicted location of the LEDs and give a bounding box and label to each detected LED. Figure 6 shows the predicted location of transmitting LED and non-transmitting LED. After that, the receiver system will only focus on transmitting LED ROI to demodulate the data and restore the data.

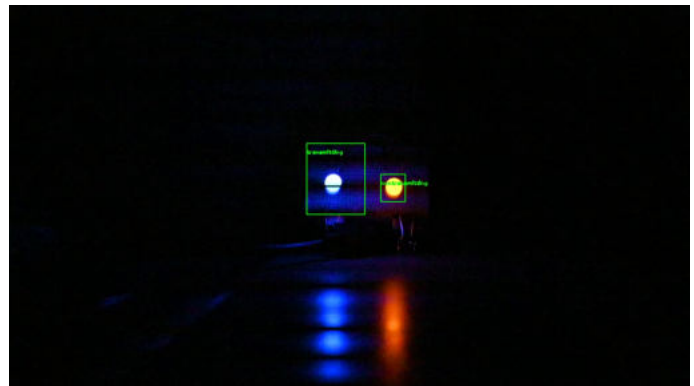


Fig. 6. AI model result predicted location of transmitting LED and non-transmitting LED.

From the experiment, we also obtain the mAP of the four developed AI models. The mAP value is shown in Figure 7. From that figure, we can see that all four AI models achieve high mAP values which are above 0.95. This indicates that all AI models can classify the transmitting LED and non-transmitting LED successfully. Also, the location prediction of each AI model is good due to the high mAP value. This means that all four AI models can be utilized in the OCC receiver system to solve issues of interference from other LEDs. However, this is depending on the hardware of the receiver. In this experiment, we use NVIDIA Jetson Xavier NX as the receiver computer. When running four AI models, we observe that each algorithm produces a different frame processing time. This is due to the complexity of each AI model. When using YOLO V4, we only get an average of 2.9 fps, then for the YOLO V4 Tiny, we get an average of 12.7 fps. After that, for MobileNet SSD, we get an average of 3 fps and 0.2 fps for Faster RCNN. Hence, although all AI models produce high mAP, when used in the receiver system, the frame processing time for each AI model is different which affects the OCC performance. If the frame processing time is very slow, it cannot capture all information transmitted by the

LED. Therefore, the fast frame processing time is a necessity to capture all transmitted information.

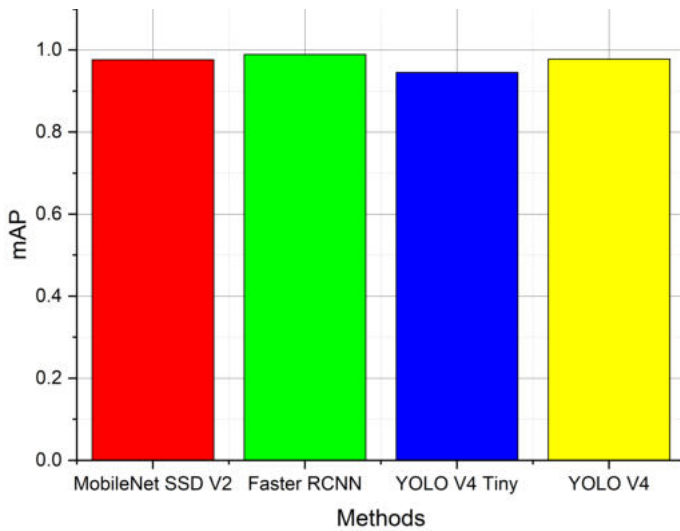


Fig. 7. mAP value of each AI models in testing process.

Besides that, in the experiment, we also calculate the data rate of the OCC systems from a distance of 10cm, 20cm, and 30cm. The result is displayed in Figure 8. We can see that the data rate is decreasing while the distance between receiver and transmitter is longer. This is caused by the emitted power of the LED is weakening when the distance is longer which causes the camera only to see fewer stripes in the long distance. Meanwhile, in a short distance, the LED power is still strong, and the camera can capture many stripes.

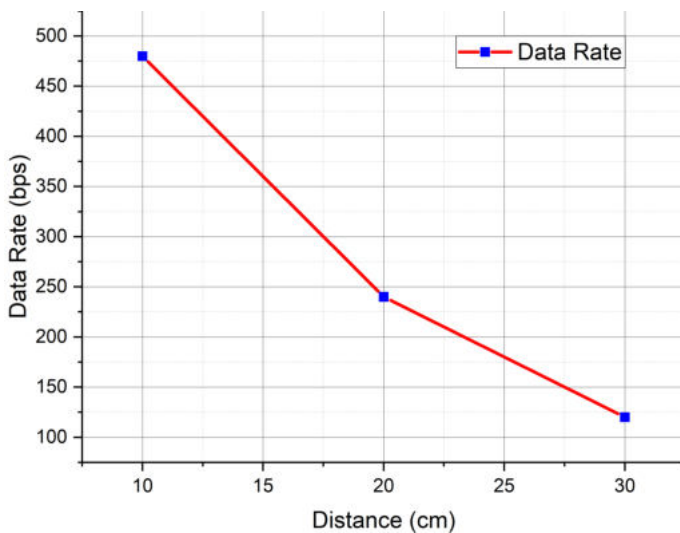


Fig. 8. Data rate from various distance.

## V. CONCLUSION

In this paper, we have implemented an OCC system that uses an AI object detection algorithm to differentiate transmitting LED and non-transmitting LED. We successfully developed the AI algorithm to predict LED location and classify

the LEDs. From the experiment result, we can see that four AI models can achieve excellent performance with high mAP. However, for implementation of OCC in embedded computers, high mAP is not enough, the AI model also should have a low computational cost to produce a fast frame processing time. Hence, the YOLO V4 Tiny is the best AI model compared to the other because it can produce a high mAP while having a low computational cost that enables the receiver system to achieve 12.9 fps. Then, the distance between transmitter and receiver is also necessary to the OCC performance. As such, the issue of interference from other LEDs in OCC can be solved by using an AI object detection algorithm to detect and classify the LED. It enables the OCC system to know the actual transmitting LED location and only focus on that location which will increase the system robustness and stability.

## ACKNOWLEDGMENT

This work was supported by the Technology development Program (S3098815) funded by the Ministry of SMEs and Startups(MSS, Korea).

## REFERENCES

- [1] Nguyen, T., Chowdhury, M. Z. and Jang, Y. M. (2013) "A novel link switching scheme using pre-scanning and RSS prediction in visible light communication networks," EURASIP journal on wireless communications and networking, 2013(1). doi: 10.1186/1687-1499-2013-293.
- [2] Chowdhury, M. Z. et al. (2019) "Convergence of heterogeneous wireless networks for 5G-and-beyond communications: Applications, architecture, and resource management," Wireless communications and mobile computing, 2019, pp. 1–2. doi: 10.1155/2019/2578784.
- [3] Ali, M. O. et al. (2021) "Current challenges in optical vehicular modulation techniques," in 2021 International Conference on Information and Communication Technology Convergence (ICTC). IEEE.
- [4] Kim, J. H. et al. (2019) "Possible effects of radiofrequency electromagnetic field exposure on central nerve system," Biomolecules & therapeutics, 27(3), pp. 265–275. doi: 10.4062/biomolther.2018.152.
- [5] Hasan, M. K. et al. (2018) "Performance analysis and improvement of optical camera communication," Applied sciences (Basel, Switzerland), 8(12), p. 2527. doi: 10.3390/app8122527.
- [6] Chowdhury, M. Z. et al. (2018) "A comparative survey of optical wireless technologies: Architectures and applications," IEEE access: practical innovations, open solutions, 6, pp. 9819–9840. doi: 10.1109/access.2018.2792419.
- [7] Nguyen, H. and Jang, Y. M. (2021) "Design of MIMO C-OOK using Matched filter for Optical Camera Communication System," in 2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC). IEEE.
- [8] Shahjalal, M. et al. (2019) "Smartphone camera-based optical wireless communication system: Requirements and implementation challenges," Electronics, 8(8), p. 913. doi: 10.3390/electronics8080913.
- [9] Younus, O. I. et al. (2020) "Data rate enhancement in optical camera communications using an artificial neural network equaliser," IEEE access: practical innovations, open solutions, 8, pp. 42656–42665. doi: 10.1109/access.2020.2976537.
- [10] Nguyen, T., Thieu, M. D. and Jang, Y. M. (2019) "2D-OFDM for optical camera communication: Principle and implementation," IEEE access: practical innovations, open solutions, 7, pp. 29405–29424. doi: 10.1109/access.2019.2899739.
- [11] Alfarozi, S. A. I. et al. (2019) "Robust and unified VLC decoding system for square wave quadrature amplitude modulation using deep learning approach," IEEE access: practical innovations, open solutions, 7, pp. 163262–163276. doi: 10.1109/access.2019.2952465.
- [12] Ren, S. et al. (2017) "Faster R-CNN: Towards real-time object detection with region proposal networks," IEEE transactions on pattern analysis and machine intelligence, 39(6), pp. 1137–1149. doi: 10.1109/TPAMI.2016.2577031.

- [13] Sandler, M. et al. (2018) “MobileNetV2: Inverted residuals and linear bottlenecks,” in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE.
- [14] Liu, W. et al. (2016) “SSD: Single Shot MultiBox Detector,” in Computer Vision – ECCV 2016. Cham: Springer International Publishing, pp. 21–37.
- [15] Bochkovskiy, A., Wang, C.-Y. and Liao, H.-Y. M. (2020) “YOLOv4: Optimal speed and accuracy of object detection,” arXiv [cs.CV]. Available at: <http://arxiv.org/abs/2004.10934>.

# Interference analysis study for coexistence between C-V2X and Wi-Fi 6E in the 6GHz band

1<sup>st</sup> Han Sol Kim  
*Department of Information and  
 Telecommunication Engineering*  
 Soongsil University  
 Seoul, Republic of Korea  
 terry5969@gmail.com

2<sup>nd</sup> Young Woon Kim  
*Department of Information and  
 Telecommunication Engineering*  
 Soongsil University  
 Seoul, Republic of Korea  
 k10193057@gmail.com

3<sup>rd</sup> Won Seok Yoo  
*Department of Information and  
 Telecommunication Engineering*  
 Soongsil University  
 Seoul, Republic of Korea  
 dnjstjr8995@gmail.com

4<sup>th</sup> Won-Cheol Lee  
*School of Electronic Engineering*  
 Soongsil University  
 Seoul, Republic of Korea  
 wlee@ssu.ac.kr

**Abstract**—Interference between C-V2X(Cellular Vehicle to Everything) usage and Wi-Fi 6E can occur in the 5.9GHz band designated for ITS(Intelligent Transportation System) purposes, so interference analysis between the vehicle and the AP(Access Point) is performed to protect C-V2X from interference to derive the distance of each RSRP(Reference Signal Received Power).

**Keywords**—C-V2X, Wi-Fi 6E, Interference Analysis, AFC

## I. INTRODUCTION

Since the existing AFC(Automated Frequency Control) is an operating system for frequency coexistence between Wi-Fi 6E in the 6GHz band and existing inbound users, it is not considering protection for C-V2X using the 5.9GHz band, so it is necessary to come up with an alternative to the protection of C-V2X. In order for the use of Wi-Fi 6E in the 6GHz sub-band(5,925MHz to 6,425MHz) to coexist without causing interference with the use of C-V2X in the 5.9GHz band, it is necessary to calculate the interference probability and protection distance through interference analysis

between Wi-Fi 6E and C-V2X[1][2][3][4].

## II. INTERFERENCE ANALYSIS

When a new radio station requests a radio station permit within the radio environment where existing radio stations exist, to obtain approval, the operation of a new radio station is permitted if interference is below the standard for mutual coexistence through radio interference analysis between existing and new radio stations. If such interference can be predicted, the transmission power or frequency band of the radio station will be adjusted to reduce interference and enable more efficient frequency use. Methods commonly used as interference analysis methods between wireless systems can be largely divided into interference analysis

using the MCL(Minimum Coupling Loss) method and MC(Monte Carlo) method[5].

### A. dRSS(desired Received Signal Strength)

dRSS, which is a  $W_t$ (Wanted transmitter) signal received from  $V_r$ (Victim receiver), may be expressed as Equation (1).

$$dRSS = P_{W_t} + G_{W_t \rightarrow V_r} + G_{V_r \rightarrow W_t} - PL \quad (1)$$

$P_{W_t}$  is the maximum transmission power of the transmitter,  $G_{W_t \rightarrow V_r}$  is the transmitter antenna gain,  $G_{V_r \rightarrow W_t}$  is the receiver antenna gain, and PL is the path loss from the transmitter to the receiver

### B. iRSS(interference Received Signal Strength)

The iRSS, which is an  $I_t$ (Interfering transmitter) signal received from  $V_r$ , may be expressed as Equation (2).

$$iRSS = P_{I_t} + G_{I_t \rightarrow V_r} + G_{V_r \rightarrow I_t} - PL \quad (2)$$

$P$  is the maximum interference transmission power,

$G_{I_t \rightarrow V_r}$  is the interference antenna gain,  $G_{V_r \rightarrow I_t}$  is the receiver antenna gain, and PL is the path loss from the interference to the receiver.

### C. Interference Probability

In this paper, interference was determined using the C/(N+I) (Carrier to Noise plus interference) technique. Interference analysis of the C/(N+I) technique includes noise in the existing C/I (Carrier to Interference) technique, and the interference analysis procedure is the same as the C/I technique that determines interference by calculating the ratio of desired signal and interference signal based on the C/I value provided by the equipment manufacturer. Whether or not to interfere is determined through Equation (3) and (4) below.

$$C/(N + I) > [C/(N + I)]_{threshold} \quad (3)$$

$$C/(N + I) < [C/(N + I)]_{threshold} \quad (4)$$

Interference does not exist in the case of Equation (3), and interference occurs in the case of Equation (4). The interference probability refers to the probability that the

Fig. 1. 6GHz frequency allocation band and adjacent band status

throughput required by the system is not satisfied when the  $iRSS$  received by the  $V_r$  is relatively larger than the  $dRSS$ . In this paper, assuming that  $dRSS$  is always received above sensitivity, Compare the interference power  $iRSS$  due interference of Unwanted Emission and Blocking. After that, the probability of satisfying Equation (4) is calculated. Equation (5) shows the process of calculating the probability of interference.

$$P = P \left\{ \frac{dRSS}{iRSS} < \frac{C}{I+N} \mid dRSS > Sensitivity \right\} \quad (5)$$

### III. SIMULATION

#### A. Interference Analysis Scenario

Fig. 2. Interference Scenario between Wi-Fi 6E and C-V2X

Victim link is a RSU(Road Side Unit) and OBU(On-Board Unit), Interfering link is a Wi-Fi 6E AP and Client. Wi-Fi 6E AP and OBU exist in urban areas, and interference was analyzed according to the separation distance between Wi-Fi 6E AP and OBU present in nearby bands in Downlink(RSU to OBU) situations. Indoor use of the next generation Wi-Fi 6E is less than 250mW, which can be used regardless of interference, so indoor interference analysis was not performed. When the Wi-Fi 6E AP was 30dB, the maximum output , a protection separation distance satisfying the interference probability of 5% or less was calculated. The WINNER II radio wave loss model, which is mostcommonly used in urban areas of urban and suburban, wheretransceivers are distributed up to 2km, was used. The Wi-Fi 6E AP parameters set as an interference in this paper are shown in Table 1 below[6][7][8].

TABLE I. Wi-Fi6E PARAMETERS

Parameters	Unit
<i>Center Frequency</i>	<b>6,005MHz</b>
<i>Transmitter Power</i>	<b>30dbm</b>
<i>Bandwidth</i>	<b>160MHz</b>
<i>Tx Antenna Type</i>	<b>Omni directional</b>
<i>Tx Antenna Gain</i>	<b>2dBi</b>
<i>Tx Antenna Height</i>	<b>1.5m</b>

The emission mask of the Wi-Fi 6E AP is shown in Figure 3.

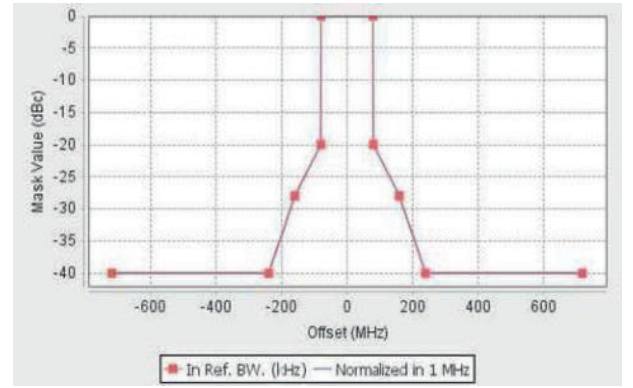


Fig. 3. Wi-Fi 6E Emission Mask

The parameters of C-V2X set as the transmitter and receiver of the Victim link are shown in Table 2[9][10][11].

TABLE II. C-V2X PARAMETERS

Parameters	Unit
<i>Center Frequency</i>	<b>5,910MHz</b>
<i>Transmitter Power</i>	<b>23dbm</b>
<i>Bandwidth</i>	<b>10MHz</b>
<i>Rx Antenna Type</i>	<b>Omni directional</b>
<i>Rx Antenna Gain</i>	<b>1dBi</b>
<i>Rx Antenna Height</i>	<b>1m</b>
<i>Noise floor</i>	<b>103dBm</b>
<i>Sensitivity</i>	<b>-90.4dBm</b>
<i>Max distance</i>	<b>107m</b>
<i>C/(I+N)</i>	<b>-1Db</b>

#### B. Interference Analysis Simulation Result

C-V2X divided each road by 20m×20m of grid and set it to be distributed within the grid, and Wi-Fi 6E AP was set to exist between min distance around the set grid and max distance. In the above situation, 20,000 events wererandomly generated using the Monte-Carlo technique to calculate a protection distance that satisfies within 5% of theinterference probability. The corresponding figure is shown in Figure 4.

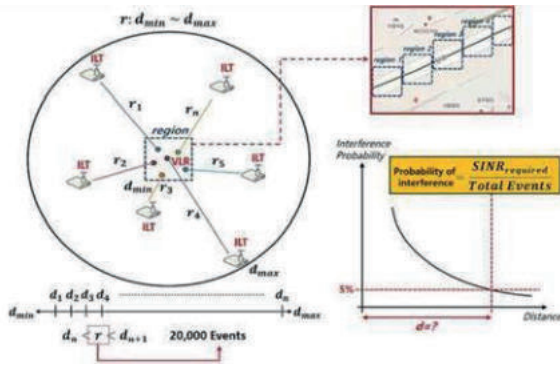
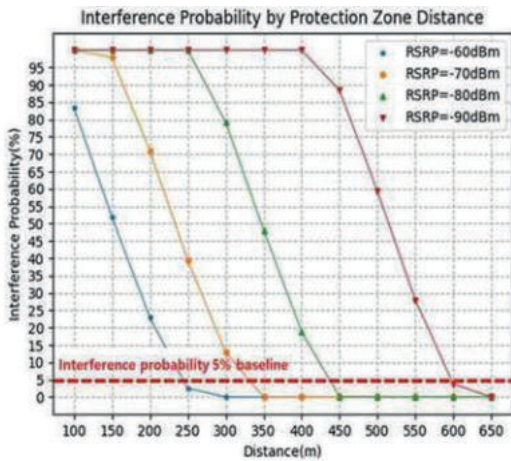


Fig. 4. Interference analysis scenatino between C-V2X and Wi-Fi 6E

When the value of SINR required as a result of interference analysis was -1dB, The protection separation distance according to the RSRP value of the C-V2X and OBU was calculated. The results showed that a minimum protective separation distance of 255m when RSRP is -60dBm, 310m when RSRP is -70dBm, 380m when RSRP is -80dBm, and 605m when RSRP is -90dBm, respectively. The result is shown in the graph of Figure 5 and 6.



Protection distance	100	150	200	250	300	350	400	450	500	550	600	650
RSRP->-60dBm	83.4	52	22.9	2.5	0	0	0	0	0	0	0	0
RSRP->-70dBm	100	97.9	71	39.5	12.8	0	0	0	0	0	0	0
RSRP->-80dBm	100	100	100	100	79.2	48.1	18.8	0.1	0	0	0	0
RSRP->-90dBm	100	100	100	100	100	100	100	88.4	59.2	27.8	3.7	0

Fig. 5. Interference analysis results between C-V2X and Wi-Fi 6E

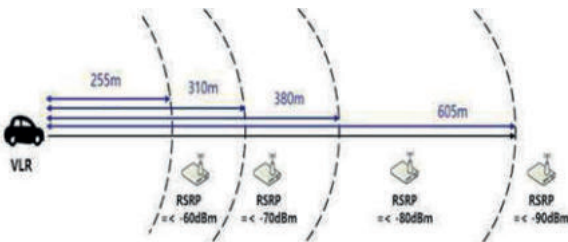


Fig. 6. Protection separation distance for each RSRP of the 5G device

#### IV. CONCLUSION

In this paper, for the mutual coexistence of C-V2X in the 5.9GHz band and Wi-Fi 6E in the 6GHz band, a study was conducted to prepare specific interference protection standards considering the potential interference of Wi-Fi 6E on C-V2X. As a result of the interference analysis between C-V2X and Wi-Fi 6E, when the RSRP of C-V2X is less than -60dBm, a minimum protective separation distance of 255m is required, and when the RSRP is less than -70dBm, a minimum protective separation distance of 310m, -80dBm, and when it is smaller than the protective separation distance of at least 380m, and when it is smaller than -90dBm, the protective separation distance of at least 605m is suggested.

#### ACKNOWLEDGMENT

This work was partially supported by an Institute for Information and Communications Technology Promotion(IITP) grant funded by the Korean government(MSIP)(No. 2018-0-00943, Study on Distributed Radio Resource Allocation Method using Hybrid Block Chain for the Beyond-5G Light Spectrum Sharing Platform) and by the National Research Foundation of Korea (NRF), which is funded by the Ministry of Education (NRF-2016R1D1A1B01007836, Research on a Multidimensional Radio Resource Allocation Method based on Big Data for Massive Autonomous Wireless Devices).

#### REFERENCES

- [1] BILL VISNIC, "Vehcile safety communications landscape clarified with controversial FCC ruling", SAE International, Nov. 2020.
- [2] "Assessment of Wi-Fi Interference to C-V2X Communication Based on Proposed FCC 5.9GHz NPRM", CAMPLLC, Sep. 2020.
- [3] "Use of the 5.850-5.925GHz Band", FCC NPRM, May 2021.
- [4] Yuanyuan Fan, Liu Liu, Shuoshio Dong, Lingfan Zhuang, Jiahui Qiu, Chao Cai, Meng Song, "Network Performance Test and Analysis of LTE-V2X in Industrial Park Scenario", Wireless Communications and Mobile Computing, vol. 2020, Dec. 2020.
- [5] ECO, "SEAMCAT Handbook", Jan. 2010.
- [6] IEEE, "802.11ax Draft 0.4", IEEE, Aug. 2020.
- [7] FCC, "Unlicensed Use of the 6GHz Band; Expanding Flexible Use in Mid-Band Spectrum Between 3.7 and 24GHz", FCC-20-51, Apr. 2020.
- [8] ITU-R, "Characteristics of broadband radio local area networks", Rec. ITU-R M.1450-5, 2014.
- [9] 3GPP, Release 16, 3GPP
- [10] 3GPP, LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment(UE) radio transmission and reception, 3GPP TS 36.101, Apr. 2017
- [11] 3GPP, V2X Services based on NR; User Equipment(UE) radio transmission and reception, 3GPP TR 38.886, July 2020



# Neural Architecture Search for Real-Time Driver Behavior Recognition

Jaeho Seong  
Department of Future Automotive and IT  
Convergence  
Kyungpook National University  
Daegu, Republic of Korea  
wogh3569@knu.ac.kr

Chaehyun Lee  
School of Electronic and Electrical  
Engineering  
Kyungpook National University  
Daegu, Republic of Korea  
hyeu333@knu.ac.kr

Dong Seog Han  
School of Electronic and Electrical  
Engineering  
Kyungpook National University  
Daegu, Republic of Korea  
dshan@knu.ac.kr

**Abstract**—Driver behavior recognition (DBR) helps to ensure driver safety by alerting drivers about potential hazards and minimizing them. In this paper, we use deep learning-based neural architecture search (NAS) to classify driver behavior. In the NAS method, a reinforcement learning algorithm is used, and the neural network architecture is quickly searched by sharing the weights of the parameters. Most DBR models focus on accuracy, while high processing speed is required in order to be applied to actual vehicles. In addition, since the driver monitoring system (DMS) includes complex algorithms based on deep learning, it requires a DBR model that takes this into account. We collect our own data set for driver behavior classification and recognize four common driving behaviors: general driving, mobile phone use, food intake, and smoking. The proposed model on our own data set collected through experiments has better performance and lower network cost than the previous lightweight classification model.

**Keywords**—deep learning, neural network architecture search, driver behavior recognition

## I. INTRODUCTION

The driver monitoring system (DMS) recognizes the driver's careless behavior and warns the driver to prevent traffic accidents. DMS includes algorithms such as driver's face detection, head pose estimation, gaze estimation, and risky behavior classification. Among them, driver behavior recognition (DBR) classifies behaviors such as smoking and eating that distract the driver while driving. Research on deep learning-based DBR using camera sensors is being actively conducted recently. However, deep learning-based models have a large amount of computation and embedded devices used in vehicle environments have limited resources. Therefore, in order to apply DBR in conjunction with algorithms used in DMS, not only high accuracy but also a lightweight classification model is required.

Recently, lightweight model architectures such as MobileNet [1], [2] and ShuffleNet [3], [4] have generally been adopted in devices with limited resources such as mobile or embedded model structures. Furthermore, remarkable progress has been made in the field of neural architecture search (NAS), which automatically generates and optimizes convolutional neural network (CNN). The classification model using NAS showed better performance than the existing human-designed

networks [5]. NAS algorithms achieve high performance despite the high computational cost for searching for a network. Most of the existing NAS studies have analyzed model performance using datasets such as CIFAR-10 and ImageNet.

In this paper, we propose a NAS algorithm for DBR. The proposed NAS algorithm uses operations to optimize the neural network architecture through reinforcement learning and to consider weight reduction and accuracy in the architecture search space. The network architecture was searched for classifying driver behavior, and the proposed model not only achieved higher accuracy than the existing deep learning-based lightweight model but also significantly reduced the cost in terms of network size.

Following the introduction, this paper is structured as follows. Section 2 introduces the NAS algorithm based on reinforcement learning, and Section 3 explains the algorithm for real-time DBR. Section 4 explains the experimental results of the existing classification model and the proposed classification model, and Section 5 concludes.

## II. RELATED WORK

The search space of the neural network architecture defines an architecture space to ensure the model's performance to be generated. The operator space of NAS includes convolution, pooling, and residual connections. In general, NAS requires very high computing costs. Reinforcement learning-based algorithms have proposed methods such as weight sharing and progressive search for efficient search. Most NAS do not consider model lightweight or use latency as a constraint for network lightweight, thus limiting the models that can be generated within the search space. In this paper, performance is improved by redefining the operator space of the NAS for accuracy improvement and weight reduction.

This paper follows ENAS [6] of the reinforcement learning algorithm-based NAS method. The neural network search space consists of a directed acyclic graph. Recurrent neural network (RNN) shares stored weight parameters before training the generated network to accelerate search time and applying previously trained weights to each network. Then, after training the network, the trained weights are stored. When the validation accuracy of the generated network is higher than the previously-stored validation accuracy, the stored cells are removed, and the

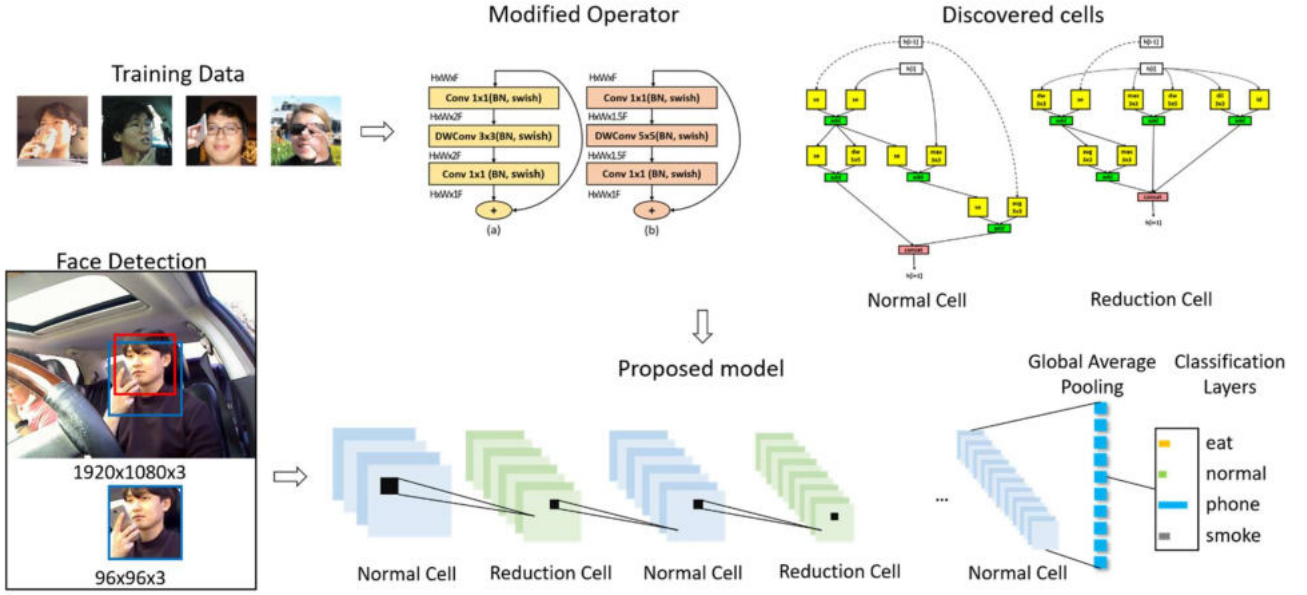


Fig. 1. Proposed DBR Algorithm.

searched cells are stored. Validation accuracy is used as a reward for the reinforcement learning algorithm to be trained in the direction that the RNN controller searches for the optimal network. This process is repeated to search for the optimal neural network architecture.

The learnable parameters are the parameter  $\theta$  of the RNN and parameter  $w$  of the generated network. The policy of reinforcement learning is defined as  $\pi(m; \theta)$ .  $m$  is the model chosen from a policy. To optimize the generated network, we use stochastic gradient descent (SGD) to learn in the direction of minimization. The loss function  $\mathcal{L}(m; w)$  uses cross-entropy, and the expected value of the loss function is  $\mathbb{E}_{m \sim \pi}(m; \theta) [\mathcal{L}(m; w)]$ . It is calculated using Monte Carlo estimation.

$$\nabla_{\mathbb{E}_{m \sim \pi}(m; \theta) [\mathcal{L}(m; w)]} \approx \frac{1}{M} \sum_{i=1}^M \nabla_w \mathcal{L}(m_i, w) \quad (1)$$

As the RNN trains, it learns in the direction of maximizing the expected value of  $\mathbb{E}_{m \sim \pi}(m; \theta) [r(m, w)]$  the reward. The reward  $r(m, w)$  uses the accuracy of the validation data set.

### III. PROPOSED ALGORITHM

#### A. Data Pre-processing

This study proposes an efficient DBR algorithm for recognizing driver behavior from in-vehicle camera sensors and applying it to DMS based on deep learning. DMS consists of complex algorithms such as face detection, facial landmark, head pose estimation, gaze estimation, emotion classification, and behavior recognition. In particular, in the case of face detection, face landmarks, emotion classification, and behavior recognition, deep learning-based models with high accuracy are mainly used. For this reason, to apply a real-time DMS by linking a deep learning-based model with a large amount of computation and a DBR model, it is essential to reduce the model's weight. Face detection in DMS is an essential algorithm

used before applying facial landmarks and emotion classification. Therefore, when the region of interest is extracted using the face detector, an object classification model may be used instead of object detection having a relatively large computational amount. As shown in Fig. 1, the original image is the size of  $1920 \times 1080 \times 3$ . By removing the surrounding background, inference speed and classification accuracy can be improved. A region of interest, including the driver's face region, is designated through the face detector. In order to classify the driver's behavior, the region of interest is expanded. Since most of the driver's behavior range is from the face to the upper body, it expands to the lower region. The image is resized to  $96 \times 96 \times 3$  to reduce the amount of computation. Pre-processed images classify driver behavior into four classes: eat, normal, phone, smoke through the proposed CNN-based classifier.

#### B. Architecture Search Space

As shown in Fig. 2, the search space consists of a network, cells, and nodes. A node is a unit constituting a cell. Node is a specific tensor having an output value of operator. An input node is defined as  $N_i$ , an intermediate node is defined as  $N_j$ , and an output node  $Y_o$ .  $N_i$  is represented by  $N_1$  and  $N_2$ . Input nodes  $N_1$  and  $N_2$  take as input the previous output tensor and the former output tensor, respectively, and are applied to the next first normal cell.  $N_j$  has  $n - 2$  output tensors from  $N_3$  to  $N_n$  if the number of nodes is  $n$ .  $N_j$  outputs the operator defined to extract features from the RNN controller and the order for connecting nodes to each other. For all nodes  $N_n$ , if the connection between nodes is defined as  $(a, b)$  and the operation is defined as  $o$ ,  $o(a, b)$  can be defined as the connection between nodes  $N_a$  and  $N_b$ . Finally,  $N_j$  ( $N_3 \sim N_n$ ) outputs  $Y_o$  by performing a concatenation operation. The RNN controller samples two operations for each node and a number to connect between the two nodes. Two output tensors are generated by the combination of each operation and node, and one output tensor is generated through element-wise addition operation.

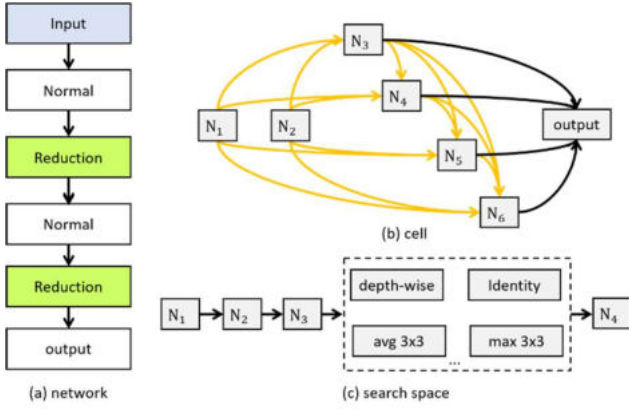


Fig. 2. Architecture Search Space. (a): Network. (b): Cell. (c): Search Space.

A cell is defined as a mapping of  $H \times W \times F$  to  $H' \times W' \times F'$ .  $H$  is the height,  $W$  is the width, and  $F$  is the number of filters. For normal cell,  $stride = 1$  is applied when the operation is applied. Therefore, it becomes  $H' = H$ ,  $W' = W$ ,  $F' = F$ . On the other hand,  $stride = 2$  is applied to the reduction cell to  $H' = H/2$ ,  $W' = W/2$ , and  $F' = 2 \times F$  increases the number of filters.

The network consists of small modular normal cells and reduction cells. This cell-based architecture search is advantageous for weight sharing because the search cost is lower than designing the entire network, and the same architecture is repeated. The entire network is constructed by repeatedly stacking normal cells and reduced cells.

### C. Operator Space

In general, the operator space in NAS includes convolution, pooling, and residual connection. For example, operators used in ENAS define a total of 7 operators using kernel-sized average pooling and max pooling of  $3 \times 3$ , kernel-sized convolution of  $3 \times 3$  and  $5 \times 5$ , kernel-sized depthwise-separable convolution of  $3 \times 3$  and  $5 \times 5$  [6]. However, according to our experimental results of this paper, these operator definitions are not effective for both accuracy and weight reduction. In addition, regular convolution operations are not suitable for weight reduction models due to their high computational cost and do not include additional attention blocks [7]-[9].

Therefore, this paper proposes two operator spaces for weight reduction and performance improvement.

The first operator space is as follows:

- inverted bottleneck conv:  $3 \times 3$ ,  $5 \times 5$
- max-pooling:  $3 \times 3$
- average-pooling:  $3 \times 3$
- se-block expansion ratio =0.25
- skip connection (identity)
- dilated conv:  $3 \times 3$

As shown in Fig. 3, the inverted bottleneck conv has kernel sizes of  $3 \times 3$  and  $5 \times 5$ , and the expansion ratios use 2 and 1.5, respectively [2]. The se-block [7] channel reduction ratio was set to 0.25. The kernel size of the dilated convolution is  $3 \times 3$  and is then replaced by regular convolution.

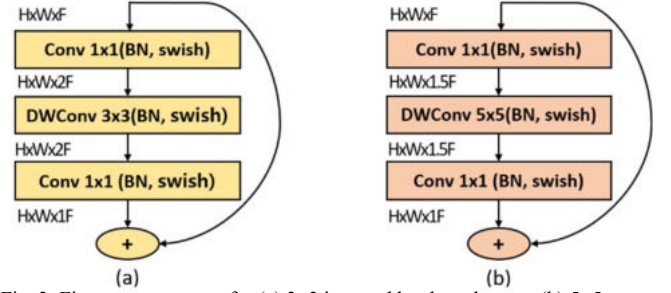


Fig. 3. First operator space for (a)  $3 \times 3$  inverted bottleneck conv, (b)  $5 \times 5$  inverted bottleneck conv.

The second operator space is as follows:

- inverted bottleneck conv:  $3 \times 3$ ,  $5 \times 5$
- depth-wise separable conv with se-block:  $3 \times 3$ ,  $5 \times 5$
- skip connection(identity)

As shown in Fig. 4(a) and Fig. 4(b), the architectures of the inverted bottleneck conv proposed by MobileNetV2. Depending on the kernel size, expansion ratio 4 was applied in  $3 \times 3$  and 2

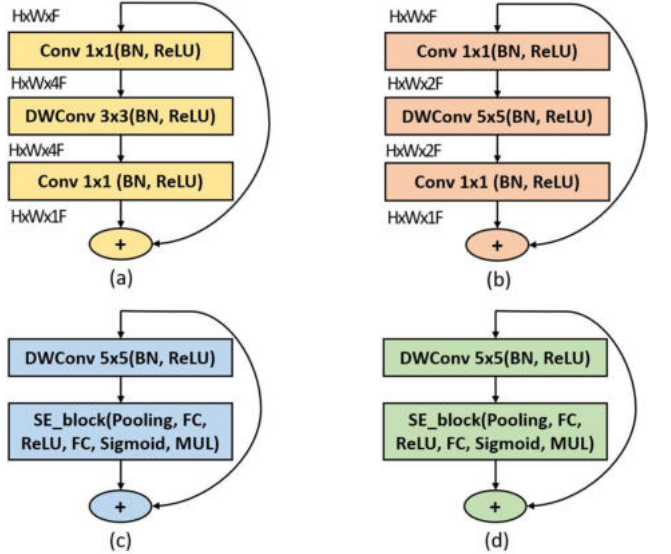


Fig. 4. Second operator space for (a)  $3 \times 3$  inverted bottleneck conv, (b)  $5 \times 5$  inverted bottleneck conv, (c)  $3 \times 3$  depth-wise separable conv with se-block and (d)  $5 \times 5$  depth-wise separable conv with se-block.

in  $5 \times 5$  in consideration of the computation amount. Fig. 4(c) and Fig. 4(d) consist of a combination of depth-wise separable conv, se-block, and skip-connection. The reduction ratio of se-block is set to 0.25. The kernel sizes of conv were  $3 \times 3$  and  $5 \times 5$ . The two-operator spaces in common include batch-normalization and activation functions after the convolution

operation. Also, He normal initialization and l2 normalization were used for each kernel.

#### IV. EXPERIMENTAL ENVIRONMENT AND RESULTS

This Section explains the experimental environment and results. In order to collect driver behavior data, original RGB images were collected using a camera sensor. Then, the face area is designated as the region of interest and the crop of the image. In order to collect driver behavior data from various angles, cameras were installed in four places in the front interior side of the car, as shown in Fig. 5 Abko Apc720 Lite and Logitech Quickcam Pro 9000 were used as camera sensors, and the image sampling rate is 30 frames per second(fps). As a data collection environment, the CPU used Intel Core i7 2.3Ghz and Python Open CV. The collected images are used as training data after extracting the region of interest through the face detector. The overall configuration of the dataset is shown in Table I.

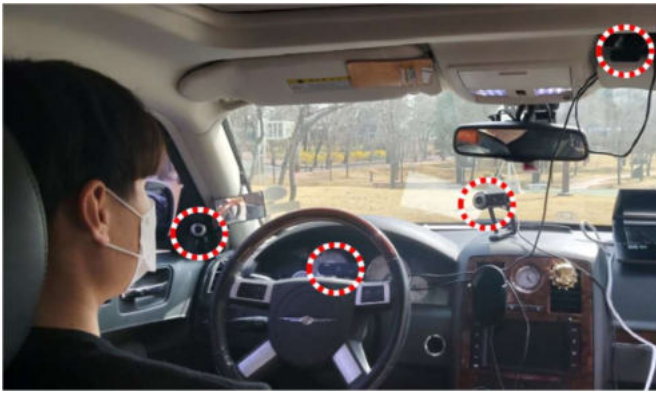


Fig. 5. Data collection environment.

TABLE I  
DRIVER BEHAVIOR DATA SET USED IN THE EXPERIMENT

Class	Training data	Validation data	Total
Eat	3,146	2,050	5196
Normal	4,782	1,130	5,912
Phone	3,204	1,400	4,604
Smoke	4,414	1,030	5,444

As the sensor used in the experiment, the Elp-Usbfhd05mt-k1170ir model was used as a single camera sensor. The Elp camera has a field of view of 180°, a resolution of 1920×1080 size, and a speed of up to 30 FPS. As an environment for NAS, Intel Core i7 3.60GHz, Nvidia GeForce RTX 3070 is used.

(1) Training for RNN: RNN consists of 32 LSTM units and uses Adam as an optimization algorithm to train the model. 0.00035, 0.9, 0.999, and 0.001 were used as the learning rate, the first momentum, the second momentum, and the L2 weight decay, respectively. The total number of nodes sampled in LSTM is 5 in the first operator space and 6 in the second operator space. The batch size of the RNN is 1, and the RNN is trained using the validation accuracy of the network generated for a total of 150 epochs as a reward.

(2) Training for the generated network: One network is generated per epoch for a total of 150 epochs. The generated network uses SGD, learning rate and momentum are 0.05 and 0.9, respectively, and the batch size is 8.

(3) Training for the optimal neural network: After the neural network search is completed, a network with the highest validation accuracy is generated. We use Adam to train the proposed model. The learning rate is 0.001, and if the validation accuracy of the model does not increase within 10 times, the learning rate is halved with a batch size of 8, and the model is trained for 100 epochs.

In this paper, Dropout [10], Stochastic Depth [11], and RandAugment [12] are used as techniques to prevent overfitting and increase model performance. First, dropout is applied only to inverted bottleneck conv, and the ratio is set to 0.1. Stochastic Depth is applied to the operator corresponding to convolution among operators and is removed while the operator is learning at a rate of 0.1. The magnitude of RandAugment is 7.

Table II shows the layer configuration of the proposed network. It was constructed by sequentially stacking normal cell and reduction cell. The number of convolutional filters starts with 8 and doubles the number of filters in the reduction cell. The input size is reduced from 96 to 6, and the driver's behavior is classified through the softmax function through global average pooling (GAP). In Fig. 6, if  $h[i - 1]$  has a larger input size than  $h[i]$ , the input size is reduced by half through  $3 \times 3$  operation, and then used as the input for the next operation.

TABLE II  
PROPOSED ARCHITECTURE

Proposed Model	Filter	Feature map size
Normal Cell	8	96×96
Reduction Cell	16	48×48
Normal Cell	16	48×48
Reduction Cell	32	24×24
Normal Cell	32	24×24
Reduction Cell	64	12×12
Normal Cell	64	12×12
Reduction Cell	128	6×6
Normal Cell	128	6×6
GAP		1×1
1028-dim FC, softmax		

Table III compares the performance of our model with the existing deep learning classification model. The proposed model#1 is a model searched using the first proposed operator space, and the proposed model#2 is the second proposed operator space. For a fair experiment, all experiments were performed under the same conditions, and in the case of accuracy, five averages were measured. MobileNetV2, used in the experiment, reduced the input size to 96×96 and the number of layers for comparison with the proposed model and then compared the performance with the proposed model. The proposed model showed about 8% higher performance than MobileNetV2 despite having fewer of parameters.

TABLE III  
PERFORMANCE COMPARISON OF CLASSIFICATION MODEL

Model	Params	Acc
MobileNetV2	0.92M	83.69
ENAS	3.6M	88.48
Proposed Model #1	0.96M	91.12
Proposed Model #2	0.77M	92.08

In addition, it showed higher performance when compared with ENAS, which is a conventional neural network structure search technique. Comparing the proposed models #1 and #2, they have similar performance, but varying the operator space shows slightly better performance.

## V. CONCLUSIONS

In this paper, the operator of the architecture search space was modified to search the neural network structure for the driver behavior classification model. The proposed algorithm constructed a more efficient network in the trade-off relationship between accuracy and inference speed. As a result, it showed higher performance than the existing classification model on our data set collected through experiments. In a future study, performance verification when using the proposed model in an actual embedded device is required, and a method for generating driver behavior data to reduce overfitting will be required.

## ACKNOWLEDGMENT

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2021-2020-0-01808) supervised by the IITP (Institute of Information & Communications Technology Planning & Evaluation); and the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2021R1A6A1A03043144).

## REFERENCES

- [1] Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." arXiv preprint arXiv:1704.04861 (2017).
- [2] Sandler, Mark, et al. "Mobilenetv2: Inverted residuals and linear bottlenecks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.
- [3] Zhang, Xiangyu, et al. "Shufflenet: An extremely efficient convolutional neural network for mobile devices." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.
- [4] Ma, Ningning, et al. "Shufflenet v2: Practical guidelines for efficient CNN architecture design." Proceedings of the European conference on computer vision (ECCV). 2018.
- [5] Elsken, Thomas, Jan Hendrik Metzen, and Frank Hutter. "Neural architecture search: A survey." The Journal of Machine Learning Research 20.1 (2019): 1997-2017.
- [6] Pham, Hieu, et al. "Efficient neural architecture search via parameters sharing." International Conference on Machine Learning. PMLR, 2018.
- [7] Hu, Jie, Li Shen, and Gang Sun. "Squeeze-and-excitation networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.
- [8] Woo, Sanghyun, et al. "Cbam: Convolutional block attention module." Proceedings of the European conference on computer vision (ECCV). 2018.
- [9] Wang, Fei, et al. "Residual attention network for image classification." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
- [10] Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." The journal of machine learning research 15.1 (2014): 1929-1958.
- [11] Huang, Gao, et al. "Deep networks with stochastic depth." European conference on computer vision. Springer, Cham, 2016.
- [12] Cubuk, Ekin D., et al. "Randaugment: Practical automated data augmentation with a reduced search space." Proceedings of the

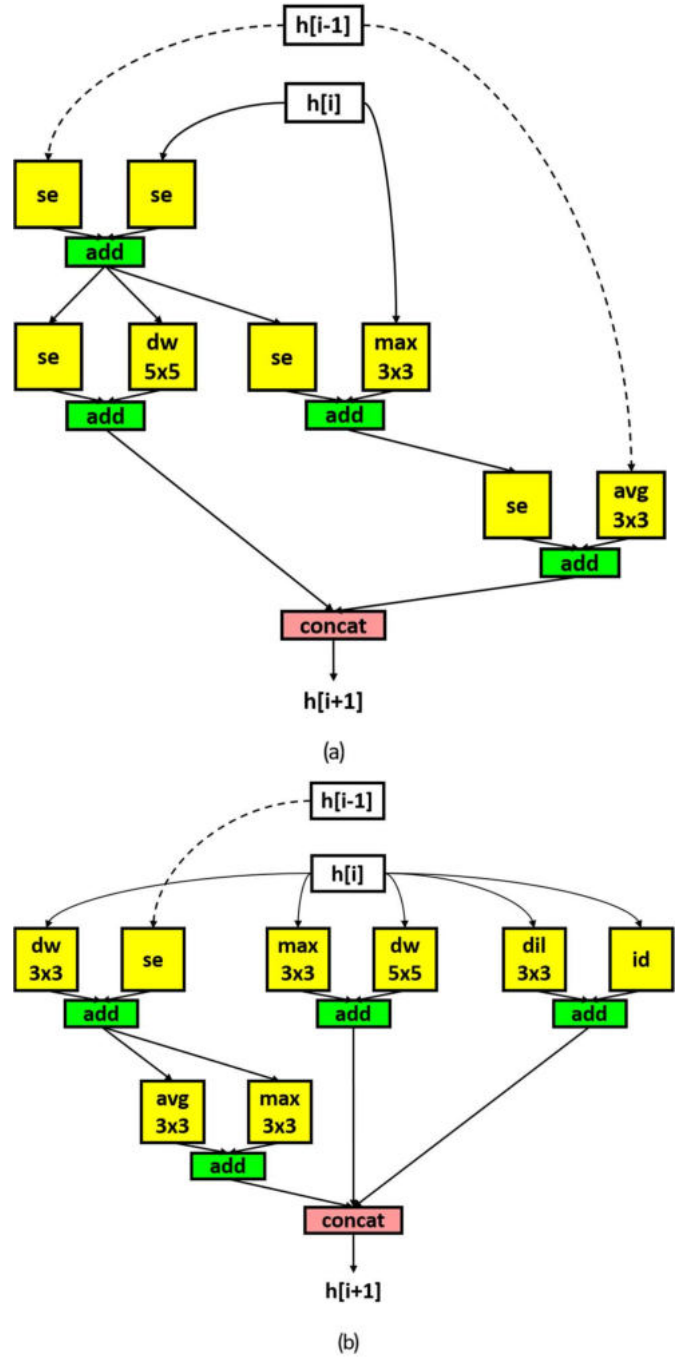


Fig. 6. Cells of proposed model#1 discovered in search space for (a) normal cell and (b) reduction cell.

IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops.

# Smart Anomaly Detection: Deep Learning modeling Approach and System Utilization Analysis

**Mourad Bouache**

*Intel*

*AI and Performance Department*

*Santa Clara, CA*

*California, USA*

*bouache@yahooinc.com*

**Benaoumeur Senouci**

*NDSU*

*ECE Department*

*North Dakota State University*

*Fargo, USA*

*Ben.senouci@ndsu.edu*

**Abstract**—The objective of this project is to perform automated classification of anomalies of system within a large scale cloud production environment using neural network based models on time series data. In large clusters of mixed workload servers, the ability to automatically identify abnormal system utilization is a challenge due to the scale of the problem. To solve this problem, we use deep learning modeling techniques, Long-Short Term Memory (LSTM) models. The data set, to build and test the models, is from production systems over the span of two weeks. The models will use utilization metrics such as CPU, Memory, Network IO, Process Run Queue and Open Files. Anomalous usage in the production cluster is classified as (1) very low usage - less than 5% across selected metrics and (2) known anomalous behaviors like memory leaks. This paper will explain how we can create a model that will identify the anomalies we want to flag, in the real world data. We are using Intel Optimized TensorFlow in containers distributed within a cluster of TensorFlow servers.

**Keywords**—*System Utilization, CPU Utilization, Performance, Deep Learning, Neural Network, LSTM, Anomaly Detection, Performance Engineering.*

## I. INTRODUCTION

A neural network is made of an input layer, a hidden layer, and an output layer. Each layer includes multiple nodes, or neurons, and dictate the input, make inferences from those inputs in the hidden layers, and then outputs the results. The synapses are the connections between all these neurons, pretty much like the brain. If we compare the typical way a computer thinks, we give them input and then an output is generated.

This study is to understand resource utilization: CPU and Memory at Verizon Datacenters, classify hosts based on workload type and further identify systems with abnormal utilization patterns, compared to their peers within the host clusters.

**The problem.** In large clusters of servers we are striving for the ability to automatically identify abnormal system utilization. With clusters that span thousands upon thousands of nodes, monitoring individual servers is typically impractical.

Therefore, we must apply automated, preferably autonomous, systems for classifying the cluster hosts and their workload patterns, to identify outliers and anomalies in utilization.

Using machine learning or deep learning approaches, we believe that it is possible to build a system that can identify the outliers with high degree of confidence. At the same time, account for seasonal patterns, unexpected peaks of traffic, or other atypical patterns of desirable system behavior within the sample server clusters.

In Section 2, we provide background about the deep learning platforms used in this study: TensorFlow and Intel Deep Learning solutions. In Section 3, we will talk about the LSTM model for time-series data. Experimentation environment is covered in Section 4 where we will discuss the classification and clustering process where we compare the normal to the abnormal operations during different resource usage. We will also delve into the experimentation environment in Section 5 with a distributed containers running the TensorFlow clusters. Results will be exposed within the same section. In Section 6 we conclude with some recommendations on anomaly detection using neural network and next steps.

## II. BACKGROUND

In this section we are presenting different technologies that we are leveraging for this work. For training we are using Tensorflow<sup>1</sup> on GPUs and inference using on general purpose CPU clusters using Intel Deep Learning Reference Stack built on Clear Linux with optimized Eigen, Intel MKL-DNN, and AVX512-DL Boost and VNNI for Tensorflow in containers.

### A. Deep Learning Platform: TensorFlow

TensorFlow [1] is one of the leading deep learning and machine learning frameworks today. Earlier in 2017, Intel worked with Google to incorporate optimizations for Intel Xeon processor based platforms using Intel Math Kernel Libraries

---

<sup>1</sup> <http://www.tensorflow.org>

(Intel MKL: see section 2.3). These optimizations resulted in orders of magnitude improvement in performance – up to 70x higher performance for training and up to 85x higher performance for inference.

### B. Intel Advanced Vector Extensions

In addition, the Intel Xeon Scalable processor includes Intel Advanced Vector Extensions 512 (Intel AVX-512) [2], originally introduced with the Intel Xeon Phi processor product line. The Intel Xeon Scalable processor introduces new Intel AVX-512 CPUID flags (AVX512BW and AVX512DQ) as well as a new capability (AVX512VL) to expand the benefits of the technology. The AVX512DQ CPUID flag is focused on new additions for benefiting high-performance computing (HPC) [4] and machine learning workloads.

### C. Intel Math Kernel Library

The optimizations discussed in this article utilize the Intel Math Kernel Library [5] for Deep Neural Networks (Intel MKL-DNN). This is an open source performance library for Deep Learning applications, intended for acceleration of DL frameworks on Intel architecture. Intel MKL-DNN includes highly vectorized and threaded building blocks for implementation of convolutional neural networks with C and C++ interfaces. Note that TensorFlow currently supports the open-sourced Intel MKL-DNN as well the DNN primitives [4] in the closed source Intel Math Kernel Library. The version to use is selected when building TensorFlow. It is expected that in the future the support for the closed source DNN primitive will be removed from TensorFlow.

## III. DEEP LEARNING MODEL: LSTM FOR TIME-SERIES

A popular choice for this type of model is Long-Short-Term-Memory (LSTM)-based models. With sequence-dependent data, the LSTM modules can giving meaning to the sequence while remembering (or forgetting) the parts it finds important (or unimportant). Sentences, for example, are sequence-dependent since the order of the words is crucial for understanding the sentence.

Time series prediction can be generalized as a process that extracts useful information from historical records and then determines future values. Learning long-range dependencies that are embedded in time series is often an obstacle for most algorithms, whereas Long Short-Term Memory [3] (LSTM) solutions, as a specific kind of scheme in deep learning, promise to effectively overcome the problem.

A time series is a sequence of discrete data values ordered chronologically and successive equally spaced in time. In this study, the values are the performance metrics such as CPU utilization and memory, Figure 1 shows an example of a time series graph for CPU and memory utilization for a specific host. The values are defined as standard or mean deviation so the idea is to record the time and then the value will help to detect anomalies. There is a combination between the LSTM deep learning model with the times series to predict if the system,

hardware interacting with the application, is entering an anomaly state or to any abnormal operation.

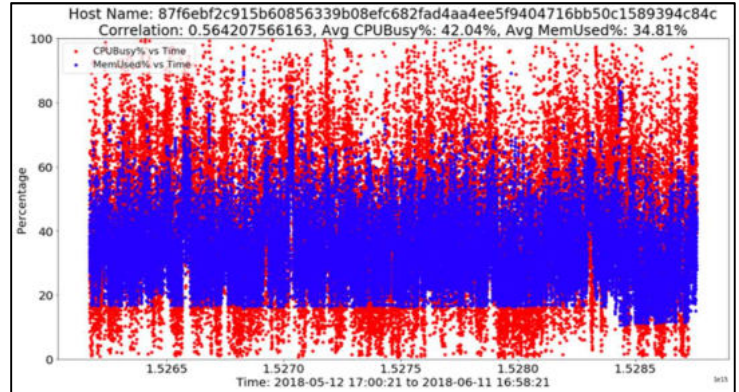


Figure 1: CPU and Memory Utilization Time series Graph

## IV. CLUSTERING AND CLASSIFICATION

In this work an anomaly is an abnormal system utilization resulting from an erroneous workload or system behavior. To identify subgroups of hosts exhibiting a particular workload patterns, we run a clustering algorithm on a set of nodes dedicated to specific compute jobs. We have observed that typically in every cluster, there is a large subgroup of nodes that is a primary workload/behavior of the cluster and a small subset of nodes that have been identified as potential anomalies. Looking at the hosts in each of the groups, the anomalous behavior was confirmed and the hosts are labeled as anomalies to be used in the training model.

Based on the system utilization graphs for the hosts in each workload we have identified the following as “normal” workloads (the color legend is below in the Appendix section ):

### 1. Sporadic workloads - see figure 2

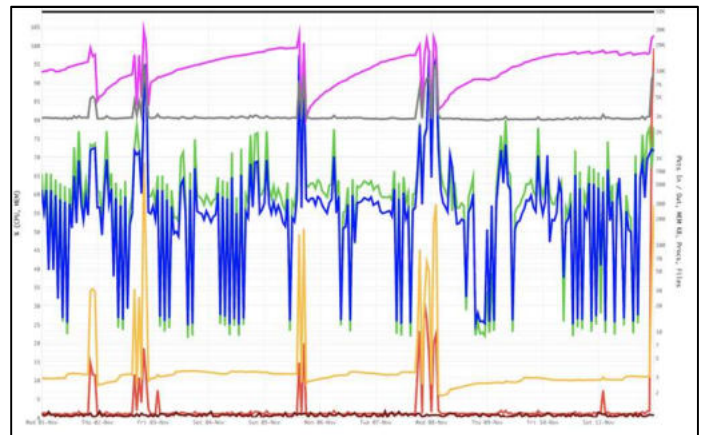
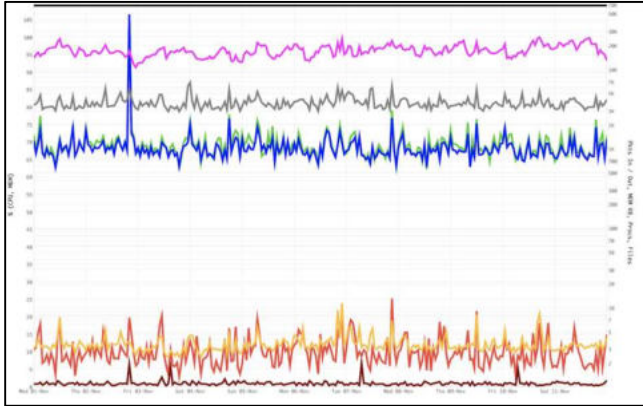


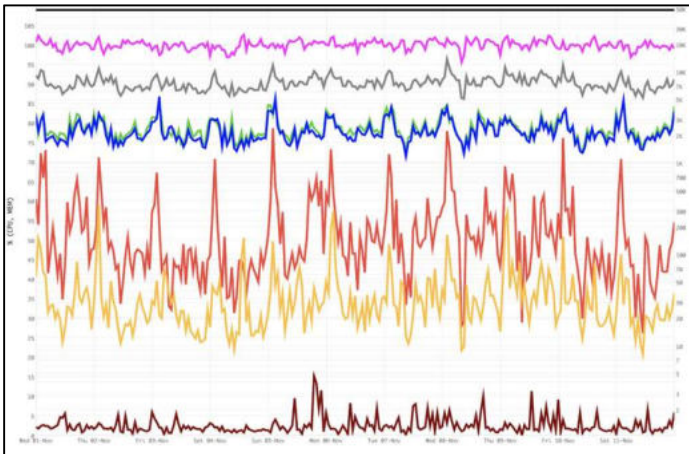
Figure 2: CPU and Memory Utilization time series representing “Sporadic” workload

2. *Low utilization workloads - see figure 3*



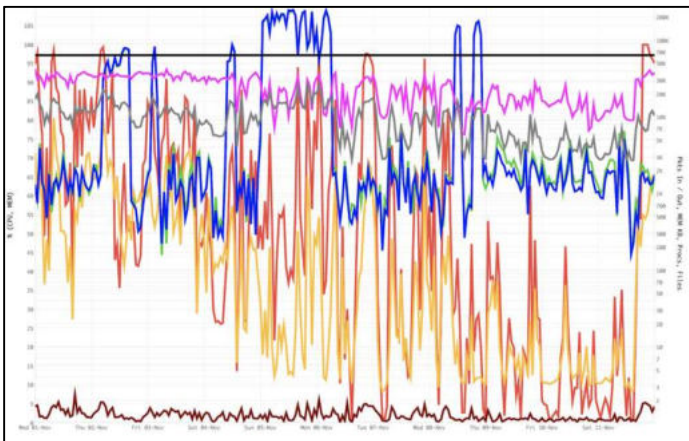
**Figure 3:** CPU and Memory Utilization time series representing “Low Utilization” workload

3. *Average utilization workloads - see figure 4*



**Figure 4:** CPU and Memory Utilization time series representing “Average Utilization” workload

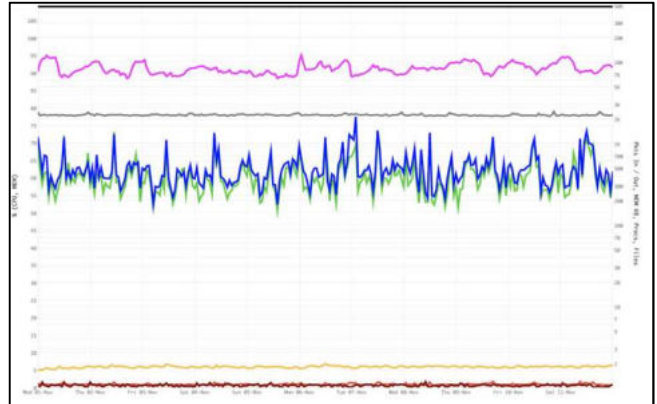
4. *High utilization workloads - see figure 5*



**Figure 5:** CPU and Memory Utilization time series representing “High Utilization” workload

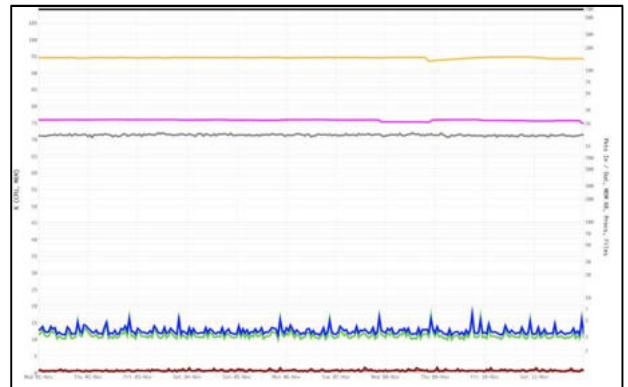
We have also examined and identified the following types of abnormal workloads (“anomalies”):

5. *Anomaly#1 - totally idle nodes (no CPU or memory utilization) - see figure 6.*



**Figure 6:** CPU and Memory Utilization time series representing “Anomaly#1”

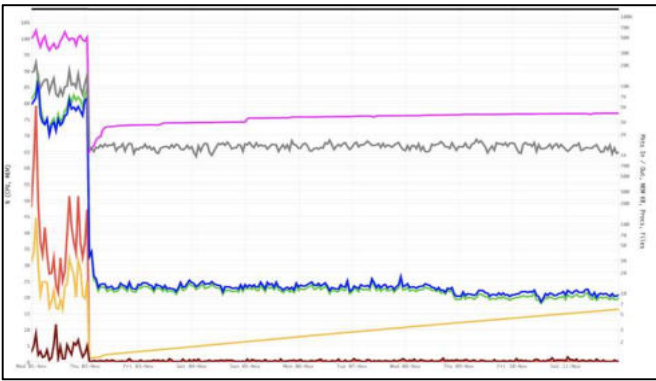
6. *Anomaly#2 - nodes that are idle, but consuming memory (there’s a process, that allocated memory, yet not consuming any significant CPU) - see figure*



**Figure 7:** CPU and Memory Utilization time series representing “Anomaly#2”

7. *Anomaly#3 - nodes that are idle, yet increasing memory consumption (there’s a process that’s not consuming any significant CPU, but is “leaking” memory) - see the right part of figure 8, showing increasing memory consumption.*



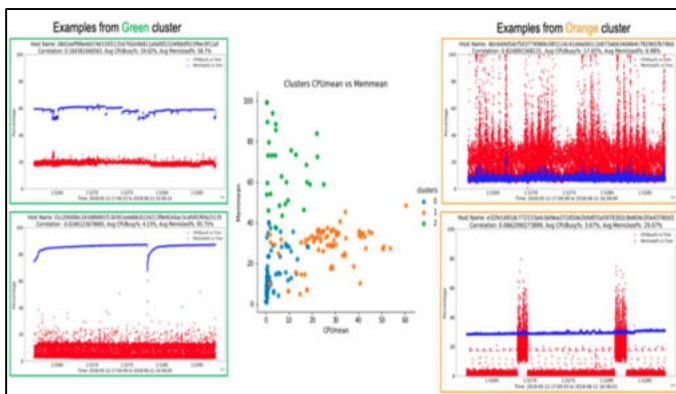


**Figure 8:** CPU and Memory Utilization time series representing “Anomaly#3” and “Anomaly#4”

- 8. **Anomaly#4 - Utilization for these nodes shows high activity for a period of time, which ends abruptly and the node becomes idle. This indicates a node taken out of service (see the change in utilization on figure 8).**

**A. File Analysis**

The files are extracted from YAMAS database in Avro format. They contain data over the course of two weeks and are consistent across all hosts for that entire period. There are about 1000 files generated each containing 20-30 hosts with approximately 20,000-30,000 hosts sampled every minute. The metrics are sampled from standard Linux kernel registers, to capture utilization metrics across the CPU, memory, network, and disk. For the processing of the data and visualization we are using Python 3.6 with the Pandas, Numpy, and SciPy libraries and seaborn for visualization.



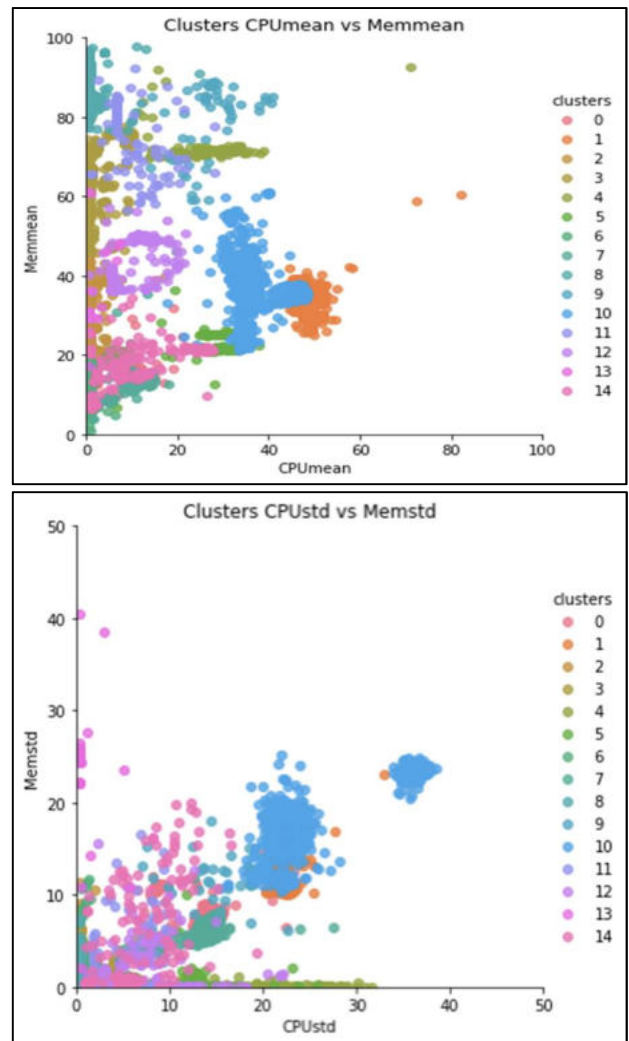
**Figure 9 :** File Analysis for 2 different Clusters (Green and Orange)

**B. Anomaly Detection Motivation**

The motivation for moving towards Deep Learning models started by first understanding the type of data and the general characteristics of anomalies in the data center environment. To get a general understanding of the data, the first step was to characterize the time series metrics by mean, min, max,

standard deviation over the entire time window. Figure 10 shows a graph of the clusters generated from the summary statistics of the metrics *CPU Busy %* and *Active Memory* using *k*-Means algorithm with *k*=15. We find that average usage throughout the data should range between 20-50% depending on the application. This accounts for time users are inactive, for example from the hours of midnight to 4am, and general server provisioning to account for peak usage times/events. The main suspects for anomalies come with the very high memory usage and very low to no CPU usage. Of the 8000 machines in this clustering study, about 250 machines fell into cluster 8 which is the most extreme set of machines that exhibit this characteristic.

Each clustering of hosts ended up being very closely linked to specific workloads, with a few exceptions. We used this clustering to help identify which hosts were assigned to different clusters and checked to see what kind of behavior those hosts were exhibiting. It turns out those hosts were acting like the anomalous behavior described above. In the next section we will discuss how we used two of those workloads to build our LSTM models.



**Figure 10:** 8000 machine cluster based on CPU and memory min, max, mean, standard deviation using *k*-Means with *k*=15

## V. EXPERIMENTATION AND RESULTS

In this section we are going to talk about the software and the hardware deep learning infrastructure to train the model that we created. We will discuss the results within the same section. The training environment was performed on an Intel Skylake CPU machine with 8 NVIDIA V100 GPUs. The inference environment is an Intel CLX server with the Clear Linux Deep Learning Reference Stack<sup>2</sup> which is a container of Clear Linux and Tensorflow with CPU optimized eigen and AVX512.

### A. The Data Set

Data is collected in a production environment through the use of Yamas2 which is a cloud monitoring system, Figure 12. The database contains hundreds of different metrics and for this experiment we have reduced it to a subset. The reduced subset is extracted into files and read into out pandas dataframe for analysis. The data set for training and testing the LSTM model is the time series data for each server with the assigned cluster information from the *k*-Means clustering algorithm. The initial experiments focus on a single LSTM model per workload to verify that it is possible to detect anomalies in a controlled environment. We identified two workloads which are good candidates to build an LSTM model on, we will call them Workload A and Workload B for anonymity. Workload A consists of 114 total hosts with 9 anomalies and Workload B consists of 78 total hosts with 4 anomalies. We reduced the full data set to metrics that are the most generally descriptive from the set. These metrics are CPU busy percent, memory total in kB, memory used percent, memory active kB, processes running, files open, IP packets, and IP packets out. Each host has 15840 measurements where each measurement represents a moment in time. The data is normalized between zero and one before building the model.

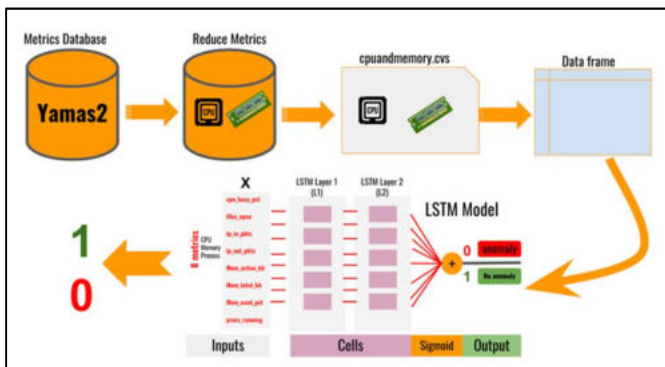


Figure 12: Data collection and manipulation flow for building LSTM models from production data

### B. Model Building

The LSTM model was chosen due to its innate characteristics in handling time series sequential data. The final

model is built with two layers where the second layer outputs to a single output which can be either an anomaly or normal behavior represented as a zero or one respectively. We choose the activation function sigmoid to keep the values bet, Figure 13.

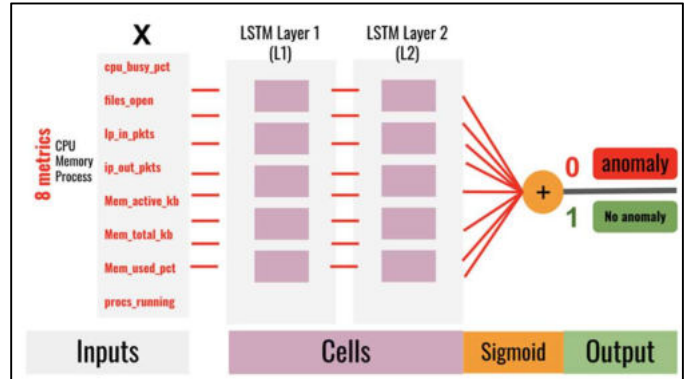


Figure 13: LSTM Model Building

The code for the model is showing in Figure 14 and written in python using keras. The model building starts with initializing a sequential model and adding LSTM layers. In this case we add two layers followed by a single dense layer since we are performing binary classification of the anomalies. The dropout of 0.5 is added to help with overfitting. We choose the Adam optimizer [6] with a loss ratio of 0.001. In future work we plan on exploring different loss ratios as well as adding more LSTM layers. The data set described in 5.a is used as the training, validation and testing data along with the binary class associated with each host represented as the \*\_tar vectors. The model is tested with the testing set and the accuracy is compared to the testing\_tar classes.

```

model = Sequential()
#4 Layer LSTM
model.add(LSTM(256, input_shape=(seq_len, 8), return_sequences=True))
model.add(LSTM(256))
#dropout to prevent overfitting
model.add(Dropout(0.5))
model.add(Dense(1,activation='sigmoid'))

#Optimizer with loss 0.001
adam = Adam(lr=0.001)
chk = ModelCheckpoint('best_model_001.pkl', monitor='val_acc',
    save_best_only=True, mode='max', verbose=1)
model.compile(loss='binary_crossentropy', optimizer=adam, metrics=['accuracy'])
model.fit(training, training_tar, epochs=100, batch_size=128,
    callbacks=[chk], validation_data=(validation,validation_tar))

#Model testing
test_preds = model.predict_classes(testing)

```

Figure 14: Python code for LSTM model

<sup>2</sup> <https://clearlinux.org/stacks/deep-learning>

### C. Results

We used the annotated data to train the LSTM model. Approximately 50-60% of the data is used for training and the rest used for testing with an equal distribution of anomalies within each set. The results from the LSTM models using one and two layers is represented in Table 1. These first experiments were performed to test out various modeling and hyper parameter configurations. With just a single layer LSTM model, with sigmoid activation function and Adam optimization, both Workload A and B had mis-predictions for 75% of the anomalies and a few mis-predictions for the non-anomalous hosts. By adding a second layer, with the same activation function and optimizer, to the network the model accuracy went to 100%. A dropout of .5 was added to the network to help with overfitting. This showed that it is possible to classify anomalies based on utilization data generated by each host.

**Table 1:** Per workload anomaly detection models and their respective accuracy

	Training Set Size (#hosts)	Validation Set Size (#hosts)	Testing Set Size (#hosts)	1 Layer Accuracy	2 Layer Accuracy
Workload A	63	25	26	86.4%	100%
Workload B	45	16	17	84.3%	100%
Mixed Workload	1244	622	626	N/A	100%

The next experiment was to combining multiple workloads into a single data set and see if it is possible to classify the anomalies. This set consists of five different workloads spanning over 2000 machines. Using the LSTM model architecture as above, only changing the batch size to 32 from 128, we tried two different hyperparameter loss ratio values for the Adam optimizer of 0.001 and 0.01. The result was a testing accuracy of 100% and 100%. It seems that even with mixed workload types, each represented in Section 4, that the LSTM model performs with extremely good accuracy. Changing the loss accuracy of the optimizer had no impact on the prediction results.

## VI. CONCLUSIONS AND FUTURE WORKS

In this study we created workload specific anomaly detection models that detect anomalous usage from within large scale production data centers. We used annotated data of two distinct workloads to train the LSTM model from two weeks of data. For each individual workload we got very good accuracy, 100%, with a two layer LSTM with a sigmoid activation layer. When combining the workloads into a single data set, the accuracy went down to 94%, but predicted all anomalies as false negatives.

The work has spawned multiple next steps that we will explore. The first thing we are going to explore the reasoning behind the accuracy levels and ensure that overfitting did not occur in the models. We would also like to explore other sequential and time series classifiers, for example Transformer.

In order to put this solution into production to monitor for anomalies on a daily cadence, the models will need to be rebuilt from daily host data with tagged anomalies instead of monthly host data. The models will continue to be built on GPU systems, but the nightly inference will be performed on a general purpose containerized CPU environment. The container will be highly optimized for inference on CPUs using the Clear Linux Deep Learning Reference Stack for Tensorflow. This will run as a best effort job on the general purpose cluster and anomalies will be reported back through the alerting system.

In conclusion, we believe that deep learning models will be a great method for classifying anomalous usage in large scale production data center.



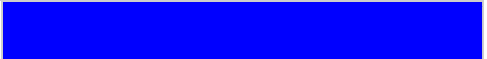

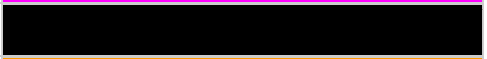



## REFERENCES

- [1] Pattanayak, Santanu. "Introduction to Deep-Learning Concepts and TensorFlow." Pro Deep Learning with TensorFlow, 2017, pp. 89–152., doi:10.1007/978-1-4842-3096-1\_2.
- [2] Kusswurm, Daniel. "Advanced Vector Extensions (AVX)." Modern X86 Assembly Language Programming, 2014, pp. 327–349., doi:10.1007/978-1-4842-0064-3\_12.
- [3] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long Short-Term Memory." Neural Computation, vol. 9, no. 8, 1997, pp. 1735–1780., doi:10.1162/neco.1997.9.8.1735.
- [4] Arora, Ritu. "An Introduction to Big Data, High Performance Computing, High-Throughput Computing, and Hadoop." Conquering Big Data with High Performance Computing, 2016, pp. 1–12., doi:10.1007/978-3-319-33742-5\_1.
- [5] Kalinkin, A., et al. "Intel® Math Kernel Library Parallel Direct Sparse Solver for Clusters." EAGE Workshop on High Performance Computing for Upstream, 2014, doi:10.3997/2214-4609.20141926.
- [6] Zhang, Zijun. "Improved Adam Optimizer for Deep Neural Networks." 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS), 2018, doi:10.1109/iwqos.2018.8624183.

## APPENDIX

### A. LEGEND TO THE UTILIZATION GRAPHS

Figure #2 to #8 contain utilization graphs - see color legend below:

cpu.busy.pct	
ip.in.pkts	
ip.out.pkts	
mem.active.kb	
mem.total.kb	
mem.used.pct	
procs.running	
sys.files.open	

# An Evaluation Framework for Machine Learning Methods in Detection of DoS and DDoS Intrusion

Temechu Girma Zewdie  
Computer Science and Engineering  
University of the District of Columbia  
Washington DC, USA  
temechu.zewdie@udc.edu

Anteneh Girma (Ph.D.)  
Computer Science and Engineering  
University of the District of Columbia  
Washington DC, USA  
anteneh.girma@udc.edu

**Abstract** — A distributed denial-of-service (DDoS) and DoS attack are the most devastating and expensive attacks among various cyber and network attacks [1] [2]. Coupled with the fact that launching such attacks could be relatively easy, it makes it a big problem in the realm of Security and Cyber Space in general. However, with the advent of advanced Artificial Intelligence / Machine Learning (AI/ML) methods and tools, we explore different research techniques and methodologies to find a better detection accuracy result and prevent many different kinds of Attacks and Intrusions. During the research process, we will address Analytical and Computational challenges, Feature Selection issues, and Machine Learning Models while paying particular attention to Feature Engineering by using Mutual Information and Principal Component Analysis in the feature construction process. Moreover, K-Nearest Neighbors, Decision Trees, Random Forests, and XGBoost for Classification are used. In General, this study will target to analyze the ability of these methods to detect DoS and DDoS attacks while also examining the capacity of the ways to distinguish between different kinds of these attacks. Finally, the research investigates and proposes a framework for simultaneous evaluation of different Machine Learning methods in detecting DoS and DDoS.

**Keyword**—*Cyber Security, Cyber Space, Decision Tree, DDoS, Dos, Feature Selection, Machine Learning, Principal Component Analysis, KNN, Random Forest, Mutual Information, XGBoost*

## I. INTRODUCTION

DoS Attacks are primarily a category of Tactics used to disrupt the traffic to a specific server. The malicious agent will cause the server to become temporarily unavailable or unresponsive to other users. It can be imagined as something clogging up the traffic in a road causing jams. Victims of such attacks can include computers, servers, or other networked resources such as IoT devices. DDoS attacks are distributed forms of DoS attacks. These tactics are usually used in tandem. DoS attacks can be precursors to DDoS attacks, and the latter is often followed by the latter. These sorts of attacks are carried out through a Network of Machines. DoS and DDoS attacks are relatively simple but extremely powerful and presently remain an immense threat to network security, and organizations spend much effort trying to address the issue. DDoS exploited the inherent nature of the Internet, and it's an open-source model, which is also its most significant advantage.

Nowadays, software and tools are developed and distributed to manage numerous types of attacks. These allow individuals to sort episodes quickly while providing a user-friendly experience. However, the attacks make the problem even worse.

This paper will investigate the use of Machine Learning methods and techniques in identifying and defending against these attacks. We will start with a definition and a brief taxonomy of these attacks. We will then outline the steps and procedures needed to use Machine Learning methods. Furthermore, we will describe some challenges and provide experimental results that validate our processes and procedures.

The paper will identify these attacks with multiple methods for the Feature Engineering phase. First, we put a set of essential features in Machine Learning models. Then, we will use methods of Dimensionality Reduction and test our main Machine Learning models to use these features.

Finally, we will use K-Nearest-Neighbors, Decision Trees, Random Forests, and XGBoost and provide experimental results. In this research, Finding the suitable dataset itself has been identified as a challenge in the field. We will be carrying out these experiments using the CIC-IDS-2017 dataset.

## II. DoS AND DDoS ATTACKS

### A. Definition

A DoS attack is an attack that can render a network incapable of providing regular services [2]. The purpose of a DoS attack is to obtain access to a network resource that has been intentionally blocked or weakened through the actions of a malicious agent. Attacks don't necessarily damage network systems or data permanently, but they temporarily compromise the availability of the resources [1]. DDoS attacks occur when multiple systems have coordinated to attack concurrently to aggravate the offense. These attacks hit the target system simultaneously. A malicious agent could obtain other systems to carry out the intended attack.

### B. Taxonomy of the attacks

The most common DoS attacks target the computer network bandwidth or connectivity [1]. These attacks compromised bandwidth "flood" the network with many network requests. So, the system will not provide resources to innocent users. It can result in degraded performance or even complete shut-down. Connectivity attacks flood a network with a high volume of connection requests, and the computer can no longer respond to innocent requests. The attacks can also be categorized based on the protocol (based on the 7-layer OSI scheme) they attack. We can also further break up each kind into subcategories [3].

### *Application Layer:*

- Some attacks render a network of machines out of order by taking advantage of specific bugs or weaknesses in the network applications hosted by the target or by requesting unnecessary resources. For example, the attacker may have launched computationally expensive requests and unnecessarily taken up resources.

### *Protocol Level Attacks:*

- At the Operating system (OS) level, DoS attacks the target and takes advantage of the weaknesses in protocol implementation. The attacker may send invalid or unnecessary protocol level requests such as ICMP (Internet Control Message Protocol) or IGMP (Internet Group Management Protocol) requests clogging up the system and halting regular protocol activity.
- Protocol Feature Attacks: these attacks exploit the inherent structure of specific standard protocols and their features. Some exploit the system by taking advantage of certain features of the IP address by spoofing their internet protocol. Others can target the structure of the DNS by trapping the victim into caching false records.

### *Volumetric Attacks:*

- Network Device Level attack is an attack caused by exploiting the weaknesses in firmware or by trying to exhaust the hardware resources of the network. I.e., One may drain the routers, hard drives, or other network hardware resources.
- Data Floods attack may include sending substantial data segments through the network bandwidth, thereby clogging up the network with the unnecessary data flow.

The attacks do not necessarily disrupt network services completely, and they may only degrade them. We can classify these attacks based on various issues such as degree of automation or rate dynamics, yet the above is adequate for our purposes.

## III. MACHINE LEARNING IN DETECTION OF DoS AND DDoS

To defend a network of systems against any intrusion or potential Cyber Attack, including DoS, we must first detect the attack and then act in time to regulate the situation. Machine Learning is used for cybersecurity measures, particularly in three essential domains: anomaly detection, intrusion detection, and misuse detection [3]. The main problem of Machine Learning in detecting DoS is that of Classification, which is a Supervised Learning problem.

In a Classification problem, we have data of previously identified attacks that can serve as blueprints for identifying new ones.

### *A. Review of relevant literature*

The issue of intrusions has been around since the advent of computation. However, with the advancement in communication technology, this problem has been severely exacerbated. The term intrusion refers to “any unauthorized

access that attempts to compromise confidentiality, integrity, and availability of information resources” [4].

Research in the field of detection of Cyber Intrusion has taken various directions. Many have applied statistical models such as Logistic Regression. Others have employed Neural Network models, yet others have used other Machine Learning Classifiers such as Bayesian classifiers, Support Vector Machines, and Lazy Learners.

In [5], Yan et al. propose a system that detects botnets in real-time by only using features from higher-level layers of the OSI. T. Cai and F. Zou [6] presented some features of HTTP Botnet and designed a detection method using clustering based on them. Chien-Hau Hung, Hung-Min Sun created a Botnet Detection System Based on Machine-Learning Using Flow-Based Features. In [7], F. V. Alejandro attempts to detect botnets using Machine Learning and the Evolution of Genetic Algorithms to select Features. In [3], Sofi, Mahajan, and Mansotra investigated the use of Machine Learning Techniques for the Detection and Analysis of Modern Types of DDoS Attacks. In [8], the authors proposed a Random Forest model for detection. The authors of [9] utilize the flow-based features of existing botnets and select 21 features for machine learning, and the outcome of the average detection rate was about 75%.

### *B. Research Methodology*

In a Machine Learning project, feature engineering is a crucial step requiring domain knowledge to extract features from raw data. Once pre-processing is accomplished, Feature Engineering is the next essential step in Machine Learning methods. We often don't have the computational ability to use all features of datasets in our models, and we must first transform or subset a set of essential attributes or features to use them in our models. Many different Supervised Learning methods can use as a model. The selection and comparison of these methods and algorithms are essential in finding the most efficient computationally and accurately system.

Apart from previous work, in this paper, we will be using different kinds of Feature Engineering (Feature Selection) approaches and testing them against other Machine Learning (Classification) algorithms to compare the results. We will be systematically looming the matter so that our work can be used as a framework for evaluating and comparing different methods in Pre-Processing, Feature Engineering and Modeling, and Testing methods to detect DoS and DDoS Intrusion. Our Framework can be adapted and extended in other Cyber Intrusion solutions that consistently achieve the best results for Feature Selection and investigate whether we can find a Classification method that consistently achieves the highest regardless of the Feature Selection method. If the result is that, we can further study the issue by focusing on the most accurate solutions. Finally, our primary purpose is to propose a framework for processing data and evaluating models and Feature Engineering methods.

### *C. The Datasets*

Finding a suitable dataset for our purposes itself is a challenge. We need to find datasets with all the features of the communication packets, and also each instance must be

correctly identified beforehand as an attack or benign activity. Most of the datasets online lack traffic diversity and volume. Besides, they do not cover the variety of known attacks, while others anonymize packet payload data, which cannot reflect the trends. Moreover, some are lacking feature sets and metadata [10].

For this research paper, we will be using the CICIDS2017 dataset from the University of New Brunswick. This dataset is an Intrusion Detection and Prevention dataset, and precisely We will be using the 'Wednesday-WorkingHours' data. The dataset has 79 columns of data. One of which is the label that indicates if the instance is an 'attack' or 'Benign' communication. It also indicates what kind of attack tools was used. The attacks are carried out using the following popular tools:

- Hulk Dos
- Slowloris Dos
- Slow Httpstest
- Dos Goldeneye

#### IV. PRE-PROCESSING

In this pre-processing stage, a data clearance must be done before starting our research. I.e., we have to handle those data with unacceptable values for our algorithms to use. I.e., Nan (Not a Number) values or Inf (Infinity) Values have to rectify before we start.

There are many methods to handle such issues, such as dropping values, imputation, and extended imputation. For simplicity and better accuracy, we will drop instances with unacceptable values. An attribute containing instances with 'Inf' values such as 'Flow Bytes/s' and 'Flow Packets/s' will drop before continuing to the next step.

#### V. FEATURE ENGINEERING

Feature Engineering refers to creating or selecting features from the data to be used in Machine Learning. In this work, we will be using three methods for Feature Engineering. After removing features with zero Variance, we will be using Mutual Information and Principal Component Analysis (PCA) to reduce the features of the datasets for use in the Machine Learning models. We will either transform these features using the PCA or select the most significant features using the other two methods.

##### A. Removal of Features with zero Variance

Before we proceed, we will remove the columns from the dataset with zero Variance. Columns with constant values don't change from instance to instance. Variance thresholding itself is a technique that can use in feature selection. A threshold is often chosen, which will be the cutoff point to include any features. If a particular column has a higher variance than the cutoff, it should be included in the model; otherwise, ignored. Here we will not be using Variance to select features for our models. Our cutoff variance will be zero, meaning that we will only drop columns of data with Variance zero, that is, a column with constant values.

##### B. Mutual Information

As referred to, an Information Gain measures how a target variable, which is the kind of DoS Attack, is dependent on other variables in the data. We will be using a sklearn-learn library to compute the Mutual Information, and we will use that to identify the most minor and most essential features for our purpose.

##### Sampling

Computation of Mutual Information can be quite expensive. It can take quite a long time to compute the Information of each Feature in our DDoS dataset. In order to achieve this, we will be using sampling. We will sample the dataset randomly at a fraction of 0.05 and then compute the Mutual Information for Features of the sample dataset. The values computed for the sample shouldn't be much different from the actual population as the sampling was random.

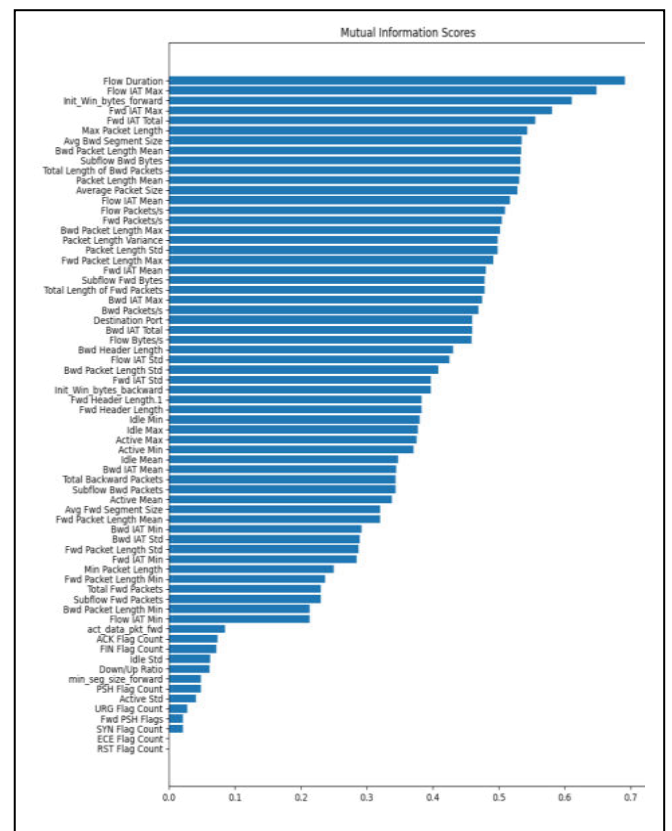


Fig. 1. Mutual Information

We will be using a range of numbers for the number of features selected for our purposes. We will be treating the number of the columns used as a hyper-parameter that will help tune in subsequent sections.

Here we will define some of the topmost and necessary features in the data. The definition will give us insight into what characteristics a potential intrusion may have also will provide us with a sense of what will be essential to us in our modeling [11]:

*Flow Duration*: The duration of communication flow. Which means the temporal length of the flow. It means the total time of the connection from beginning to end. It has been used a lot in Intrusion Detection.

*Fwd IAT Max*: The Inter-Arrival Time is the amount of time that elapses after receiving a packet until the next one arrives. Fwd IAT Max refers to the maximum of all IAT's forward direction.

*Packet Length Mean* : Average length of packets transferred in the connection flow.

*Subflow Bwd Bytes*: The average number of bytes in a sub-flow in the backward direction.

*Flow IAT Mean*: Mean of Inter-Arrival Times in the flow.

*Packet Length Std*: Standard deviation of the length of packets in the connection.

*Fwd Packet Length Max*: The maximum length of all packets in the forward flow.

*Fwd IAT Mean*: Average mean of Inter-Arrival Times in the forward direction.

To interpret the meaning of this, we will first have to understand the importance of features mentioned above to our Intrusion Detection models. These features suggest that the most decisive features that can determine the banality of malice of a connection concerning to Dos attacks are usually related to durations of flow, duration of Inter-Arrival Times, and length of packets. We can further statistically analyze these features to instigate further insight.

### C. Principal Component Analysis

PCA is a method for transforming the features of a dataset into a new set of features. It is a mathematical mapping that will map a location of points from one space to another. However, the features will be ordered decreasingly regarding importance (i.e., the first features will carry more weight than those coming after). Thus, reducing the need for too many features. We can capture the essential features of the data using fewer features in a different space. Figure 2 shows the variance explained by the first ten components after transformation. We can see from this and figure 3 that we capture nearly all of the variance in our data using only a few (perhaps 5 or 6) components.

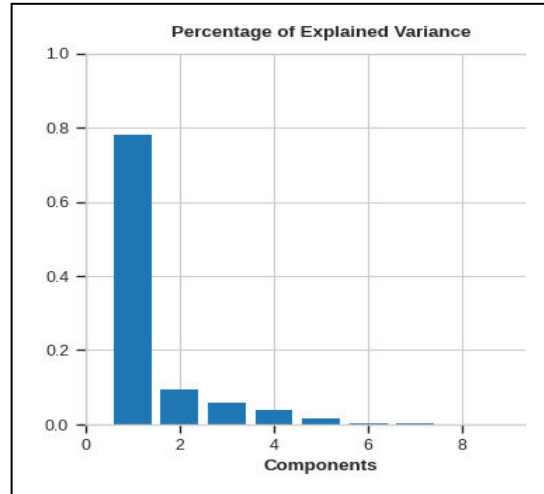


Fig. 2. Percentage of Explained Variance

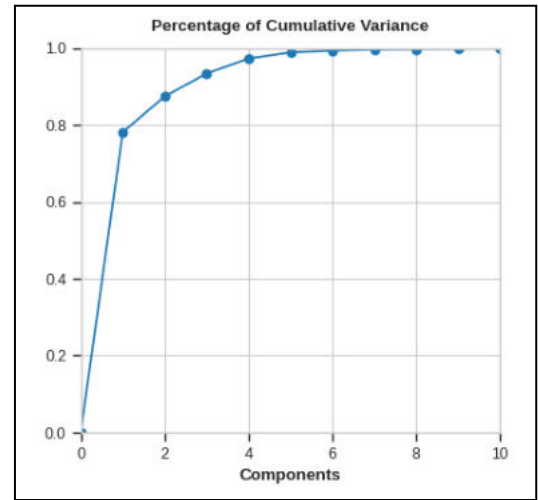


Fig. 3. Cumulative Variance

## VI. MACHINE LEARNING

In this section, we will be using Machine Learning algorithms to classify the activities, i.e., primarily address a classification problem. Each instance of the data will be labeled either a benign activity or labeled as an attack.

### A. Decision Tree Classifier

The Decision Tree Classifier is one of the most known and used Machine learning algorithms. The simplicity and efficiency make it a great candidate in any application. A decision tree is a classifier for determining an appropriate action (among a predetermined set of steps) for a given case [12]. It helps for Classification and Regression purposes. The algorithm works by creating certain simple decision rules that predict the value of a target variable. Representations of the Decision Tree can look like an inverted tree hence the name.

## *An Architecture of the Decision Tree*

A decision tree has three kinds of Nodes. The Root Node is the highest, and It has no incoming Edge but can have none or more outgoing Edges. Internal Nodes have exactly one incoming edge and two or more outgoing Edges. Leaf Nodes have precisely one incoming Node and no outgoing Edges [13].

### *B. Random Forest Classifier*

The Random Forest is an Ensemble of Decision Trees. Random forests [14] are the most popular bagging technique in machine learning, where we use decision trees as the base models. In other words, a random forest consists of many iterations of the Decision Tree, each of them constructed using a bootstrap sample. The outcome will usually be the average of all the outputs of the Decision Trees. The Random Forest Classifier achieves very highly and is very efficient due to its simplicity.

### *C. k-Nearest Neighbors Classifier*

Perhaps the simplest of all Supervised Learning methods is the KNN. This algorithm is in the category of Lazy Classifiers. As opposed to Eager Classification, which attempts to capture deep structures in the data, this kind of algorithm imitates the nearest or most similar (in terms of Classification Features) examples. So the KNN will look for the k most similar measures in the dataset and assign the label per those instances. It could output the weighted average or simply the most occurring label.

### *D. XGBoost Classifier*

Arguably the most accurate modeling technique for structured data. XGBoost is an optimized Distributed Gradient Boosting framework designed to be highly efficient, flexible, and portable [15]. It provides Machine Learning algorithms using the Gradient Boosting framework. It is capable of Automatic Feature selection, cleverly penalizes inaccurate trees, shrinks leaf nodes, makes use of randomization parameters, it can also utilize distributed and out-of-core computation.

## VII. RESULT

Like any other scientific or engineering discipline, Machine Learning also requires experimentation. Our experiments are usually only computational. Nevertheless, it is an essential part of our process. In this research, we achieved very high accuracy results in our experiments. Here we will present these results and provide an analysis. Here, we will represent the results in two separate graphs, one for features selected using Mutual Information and one for features based on Principal Component Analysis. The accuracies shown are accuracies on the Test Set. The accuracy for each algorithm is juxtaposed alongside the others.

### *Two-step process of Classification*

In any Classification problem, we have two steps to be accomplished: Optimization, also known as Training and Prediction. These processes are carried out on the datasets using different algorithms. The first one, also called the Optimizer, will run through the Train Set and capture its

essential features. The Predictor then uses this structure to Predict the labels of new data.

We are concerned about the accuracy of prediction on new data. To test our Classification models, we will need two separate datasets. We must first Train the program on the Train dataset and then evaluate the accuracy of the Test dataset.

### *Train-Test Splitting of the Dataset*

To Train and then Evaluate our model, we must first create two separate datasets, one for the Training and one for evaluation of the model. To achieve this, we will split our dataset into a Train and Test set using randomly chosen samples. The fraction of the data used for Testing is 30 percent, while the rest will use for Training.

### *A. Evaluation Metrics*

The metric we use in this paper is classification accuracy on the Train set. We must keep in mind that our models also predict the tool used in the attack. The accuracy that we measure here is the Accuracy, Precision, and Recall of Classification. The latter two will use in a binary way. That is, we will be treating the output in a binary format. If the output is Benign, the binary value is set to False if it is an Attack, the binary value is set to True. Here are the formal definitions of the metrics we will be using in the present work:

$$Accuracy = \frac{Number\ of\ correct\ Predictions}{Total\ Number\ of\ Predictions}$$

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

In the figures below, we can see the Accuracy of the Classifiers plotted versus the number of Features chosen. There are two graphs, one for Features selected based on PCA and one for Features based on Mutual Information. These plots allow us to compare the performance of different Classifiers and also allow us to compare the performance of a single Classifier for different numbers of Features. The Classifications here are very accurate.

We can see that the KNN always has the upper hand for PCA-based Features, followed by the Decision Tree and Random Forest. The optimal number of Features we need to use here for PCA-based features is 5 or 6. Beyond that, we don't need anymore. We can see that as we were expecting from the PCA plots, we don't need more Features to capture the essential information of the data. We have effectively reduced the number of Features needed while preserving Accuracy.

Figure 4 shows the Accuracy of the Classifiers for Mutual Information Feature selection. Here we can see that Random Forest has the upper hand, followed closely by the Decision Tree and K-Nearest Neighbor. XGBoost has the lowest performance in both scenarios. Here again, we can reach near-perfect Accuracy by using six or more Features based on Mutual Information. In effect, we have chosen a small subset



of features instead of all of the classification features, improving efficiency computationally.

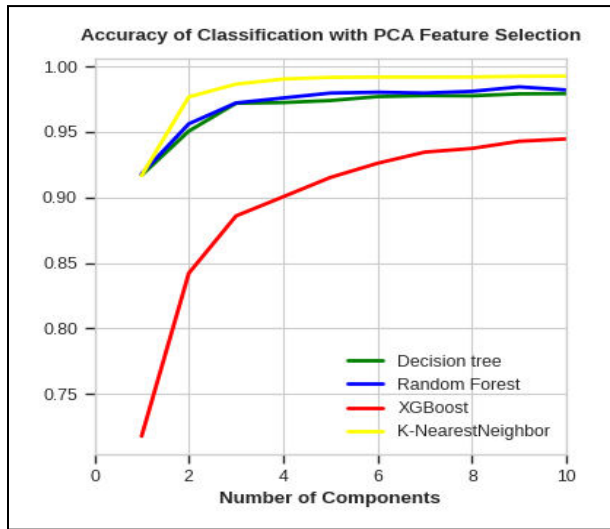


Fig. 4. Accuracy of Classification with PCA Feature Selection

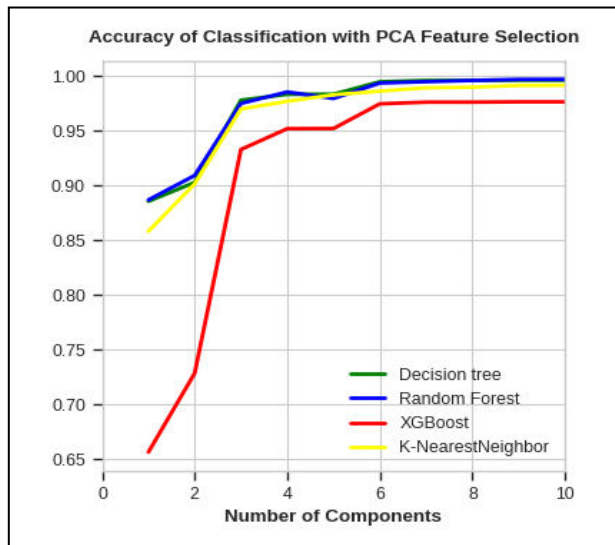


Fig. 5. Accuracy of Classification with Mutual Info Feature Selection

Here we can see the precision and recall tables of our Classifiers. Again, one table is the results for our PCA Features selection, and the other is for Mutual information Feature selection. We can generally see that precision and recall for Mutual Information based selection is always higher except for KNN. For the KNN, it seems like our precision and recall are improving with PCA Features.

Features	DTC precision	DTC recall	RFC precision	RFC recall	KNN precision	KNN recall	XGB precision	XGB recall
1	0.921936	0.95282	0.921676	0.95448	0.908277	0.969902	0.902277	0.962379
2	0.95047	0.97398	0.957333	0.97462	0.974629	0.989307	0.945196	0.977386
3	0.981242	0.97545	0.975595	0.98069	0.985818	0.993129	0.963526	0.981792
4	0.974674	0.98208	0.976446	0.98588	0.989795	0.995131	0.965017	0.986335
5	0.977144	0.98232	0.981042	0.98695	0.99131	0.995738	0.96552	0.988693
6	0.981258	0.98266	0.98191	0.98705	0.991618	0.995799	0.967784	0.989376
7	0.981142	0.984	0.979021	0.98911	0.991618	0.995768	0.968225	0.990369
8	0.980636	0.98427	0.981183	0.98895	0.991886	0.995662	0.969267	0.990635
9	0.980478	0.98684	0.987175	0.9882	0.992502	0.995753	0.97017	0.991461
10	0.981569	0.98583	0.982497	0.98925	0.99266	0.995829	0.970406	0.992894

Tab. 1. Precision and Recall of Classification for different number of Features for PCA Feature Selection

Features	DTC precision	DTC recall	RFC precision	RFC recall	KNN precision	KNN recall	XGB precision	XGB recall
1	0.920722	0.90124	0.921243	0.90266	0.846989	0.953726	0.909032	0.913216
2	0.93328	0.91596	0.935068	0.92311	0.893082	0.963077	0.928154	0.924409
3	0.980731	0.98574	0.975868	0.98593	0.970072	0.985175	0.969274	0.992758
4	0.982642	0.9947	0.982282	0.99599	0.977598	0.988496	0.973095	0.996163
5	0.982187	0.99514	0.979318	0.98962	0.983512	0.991545	0.973842	0.996049
6	0.996096	0.99645	0.993202	0.99717	0.986822	0.992629	0.975939	0.998741
7	0.996478	0.99763	0.995262	0.99716	0.990699	0.993539	0.975996	0.998696
8	0.996508	0.99762	0.996313	0.99792	0.991107	0.993948	0.975996	0.998696
9	0.996652	0.99783	0.997386	0.99823	0.992197	0.995147	0.976004	0.998749
10	0.996652	0.99782	0.997696	0.99817	0.992311	0.995276	0.976004	0.998749

Tab. 2. Precision and Recall of Classification for different number of Features for Mutual Information Feature Selection

PCA-based features and the other is for Mutual Information base features. The mistakes (or confusion) for the Classifier with Mutual Information features are lower than those with PCA-based features. In general, it seems like Mutual Information based selection is the better choice for Decision Trees in this problem.

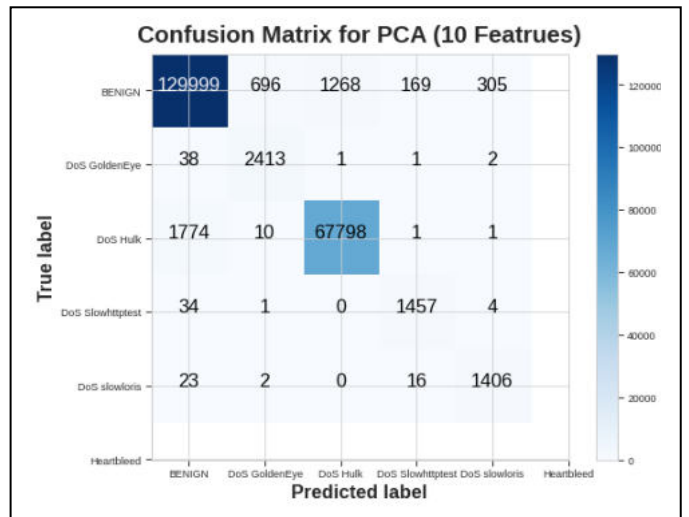


Fig. 6. Confusion Matrix of Decision Tree Classification with 10 PCA Features

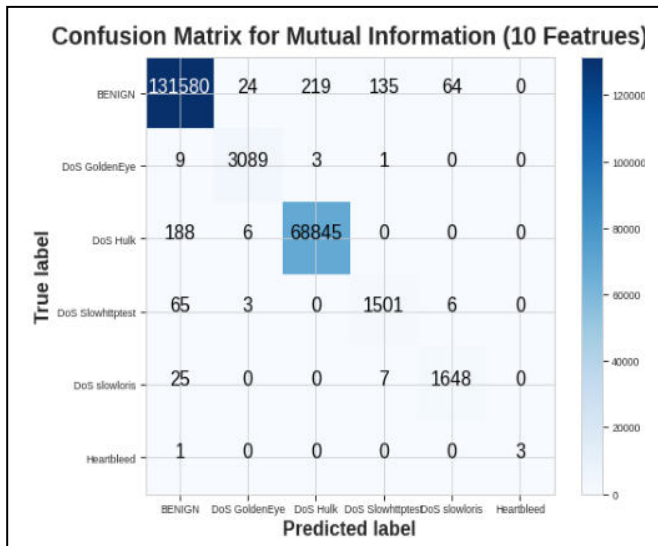


Fig. 7. Confusion Matrix of Decision Tree Classification with 10 Mutual Information Features

### B. Discussion

The algorithms were evaluated, and the results were juxtaposed and plotted above gives us a chance to judge and compare the different approaches.

## VIII. CONCLUSION

Herein we have shown the adequacy of our modeling and Machine Learning techniques to deal with the issue of Intrusion and, in particular, DoS and DDoS Intrusion. We have shown that our algorithms not only discern the fact that a DoS Intrusion is being made but also determine which tool was used in the attack. However, our focus was to show how we can compare different Classifiers and different Feature Engineering methods. To do that, we have demonstrated and proposed a framework for comparing these different methods. The results determined that Mutual Information is perhaps a slightly better tool, in the long run, to use for Feature Selection of such problems.

Most importantly, we achieved high accuracy results with all the algorithms, with XGBoost being the worst. Moreover, we developed and expounded a framework for analyzing, pre-processing, modeling, benchmarking, evaluating multiple Feature Engineering and Machine Learning models. In addition, we outlined all the steps needed to use Machine Learning and presented a framework upon which future research can be based. Researchers may use our framework to experimentally investigate the adequacy of their models for their specific problems. The problems don't have to be confined to DoS Intrusions, but they can be applied to similar problems.

## REFERENCES

- [1] C. Douligeris and A. Mitrokotsa, "DDoS attacks and defense mechanisms: classification and state-of-the-art," *Computer Networks*, vol. 44, no. 5, pp. 643 - 666, 2004.
- [2] L. Stein and J. Stewart, "The Worldwide Web Security FAQ," 4 February 2002. [Online]. Available: <https://www.w3.org/Security/Faq/>.
- [3] I. Sofi, A. Mahajan and V. Mansotra, "Machine Learning Techniques used for the Detection and Analysis of Modern Types of DDoS Attacks," *International Research Journal of Engineering and Technology (IRJET)*, vol. 4, no. 6, 2017.
- [4] K. Scarfone and P. Mell, "Guide to Intrusion Detection and Prevention Systems (IDPS)," February 2007. [Online]. Available: <https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-94.pdf>. [Accessed 4 December 2021].
- [5] Q. Yan, Y. Zheng, T. Jiang and W. Lou, "PeerClean: Unveiling peer-to-peer botnets through dynamic group behavior analysis," in *Computer Communications (INFOCOM)*, 2015.
- [6] T. Cai and F. Zou, "Detecting HTTP Botnet with Clustering Network Traffic," 2012.
- [7] F. V. Alejandro, N. C. Cortés and E. A. Anaya, "Feature selection to detect botnets using machine learning algorithms," in *in Electronics, Communications and Computers (CONIELE- COMP)*, 2017, 2017.
- [8] J. Pei, Y. Chen and W. Ji, "A DDoS Attack Detection Method Based on Machine Learning," *Journal of Physics*, vol. 032040, no. 1237, 2019.
- [9] E. Samani, H. H. Jazi and A. Ghorbani, "Towards effective feature selection in machine learning-based botnet detection approaches," in *Communications and Network Security*, 2014.
- [10] University of New Brunswick, "Intrusion Detection Evaluation Dataset (CIC-IDS2017)," 2017. [Online]. Available: <https://www.unb.ca/cic/datasets/ids-2017.html>. [Accessed 01 12 2021].
- [11] H. Lashkari, "CICFlowMeter/ReadMe.txt at master · ahlashkari/CICFlowMeter.," 2021. [Online]. Available: <https://github.com/ahlashkari/CICFlowMeter/blob/master/ReadMe.txt>. [Accessed 12 December 2021].
- [12] N. Sharma, A. Mahajan and V. Mansotra, "Identification and analysis of DoS attack Using Data Analysis tools," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 4, no. 6, pp. 11368-11375, 2016.
- [13] N. Sharma, A. Mahajan and V. Mansotra, "Machine Learning Techniques used for the Detection and Analysis of Modern Types of DDoS Attacks," *International Research Journal of Engineering and Technology (IRJET)*, vol. 4, no. 6, pp. 1086-1092, 2017.
- [14] H. Jiang, "Machine Learning Fundamentals A Concise Introduction.," Toronto, Cambridge University, 2021, pp. 111-112.
- [15] xgboost developers., "XGBoost Documentation," 2021. [Online]. Available: <https://xgboost.readthedocs.io/en/stable/>. [Accessed 07 12 2021].

# A study on the application of mission-based cybersecurity testing and evaluation of weapon systems

Ikjae Kim  
Dept of Computer Engineering,  
Sejong University  
R.O.K Cyber Operations Command  
Seoul, South Korea  
kij397@mnd.go.kr

Hansung Kim  
R.O.K Cyber Operations Command  
Seoul, South Korea  
khs4284@mnd.go.kr

Dongkyoo Shin\*  
Dept of Computer Engineering,  
Sejong University  
209 Neungdong-ro, Gwangjin-gu,  
05006  
Seoul, South Korea  
shindk@sejong.ac.kr  
\*Corresponding author: Dongkyoo Shin

**Abstract**— In this paper, we investigate the ongoing research on ways to improve cybersecurity throughout the life cycle of weapon systems applied in advanced countries such as the United States(U.S.), and present effective security evaluation measures by analyzing restrictions on acquiring weapon systems in Republic of Korea. We consistently performed mission-based risk assessment in cybersecurity tests and evaluation plans at the entire stage to support decision-making by providing key information to major decision-making organizations in a timely manner. We propose a plan to carry out simulated penetration by establishing rules of engagement so that protection measures can be verified for vulnerabilities identified in terms of cybersecurity. In addition, we identified the areas where artificial intelligence can be applied to the proposed cybersecurity test and evaluation system, and suggested future development plans. Through this, we supplemented our ability to support major decisions by integrating mission-based risk assessment factors into the cybersecurity test and evaluation system research conducted so far to identify risks in a timely manner between acquisition projects.

**Keywords**—mission, cybersecurity, test and evaluation

## I. INTRODUCTION

With the rapid development of information technology (IT) around the world, the use of software is increasing in many parts of society throughout the world. Software is increasing in all fields such as home appliances and automobiles, and the same is true of the defense weapon systems. This increase in the proportion of such software in the field of defense weapon systems helps to implement the excellent performance, but vulnerabilities may be inherent in the software, and the possibility of exposure to cyber threats using these vulnerabilities increases. In particular, since South Korea has a higher software utilization rate than North Korea, North Korea is using this asymmetry to openly attack cyberattacks, which is a factor that can cause serious damage to the allies in a war situation.

In order to respond to such cyber-attacks, research to remove security vulnerabilities in software is increasing, and in particular, research on software development security to remove factors that cause security vulnerabilities in advance in the software development stage is being actively

conducted.

The U.S. applies the cybersecurity test and evaluation system to efficiently identify and eliminate vulnerabilities throughout the weapon system acquisition stage to safely deploy it, and operates it by integrating it with a cybersecurity system called Risk Management Framework (RMF) are doing.

In this regard, the paper [1] proposed the introduction of a “weapon acquisition cybersecurity test and evaluation system” that applies the US cybersecurity test and evaluation to the defense weapon system acquisition stage. In this paper, we supplement and propose procedures to enable the identification and management of cyber-related risks by performing mission-based risk assessment in addition to the identification of technical and management vulnerabilities in the “Weapon Acquisition Cybersecurity Test and Evaluation System”.

Following the introduction, this paper examines the related work in Chapter 2, the limitations of the “Weapon Acquisition Cybersecurity Test and Evaluation System” proposed in [1] in Chapter 3, and combines the mission-based risk assessment in Chapter 4. Suggest ways to improve the security evaluation system. Chapter 5 identifies the areas where artificial intelligence can be applied to the proposed cybersecurity test and evaluation system and presents the application plan.

## II. RELATED WORK

### A. U.S. military cybersecurity test and evaluation

The United States government defines cybersecurity as preventing, protecting, and restoring damage to computers, electronic communications systems, electronic communications services, wireline communications, electronic communications, and information contained therein to ensure availability, integrity, authentication, confidentiality, and non-repudiation [2].

Test and evaluation is a compound word of test and evaluation as a field in the weapon system acquisition stage. The main purpose of test and evaluation is to provide timely information necessary for decision-making to policy makers [3]. Test and evaluation verifies whether the target weapon system conforms to the user's requirements and meets the operational capability by obtaining basic data for verifying and evaluating the objective performance of the weapon

---

This work was supported by the National Research Foundation of Korea through the Basic Science Research Program, Ministry of Education, under Grant 2018R1D1A1B07047395.

system and comparing and analyzing it with preset test standards through various tests. to judge suitability. Through this, it is the decision-making support stage to determine whether the purchase, R&D, design, and manufacture of weapons systems meet the requirements of the military [1].

The U.S. guarantees the rationality and efficiency of test evaluation by defining NIST SP800-related reference documents related to cybersecurity by the National Institute of Standards and Technology (NIST) to apply the same criteria and share evaluation results at the national level for rational cybersecurity test and evaluation [1][4][5][6][7][8][9].

The U.S. Department of Defense incorporates a risk management framework (RMF) and a cybersecurity test and evaluation system into the defense acquisition system to effectively realize cybersecurity in the entire life cycle of weapons systems [1][3][10][11][12].

The U.S. is carrying out a six-step cybersecurity test and evaluation process to implement cybersecurity throughout the entire life cycle of a weapon system. This process proceeds continuously from the initial required analysis stage to the final electrification stage during the weapon system acquisition stage, and is performed by integrating with the system engineering and RMF process processes. This cybersecurity test and evaluation process consists of six steps and is as follows [13].

- Step 1, Understanding cybersecurity requirements
- Step 2, Identifying the cyber-attack surface
- Step 3, Identifying vulnerabilities through collaboration
- Step 4, Adversary cybersecurity development test and evaluation
- Step 5, Vulnerability assessment and penetration assessment through collaboration
- Step 6, Hostile evaluation

#### B. Korea's cybersecurity test and evaluation

The acquisition of weapons systems for the Republic of Korea (ROK) military is defined in the 'Defense Forces Power Generation Work Order' [14], which defines guidelines for the overall life cycle, including the requirement, acquisition, operation and maintenance of weapons systems. In Article 52 of the Ordinance (Weapon System Research and Development), the weapons system research and development process is divided into the exploration and development stage, the system development stage and the mass production stage. In particular, the following detailed guidelines are provided for cybersecurity in the weapon system introduction stage.

- Article 52 (Research and Development of Weapon Systems) ⑤ In the case of research and development of weapons systems, the following activities are carried out in relation to security.
  - 1.The Defense Acquisition Program Administration (DAPA) submits the search and development result report including the results of the security support company's review of the protection measures for the information system of the weapon system.

- 2.The DAPA requests the security support company to review the protection measures for the information system of the weapon system before starting the system development, and reflects the review results in the system development plan.

- 3.In the system development stage, the DAPA requests the security support company to review the protection measures for the built-in SW, and when a change in the development plan is required, such as when a change in operational performance is required or when jointness and interoperability are affected, the Ministry of National Defense (Informatization Planning Office), the Joint Chiefs of Staff, and the armed forces should be consulted in advance, and re-requested to the Security Support Agency to review the protection measures for the weapon system information system and embedded SW.

In addition, in relation to Articles 25 and 26 of the 'Defense Cybersecurity Ordinance' of Korea, the security support officer reviews the protection measures for the information system and embedded SW of the weapon system during the weapon system search and development and system development stage, and transfers weapons system test and evaluation, etc. to full power. In this step, security measures are performed [14].

As for the cyber security test and evaluation items of the ROK military, the details of interoperability test and evaluation in Article 81 (Classification and Method of Test and Evaluation) of the Defense Power Generation Work Order are set to follow the Defense Interoperability Management Directive. Test and evaluation items related to information assurance and cyberthreat response are included [14].

In the Power Generation Work Order, the responsibility for compiling the interoperability field is defined as the Joint Interoperability Technology Center. In the development test and evaluation, software reliability test and information protection are evaluated, and in the operation test and evaluation, only information protection is evaluated.

Information protection test and evaluation items for weapon systems specified in the Defense Interoperability Management Directive include information protection level, network information protection, control system establishment, key management system establishment, application system, server, terminal, encryption equipment application, cyber threat response capability, It is divided into SW vulnerability removal, and details are shown in "Table I." [15].

#### C. Cyber security test and evaluation system for weapon system

##### D.

The "weapon acquisition cybersecurity test and evaluation system" proposed in [1] proposes the advantage of systematically conducting cybersecurity test and evaluation from the early stage of weapon systems applied in the United States as ROK domestic cybersecurity process. The application of cyber security within the weapon system has been improved by increasing connectivity. The article referenced in [1] divides the cybersecurity stage in the defense acquisition system into four stages and suggests the

process to be performed at each stage. In particular, the process of actively identifying and removing vulnerabilities was added by applying vulnerability analysis, evaluation and simulated penetration in the development/operation test and evaluation stage. This is to strengthen cybersecurity by extending vulnerability analysis and evaluation and simulated penetration, which are currently applied only in the operational stage in the defense field, to the acquisition stage.

The “weapon acquisition cybersecurity test and evaluation system” in [1] is divided into stage 1 cybersecurity requirements identification, stage 2 cyber - attack surface identification, stage 3 cybersecurity development test and evaluation, and stage 4 cybersecurity operation test and evaluation. Is as follows.

- Stage 1, Identifying Cybersecurity Requirements: Develop an initial approach and plan to identify cybersecurity requirements by looking at all target system-related documentation and conduct cybersecurity testing and evaluation
- Stage 2, Identifying of the cyber-attack surface: Identifies the attack path that an attacker can access to the target system's network, hardware, firmware, physical interface, software, etc., and identifies possible vulnerabilities in that path
- Stage 3, cybersecurity development test and evaluation: Perform test and evaluation of the target system using the vulnerability analysis/evaluation report, security evaluation report, and development test and evaluation output
- Stage 4, cybersecurity operation test and evaluation: Perform vulnerability penetration test from the attacker's perspective by referring to the vulnerability analysis/evaluation report, cybersecurity development test and evaluation output, etc. and evaluate the cybersecurity level of the target system

TABLE I. ROK MILITARY WEAPON SYSTEM INFORMATION SECURITY TEST EVALUATION ITEMS

Evaluation items	
Information protection level	
Network information protection	
Establishment of control system	
Establishment of key management system	
Application system information protection	
Server information protection	
Terminal information protection	
Encryption equipment application	
Cyber Threat Response Capability	Cyber Threat Response Capability
	Ability to respond to identity-disguised threats
	Ability to respond to data tampering threats
	Denial of aggression threat response capability
	Ability to respond to threats of information leakage
	Denial of Service (DoS) Threat Response Capability
SW Vulnerability Removal	Ability to respond to elevation of privilege threats
	SW Vulnerability Removal
	Appropriateness of applying secure coding rules
Relevance of removing open-source vulnerabilities	

### III. LIMITATIONS OF “WEAPON ACQUISITION CYBERSECURITY TEST AND EVALUATION SYSTEM”

The “weapon acquisition cybersecurity test and evaluation system” proposed in [1] consists of four stages. Each stage identifies cybersecurity requirements by examining all target system-related documents, and identifies the target system's network, server, firmware, and physical Identifies the attack surface using interfaces, etc.

In the cybersecurity development and operation test and evaluation stage, the level of cybersecurity is evaluated by using the vulnerability analysis/evaluation report and development test and evaluation output, and by adding the technical vulnerability penetration test to it. As a result of these activities, technical and managerial weaknesses can be derived, but information on risks arising from the identified weaknesses to project managers is not considered.

The purpose of test and evaluation is to provide information for key decision-making to decision-making organizations in a timely manner.

To this end, it is necessary to supplement the mission-based risk assessment process that can identify and manage risks by identifying cyber-dependent missions based on the mission performed by the target weapon system and deriving major cyber threats related thereto.

### IV. PROPOSED MISSION-BASED CYBERSECURITY TEST AND EVALUATION

In this chapter, we propose a mission-based cybersecurity test and evaluation procedure suitable for the acquisition stage of the ROK military weapon system. The proposed mission-based cybersecurity test and evaluation is performed as shown in “Table II.”.

TABLE II. CLASSIFICATION OF CYBERSECURITY TEST AND EVALUATION STAGES

Division	Weapon system acquisition	Mission-based cybersecurity test and evaluation
Step 1	Understanding cybersecurity requirements	Risk/threat modeling
Step 2	System development	Attack Surface Listing
Step 3	Development test evaluation	Attack surface vulnerabilities analysis and evaluation
Step 4	Operational test evaluation	Analysis and evaluation of simulated penetration and vulnerability based on rules of engagement

#### A. Risk/threat modeling

Step 1, risk/threat modeling is performed in the cybersecurity requirements identification step in weapon system acquisition stage, and initial risk/threat modeling

based on power requirements is performed for mission-based risk assessment.

For the mission-based risk assessment, the cyber-dependent mission is identified, the detailed functions for the mission are subdivided, and the diagram is shown in “Fig. 1”.

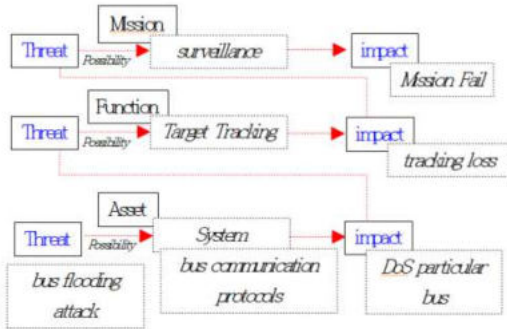


Fig. 1. Risk/threat modeling as an example

### B. Attack Surface Listing

Step 2, attack surface inventorying is performed in the system development step, and risk/threat modeling is performed again based on supplementary documents such as RFP and the attack surface is listed.

Identification of the attack surface establishes a cyber boundary around the weapon system, identifies the entry point that approaches the system from the outside through this attack surface, and divides and expresses the system nodes from the assets entering the system step by step. Currently, the node with the vulnerability becomes the main node that becomes the attack target. If this is expressed as a figure, it is shown in “Fig. 2”.

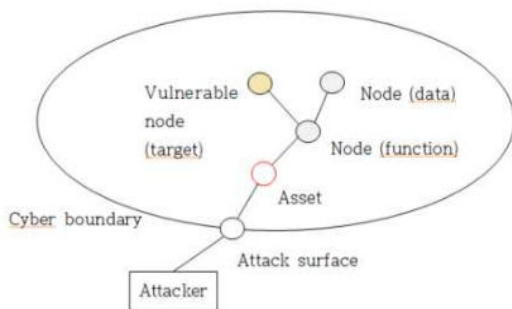


Fig. 2. Identification of the attack surface of a weapon system

The identified attack surface can be schematized as shown in “Fig. 3” by identifying sub-functions, data, and assets from the mission of the weapon system. This allows the attack surface to connect which assets, which data, which functions, and which missions it ultimately wants to achieve.

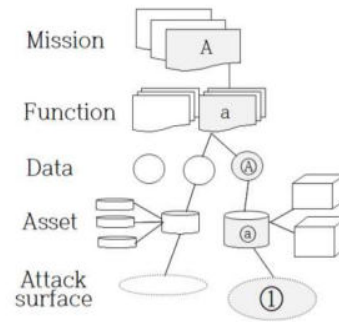


Fig. 3. Identify the attack surface of cyber-dependent missions

### C. Attack surface vulnerability analysis and evaluation

Step 3, attack surface vulnerability analysis and evaluation is performed in the development test and evaluation step, and risk/threat modeling is performed again based on the development document, and vulnerability analysis and evaluation of the attack surface is performed.

By performing vulnerability analysis and evaluation on the attack surface, vulnerable assets are identified in the asset layer, and related data and functions are identified. Based on this, weak missions can be identified.

Using this to supplement the threat scenario, “Asset (a) has identified vulnerability (1), and using it, assets (b) and (c) can be seized, and related data (A) can be stolen or altered. This limits function a and consequently cannot perform mission A”.

Accordingly, the project manager can take measures to mitigate risks by devising protection measures for vulnerabilities.

“Fig. 4” derives protection measures to mitigate vulnerability(1) for vulnerable asset (a), and the derived protection measures are indicated by a blue dotted line on the border of asset (a).

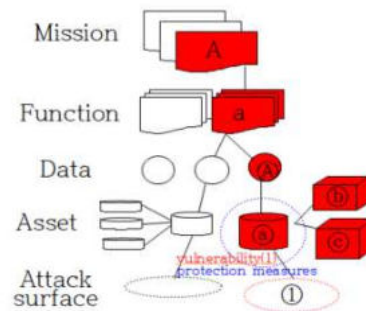


Fig. 4. Deriving protection measures for vulnerable assets

The vulnerabilities supplemented by these protection measures can mitigate the risk of the entire weapon system, and by taking protection measures for asset (a), assets (b) and (c) were returned to their normal state as shown in “Fig. 5”. As a result, data (A), function a, and task A all show a normal state.

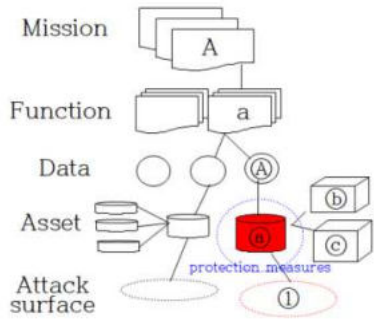


Fig. 5. Mitigation of vulnerabilities through protection measures

Based on these measures, a mission-based risk assessment can be performed, and the results are reported to the decision-making organization and provided for quick decision making, thereby facilitating the communication of the unplanned acquisition process.

In particular, if it takes a long time to acquire a weapon system, it is realistically limited to completely eliminate all threats that develop rapidly within a limited project period and within the project budget. Therefore, it is necessary to manage risk through this process.

In this way, all possible threat scenarios for each attack surface can be set as rules of engagement for simulated infiltration, and become a standard to verify the effectiveness of protection measures for each asset.

#### D. Simulated penetration based on rules of engagement and vulnerability analysis / evaluation

Step 4, the simulation penetration based on rules of engagement and vulnerability analysis/evaluation is performed in the operational test and evaluation step, and simulated penetration is performed using the threat scenario for each attack surface identified above as the rules of engagement.

The threat scenario is to neutralize the protection measures of the blue dotted line through the attack surface ①, and to perform a simulated penetration so that the mission A cannot be performed using the vulnerability (1) of the asset a.

Through this, using the vulnerability of the initially identified attack surface, simulated penetration is performed following the threat scenario to verify the effectiveness of the devised security measures. It verifies the risk mitigation status of assets, data, functions, and missions through simulated penetration based on rules of engagement and performs mission-based risk assessment based on the results. At this time, if vulnerabilities are continuously identified, security measures are supplemented and re-verified through retesting to ensure a safe state before electrification.

#### V. "MISSION-BASED CYBERSECURITY TEST AND EVALUATION" AND FIELDS OF APPLICATION OF ARTIFICIAL INTELLIGENCE TECHNOLOGY

The "Mission-based cybersecurity test and evaluation" is a proposal to strengthen cybersecurity by combining the US cybersecurity process with ROK domestic weapon system

test and evaluation from a cybersecurity point of view. In addition, it emphasized the implementation of reinforced cybersecurity by setting the rules of engagement as a guideline for conducting test and evaluation and supplementing the protection measures of the weapon system through retesting if the response was insufficient.

In this paper, iterative risk/threat modeling is performed step by step by adding risk assessment to the domestic weapon systems test and evaluation, and vulnerability analysis and evaluation based on mission-based threat scenarios. It was proposed by improving the system that can verify the effectiveness of security measures in the operational test and evaluation stage by executing simulated penetration under the rules of engagement and settling.

"Table III." compares the test and evaluation of domestic weapon system and the proposed method from the perspective of cyber security.

TABLE III. COMPARISON WITH DOMESTIC WEAPON SYSTEM TEST AND EVALUATION

Division	Domestic weapon system test and evaluation	Mission-based cybersecurity test and evaluation
Threat Scenarios	None	O
Risk Assessment	None	O
Vulnerability analysis and evaluation	1 time	3 times
Simulated Penetration	None	1 time

If artificial intelligence technology, which is being actively researched, is applied to the mission-based cybersecurity test and evaluation of the weapon system proposed in this paper, the level of cybersecurity activity will increase.

In particular, with the advancement of the weapon system, the vast attack surface is continuously increasing, and numerous networks are formed between the asset nodes constituting the weapon system. Therefore, it is necessary to check all cases by applying artificial intelligence technology including machine learning and deep learning rather than activities by professional personnel to identify all paths available for attack among the networks between all asset nodes centering on this attack surface and perform simulated penetration [16][17].

The use of artificial intelligence technology can evaluate and predict all possible penetration paths for an attack, and it will be possible to derive priorities for establishing security measures according to the evaluation results, thereby supporting the timely decision of policy makers.

It will be possible to implement more effective protection measures if a series of procedures for identifying threat scenarios and performing simulated infiltration against the

vast attack surface of increasingly complex weapon systems are made with artificial intelligence technology.

## VI. CONCLUSION

In this paper, based on the cybersecurity test and evaluation of the United States, the mission-based risk assessment is consistently performed in the cybersecurity test and evaluation method suitable for ROK domestic situation studied earlier, and important information is provided to major decision-making organizations in a timely manner. To support decision-making and to verify protection measures against identified vulnerabilities in terms of cybersecurity, it is proposed to set up rules of engagement to conduct simulated infiltration.

This can facilitate communication between related organizations throughout the acquisition phase, and improve the cybersecurity capabilities of acquired weapon systems by improving vulnerabilities through information sharing, verifying their effectiveness, and institutionalizing them so that they can be retested if insufficient.

In particular, if AI technology is used for vulnerability analysis and evaluation and simulated penetration in the mission-based weapon system cybersecurity test evaluation, it is judged that cybersecurity will be able to develop dramatically through evaluation and verification of all possible attackable weapon system access paths.

Considering that efforts to strengthen cybersecurity activities and manage risks throughout the entire life cycle of weapon systems pursued in the United States and advanced countries are continuing, this paper is one way to evaluate cybersecurity tests applicable to domestic weapon systems. It can be said that it contributed to the specification of the methodology.

## REFERENCES

- [1] Ji-seop Lee, Sung-yong Cha, Seung-soo Baek, Seung-joo Kim, "Research for Construction Cybersecurity Test and Evaluation of Weapon System," *Journal of The Korea Institute of Information Security & Cryptology* Vol.28, No.3, Jun. 2018. <https://doi.org/10.13089/JKIIISC.2018.28.3.765>
- [2] THE WHITE HOUSE WASHINGTON, "National Security Presidential Directive-54/Homeland Security Presidential Directive-23," January 8, 2008.
- [3] Jong Wan Park, "The Action of the Reliability Enhancement in Test and Evaluation of the Weapon Systems," *Journal of Applied Reliability* Vol. 15-2, pp. 108-123, 2015.
- [4] Congressional Research Service, "Defense Acquisitions: How DOD Acquires Weapon Systems and Recent Efforts to Reform the Process," May 23, 2014.
- [5] "Guide for Conducting Risk Assessment," NIST SP 800-30 Rev.1. 2012.
- [6] "Guide for Applying the Risk Management Framework to Federal Information systems," NIST SP 800-37 Rev.1. 2010.
- [7] "Managing Information Security Risk," NIST SP 800-39, 2011.
- [8] "Security & Privacy Controls for Federal Information Systems and Organizations," NIST SP 800-53 Rev.4, 2013.
- [9] "Guide for Assessing the Security Controls in Federal Information Systems and Organizations," NIST SP 800-53A Rev.1, 2010.
- [10] "Cybersecurity & Acquisition Lifecycle Integration Tool(CALIT)," DAU ver 3.1, sep 2018.
- [11] Hyun-suk Cho, Sung-yong Cha, Seung-joo Kim, "A Case Study on the Application of RMF to Domestic Weapon System," *Journal of The Korea Institute of Information Security & Cryptology* Vol.29, No.6, Dec.2019.
- [12] Sungyong Cha, Seungss Baek, Sooyoung Kang and Seungjoo Kim, "Security Evaluation Framework for Military IoT Devices," *Security and Communication Networks*. Vol. 2018, Article ID 6135845, 12 pages, Jul. 2018.
- [13] Department of Defense, "Cybersecurity Test and Evaluation Guidebook," 2015.
- [14] "National Defense Power Generation Business Instruction," Ordinance of the Ministry of National Defense, 2021.
- [15] "Defense Interoperability Management Directive," Ministry of National Defense, Jan. 2021.
- [16] Philip W. Shin, Jack Sampson, V Narayanan "Context-Aware Collaborative Object Recognition For Distributed Multi Camera Time Series Data," in *Tenth International Symposium on Information and Communication Technology (SoICT) Dec. 2019*, Hanoi - Halong Bay, Vietnam, pp. 154-161. <https://doi.org/10.1145/3368926.3369666>
- [17] Philip W. Shin, Jinhee Lee, Seung Ho Hwang, "Data Governance on Business/Data Dictionary using Machine Learning and Statistics," in *International Conference on Artificial Intelligence in Information and Communication (ICAIIIC) 2020*, Fukuoka, Japan, pp.547-552. <https://doi.org/10.1109/ICAIIIC48513.2020.9065194>



# Grey Wolf Optimizer-Based Automatic Focusing for High Magnification Systems

Islam Helmy  
Dept. of Computer Engineering  
Chosun University  
Gwangju, South Korea  
islam.helmy@chosun.kr

Wooyeol Choi  
Dept. of Computer Engineering  
Chosun University  
Gwangju, South Korea  
wyc@chosun.ac.kr

**Abstract**—High-quality astronomical images significantly affect the results of astronomical scientific research. Since the accurate focus is one of the principal factors that vary the observation quality, automatic focusing is essential. The automatic focusing finds the best position by measuring the images' focus level at different positions and considering the best one. It applies a focus measure operator to evaluate the focus level. However, the astronomical images suffer from high blur due to the high magnification of the telescope. Thus, a proper focus measure is tricky due to a high blur. In this paper, we firstly investigate a focus operator based on fuzzy logic because of its ability to deal with imprecise data. We secondly optimized the parameters of the membership functions using the grey wolf optimizer (GWO). We acquired two data sets using the 74-inches telescope of the Kottamia astronomical observatory (KAO) at good seeing conditions and composed of in-focus and out-of-focus images. After that, we compare the measures using four criteria. The results show that the optimized one outperforms the other operators.

**Index Terms**—Grey wolf optimizer, fuzzy logic, focus measure, astronomical images

## I. INTRODUCTION

The precise automatic focusing for high magnification systems like the telescope is principle due to the significant effect on the observation quality, which is principal for astronomical scientific research. The automatic focusing systems search the positions until it finds the best one which has the highest focus level. In the best one, the maximum amount of the light converges, while in out-of-focus, the light rays are spreading on a large area on the charge-coupled device (CCD) resulting, in the loss of faint celestial objects and improper image for scientific research. The automatic focusing estimates the focus level by applying a focus level operator into the image.

Various focus measure operators (FM) are investigated for automatic focusing. The focus measures are traditionally classified into five main categories [1], which are Gradient-based, Auto-correlation-Based, Statistical-Based, Transform-Based, and Edge-Based. However, a proper focus measure for high-magnificent images like astronomy images is tricky due to a high blur. A focus measure based on fuzzy logic [2] obtains attention because of the ability of fuzzy to deal with

imprecise data. It is based on applying fuzzy logic into the images then calculating a statistical called Modified Histogram as a focus measure. In this research, we firstly investigate the focus measure based on fuzzy logic on astronomical images. Besides, we compare the Modified Histogram focus measure with other traditional operators on two data sets of astronomical images. We acquired two star-clusters sequences using the 74-inches telescope of the Kottamia astronomical observatory (KAO) at good seeing conditions.

It is well-known that the parameters' values of the membership functions are selected depending on experience. As a result, an optimization technique for optimizing the fuzzy parameters has been widely used to ensure better performance [3], [4]. In literature, several optimization techniques have been introduced, such as genetic algorithm (GA) [5], ant colony optimization (ACO) [6], particle swarm optimization (PSO) [7], and so on. However, the grey wolf optimizer (GWO) [8] achieves a competitive result. The GWO algorithm imitates the leadership structure and hunting ways of grey wolves in nature. Four types of wolves, alpha, beta, delta, and omega, are manipulated to mimic the leadership hierarchy. They are hunting by firstly searching for prey, then they surround it for the attack.

In this context, we secondly propose an optimization of the fuzzy parameters using GWO. The remainder of this paper is organized as follows. Section II presents the common focus measure operators. In Section III, the description of the modified histogram focus measure is introduced. The methodology is presented in Section IV. Finally, the analysis of experimental results and conclusions are presented in Section V and VI, respectively.

## II. FOCUS MEASURE OPERATORS

In the state-of-the-art, several comparative studies, including various focus measures, have been introduced [1], [9], [10]. In this work, we investigate several existing focus measures, such as absolute gradient [11], squared gradient [12], Tenengrad [13], Brenner [15], auto-correlation [16], amplitude [17], histogram [18], discrete cosine transform (DCT) [19], and fast Fourier transform (FFT) [20].

This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. NRF-2021R111A3050535).

### A. Absolute gradient

The absolute gradient [11], which is also known as sum modulus difference (SMD), is the sum of absolute of image gradients, horizontal and vertical, along image  $M$  rows and  $N$  columns in Eq. (1), where FM is the focus measure and  $I(x, y)$  is the image intensity at pixel located at  $(x, y)$ . The horizontal gradient is the intensity differences between neighboring pixels along image rows. However, the vertical gradient is the intensity differences between neighboring pixels along image columns.

$$\text{FM} = \sum_{x=1}^M \sum_{y=1}^N (|I(x, y+1) - I(x, y)| + |I(x+1, y) - I(x, y)|). \quad (1)$$

### B. Squared gradient

The squared gradient [12] is the sum of squares of gradients, horizontal and vertical. It is computed as shown in Eq. (2). Indeed the absolute gradient is similar to the squared gradient. However, the larger gradients have more influence on the squared gradient than the absolute gradient. Whereas, the absolute gradient is computed much faster than the squared gradient measure.

$$\text{FM} = \sum_{x=1}^M \sum_{y=1}^N (|I(x, y+1) - I(x, y)|^2 + |I(x+1, y) - I(x, y)|^2). \quad (2)$$

### C. Tenengrad

Tenenbaum and Schlag are proposed focus measure similar to the squared gradient called Tenengrad [13]. However, the horizontal and vertical gradients are obtained using the Sobel operator [14]. The Tenengrad focus measure can be expressed as

$$\text{FM} = \sum_{x=1}^M \sum_{y=1}^N (I_x^2 + I_y^2), \quad (3)$$

where  $I_x(x, y)$  and  $I_y(x, y)$  are the horizontal and vertical gradient at pixel located at  $(x, y)$ , respectively, and can be given by

$$\begin{aligned} I_x(x, y) &= I(x, y) \times g_x(x, y), \\ I_y(x, y) &= I(x, y) \times g_y(x, y), \\ g_x &= \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, \quad g_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}. \end{aligned}$$

### D. Brenner

Another measure was developed by Brenner, which was devised for automated microscopy. The author [15] notes that the differences between the image pixels and their neighbors that are located two pixels away increase as the focus increases. The focus measure can be expressed by

$$\text{FM} = \sum_{x=1}^M \sum_{y=1}^N |I(x, y) - I(x+2, y)|^2. \quad (4)$$

### E. Auto-correlation

The correlation family is based on the evaluation of neighboring pixels' dependency. The image auto-correlation is expected to contain sharpness information. The auto-correlation [16] focus measure is computed as presented in Eq. (5).

$$\text{FM} = \sum_{x=1}^{M-1} \sum_{y=1}^N I(x, y)I(x+1, y) - \sum_{x=1}^{M-2} \sum_{y=1}^N I(x, y)I(x+2, y). \quad (5)$$

### F. Amplitude

The sum of the absolute of differences between image intensities and image mean ( $\bar{I}$ ) is suggested in the amplitude focus measure, which is also known as absolute central moment [17]. Eq. (6) and (7) show the Amplitude focus measure and the image mean.

$$\text{FM} = \sum_{x=1}^M \sum_{y=1}^N |I(x, y) - \bar{I}|, \quad (6)$$

$$\bar{I} = \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N I(x, y). \quad (7)$$

### G. Histogram

The histogram focus measure is proposed in [18]. It is defined as the difference between the maximum and minimum grey levels in Eq. (8).

$$\text{FM} = \max \{k | P_k > 0\} - \min \{k | P_k > 0\}, \quad (8)$$

where  $k$  is the grey level and  $P_k$  is the relative (normalized) frequency.

### H. Discrete cosine transform

In [19], the DCT is used to transform the image into the DCT domain, and the average of alternative current (AC) components  $E_{AC}(x, y)$  is used as an indication of the focus measure. It is equivalent to the image variance and can be calculated as

$$\text{FM} = \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N E_{AC}(x, y), \quad (9)$$

where

$$\begin{aligned} E_{AC}(x, y) &= \left( \sum_{u=1}^8 \sum_{v=1}^8 I(u, v)^2 \right) - I(1, 1)^2, \\ I(u, v) &= \frac{4c(u)c(v)}{8^2} \sum_{i=1}^8 \sum_{j=1}^8 \left[ I(i, j) \times \right. \\ &\quad \left. \cos \left( \frac{(2i+1)u\pi}{2 \times 8} \right) \cos \left( \frac{(2j+1)v\pi}{2 \times 8} \right) \right], \\ c(u) &= \begin{cases} \frac{1}{\sqrt{2}}, & \text{if } u = 1 \\ 1, & \text{if } u > 1 \end{cases}. \end{aligned}$$

### I. Fast Fourier transform

A focus measure based on the FFT has been suggested in [20]. The idea in using this criterion is that sharp edges have high spatial frequencies. So measuring frequencies provides an image focus level. The focus measure using the FFT is defined as the sum of absolute of product of image magnitude spectrum ( $\text{Mag}(u, v)$ ) and phase ( $\text{Ang}(u, v)$ ), and can be expressed as

$$\text{FM} = \sum |\text{Mag}(u, v) \times \text{Ang}(u, v)|. \quad (10)$$

### III. MODIFIED HISTOGRAM FOCUS MEASURE

In [2], the authors propose a focus measure called modified histogram. It is based on firstly applying fuzzy logic to the image, then the focus measure is estimated from the output of the fuzzy logic (fuzzy image). The modified histogram is a modification of the Histogram focus measure, which has the advantage of simplicity and fast computation. The basic idea of the histogram is that as the image goes focus, its maximum increases, which consequently increases the difference between the maximum and minimum. Indeed, the image minimum is assumed to be fixed since it is affected by the sky background. The maximum may not be increased due to the effects of the observation conditions like temperature, cloud, and humidity. Consequently, the authors substitute the minimum with the median of maximum, and the focus measure can be expressed by

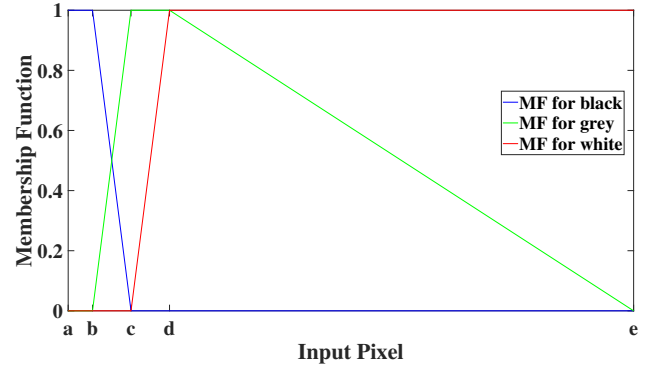
$$\text{FM} = \max \{k | P_k > 0\} - \text{med} \left\{ \max_{(x,y) \in S_i} I(x, y) \right\}, \quad (11)$$

$$i = 1, 2, 3, \dots, N_S,$$

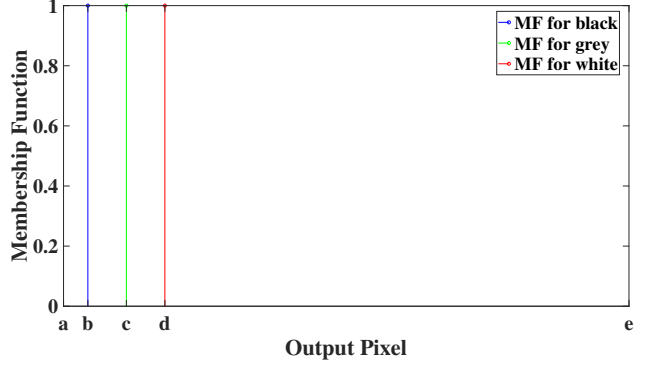
where  $N_S$  is the number of samples. Samples  $S_i$  from the image are randomly selected, then the maximum of samples is found, and the median of those values is calculated.

#### A. Fuzzy logic

The fuzzy logic is a single-input-single-output system. The linguistic values are black, grey, and white. The adopted membership functions (MF) of the input and the output are trapezoids and singletons, as shown in Fig. 1a and 1b, respectively. **a** is the image minimum, **b** is the mean of minimum, **c** is the median of maximum, **d** is the mean of maximum, and **e** is the image maximum. The mean of minimum and maximum is computed similarly to the median of maximum, except the mean is used. The fuzzy rules are as follows. If the input is dark, make it darker. If it is bright, make it brighter. Finally, if it is grey, make it grey. In conclusion, the defuzzification is calculated by the center of gravity method (CoG) [21].



(a) Input membership function



(b) Output membership function

Fig. 1: Input and output membership functions.

$$\mathbf{a} = \min_{(x,y) \in (M,N)} I(x, y),$$

$$\mathbf{b} = \frac{1}{N_S} \sum_{i=1}^{N_S} \min_{(x,y) \in S_i} I(x, y),$$

$$\mathbf{c} = \text{med} \left\{ \max_{(x,y) \in S_i} I(x, y) \right\}, i = 1, 2, 3, \dots, N_S,$$

$$\mathbf{d} = \frac{1}{N_S} \sum_{i=1}^{N_S} \max_{(x,y) \in S_i} I(x, y),$$

$$\mathbf{e} = \max_{(x,y) \in (M,N)} I(x, y).$$

#### B. Grey wolf optimizer

Grey wolf belongs to the family Canidae, and they are considered apex predators [8]. The leadership hierarchy of the grey wolves is manipulated into four types, alpha, beta, delta, and omega. The alpha is liable for making decisions about hunting, accommodation, and wake time. The beta is the second level in the hierarchy, and they play the role of advisor to the alpha in decision-making. The omega is the lowest rank grey wolf, and they always have to capitulate to all the other powerful wolves. However, the omegas seem insignificant internal problems have been observed in the case of losing the omegas. Finally, the delta is a wolf who is not

an alpha, beta, or omega, and they have to obey alphas and betas; however, they control the omegas.

The grey wolves search for prey by making use of the position of the alpha, beta, and delta. In the beginning, they diverge from each other to find it, then they converge to attack it. The GWO algorithm runs as follows. Firstly, a random population of grey wolves (candidate solutions) is initiated. Throughout iterations, alpha, beta, and delta wolves estimate the probable position of the prey, then each wolf (solution) updates its distance from it. Eventually, the GWO algorithm is stopped by an end criterion. In this context, we use the GWO to optimize the parameters of the fuzzy membership functions, which are **b**, **c**, and **d**, to maximize the fitness function modified histogram focus measure. The search interval for each parameter is  $\mathbf{a} \leq \mathbf{b} \leq \mathbf{c}$ ,  $\mathbf{b} \leq \mathbf{c} \leq \mathbf{d}$ , and  $\mathbf{c} \leq \mathbf{d} \leq \mathbf{e}$ .

#### IV. METHODOLOGY

In this paper, we firstly investigate the focus measure based on the modified histogram on two sequences of star-clusters observations, which are M103 and N7067. Then, we optimize the parameters of the fuzzy membership functions using the GWO. The data sets are acquired using the 74-inches telescope of the KAO [23] shown in Fig. 2. Each sequence contains in-focus and out-of-focus images of size 2K. The M103 and N7067 sequences consist of 66 and 55 images, respectively. In addition, we use an exposure time of 60 (sec/frame).

The modified histogram and the proposed optimized modified histogram are compared to the operators mentioned in section II. In literature, various ranking criteria are used. In [22], they are ranked according to four criteria as follows.

- Accuracy: This determines how the operator is distant from the best position. It is the distance between the operator's best position and the data set's best one.
- Range: This measures the distance between two neighboring local minimums around the global maximum. In other words, it describes the region where exists no local maximums.
- The number of false maximum: This counts the spurious maximum arising in a focus function.
- Width: This indicates the rate of change of the focus function. It is the width of the focus function at half of the height.

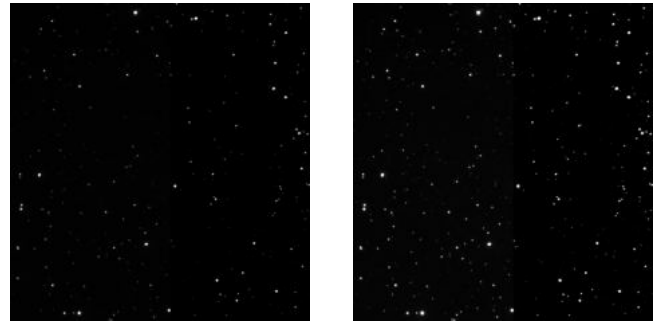
The operator's rank depends on its score, which is the mean of normalized criterion values. Hence. A lower score indicates higher performance.

#### V. EXPERIMENTAL RESULTS AND ANALYSIS

In this paper, we firstly investigate the effect of the fuzzy logic introduced in Section III on the astronomical data sets described in Section IV. We find it has a significant improvement in image contrast, as shown in Fig. 3. The fuzzy enhance the image's sharpness beside the faint details appearing. Furthermore, we optimize the parameters of the fuzzy membership functions using GWO. Fig. 4 shows a representative example of optimized membership functions for inputs and outputs.



Fig. 2: 74-inches Kottamia telescope.



(a) Input image

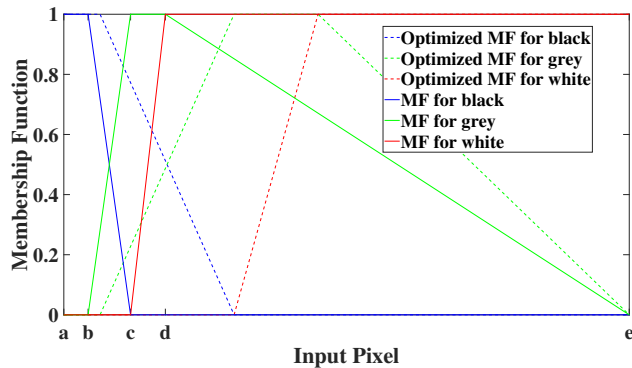
(b) Fuzzy logic output image

Fig. 3: A representative example of sequence M103.

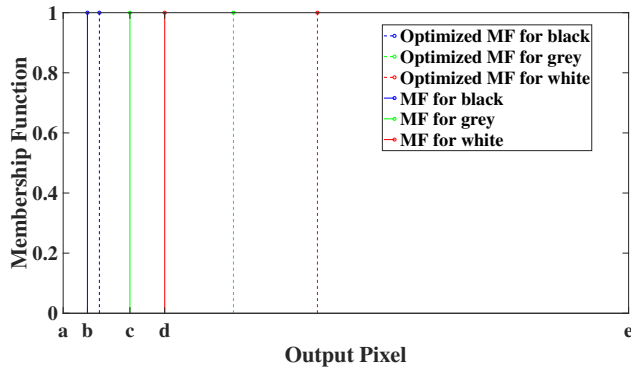
TABLE I: Rank summary for sequence M103.

Measure operator	Accuracy ( $\mu\text{m}$ )	Range ( $\mu\text{m}$ )	# of false	Width ( $\mu\text{m}$ )	Score
Optimized modified histogram	0	5800	13	3842.272	2.188442
Modified histogram	0	5800	13	4276.637	2.277336
Absolute gradient	0	5800	18	3627.112	2.329595
DCT	0	6200	20	3567.982	2.45506
Auto-correlation	0	6200	20	3701.837	2.482454
Squared gradient	800	6300	20	3372.492	2.738618
FFT	800	6200	20	3469.49	2.742596
Brenner	800	6200	20	3490.649	2.746926
Tenengrad	800	6300	20	3574.423	2.779943
Histogram	1200	5900	20	4886.36	3.138787
Amplitude	2600	6100	27	1970.705	3.371561

In addition, we analyze the performance of the optimized modified histogram compared to the operators described in Section II. The operators are ranked according to their score. Table I shows the rank summary for all mentioned operators for the sequence M103. The results show that the optimized modified histogram exceeds the others according to the ranking criteria. That is, the modified histogram before optimization achieves the highest performance. The rank summary for the sequence N7067 is presented in Table II. The results clarify that the optimized modified histogram also obtains the best performance.



(a) Membership function of the input



(b) Membership function of the output

Fig. 4: A representative example of the optimized memberships functions of the fuzzy inputs and outputs.

TABLE II: Rank summary for sequence N7067.

Measure operator	Accuracy ( $\mu\text{m}$ )	Range ( $\mu\text{m}$ )	# of false	Width ( $\mu\text{m}$ )	Score
Optimized modified histogram	0	2600	2	2737.269	0.756599
Modified histogram	0	2600	2	2742.576	0.756651
Squared gradient	0	2600	3	1988.294	0.811725
Tenengrad	0	2600	3	2078.289	0.812611
Brenner	0	2600	4	2133.232	0.875652
Auto-correlation	0	2600	4	2237.403	0.876678
FFT	0	2800	4	2648.115	0.927233
Histogram	700	3500	7	3009.135	1.540337
Absolute gradient	0	4300	9	2926.615	1.591312
DCT	0	4300	12	2649.596	1.776084
Amplitude	2700	2700	16	101578	3.627907

## VI. CONCLUSION

Precise automatic focusing for the high magnification astronomical system significantly affects the observation quality. In this paper, we firstly investigate a focus measure based on fuzzy logic, namely modified histogram on two star-clusters sequences. The data sets are acquired using the 74-inches telescope of the KAO at good seeing conditions and different focus positions. We apply the modified histogram and other common operators into the data sets and compare them based on four evaluation criteria. In addition, we have presented the parameter optimization of the fuzzy membership functions using GWO. The results show that the optimized modified histogram can achieve the highest performance compared to other operators.

## REFERENCES

- [1] Y. Yao, B. Abidi, N. Doggaz, and M. Abidi, "Evaluation of Sharpness Measures and Search Algorithms for the Auto-Focusing of High Magnification Images", Proc. of SPIE, Vol. 6246, 62460G, 2006.
- [2] A. Hamdy, F. Elnagahy, I. Helmy, "Application of Fuzzy Logic on Astronomical Images' Focus Measures", Turkish Journal of Electrical Engineering & Computer Sciences, Vol. 27, 2019, pp. 3815–3822.
- [3] D. Eid, A. Attia, S. Elmasry, I. Helmy, "A Hybrid Genetic-Fuzzy Controller for a 14-inches Astronomical Telescope Tracking", Journal of Astronomical Instrumentation, Vol. 10, 2021, pp. 1–10.
- [4] M. Navabi, S. Hosseini, "A Hybrid PSO Fuzzy-MRAC Controller Based on EULERINT for Satellite Attitude Control", Proceedings of IEEE International Conference on Iranian Joint Congress on Fuzzy and Intelligent Systems, 2021, pp. 33–38.
- [5] E. Bonabeau, M. Dorigo, G. Theraulaz, "Swarm Intelligence: from Natural to Artificial Systems", OUP USA, 1999.
- [6] M. Dorigo, M. Birattari, T. Stutzle, "Ant Colony Optimization", IEEE Computational Intelligence Magazine, vol. 1, 2006, pp. 28–39.
- [7] J. Kennedy, R. Eberhart, "Particle Swarm Optimization", Proceedings of IEEE International Conference on Neural Networks, 1995, pp. 1942–1948.
- [8] S. Mirjalili, S. irjalili, A. Lewis, "Grey Wolf Optimizer", Advances in Engineering Software, vol. 69, 2014, pp. 46–61.
- [9] S. Pertuz, D. Puig, M. Angel Garcia, "Analysis of Focus Measure Operators for Shape-From-Focus", Pattern Recognition, 2013, PP.1415–1432.
- [10] I. Helmy, F. Elnagahy, A. Hamdy, "Focus Measures Assessment for Astronomical Images", Proceedings of IEEE International Conference on Innovative Trends in Communication and Computer Engineering (ITCE), 2020, pp. 6–10.
- [11] I. Sobel, "Focusing", International Journal of Computer Vision, Vol. 1, 1987, pp. 223–237.
- [12] M. Subbarao, J. Tyan, "Selecting the Optimal Focus Measure for Autofocusing and Depth-From-Focus", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, Issue 8, 1998, pp. 864–870.
- [13] J. Schlag, A. Sanderson, C. Neuman, F. Wimberly, "Implementation of Automatic Focusing Algorithms for a Computer Vision System with Camera Control", Carnegie-Mellon University, 1983.
- [14] I. Sobel, "Camera Models and Machine Perception", Ph.D. Thesis, Stanford University, 1970.
- [15] A. Santos, C. Solorzano, J. Vaquero, J. Pena, N. Malpica, F. Pozo, "Evaluation of Autofocus Functions in Molecular Cytogenetic Analysis", Journal of Microscopy, Vol. 188, Issue 3, 1997, pp. 264–272.
- [16] D. Vollath, "The Influence of the Scene Parameters and of Noise on the Behaviour of Automatic Focusing Algorithms", Journal of Microscopy, Vol. 151, Issue 2, 1988, pp. 133–146.
- [17] M. Shirvaikar, "An Optimal Measure for Camera Focus and Exposure", Proceedings of IEEE Southeastern Symposium on System Theory (SSST), 2004, pp. 472–475.
- [18] L. Firestone, K. Cook, K. Culp, N. Talsania, K. Preston, "Comparison of Autofocus Methods for Automated Microscopy", Cytometry, Vol. 12, Issue 3, 1991, pp. 195–206.
- [19] J. Baina, J. Dublet, "Automatic Focus and Iris Control for Video Cameras", Proceedings of IEEE International Conference on Image Processing and its Applications, 1995, pp. 232–235.
- [20] N. Chern, P. Neow, M. Ang, "Practical Issues in Pixel-Based Autofocusing for Machine Vision", Proceedings of IEEE International Conference on Robotics and Automation, Vol. 3, 2001, pp. 2791–2796.
- [21] D. Kim, I. Cho, "An Optimal COG Defuzzification Method for A Fuzzy Logic Controller", Soft Computing in Engineering Design and Manufacturing, 1998.
- [22] F. Groen, I. Young, and G. Lighthart, "A Comparison of Different Focus Functions for Use in Autofocus Algorithms", Cytometry 6.1985, PP. 81–91.
- [23] Y. Azzam, G. Ali, F. Elnagahy, H. Ismail, A. Haroon, I. Selim, A. Essam, "Current and Future Capabilities of the 74-Inch Telescope of Kottamia Astronomical Observatory in Egypt", NRIAG Journal of Astronomy and Astrophysics, Special Issue, 2008, pp. 271– 285.

# Research and examination on implementation of super-resolution models using deep learning with INT8 precision

Shota HIROSE, Naoki WADA, Jiro KATTO

*School of Fundamental Science and Engineering Faculty of Science and Engineering, Waseda University*  
Shinjuku, Tokyo, Japan  
syouta.hrs@akane.waseda.jp

Heming SUN

*Waseda Research Institute for Science and Engineering*  
Waseda University, Tokyo, Japan  
JST, PRESTO, Saitama, Japan  
hemingsun@aoni.waseda.jp

**Abstract**— Fixed-point arithmetic is a technique for treating weights and intermediate values as integers in deep learning. Since deep learning models generally store each weight as a 32-bit floating-point value, storing by 8-bit integers can reduce the size of the model. In addition, memory usage can be reduced, and inference can be much faster by hardware acceleration when special hardware for int8 inference is provided. On the other hand, when inferences are carried out by fixed-point weights, accuracy of the model is reduced due to loss of dynamic range of the weights and intermediate layer values. For this reason, inference frameworks such as TensorRT and TensorFlow Lite, provide a function called “calibration” to suppress the deterioration of the accuracy caused by quantization by measuring the distribution of input data and numerical values in the intermediate layer when quantization is performed. In this paper, after quantizing a pre-trained model that performs super-resolution, speed and accuracy are measured using TensorRT. As a result, the trade-off between the runtime and the accuracy is confirmed. The effect of calibration is also confirmed.

**Keywords**— *Tensor RT, Quantization, Super resolution, Real-time inference*

## I. INTRODUCTION

Image super-resolution is an ill-posed problem for reconstructing the original image from downsampled image. To achieve this, it is needed to add the lost information (mainly high frequency signal). SRCNN[1] is the first successful approach in performing super-resolution using convolutional neural network(CNN). SRCNN itself enhances algorithmically upsampled images. On the other hand, ESPCN[2] uses PixelShuffle (called DepthToSpace in TensorFlow) for downsampled image inputs. ESPCN is much faster than SRCNN because the computational cost of ordinary CNN is proportional to the resolution of the input image. In addition, directly using downsampled images is better in Peak Signal-to-Noise Ratio (PSNR) because refining upsampled images needs larger size of convolution kernel. Therefore, ESPCN is better than SRCNN in throughput and PSNR. After the invention of ESPCN, super-resolution models became deeper and deeper for better PSNR. For example, RCAN[3] is a very deep model for achieving very good PSNR. It has 400 convolutions in total and residual structure for extracting very detailed features of images. It has the significant accuracy but results in longer runtime due to the model structure.

Recently, super-resolution becomes convenient for enhancing images taken by mobile devices, which does not have powerful processor to run deep models. Therefore, the

trade-off between runtime and PSNR is very important. This trend leads to Mobile AI 2021[4], which is a new competition style workshop as sub-area of NTIRE[5], where objective scores are calculated with consideration of runtime and accuracy. Memory usage is also a problem in using deeper model. To solve the problem, quantization is helpful. In quantization, the precision of the parameters of the model is reduced. In most cases, 8bit fixed-point(INT8) value is used in quantization because many devices can accelerate the calculation by using parallel INT8 accelerator. Because pretrained models in PyTorch, TensorFlow and so on, usually use 32bit floating-point(FP32) values to represent parameters, INT8 quantized model becomes about 4 times smaller than FP32 model without quantization. However, the accuracy of the model will significantly drop due to reduction of precision because float to int conversion loses dynamic range of the parameters. To avoid this, calibration is one of the solutions, in which the dynamic range is appropriately adjusted by measuring the distribution of tensors that flow through the model and calculating proper scaling value. In some frameworks including TensorRT, this calibration is automatically done when we have a representative dataset for calibration.

## II. TENSORRT AND CALIBRATION

TensorRT [6] is a framework for fast inference, provided by NVIDIA. TensorRT reduces the inference time by optimizing the kernel by automatically specifying kernels suitable for the devices that perform inferences. TensorRT can fuse layers (such as series of convolutions, batchnorm, and ReLU), and has more optimizing techniques. In addition, it can convert the precision of models from 32-bit floating-point to 16-bit floating-point. In addition, on devices with Tensor cores that support INT8, the precision of models can be converted to INT8. Using INT8 precision and tensor cores, the speed of inference can be improved more than the inferences using 32-bit and 16-bit floating-point. However, a simple conversion from FP32 to INT8 will cause a loss of model accuracy. TensorRT has functions of calibration of models for avoiding significant loss of the accuracy.

## III. EXPERIMENTS

### A. Method

A trained super-resolution model (modified from ESPCN[2]) is converted to a TensorRT model, and super-resolution is performed. Calling CUDA Kernel many times can be bottleneck and we decided to use ESPCN-based model since ESPCN has only three convolutions. To convert

PyTorch’s pretrained model to TensorRT’s model, we used torch2trt [7]. Torch2trt is a PyTorch to TensorRT converter. It uses Python API of TensorRT, and it can make the usage of TensorRT model easy because it automatically converts PyTorch’s model to TensorRT’s model. However, direct conversion from PyTorch to TensorRT fails for our model because the parser of torch2trt for PyTorch is not complete. Torch2trt also supports the model conversion from Open Neural Network Exchange(ONNX) model because torch2trt has the parser for ONNX. ONNX is an universal format to write the structure of neural network, so it is used when pretrained model is exported to other platform. We found that the parser of torch2trt for ONNX can convert our model. Even from ONNX model, the model can be converted and INT8 precision is supported. Therefore, we first converted our pretrained model to ONNX, and then converted ONNX model to TensorRT model. We investigate how inference time and PSNR between original images and reconstructed ones varied depending on precision of the parameters of the model. NVIDIA’s Jetson Xavier NX[8] is used as the device to perform inference. It has tensor cores, and it can infer using models with INT8 precision. To prepare images to be super-resolved, we create a dataset from a video called crowd\_run, distributed by xiph.org [9]. Crowd\_run is a video of people running, and consists of 500 frames. We divide each frame of the video and reduce the resolution of each frame from 1920x1080 to 960x540, then process the reduced image with a super-resolution model to measure the PSNR. In this experiment, we use the ESPCN\_RGB model and the ESPCN\_YCbCr model. ESPCN\_RGB is based on ESPCN with 96 channels in the first layer, 48 channels in the second layer, and 12 channels in the third layer. ESPCN\_YCbCr extracts only the luminance channel obtained by transforming the input image from RGB to YCbCr, and super-resolves only the luminance channel. The numbers of channels are 96, 48, and 4 for the first layer, the second layer and the third layer,

respectively. When the precision of the parameters of the model is converted to INT8 by TensorRT, a calibration dataset is required. In this experiment, we use the validation dataset of DIV2K[10] as the correction dataset, with each image cropped to 960x540 from the center. As a comparison, we also measure the PSNR and speed of PyTorch, which supports FP32 and FP16 inference. To calculate PSNR, we used the formula shown in Eq.1 and Eq.2. The range of the value of images is from 0 to 255, so we divide squared error by 255. It is assumed that the resolution of the picture is H\*W.

$$PSNR = -10 \log_{10} \sum_{i,j} \sum_{RGB} \frac{(HR[i,j][RGB] - Recon[i,j][RGB])^2}{255^2 * H * W * 3}$$

Eq.1 The calculation of PSNR (RGB)

$$PSNR_Y = -10 \log_{10} \sum_{i,j} \frac{(HR[i,j][Y] - Recon[i,j][Y])^2}{255^2 * H * W}$$

Eq.2 The calculation of PSNR (Y)

### B. Results

The result is shown in Table.1. In this table, “PSNR\_Y” represents PSNR of the luminance channel only.

Comparing ESPCN\_RGB and ESPCN\_YCbCr, PSNRs of ESPCN\_YCbCr and the luminance channel are higher than those of RGB channels, regardless of the precision of models. In addition, in all cases except FP32 (TensorRT) and FP16 (PyTorch), the FPS of ESPCN\_YCbCr is lower than ESPCN\_RGB. ESPCN\_YCbCr super-resolves only its luminance channel, and theoretical computational complexity of the convolutions is lower than ESPCN\_RGB. However, there is an overhead of converting the color spaces between RGB and YCbCr, that makes ESPCN\_YCbCr slower.

Table.1 PSNR and FPS (frames per second) among two models and bicubic upsampling

		Pytorch		TensorRT			
		FP32	FP16	FP32	FP16	INT8 (w/o calibration)	INT8 (w/ calibration)
PSNR [dB]	ESPCN_RGB	30.0797	30.0817	30.0797	30.0772	27.9382	29.7454
	ESPCN_YCbCr	30.7626	30.7586	30.7626	30.7587	29.9882	<b>30.5667</b>
	Bicubic				28.9282		
PSNR_Y [dB]	ESPCN_RGB	30.6808	30.6829	30.6808	30.6782	28.3645	30.4385
	ESPCN_YCbCr	30.7575	30.7534	30.7575	30.7536	29.9838	<b>30.5617</b>
	Bicubic				29.4030		
FPS [frames/sec]	ESPCN_RGB	10.39	16.01	14.08	28.16	51.95	<b>51.95</b>
	ESPCN_YCbCr	10.27	16.97	14.82	28.01	47.37	<b>47.37</b>
	bicubic				78.5102		

We visualize the model structure of ESPCN\_RGB and ESPCN\_YCbCr, using netron[11]. Netron can analyze ONNX model and visualize the structure of the model. First, the model structure of ESPCN\_RGB is shown in Fig.1. There are only convolutions, ReLUs, clip function, and DepthToSpace (i.e. PixelShuffle). Next, the model structure

of ESPCN\_YCbCr is shown in Fig.2. We dividedly show the structure of ESPCN\_YCbCr that is more complex than ESPCN\_RGB since it has a function to convert color-space in itself. According to Fig. 2(a), ESPCN\_YCbCr divides RGB signals to each color channel to calculate YCbCr channels. In addition, according to Fig.2(b), ESPCN\_YCbCr concatenates

super-resolved luminance channel and algorithmically upsampled chroma channels. This conversion has not much computational complexity compared to three convolutions, but the calculation calls CUDA kernels multiple times, resulting in a bottleneck. In fact, the convolutions in ESPCN\_YCbCr are not very computationally demanding, and the overhead of calling CUDA kernels multiple times by converting color-spaces cannot be ignored. To prove that the conversion is a bottleneck, we show the result of the analysis using Nsight Systems[12], that is a performance analysis tool

to visualize CPU/GPU interworking. The result for ESPCN\_RGB is shown in Fig. 3, and the result for ESPCN\_YCbCr is shown in Fig. 4. It is confirmed that ESPCN\_YCbCr calls CUDA kernel more than ESPCN\_RGB and runtime per image is longer. In addition, we show Fig.5 to prove color-space conversion from YCbCr to RGB needs 2.5ms. For readability, we divided Fig. 5. into four figures and show as Fig. 6. According to Fig. 6, color-space conversion is not light computation.

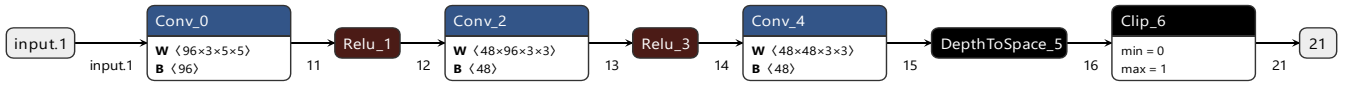
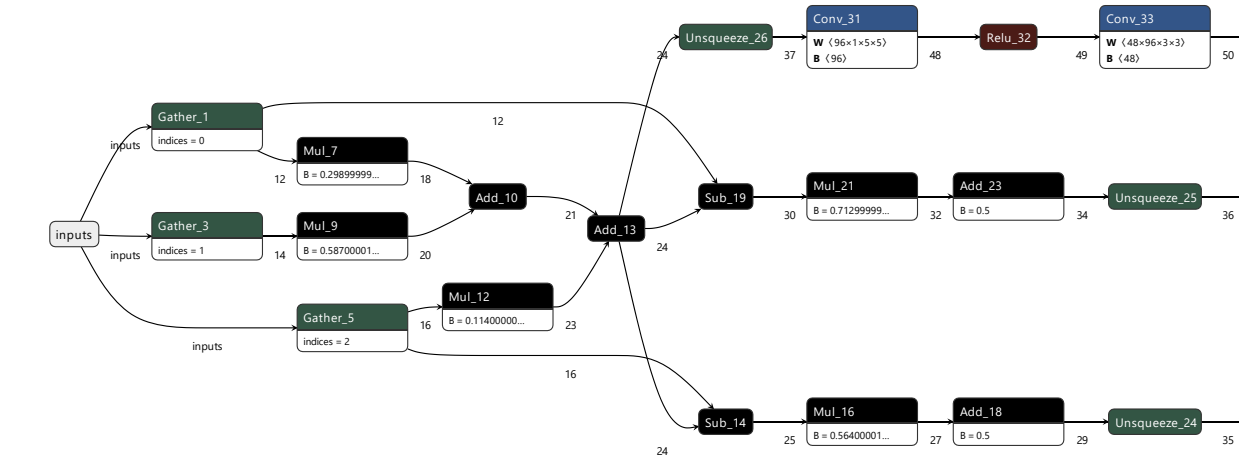
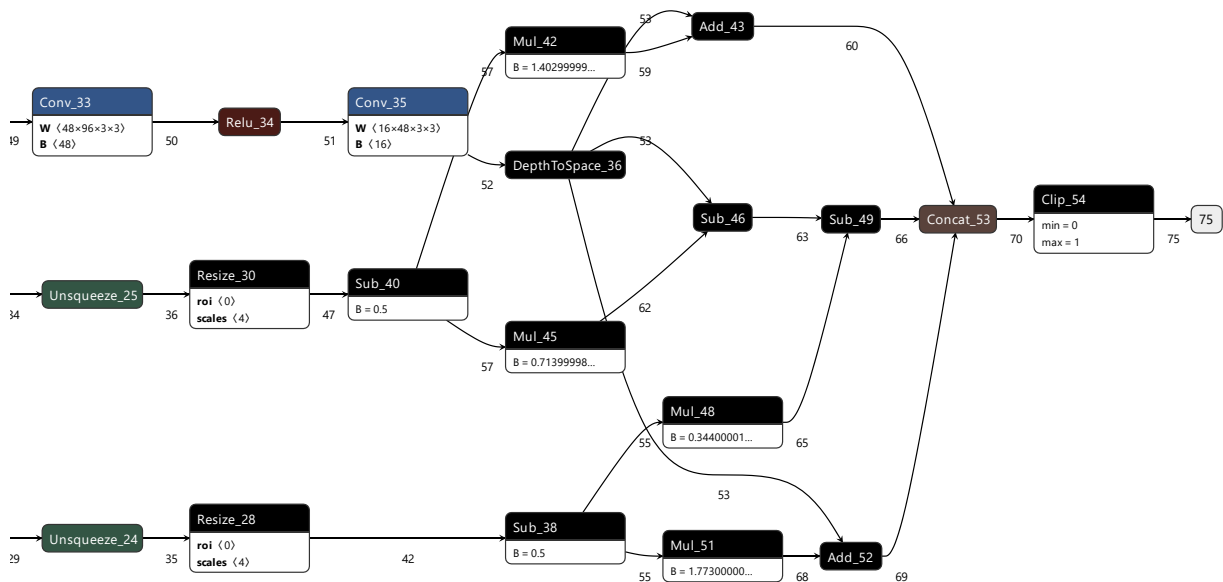


Fig.1 The structure of ESPCN\_RGB



(a) first half



(b) second half

Fig.2 The structure of ESPCN\_YCbCr



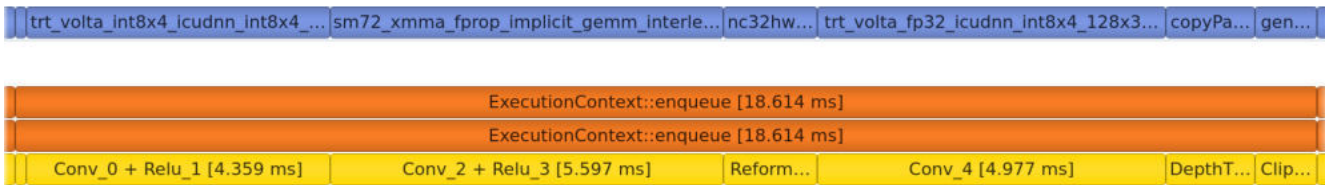


Fig.3 The result of the performance analysis of ESPCN\_RGB

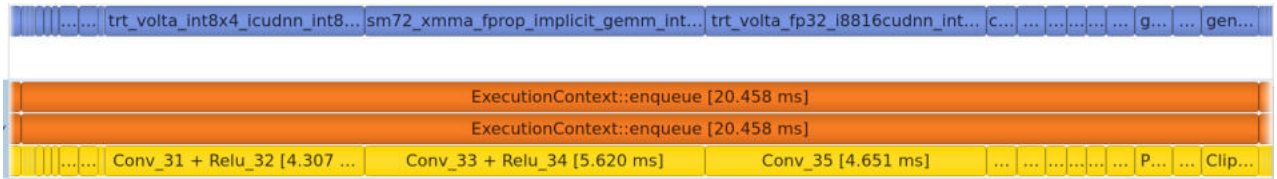


Fig.4 The result of the performance analysis of ESPCN\_YCbCr

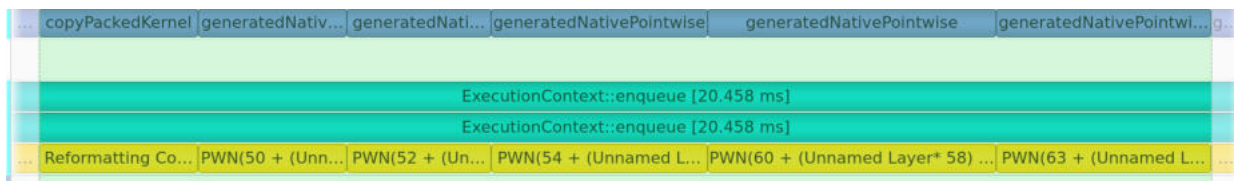


Fig.5 The runtime analysis of the conversion color-space from YCbCr to RGB

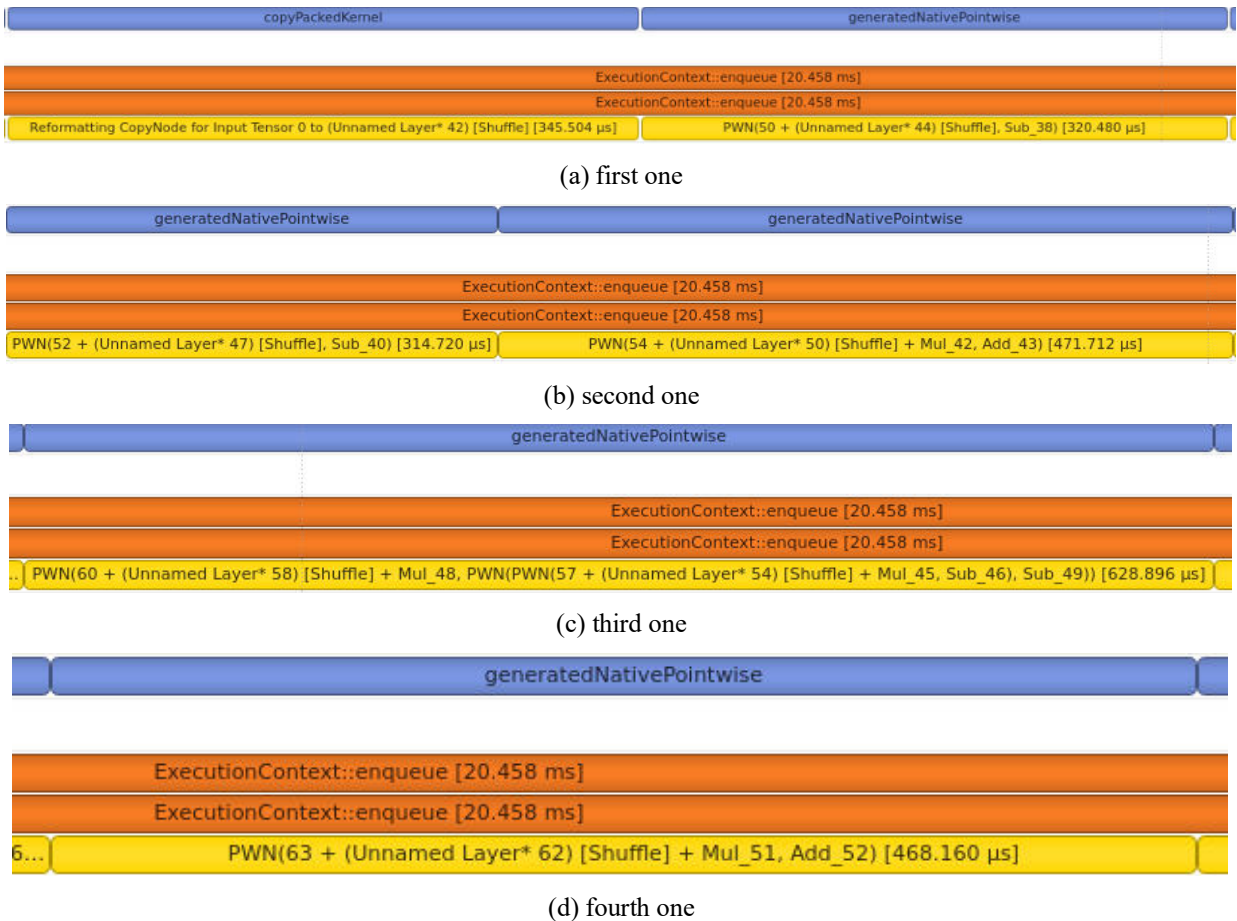


Fig.6 The detailed analysis of the runtime of the functions for converting color-space from YCbCr to RGB

Although ESPCN\_YCbCr has little more computational cost, the PSNR is much better than ESPCN\_RGB. Particularly,

ESPCN\_YCbCr has good performance when it is quantized. As shown in Table. 1, without calibration, quantized

ESPCN\_RGB lost PSNR significantly (2.1415 [dB]) from pretrained. However, ESPCN\_YCbCr lost PSNR less (0.7744 [dB]) from pretrained. From the viewpoint of the speed, we found that inference using TensorRT and INT8 precision is more than four times faster than inference using PyTorch's FP32 precision. This reason can be attributed to the Tensor core in Xavier NX, which is capable of performing 4x4x4 matrix products at high speed, and it is thought that TensorRT converts convolutional operations into matrix products and uses the acceleration provided by the Tensor core.

#### IV. CONCLUSIONS

In this paper, TensorRT was used to infer for the super-resolution model. Although conversion of the FP32 model to INT8 precision generated a decrease in PSNR, the calibration function was able to suppress the decrease of the accuracy. This is because TensorRT measures the distribution of the input tensor values when converting from floating-point to fixed-point and calculates how to scale the tensor to fit into the INT8 range and sets an appropriate dynamic range. We also found that converting the model to INT8 was more than four times faster than PyTorch's (FP32) inference. We can conclude that the quantization provided by TensorRT is useful in situations that require fast inferences.

In conclusion, we successfully implemented super-resolution running on mobile devices by 50 fps. As future work, we try to compare the automatic calibration with manual optimization of fixed point arithmetic, and incorporate other super-resolution models under tradeoff between runtime and super-resolution accuracy.

#### ACKNOWLEDGMENT

This work was supported in part by NICT, Grant Number 03801, Japan.

#### REFERENCES

- [1] Chao Dong, Chen Change Loy, Kaiming He, Xiaoou Tang, "Image Super-Resolution Using Deep Convolutional Networks", TPAMI 2015, May 2015.
- [2] Wenzhe Shi, Jose Caballero, Ferenc Huszar, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, Zehan Wang: "Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network", IEEE CVPR 2016, June 2016.
- [3] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, Yun Fu, "Image Super-Resolution Using Very Deep Residual Channel Attention Networks", ECCV 2018, September 2018.
- [4] Computer Vision Laboratory, ETH Zurich: Accessed on Nov. 30, 2021. [Online] Available: <https://ai-benchmark.com/workshops/mai/2021/>
- [5] Computer Vision Laboratory, ETH Zurich: Accessed on Nov. 30, 2021. [Online] Available: <https://data.vision.ee.ethz.ch/cvl/ntire21/>.
- [6] NVIDIA: "NVIDIA TensorRT | NVIDIA Developer", <https://developer.nvidia.com/tensorrt>.
- [7] "NVIDIA-AI-IOT/torch2trt: An easy to use PyTorch to TensorRT converter", Accessed on: Jan. 10, 2022. [Online]. Available: <https://github.com/NVIDIA-AI-IOT/torch2trt>.
- [8] NVIDIA: "Jetson Xavier NX for embedding/edge systems | NVIDIA", <https://www.nvidia.com/ja-jp/autonomous-machines/embedded-systems/jetson-xavier-nx/>.
- [9] Xiph.org: "Xiph.org : Derf's Test Media Collection", <https://media.xiph.org/video/derf/>.
- [10] Agustsson, Eirikur and Timofte, Radu: "NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study", CVPR workshops, July 2017.
- [11] Lutz Roeder: Accessed on: Nov. 28, 2021. [Online]. Available: <https://netron.app/>.
- [12] NVIDIA: "NVIDIA Nsight Systems | NVIDIA Developer", <https://developer.nvidia.com/nsight-systems>.

# Mitigating Overflow of Object Detection Tasks Based on Masking Semantic Difference Region of Vision Snapshot for High Efficiency

Heuijee Yun<sup>1</sup> and Daejin Park<sup>1\*</sup>

<sup>1</sup>School of Electronics Engineering, Kyungpook National University, Daegu, Republic of Korea

\*Correspondence to: Daejin Park (boltanut@knu.ac.kr)

**Abstract**—Object recognition functions are essential to properly perform safety and autonomous driving functions. However, sophisticated object recognition work requires extensive computation. It is difficult to handle a large amount of computation on the lightweight embedded boards currently used in vehicles. In this paper, we propose a method using machine learning and deep learning for lightweight object recognition algorithm in lightweight embedded boards. We created an algorithm suitable for lightweight embedded boards by appropriately using deep neural network architecture that requires small computational volumes but provides low accuracy, as well as deep-learning algorithms that require large computational volumes but provide high accuracy. After determining the area using a deep neural network architecture algorithm with a relatively small amount of computation, we improved the accuracy by using a more accurate deep learning algorithm. We used OpenCV to process input images in Python, and we processed image by using efficient neural network (ENet) and You Only Look Once (YOLO). By executing this algorithm, we can realize more accurate and lightweighted object recognition.

**Index Terms**—Autonomous driving, object detection, OpenCV, ENet, YOLO, deep learning

## I. INTRODUCTION

Currently, with the development of artificial intelligence technology, its application is expanding. For most functions, image processing is essential. However, to process images in real time, the amount of computation must be small. Although current image processing technologies are developing, they have been unable to effectively process the required amount of computation with required accuracy. In addition, to run on an embedded board with relatively little memory, the computation volume must be very small. In this study, we used a combination of deep neural network architecture and deep-learning techniques to efficiently increase the amount of computation and accuracy. We used semantic segmentation using deep neural network architecture and YOLO using deep learning to highlight the advantages of computational amount and accuracy when each is independently executed.

This study was supported by the BK21 FOUR project funded by the Ministry of Education, Korea (4199990113966, 10%), and the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (NRF2019R1A2C2005099, 10%), and Ministry of Education (NRF2018R1A6A1A03025109, 10%), and Institute of Information & communication Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (no. 2021-0-00944, Metamorphic approach of unstructured validation/verification for analyzing binary code, 70%), and the EDA tool was supported by the IC Design Education Center (IDEC), Korea.

## II. PROPOSED METHOD

### A. Overall Structure

Fig 1 (a) illustrates the system's structure. After receiving the image input to the webcam in units of frames, semantic segmentation is executed using ENet [1]. After inserting an image masked only with ROI as an input of YOLO [2], the result of YOLO is written on the image. It is a structure that combines these frames to output results in real time. Therefore, by setting the ROI (region of interest) using semantic segmentation, we improved the accuracy in YOLO and made it possible in real time using Python.

### B. Image Detection Structure

We used ENet program for semantic segmentation. 1 (b) illustrates the ENet segmentation program's structure. First, the image frame from the webcam is set to be processed on the ENet. Because the trained model processes with a size of 256, it is set accordingly. It then loads through the path of the trained model and processes the image. When the image is processed, the resulting image represents the label suitable for each object and its color in the form of a matrix. To filter for only human images in the results, all information except for humans is binarized as 0, and information about humans is binarized as 1. Semantic segmentation is completed only when the mask, which is a binarized matrix, and the original image are combined. In Python, you can use NumPY to represent an image as a matrix. With the used of this term, the binarized matrix and the original image are composed into a matrix of the same format and then masked using the OpenCV function. The original image is depicted in Fig 1 (d) (1), the mask in Fig 1 (d) (2), and the result in Fig 1 (d) (3).

Fig 1 (c) illustrates the structure of the YOLO object detection program. Among the programs that use YOLO, we used darkflow [3], which can be used with Tensorflow. It is convenient to use the functions of Tensorflow which reduces the amount of code. Earlier in the process, the masked image using the ENet is received as an input. Afterward, the YOLO model is loaded into the appropriate path. Because the ROI is set by semantic segmentation, the confidence threshold of YOLO is set high to reduce the amount of computation in YOLO. When the image is processed using the model loaded with the set threshold, each object's label, confidence and the position are displayed as a matrix. With the OpenCV function in Python, the detected object's position is drawn as a box, and the object label and confidence are written on the box. [4]

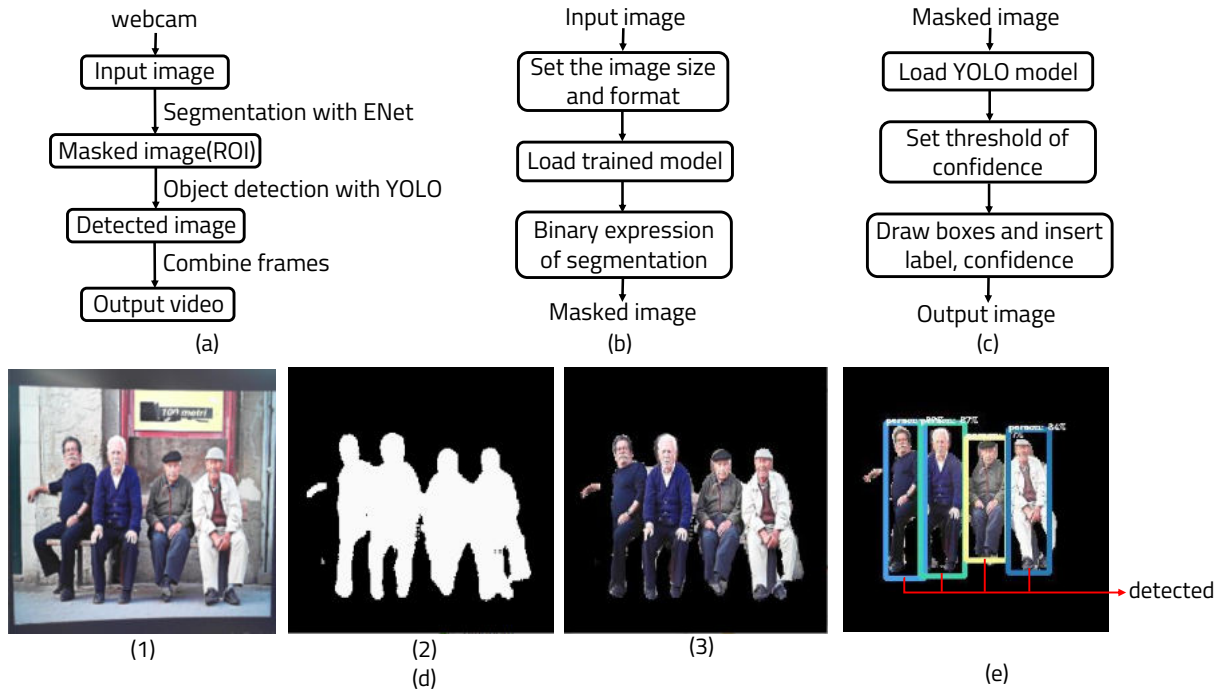


Fig. 1. Structure of program and result (a) Overall structure (b) Semantic segmentation structure (c) YOLO structure (d) LS1028a board (e) Semantic segmentation result (d) YOLO result

### C. Measurement

Fig 2 (a) illustrates YOLO time measurement and Fig 2 (b) demonstrates time of ENet and YOLO program. Since it is difficult to photograph a place where the number of people changes in real time, it was measured while continuously showing pictures with different numbers of people on the webcam. In the program using YOLO alone, processing time and FPS measurements are clearly unstable, as they depend on the number of people being counted. However, the program used by integrating ENet and YOLO is clearly stable, even when the number of people changes in FPS and processing time.

We measured memory use as each program ran. "Working set" represents the amount of physical memory the program is currently using. "Private bytes" indicates the amount of memory allocated to the program which includes physical memory and swapped memory. "Page Faults/sec" shows the use of virtual memory. Fig 2 (c) illustrates YOLO's memory measurement and Fig 2 (d) shows YOLO's and ENet's memory usage. In both programs, the value of Page Faults/sec was used small, so interrupts occurred less frequently. The average amount of physical memory used to process YOLO alone was 1.558 GB, and the average working set required to use YOLO and ENet together was 2.223 GB. When ENet and YOLO are used together, the models of the two programs need to be loaded separately, so memory use is of course higher than when YOLO is used alone. However, this method is valid because the increased number of memory bytes greatly improves accuracy, and it can be used on an embedded board.

Fig 3 (a) illustrates the programs' average time and fps measurement. With much fewer convolution layers than YOLO and a fast and compact encoder decoder structure, the lowest average processing time was 0.156 seconds per frame, and the average fps was highest at 6.41. When YOLO was used alone, the average processing time per frame was 0.281 seconds, and the average fps was 3.645. When processing the results of ENet with YOLO, the average time spent in YOLO was 0.269 seconds and the average fps was 3.701. We confirmed that the method of setting the ROI using ENet and processing it as YOLO's input can reduce the burden on YOLO's execution. This result is an advantage of noticeably lowering the threshold of YOLO.

Fig 3 (b) demonstrates the programs' accuracy and average errors. Several methods are available to measure a model's accuracy in deep learning, among which it was measured using a confusion matrix [5]. We can divide into 4 states as TP when the model corrects the correct answer, TN when the model incorrectly predicts the correct answer, FP when the model incorrectly predicts an incorrect answer as a correct answer, and FN when the model incorrectly predicts the correct answer as an incorrect answer. Accuracy can be calculated using these states by this equation  $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$ . To measure the object recognition program's accuracy, we divided the total number of recognition frames by the number of frames that accurately matched the number of people. The accuracy is clearly better when YOLO and ENet are used together. However, since we recognized the part where the number of people was accurately matched with TP and TN, we calculated the error by comparing the number of

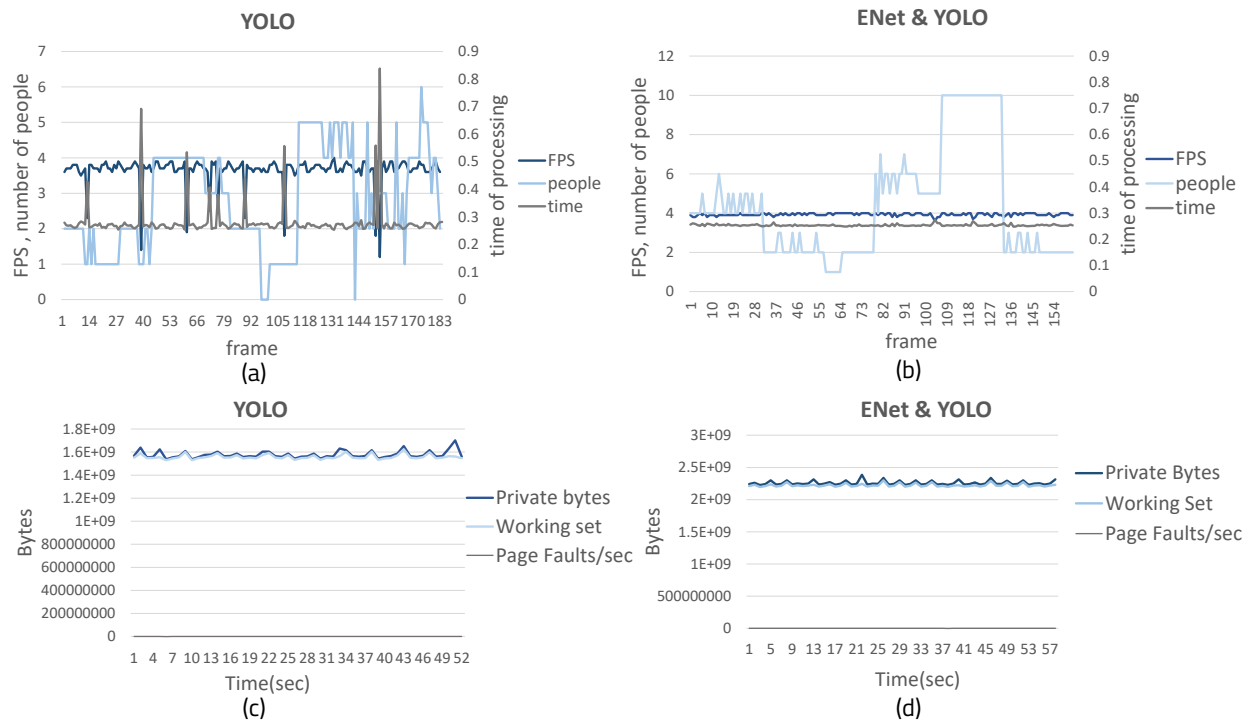


Fig. 2. Time and memory usage measurement (a) Time measurement of YOLO only (b) Time measurement of YOLO and ENet (c) Memory usage measurement of YOLO only (d) Memory usage measurement of ENet and YOLO

recognized people with the number of real people to calculate with more appropriate accuracy. When we calculated accuracy, our method did not improve significantly. However, regarding the average recognition error, when YOLO was used alone, it was 7.097, and when ENet and YOLO were used together, it was 2.913, a clear difference. As the number of people in the photo increases, signs emerge that the errors increase when YOLO is used alone. However, if ENet and YOLO are used together, errors can be reduced and accurate recognition can be achieved.

Program	Average time	Average fps
ENet	0.156s	6.41
YOLO	0.281s	3.645
ENet + YOLO	0.269s	3.701

(a)

Program	Accuracy	Average error of counting
ENet + YOLO	0.574	2.913
YOLO	0.549	7.097

(b)

Fig. 3. Average time and accuracy measurement of programs (a) Average time and fps measurement of ENet, YOLO and ENet and YOLO (b) Accuracy of YOLO, ENet and YOLO

### III. CONCLUSION

In this paper, we propose a structure of real-time object detection using semantic segmentation and YOLO for advanced accuracy and small amount of computation. We proposed this structure because it is difficult to implement high accuracy in real time due to computational volume and memory limitations. When a webcam image is input, the ROI is set by semantic segmentation using the trained model of ENet. Then we can run YOLO to mark the object's position, label and confidence. As a result of measuring processing time, FPS, accuracy and error, respectively, our program clearly recognizes objects with much fewer errors and greater accuracy. With this structure, we can optimize real-time object detection with great accuracy and less computation.

### REFERENCES

- [1] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: A deep neural network architecture for real-time semantic segmentation," 2016.
- [2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.
- [3] T. H. Trieu, "Darkflow," *GitHub Repository*. Available online: <https://github.com/thtrieu/darkflow> (accessed on 14 February 2019), 2018.
- [4] P. Ren, W. Fang, and S. Djahel, "A novel yolo-based real-time people counting approach," in *2017 International Smart Cities Conference (ISC2)*, 2017, pp. 1–2.
- [5] J. T. Townsend, "Theoretical analysis of an alphabetic confusion matrix," *Perception & Psychophysics*, vol. 9, no. 1, pp. 40–50, 1971.

# Calibration-Net:LiDAR and Camera Auto-Calibration using Cost Volume and Convolutional Neural Network

1<sup>st</sup> An Nguyen Duy

Department of Information Communication Convergence  
Soongsil University  
Seoul, Korea Republic of.  
nguyenduyan710@gmail.com

2<sup>nd</sup> Myungsik Yoo

School of Electronic Engineering  
Soongsil University  
Seoul, Korea Republic of.  
myoo@ssu.ac.kr

**Abstract**—A fusion of multi-sensor has been utilized widely for improving the environment perception in autonomous vehicles and robot navigation. Calibration is an essential procedure for preprocessing the data fusion between multiple sensors. Most target-based calibration techniques require manual works and specific calibration targets to achieve high accuracy. It gradually becomes outmoded for Light Detection and Ranging (LiDAR) and camera with the development of deep learning techniques. This paper proposed an online LiDAR-camera calibration that automatically predicts the extrinsic parameters by taking advantage of convolutional neural networks (CNNs). We take depth maps of stereo camera prediction and depth maps of the LiDAR projection as two separated branches as inputs for the proposed network. Unlike the current CNN-based calibration method, we construct a cost volume of the correlation between two corresponding pixels of depth maps in stereo camera and LiDAR, respectively. The proposed model gains a reasonable capability to adjust to different initial calibration error ranges. We evaluate the proposed architecture on the KITTI dataset and achieve 0.378 degree in rotation error and 2.353cm translation error.

**Index Terms**—sensor fusion, LiDAR, stereo camera, supervised learning, deep-learning,

## I. INTRODUCTION

In the last decade, multi-sensor fusion has developed rapidly in many applications such as object detection, 3D reconstruction, classification, and depth prediction. LiDAR and camera are the most widely used sensors to provide accurate and stable perception for the surrounding environment. While LiDAR obtains the spatial depth information with high accuracy, it lacks color and texture information at low resolution. The camera brings the benefits of high-resolution RGB images but no distance information. Therefore, calibration is an essential preprocessing of data fusion to ensure the precision to transform the LiDAR coordinate to camera coordinate.

Most early LiDAR-camera calibration methods [1-3] used specific calibration targets and complex manual setups to extract the 2D-3D corresponding feature to find the external parameters. However, these methods operate offline and are not suitable for running in real-time for the autonomous vehicle. Deep learning techniques [4-7] are raising currently to give accurate calibration between LiDAR-camera through the

driven-data source collected in real-time (KITTI) [9] which does not require any specific calibration target or manual setups. In this paper, we proposed a novel deep learning network to automatically estimate the 6 DoF transformation.

Our design consists of a stereo depth map estimated by a stereo camera and a LiDAR depth map projected from the point cloud as inputs. The network extracts multiscale features, the correlation layer is used to match the information from both multiscale features of LiDAR and stereo camera. Then, to predict the transformation, we stack two fully connected layers for global regression and a loss function to optimize the learning process.

## II. METHODOLOGY

In this section, we introduce our proposed model for estimating extrinsic calibration with stereo and LiDAR depth maps as inputs. This work aims to find the rigid transformation by minimizing the loss function compared to the ground truth. The ground truth consists of projected depth maps from the 3D point clouds to the image plane, which provides the same depth at each arbitrary pixel location of the depth map derived from the stereo camera. The data representation of the network inputs, the network architecture, and the training are discussed in detail in the following sections.

### A. Data Representation

The first input of our network is the stereo depth maps. Based on the known intrinsic and extrinsic parameters of stereo cameras, the depth information of a point from the left-right camera can be calculated as the following equation:

$$depth = \frac{B \cdot f}{disparity} \quad (1)$$

where  $B$  and  $f$  are the baseline and focal length of the stereo camera, respectively, and the disparity is the difference of two corresponding pixels present for the same point in the world coordinate. By given initial LiDAR-camera transformation  $H_{init}$  and camera intrinsic  $K$ , we can project each 3D point cloud  $P_i = [X_i, Y_i, Z_i] \in \mathbb{R}^3$  into a virtual image plane

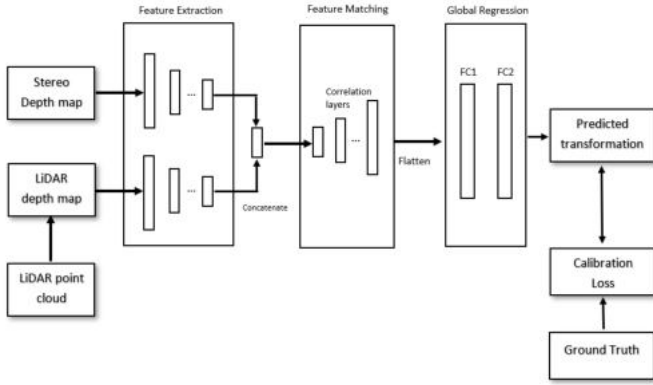


Fig. 1. The working flow of the proposed model.

with corresponding pixels  $p_i = [u_i, v_i] \in \mathbb{R}^2$ . The projection process is described as follows:

$$\begin{aligned} Z_i^{init} \cdot p_i &= K \cdot H_{init} \cdot P_i \\ H_{init} &= \begin{bmatrix} R_{init} & t_{init} \\ 0 & 1 \end{bmatrix} \end{aligned} \quad (2)$$

where  $R_{init}$  and  $t_{init}$  are the initial rotation matrix and translation vector of the transformation  $H_{init}$ . At each pixel  $p_i$  the depth value  $Z_i^{init}$  is preserved. If the pixel does not match any LiDAR point, this pixel will be set as zero.

### B. Network Architecture

The proposed network architecture includes of three parts to solve the tasks of feature extraction, feature matching and global regression of the calibration. Since the three parts are merged as one CNN but with different parameter for each part, the network can be trained end-to-end. The working flow of the network is shown in the Fig.1 and the function of each part will be described as following section

1) *Feature extraction*: The depth maps of the RGB prediction and LiDAR projection are calculated individually as mention in the previous section. However, the formats of the depth maps are different because of two different sensor modalities. Since the depth map are pre-processing individually mentioned in previous section. The data is calculated in different sensors with different modalities. There are two parallel feature extraction network to use to extract the rich features and reduce their dimensions. The output of the final feature maps will be down-sampled by six times and has 196 channels which extract the high-level features of the original inputs.

2) *Feature Matching*: The feature maps are concatenated along the channel dimension after extracting features from both input modalities. This network is motivated by PWC-Net [12] who introduces a correlation layer for feature matching. A cost volume is constructed to calculate the matching cost for connecting depth value of a pixel in the RGB depth prediction branch  $x_D^{rgb}$  with its corresponding pixel in depth feature maps projected from LiDAR  $x_D^{lidar}$ . The matching cost can be defined as:

$$cv(p_1, p_2) = \frac{1}{N} (c(x_D^{rgb}(p_1)))^T c(x_D^{lidar}(p_2)) \quad (3)$$

where  $c(x)$  is the flattened vector of the feature map  $x$  and  $T$  is the transpose operator,  $N$  is the length of the column vector  $c(x)$ . For different level of the pyramid layer setting, the cost volumes is needed to compute with a limited range of  $d$  pixels, i.e.,  $|p_1 - p_2|_\infty \leq d$ . The size of feature maps (conv6 in PWC-Net) are very small. Therefore, we set the value of the range  $d$  to be small. The dimension of the 3D cost volume  $cv(p_1, p_2)$  is  $d^2 \times H \times W$ , where  $H$  and  $W$  are denoted as the height and width of the final pyramid feature maps  $x_D^{rgb}$  and  $x_D^{lidar}$ , respectively.

3) *Global Aggregation*: According to the regression network, it consists of two fully connected layers with 512 neurons and 256 neurons to regress the rotation and translation. The output of the network is 1x4 rotation  $r_{pred}$  and 1x3 translation vector  $t_{pred}$ . The results of the estimated rotation  $r_{pred}$  and the estimated translation  $t_{pred}$  can be evaluated by calculating the loss function compared to the ground truth with well-calibrated scenes.

### C. Loss Function

Given an input pair of a depth image predicted by RGB image  $D_{rgb}$  and a depth image projected from LiDAR point cloud  $D_{lidar}$ , we used the following loss function described as Eq. (4):

$$\mathcal{L}(D_{rgb}, D_{lidar}) = \lambda_1 \mathcal{L}_r(D_{rgb}, D_{lidar}) + \lambda_2 \mathcal{L}_t(D_{rgb}, D_{lidar}) \quad (4)$$

where the  $\mathcal{L}_r(D_{rgb}, D_{lidar})$  is the rotation loss and the  $\mathcal{L}_t(D_{rgb}, D_{lidar})$  is the translation loss,  $\lambda_1$  and  $\lambda_2$  denote the respective loss weight to the rotation and translation loss. According to the rotation loss, the predicted rotation and the ground truth present in quaternions which are difficult to evaluate through Euclidean different distance between prediction and the ground truth. Therefore, we need to present the difference between quaternions into angular distance to evaluate the rotation loss:

$$\mathcal{L}_r = \mathcal{D}_a(r_{gt}, r_{pred}) \quad (5)$$

where  $r_{gt}$  and  $r_{pred}$  are the ground truth and prediction of quaternion, respectively.  $\mathcal{D}_a$  is the angular distance of two quaternions [4] For the translation loss we use a smooth  $\mathcal{L}_1$  loss [8] which is much smoother regarding to the square function's usage near zero.

## III. EXPERIMENT AND RESULT

### A. Experimental Setup

1) *Dataset*: Our proposed network is evaluating on the raw branch of the KITTI dataset [9]. The ground truth of the extrinsic parameters are provided by [11] for each sensor. The camera depth will be generated by using the left and right images of the camera images. The depth maps of the LiDAR will be obtained by projecting point cloud into a virtual image



Fig. 2. The initial calibration

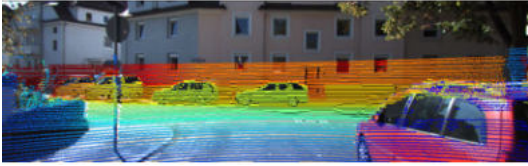


Fig. 3. The calibration result

plane with random initial transformation  $H_{init}$  and known intrinsic parameters of the camera  $K$ . There are 15967 frames for training and 4541 frames for testing. The testing set is spatially separated from the training set.

2) *Evaluation Metrics*: The calibration results are evaluated regarding to the rotation and translation errors of the predicted extrinsic parameters compared to the ground truth transformation. The rotation error can be described as follows:

$$E_r = \mathcal{D}_a(r_{gt} * inv(r_{pred})) \quad (6)$$

$$\mathcal{D}_a(m) = atan2(\sqrt{b_m^2 + c_m^2 + d_m^2}, |a_m|) \quad (7)$$

where  $\{a_m, b_m, c_m, d_m\}$  is four components of the quaternion  $m$ , and  $*$  denotes as the quaternion multiplication and  $inv$  presents the inverse of a quaternion. The translation error is evaluated by the difference of the Euclidian distance between the predicted translation vector and the ground truth. It can be expressed as follows:

$$E_t = \|t_{gt} - t_{pred}\|_2 \quad (8)$$

3) *Training Details*: We implemented our model with PyTorch (1.10.1) and trained on a RTX 3060 GPU. During the training, we choose Adam Optimizer [10] with learning rate  $1e^{-4}$  with batch size 24 and total epoch 100.

## B. Results and Discussion

In this section, the visual results of the calibration are shown in the Figs. 2 and 3. We sampled the decalibration in range of  $[-20^\circ, 20^\circ] / [-1.5m, 1.5m]$ . TABLE I expresses that our method is superior to other architecture due to the same training dataset. However, it still contains a large gap between our performance and the state-of-the-art CFNet in both rotation and translation errors. After investigating, the reason for the performance differences is the iterative calibration refinement. By predicting the calibration flow and valid 2D-3D corresponding set, CFNet applies the EPnP algorithm with the RANSAC scheme to refine the initial extrinsic parameters, which improves the extrinsic calibration accuracy after five times refinement. Despite not having the best performance,

TABLE I  
COMPARISON WITH OTHER METHODS

Methods	Rotation ( $^\circ$ )			Translation (cm)		
	Roll	Pitch	Yaw	X	Y	Z
CFNet [13]	0.059	0.110	0.092	1.025	0.092	1.042
Ours	0.105	0.21	0.19	2.82	2.35	1.89
CalibRCNN [14]	0.19	0.64	0.44	6.2	4.3	5.4
CalibNet [6]	0.15	0.9	0.18	4.2	1.6	7.22

our proposed architecture points out some improvements compared to other models with a mean calibration error  $0.378^\circ$  in rotation and 2.353cm in translation.

## IV. CONCLUSION

In this paper, we have proposed a novel method for 3D LiDAR-Camera extrinsic calibration using a deep neural network. By extracting the depth maps of the stereo camera and LiDAR point cloud, the model construct a cost volume between two depth maps. Our model achieves an improvement in performance comparing to other methods.

## ACKNOWLEDGMENT

This research was supported by the National Research Foundation of Korea (NRF) grant funded by the Government of South Korea (MSIT)(NRF-2021R1A2B5B01002559)

## REFERENCES

- [1] S. Verma, J.S. Berrio, S. Worrall, and E. Netbot, "Automatic extrinsic calibration between a camera and a 3D LiDAR using 3D point and plane correspondences," *arXiv preprint arXiv:1904.12433*, 2019
- [2] P. An, T. Ma, K. Yu, B. Fang, J. Zhang, W. Fu, and J. Ma, "Geometric calibration for LiDAR-camera system fusing 3D-2D and 3D-3D point correspondence," *Optics Express*, vol. 28, no. 2, pp. 2122-2141, 2020
- [3] A.-S. Vaida and S. Nedevschi, "Automatic extrinsic calibration of LiDAR and monocular camera images," in *IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP)*, Cluj-Napoca, Romania, pp. 117-124, 2019
- [4] Kendall, A. Grimes, M. and Cipolla, R. "Posenet: A convolutional network for real-time 6 DoF camera relocalization" in *Proceeding of the IEEE international conference on computer vision*, 2938-2946 (2020)
- [5] Schneider, Piewak, F, Stiller, C and Franke, U. "Regnet: Multimodal sensor registration using deep neural networks," in *IEEE intelligent vehicles symposium (IV)*, 1803-1810 IEEE (2017)
- [6] Iyer, G. Ram, R. Murthy, J.K and Krishna, "Calibnet: Geometrically supervised extrinsic calibration using 3D spatial transformer networks" *arXiv preprint arXiv: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1110-1117, IEEE (2018)
- [7] Lv, X., Wang, B., Ye, D., and Wang, S., "Lidar and camera self-calibration using costvolume network," *arXiv preprint arXiv:2012.13901* (2020).
- [8] R. Girshick, "Fast R-CNN" in *Proceeding of the IEEE international conference on computer vision*, 2015, pp. 1440-1448
- [9] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR*, 2012.
- [10] D. Kingma and J. Ba, "Adam: A method for stochastic optimization." *arXiv preprint arXiv:1412.6980*, 2014.



- [11] Andreas Geiger, Frank Moosmann, and Bernhard Schuster "Automatic camera and range sensor calibration using a single shot" in 2012 *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3936-3943, IEEE, 2012.2,5
- [12] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8934–8943, 2018
- [13] Lv, X.; Wang, S.; Ye, D. CFNet: LiDAR-Camera Registration Using Calibration Flow Network. *Sensors* 2021, 21, 8112. <https://doi.org/10.3390/s21238112>
- [14] J. Shi et al., "CalibRCNN: Calibrating Camera and LiDAR by Recurrent Convolutional Neural Network and Geometric Constraints," 2020 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 10197-10202, doi: 10.1109/IROS45743.2020.9341147.

# Granular Analysis of Pretrained Object Detectors

Eric Xue  
University of Toronto  
Toronto, Canada  
e.xue@mail.utoronto.ca

Tae Soo Kim  
Johns Hopkins University  
Baltimore, USA  
tkim60@jhu.edu

**Abstract**—Object detectors have become the fundamental building blocks of many real-world machine learning applications. Even though different problem domains require their own unique object detector specifications, it is common practice to take a pretrained object detector off the shelf and either use it as-is or fine-tune it with limited amounts of labeled training data. However, the image distribution that such object detectors are trained on is more often times than not different from the targeted problem domain of interest. In this work, we scrutinize whether existing state-of-the-art object detectors have the ability to generalize across different domains. Specifically, we evaluate whether widely used pretrained state-of-the-art object detectors such as Faster-RCNN and YOLOv3 generalize to images sampled from an autonomous vehicle application. For this purpose, we evaluate the performance of detectors on localizing humans and vehicles on images from the KITTI dataset and report results of detailed subgroup analysis on multiple factors. Our analysis shows that the detectors exhibit different levels of performance on varying levels of object-object occlusion and object size. Moreover, we report the performance drop of the object detectors with different image-altering hazardous factors.

**Index Terms**—(Object Detection, Subgroup analysis, Autonomous Vehicle)

## I. INTRODUCTION

The ability to detect objects from images is arguably one of the most important aspects of many modern day vision based applications. The growing capability of recent object detectors [1]–[5] enabled them to be applied to a diverse set of problem domains ranging from applications in autonomous vehicles [6] to medical image analysis [7], only to name a few. What is common to all such object detectors is the use of a deep convolutional neural network based general purpose visual feature extractor backbone [8] combined with modules for detecting objects which often include a module responsible for localizing objects, a bounding-box regressor and a classifier [2].

Such modern state-of-the-art object detectors are very heavily parameterized neural networks which require large amounts of carefully labeled annotated data for training. Thus, the advances in curating larger datasets such as [9]–[11] with accurate object level annotations have fueled the progress in the field of visual object detection. Most of these large scale datasets consist of everyday images which cover a large set of common objects and serve as a general purpose pre-training datasets for object detectors. However, many realistic application of object detectors such as autonomous vehicles focuses on much narrower distribution of images and objects.

For example, in autonomous vehicle applications, vehicles and humans are observed from a particular point of view, all objects are viewed in an outdoor setting and actor positions and orientations adhere to a specific distribution which may be different to those of objects found in a common household. However, in practice we often assume that the performance of these object detectors readily transfers to our target applications. Therefore, many times the detectors are used as-is or fine-tuned using small amounts of available training data.

In this work, we wish to scrutinize this assumption by performing an in-depth subgroup analysis of the performance of commonly used pretrained object detectors such as Faster-RCNN and YOLOv3. We take the networks pretrained on MS COCO and test them on images from KITTI to test the detectors’ ability to generalize to images drawn from a different distribution. More specifically, we are interested in identifying in detail the strengths and weaknesses of the model with respect to different subgroups. We choose object occlusion level and object size as the main subgroups that we perform analysis on. We identify vehicles and humans to be the most important object types for many applications and perform subgroups analysis separately for the two object classes. Our analysis shows that our common assumption that object detectors transfer well across datasets is not always true. We find that the object detectors perform better for certain subgroups than others and the results provide helpful insights into potential directions to improve existing models as well as datasets.

## II. RELATED WORK

**Object detections:** In the object detection literature, there are two mainstream philosophies in designing object detectors. The first is a region proposal based architectures where the model first generates region proposals and later classifies them. The most notable architecture that follows this pipeline is the Faster-RCNN [2] and the Mask-RCNN [12]. The second type includes object detectors that pose the detection problem as a regression or classification problem by jointly predicting categories and locations directly. For this case, YOLO [3] is a well known architecture with very efficient implementations available. We refer to [13] for a thorough survey on the field of object detection. In this work, we perform our subgroup analysis on the two representative object detectors, Faster-RCNN and YOLOv3.

Faster R-CNN [2] adopts a new region proposal approach, using a Region Proposal Network that share convolution features with the Fast R-CNN detector, rather than using the traditional Selective Search algorithm. The approach allows for the increase in efficiency and accuracy due to the increased region proposal quality.

YOLOv3 [14] is an one-stage detector based on its predecessor: YOLOv2. It simultaneously predicts the class and location, making it considerably faster than some other state-of-the-art methods. YOLOv3 comes with many architectural changes compared to YOLOv2, such as multilabel classification instead of softmax and a new feature extraction network (Darknet-53), which is slower than the previously used Darknet-19, but much more accurate.

**Measuring the performance of object detectors:** The field of object detection has converged towards an universal metric to measure object detector performance, namely the mean Average-Precision (mAP). One computes mAP by measuring the area under the precision-recall curve for detections over multiple intersection-over-union (IoU) thresholds with which is then averaged over all classes to produce a single evaluation criteria [15]. While mAP provides a great overview of the general performance of a detector on a particular dataset, it hinders analysis of detection errors at a granular level. For example, a practitioner cannot intuitively isolate certain error types and cannot identify different factors that contribute to detection errors. In this work, we isolate different subgroups within the dataset, observe how the performance of a detector is affected by different image perturbations for each subgroup and thus provide much granular analysis of strengths and weaknesses of object detectors.

**Analyzing strengths and weaknesses of object detectors:** There has been many attempts to diagnose the errors of deep learning based object detectors in recent years. The seminal work of [16] provided tools necessary to perform a more in-depth analysis of false positive detections of the detector. Tools such as the COCO evaluation toolkit<sup>1</sup> extends the analysis of [16] by analyzing errors with respect to their effects on model's precision-recall characteristics. There also exists a recent work [17] that improves usability and interpretability while decreasing dataset dependency of the error analysis. However, all analyses mentioned above assume that the object detector is trained adequately on the target dataset using a adequately large set of annotated training images from the same dataset. However, there lacks detailed error analysis on widely used pretrained object detectors in their off-the-shelf form. In this work, we expose detailed performance characteristics of popular pretrained object detectors and compare how various image perturbations effect detector performance. We also provide granular analysis of object detector performance per different object subgroups such as object sizes and occlusion levels.

**Image datasets:** The pretrained object detectors used in the experiment were both trained on MS COCO [9]. It contains

a total of 2.5 million labeled instances over 328 thousand images covering 91 object types in their natural context. All images in MS COCO were collected from Flickr, a website hosting videos and photos shot by photographers, meaning most images in MS COCO are taken from a typical human eye perspective.

On the other hand, we are testing the object detectors on KITTI. KITTI is a dataset focused on providing annotated images for training and evaluating models in mobile robotics and autonomous driving applications [18]. Its 2D object detection benchmark contains 80,256 labeled instances across 14999 images in total. Unlike MS COCO, all images in KITTI were collected by high-resolution cameras mounted on a vehicle while driving around a mid-sized city. This implies that there will be fundamental differences between the context and perspective of the images between MS COCO and KITTI.

### III. GRANULAR ANALYSIS OF PRETRAINED OBJECT DETECTORS

Existing methods assess object detector performance using mAP which provides an overall summary of detector performance for all defined object classes averaged over multiple operating points. Instead, we wish to provide a more granular analysis of detector performance by measuring the effect of isolated factors. Thus, we fix the operating point of detectors at intersection-over-union (IoU) threshold of 0.5 but measure the performance of the detector across various subgroups. In this section, we define the subgroups and various image perturbations that we perform to measure how robust or fragile the pretrained object detectors are for each category.

#### A. Area Under the Curve as the Performance Metric

Area Under the Curve (AUC) is a commonly used performance metric for classification problems. We plot Receiver Operating Characteristic (ROC) curves showing precision-recall trade-offs for each subgroup and type of image perturbation. The precision and recall values of each image is calculated independently. The average precision and recall among images contained in a subgroup is used to plot its precision-recall curve. We then report AUC as the summary of the detector performance.

#### B. Defined Subgroups

To allow us to examine the performance of the pretrained object detectors in detail, we divided the dataset into many subgroups. On a more general level, all the objects in the dataset are divided into two subgroups: cars and humans. These two subgroups are arguably the most crucial prediction targets in autonomous vehicle applications. Those that aren't in either of the subgroups are excluded from the experiment. Among cars and humans, each object is further categorized into different subgroups according to their occlusion level and relative object size within cars/humans. The KITTI dataset provides each ground truth with four possible occlusion labels: visible, semi-occluded, fully occluded, and truncated, with

<sup>1</sup><http://cocodataset.org/#detection-eval>

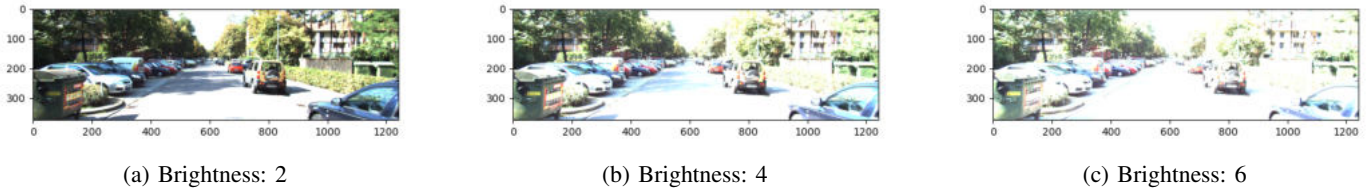


Fig. 1: Effect of Brightness Transformations on the Image

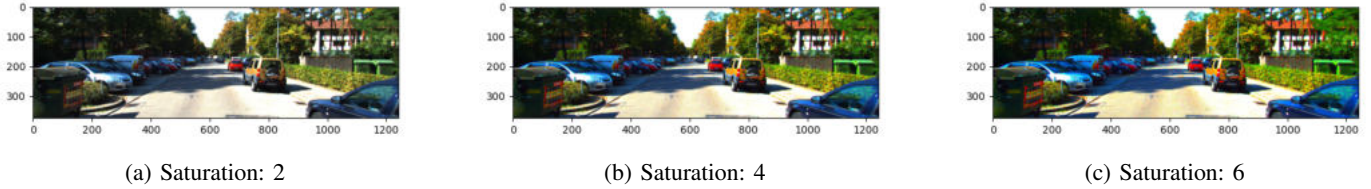


Fig. 2: Effect of Saturation Transformations on the Image

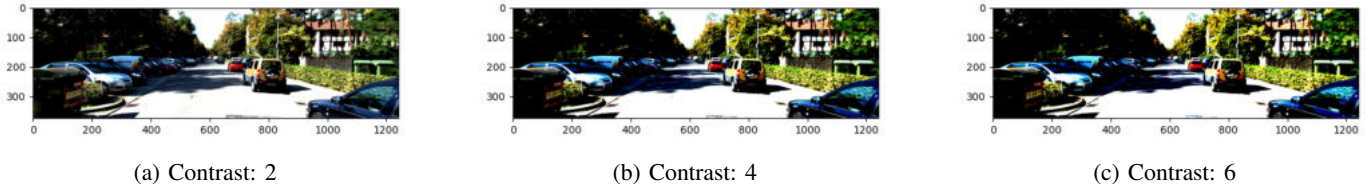


Fig. 3: Effect of Contrast Transformations on the Image

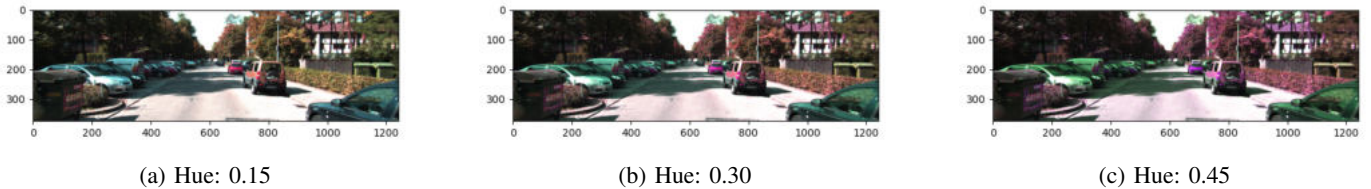


Fig. 4: Effect of Hue Jitters on the Image

each label assigned an occlusion level between 0 to 3, respectively. The occlusion level of an object in an image is defined by the average occlusion level of all ground truths in a given image, rounded down. Objects in a given image are labeled as either visible, semi-occluded, or fully occluded according to their average occlusion levels. No image contained only truncated objects, hence no objects were labeled as truncated. Objects in an image is labeled as being large if the average object size is in the top 50% among the all objects in that particular subgroup. Conversely, objects in an image with an average object size in the lower 50% are classified as being small.

### C. Performed Image perturbations

We selected a range of different image perturbations to discover how robust pretrained object detectors are when the image quality isn't ideal. More specifically, the image perturbations we employed includes Gaussian blur, brightness scaling, contrast scaling, saturation scaling, and hue jitter.

Although color jitter is a common technique in data augmentation, the result it produces would be too inconsistent for analysis, therefore all but hue transformation were done by applying a fixed scaling factor. Three scaling factor values are used for this experiment, ranging from 2, 4, and 6. Hue transformation was done using hue jitter because setting the entire dataset to a certain hue is unquantifiable. The hue jitter value is chosen uniformly from a range of  $[-n, n]$ , where  $n$  is the jitter factor. Since a value of  $-0.5/0.5$  is enough to transform the hue to the opposite side of the color wheel, 0.5 is regarded as the maximum value for the jitter factor. Hence, in this experiment, the values of jitter factor was chosen to be 0.15, 0.30, and 0.45. Lastly, Gaussian blur takes in two parameters: sigma and the corresponding kernel size. The sigma values used were chosen to be 1, 2, and 3, and the kernel size that matches each sigma value ranges from 5, 9, 13, respectively.

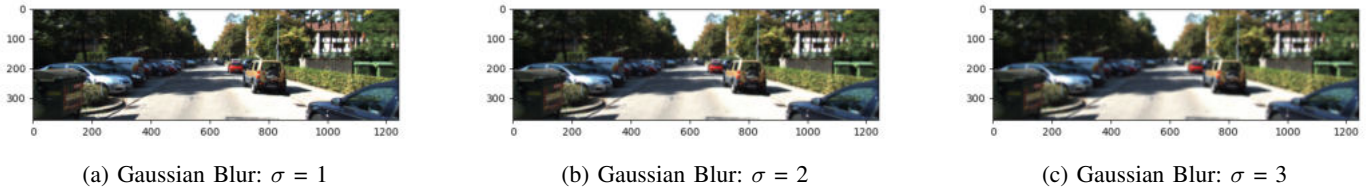


Fig. 5: Effect of Gaussian Blur on the Image

Occlusion Level	Vehicles														
	Gaussian Blur			Brightness			Contrast			Saturation			Hue		
	$\sigma = 1$	$\sigma = 2$	$\sigma = 3$	2	4	6	2	4	6	2	4	6	0.15	0.30	0.45
visible	0.686	0.527	0.303	0.675	0.616	0.537	0.562	0.510	0.469	0.697	0.674	0.649	0.696	0.704	0.710
semi-occluded	0.530	0.422	0.273	0.538	0.498	0.450	0.450	0.414	0.382	0.557	0.536	0.521	0.544	0.550	0.557
fully occluded	0.176	0.128	0.077	0.205	0.209	0.212	0.127	0.122	0.094	0.192	0.191	0.172	0.188	0.185	0.195
Object Size															
small	0.604	0.416	0.188	0.598	0.535	0.460	0.474	0.419	0.383	0.622	0.596	0.566	0.621	0.627	0.647
large	0.753	0.690	0.557	0.735	0.699	0.624	0.659	0.632	0.601	0.751	0.735	0.725	0.736	0.744	0.744

TABLE I: Overview of Faster R-CNN Performance on Vehicle Subgroup

Occlusion Level	Humans														
	Gaussian Blur			Brightness			Contrast			Saturation			Hue		
	$\sigma = 1$	$\sigma = 2$	$\sigma = 3$	2	4	6	2	4	6	2	4	6	0.15	0.30	0.45
visible	0.312	0.242	0.156	0.325	0.235	0.138	0.258	0.201	0.171	0.349	0.338	0.322	0.338	0.338	0.338
semi-occluded	0.099	0.072	0.040	0.100	0.087	0.056	0.074	0.058	0.052	0.105	0.101	0.099	0.107	0.112	0.110
fully occluded	0.047	0.033	0.022	0.035	0.020	0.008	0.027	0.022	0.019	0.046	0.044	0.039	0.044	0.045	0.041
Object Size															
small	0.156	0.094	0.042	0.171	0.128	0.072	0.119	0.093	0.076	0.178	0.170	0.162	0.170	0.173	0.172
large	0.547	0.486	0.392	0.489	0.335	0.207	0.435	0.349	0.303	0.540	0.525	0.504	0.532	0.532	0.525

TABLE II: Overview of Faster R-CNN Performance on Human Subgroup

Occlusion Level	Vehicles														
	Gaussian Blur			Brightness			Contrast			Saturation			Hue		
	$\sigma = 1$	$\sigma = 2$	$\sigma = 3$	2	4	6	2	4	6	2	4	6	0.15	0.30	0.45
visible	0.507	0.320	0.147	0.566	0.521	0.467	0.487	0.425	0.381	0.586	0.566	0.548	0.591	0.605	0.611
semi-occluded	0.430	0.299	0.153	0.474	0.436	0.395	0.395	0.351	0.311	0.473	0.460	0.454	0.481	0.481	0.481
fully occluded	0.186	0.121	0.046	0.229	0.181	0.142	0.140	0.082	0.078	0.218	0.214	0.212	0.223	0.213	0.224
Object Size															
small	0.436	0.240	0.082	0.502	0.453	0.403	0.417	0.348	0.311	0.526	0.494	0.478	0.525	0.542	0.554
large	0.606	0.481	0.320	0.620	0.589	0.544	0.560	0.525	0.478	0.635	0.635	0.626	0.641	0.643	0.643

TABLE III: Overview of YOLOv3 Performance on Vehicle Subgroup

#### IV. RESULTS

In this section, we first report our findings regarding the performance of pretrained Faster R-CNN on KITTI without any finetuning. We experiment with how Gaussian blur and brightness/hue/contrast/saturation transformations affect the model performance with respect to varying occlusion levels and object sizes. Tables I and II demonstrate that the effect of image perturbations is similar in both vehicles and humans, other than the fact that the model performance is generally lower for detecting humans. We visualize the effect of all image perturbations in Figures 1, 2, 3, 4 and 5.

We find that Gaussian blur has a large impact on the performance of the model. As the kernel size and sigma gets larger, we observe that there is large drop in the performance. The drop is also shown to be larger when sigma increases from 2 to 3 when compared to the increase from 1 to 2, suggesting that the impact of Gaussian blur increases exponentially as the level of blur increases. This increase is particularly evident in small

objects; the highest level of Gaussian blur caused the largest difference in performance between large and small objects. On the other hand, although objects that are completely visible and semi-occluded seem to be rather robust against Gaussian blur, objects that are fully occluded seem to suffer a lot more.

In Tables I and II, we also report the performance of the detector across transformations in brightness, contrast, saturation and hue. The Faster-RCNN model is generally robust against these types of color jitter and the performance is generally higher than that of under Gaussian blur. However, there are subtle differences between the effect of each color transformation. Contrast impacted model performance the most, resulting in the lowest AUC scores among color transformations across all occlusion levels and object sizes. In contrast, hue jitter had the least impact on model performance, resulting in either similar or even higher performance in all subgroups.

Next, we report our findings regarding the performance of

Occlusion Level	Humans														
	Gaussian Blur			Brightness			Contrast			Saturation			Hue		
	$\sigma = 1$	$\sigma = 2$	$\sigma = 3$	2	4	6	2	4	6	2	4	6	0.15	0.30	0.45
visible	0.224	0.162	0.092	0.242	0.212	0.160	0.177	0.141	0.118	0.238	0.238	0.219	0.237	0.237	0.237
semi-occluded	0.052	0.039	0.016	0.059	0.052	0.040	0.044	0.034	0.026	0.054	0.052	0.052	0.058	0.060	0.065
fully occluded	0.030	0.020	0.008	0.035	0.038	0.031	0.017	0.015	0.013	0.033	0.022	0.022	0.031	0.037	0.027
Object Size															
small	0.088	0.053	0.023	0.101	0.092	0.068	0.059	0.044	0.038	0.092	0.084	0.078	0.095	0.095	0.091
large	0.448	0.372	0.245	0.448	0.403	0.298	0.366	0.310	0.264	0.461	0.462	0.448	0.461	0.461	0.468

TABLE IV: Overview of YOLOv3 Performance on Human Subgroup

pretrained YOLOv3 without any finetuning on KITTI in Tables III and IV. Overall, YOLOv3 shows similar performance characteristics when compared against Faster R-CNN in many aspects. However, we observe that the general performance of YOLOv3 across all subgroups is lower than that of Faster R-CNN. While both being vulnerable to Gaussian blur, all other subgroups (including those from Faster R-CNN) only show a large performance drop from semi-occluded to fully occluded cases, but YOLOv3 already shows a large performance drop when going from visible to semi-occluded in the human subgroup. In the scope of color transformation, the results show that just like Faster R-CNN, YOLOv3 is also most vulnerable to contrast while being least affected by hue jitter.

## V. CONCLUSION

In this paper, we studied the performance of widely used pretrained object detectors, Faster-RCNN and YOLOv3. There are important conclusions that can be made based on our experimental results. First, both detectors show a performance drop from detecting cars compared to when detecting humans. We suspect this is because the humans are inherently smaller than vehicles and this leaves less margins of error for the models to draw prediction boxes that meet the IoU threshold. Secondly, even as some levels of perturbation have been shown to greatly distort the image, the effect on performance is still minimal compared to the effect of high occlusion levels and variations in object sizes. This suggests that we should focus primarily on guaranteeing the model’s consistency to detect occluded objects and smaller objects rather than potentially focusing on solving issues regarding color distortion and low-resolution images.

## REFERENCES

- [1] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick, “Detectron2,” <https://github.com/facebookresearch/detectron2>, 2019.
- [2] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds. 2015, vol. 28, Curran Associates, Inc.
- [3] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [4] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg, “Ssd: Single shot multibox detector,” in *ECCV (1)*, Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, Eds. 2016, vol. 9905 of *Lecture Notes in Computer Science*, pp. 21–37, Springer.
- [5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko, “End-to-end object detection with transformers,” *CoRR*, vol. abs/2005.12872, 2020.
- [6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele, “The cityscapes dataset for semantic urban scene understanding,” in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [7] Hao Dong, Guang Yang, Fangde Liu, Yuanhan Mo, and Yike Guo, “Automatic brain tumor detection and segmentation using u-net based fully convolutional networks,” in *Medical Image Understanding and Analysis*, María Valdés Hernández and Víctor González-Castro, Eds., Cham, 2017, pp. 506–517, Springer International Publishing.
- [8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
- [9] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár, “Microsoft coco: Common objects in context,” 2014, cite arxiv:1405.0312Comment: 1) updated annotation pipeline description and figures; 2) added new section describing datasets splits; 3) updated author list.
- [10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, June 2010.
- [11] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalanditis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei, “Visual genome: Connecting language and vision using crowdsourced dense image annotations,” 2016.
- [12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, “Mask r-cnn,” 2018.
- [13] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu, “Object detection with deep learning: A review,” 2018, cite arxiv:1807.05511.
- [14] Joseph Redmon and Ali Farhadi, “YOLOv3: An Incremental Improvement,” Tech. Rep., University of Washington, 04 2018.
- [15] Daniel Bolya, Sean Foley, James Hays, and Judy Hoffman, “Tide: A general toolbox for identifying object detection errors,” in *European Conference in Computer Vision (ECCV)*, 2020.
- [16] Derek Hoiem, Yodsawalai Chodpathumwan, and Qieyun Dai, “Diagnosing error in object detectors,” in *Computer Vision – ECCV 2012*, Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, Eds., Berlin, Heidelberg, 2012, pp. 340–353, Springer Berlin Heidelberg.
- [17] Daniel Bolya, Sean Foley, James Hays, and Judy Hoffman, “TIDE: A general toolbox for identifying object detection errors,” in *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part III*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, Eds. 2020, vol. 12348 of *Lecture Notes in Computer Science*, pp. 558–573, Springer.
- [18] Andreas Geiger, Philip Lenz, and Raquel Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361.

# Irregular Repetition Slotted ALOHA Scheme with Multi-Packet Reception in Packet Erasure Channel

Chundie Feng\*, Xuhong Chen<sup>‡</sup>, Zhengchuan Chen\*<sup>§</sup>, Zhong Tian\*, Yunjian Jia\*, and Min Wang<sup>†</sup>

\*School of Microelectronics and Communication Engineering, Chongqing University, Chongqing, China

<sup>‡</sup>China Development Bank, China

<sup>§</sup>National Mobile Communications Research Laboratory, Southeast University, Nanjing, China

<sup>†</sup>School of Optoelectronics Engineering, Chongqing University of Posts and Telecommunications, China

Email: \*{fcd, czc, ztian, yunjian}@cqu.edu.cn, <sup>‡</sup>chenxh13@mails.tsinghua.edu.cn, <sup>†</sup>wangm@cqupt.edu.cn

**Abstract**—In massive machine-type communications (mMTC), a large amount of devices demand to access to the network via a commonly sharing wireless channel. Uncoordinated frequent channel contention associated with channel variation bring a big challenge for improving the performance of massive access system. Irregular repetition slotted ALOHA (IRSA), as a typical grant-free random access protocol, exploits collided signals for packets recovery and possesses a great potential in improving the access capability and throughput of the mMTC system. In this paper, we analyze the performance of IRSA scheme with multi-packet reception where the access point (AP) can simultaneously retrieve multiple packets in a collided signal under packet erasure channel. In particular, decoding failure probabilities during each round of iterative packet recovery are expressed clearly, based on which the packet loss ratio and the throughput of the system are characterized in detail. Simulation results validate the correctness of the theoretical analyses. By selecting appropriate distribution of the replica number of packet, the error floor caused by the packet erasure can be decreased. Besides, the throughput of the massive access system can be improved by optimizing the link load of the IRSA scheme.

**Index Terms**—Irregular repetition slotted ALOHA, multiple packets reception, packet erasure channel, successive interference cancellation.

## I. INTRODUCTION

As one of the most promising applications of 5G, the Internet of Things (IoT) aims to connect billions of devices to completely change our current lifestyle [1]. To meet the massive connection demand of the IoT, massive machine type communication (mMTC) is regarded as one of three application scenarios for 5G [2].

However, in wireless networks where bandwidth is limited and multipath fading exists, it is still a challenge to provide reliable massive access for mMTC [3]. Uplink transmissions from the devices to the access point meet frequent signal collisions, bringing difficulties in improving the performance of massive access system [4]. It is noted that when the number of user equipments (UEs) reaches a certain level, scheduling-based access protocols would cause extremely high access

delay and utility loss due to complex signaling information interaction [5]. In contrast, ALOHA-based access protocol family, allowing UEs sharing wireless resources without specific scheduling, becomes popular in forming grant-free random access system [6].

Traditional ALOHA-based random access schemes, such as slotted ALOHA scheme [7], directly discard collided signals caused by uncoordinated packet transmission. Hence, the channel utility, i.e., the normalized throughput of the random access system is limited and is insufficient to support mMTC [8]. In recent years, some improved ALOHA schemes proposed to make full use of collided signal, promoting the normalized throughput significantly [9]–[12]. For example, to make use of the collided signals, contention resolution diversity slotted ALOHA (CRDSA) was proposed in [9] where a UE would send its packet and a packet replica to different slots. When any packet replica is successfully demodulated, the other replica can be cancelled with the help of the replica position pointer enclosed by the packet. That is, the interference caused by the demodulated packet is removed.

The irregular repetition slotted ALOHA (IRSA) proposed in [10] adopted the similar idea as CRDSA except that a UE can send a random number of replicas of a packet rather than two. In particular, the number of replicas is determined by the corresponding UE according to the probability distribution function, which is the so-called irregular repetition [11]. Moreover, the pointer owned by the packet can find the location of all other packets sent by the same UE [13]. By using successive interference cancellation (SIC) adopted in CRDSA, collision among packets is well resolved and the throughput of the system is enhanced remarkably. It can be seen that CRDSA and IRSA are both simple repetition of the burst, using interference elimination to turn the conflict burst into a treasure. Furthermore, [14] considered the combination of multi-antenna technology and IRSA scheme. Compared with traditional IRSA scheme, this scheme can support greater system link load. It is worth noting that IRSA can also be used in the industrial IoT. The age of information of IRSA protocol was studied for the first time in [15], which proved the potential of modern random access technology in information freshness.

Then, to further improve the successful packets delivery

This work was supported in part by the National Natural Science Foundation of China under Grant 61901066, Grant 61971077, in part by Graduate Research and Innovation Foundation of Chongqing, China (Grant No. CY-B21067), in part by the Chongqing Science and Technology Commission under Grant cstc2019jcyj-msxmX0575, and in part by the open research fund of National Mobile Communications Research Laboratory, Southeast University (No. 2021D13).

probability, coded slotted ALOHA (CSA) was proposed in [12] based on IRSA. Different from the repetition of raw packets in IRSA scheme, the raw packets would be encoded before transmission by using well-designed local packet-level codes in the CSA scheme. It is noticed that the IRSA scheme can be regarded as one typical case of the CSA scheme, where all local coding scheme at devices are repetition coding. Taking the multipath fading and complex interference environment of wireless networks into account, the authors in [16] evaluated the performance of CSA scheme in both packet erasure channels and slot erasure channels.

While the aforementioned works considered that a packet only can be recovered in a slot where no collision signal exists after implementing SIC, advanced communication technologies such as multi-antenna receiver and power capture effect provide for access point (AP) a capable of retrieving multiple packets simultaneously in a collided signal. Motivated by this, the authors in [14] explored the packet loss ratio and the throughput of IRSA scheme in a system where the AP has the multi-packet reception capability. In a recent work, the optimal distribution of the replica number of packet in IRSA scheme have been explored for mMTC system where the AP can retrieve two packets simultaneously [17].

To the best of our knowledge, there is no existing work evaluating the transmission performance of improved ALOHA-based protocol in wireless access systems with multi-packet reception capability and packet erasures. In fact, as the instability of wireless channel is inherent, incidental decoding failure is commonly observed at the AP end even though the multi-packet reception is adopted. While the CSA scheme requires extra computing complexity at all devices, the IRSA scheme avoids the packet level computing at the transmitter side. Motivated by this, we in this work investigate the overall packet loss ratio and throughput performance of the IRSA scheme in a massive wireless access system with multi-packet reception under packet erasure channel. In particular, the incidental decoding failure caused by channel variation is modeled as packet erasure during the transmissions. The main contributions of this work are summarized as follows.

- 1) We derive the decoding error probabilities of the IRSA scheme during the packet recovery process when the AP can demodulate multiple packets simultaneously under packet erasure channel.
- 2) We characterize the packet loss ratio and the throughput in detail. Numerical results show that by selecting appropriate replica distribution of packets, the packet loss ratio floor can be decreased. Moreover, the throughput can be improved by optimizing the link load.

The rest of this paper is organized as follows. In Section II, the system model for IRSA scheme with multiple packets reception capability of AP under packet erasure channel is established. In Section III, we propose the implicit conditions of decoding failure probability during packets recovery when the AP can demodulate multiple packets under packet erasure channel. In Section IV, the system performance is analyzed

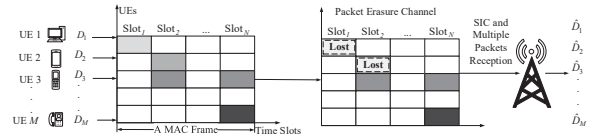


Fig. 1. The schematic diagram of the IRSA-scheme-enabled multi-access system with multi-packet reception under erasure channels. For  $i \in \{1, 2, \dots, M\}$ ,  $D_i$  and  $\hat{D}_i$  represent the raw data and the recovered data at the AP corresponding to UE  $i$ .

under packet erasure channel. Numerical results are presented in Section V. The conclusions are given in Section VI.

## II. SYSTEM MODEL

A multi-access system where  $M$  UEs attempt to transmit data to an AP by sharing the same wireless medium is considered as shown in Fig. 1. Synchronization is assumed for all the communication patterns. Transmissions are organized into consecutive medium access control (MAC) layer frames where each frame is further divided into  $N$  slots. IRSA scheme is adopted at all UEs for countering possible signal collisions and packet recovery failures at the AP, i.e., each UE would repetitively send its packet in multiple slots while the number of packet replicas follows a designed distribution. For clarity, the slot length is assumed to be the same as the packet length, i.e., a packet just fills one slot. Since the wireless channel between UEs and the AP usually experience fading, the successful recovery of the encoded packet from the AP can not be guaranteed even when there is no signal collision in each slot. We model this event as packet erasure and denote the probability of such packet recovery failure by  $\epsilon$  [16]. By employing multi-antenna technology, the AP is assumed to retrieve up to  $K$  multiple packets from each slot. That is, multi-packet reception capability is considered at the AP. To further improve the access efficiency, SIC is adopted at the AP, combating the mutual signal pollution by cancelling the known message from the mixed signals.

### A. Iterative Principle of SIC at the AP

Note that it is possible that there are more than  $K$  UEs choosing to transmit packets in a particular slot. Hence, even though multi-packet reception is adopted, the AP might fail to recover all the packets transmitted in a particular slot. To further improve the decoding capability, SIC is introduced in the IRSA-based random access framework. Specifically, based on some retrieved decoded packets, the AP tries to recover other packets, reconstruct the signal of the recovered packets, and subtract those signals in the received noisy signal of the corresponding slot the signal being transmitted. This procedure would bring more slots where the AP can retrieve decoded packets by multi-packet reception and can be iteratively operated at the AP until no slot can provide new recovered packets.

To analyze the SIC procedure, graph  $\mathcal{G} = (\mathcal{M}, \mathcal{N}, \mathcal{E})$  is usually introduced to characterize the IRSA scheme. Specifically,



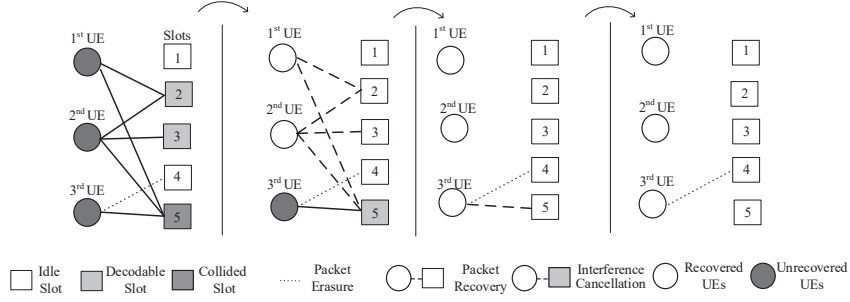


Fig. 2. Bipartite graph representation of the SIC iterative process under packet erasure channel,  $K = 2$ . In particular, encoded packets transmitted in slot 2 and slot 3 can be recovered in the first iteration because of multiple packets reception ability of the AP.

$\mathcal{M}$ ,  $\mathcal{N}$ , and  $\mathcal{E}$  represent the set of  $M$  UE nodes (UNs), the set of  $N$  slot nodes (SNs), and the set of edges connecting UNs and SNs, respectively. If  $i$ th UE choose to transmit a packet in  $j$ th slot, then there is an edge connecting  $i$ th UN with  $j$ th SN. Fig. 2 presents an example of the SIC procedure where  $M = 3$  and  $N = 5$ .  $K = 2$  is considered for the multi-packet reception capability at the AP side. The packet recovery begins from slot 2 and slot 3 using multi-packet reception, which retrieves the raw packets transmitted from 1st and 2nd UEs. According to IRSA scheme, the transmitted signals of 1st and 2nd UEs can be reconstructed, which are used for subtracting known signals from that received in slot 5. As the signals collided in slot reduces, more packets can be recovered from slot 5 and this decoding-subtracting process can be iteratively operated. Finally, all the raw data of the three UEs are recovered. It is noted that one of the packets transmitted by 3rd UE is missed due to packet erasure of slot 4 in the example.

### B. Degree Distribution

As shown in Fig. 2, there are edges connected to UNs and SNs in graph  $\mathcal{G}$ . Let us define the number of edges connected to a node as node degree. Accordingly, the polynomial representations of UN degree distribution  $\Lambda(x)$  and SN degree distribution  $\Psi(x)$  can be expressed respectively as

$$\Lambda(x) = \sum_{l=1}^N \Lambda_l x^l, \quad \Psi(x) = \sum_{l=1}^M \Psi_l x^l, \quad (1)$$

where  $\Lambda_l$  represents the probability that the degree of UN is  $l$  and  $\Psi_l$  represents the probability that the degree of SN is  $l$ . From Eq. (1), one can quickly get the average degree of UN  $\Lambda'(1) = \sum_{l=1}^N \Lambda_l l$  and the average degree of SN  $\Psi'(1) = \sum_{l=1}^M \Psi_l l$ . In addition, the degree distributions can also be defined from an edge's perspective as

$$\lambda(x) = \sum_{l=1}^N \lambda_l x^{l-1}, \quad \rho(x) = \sum_{l=1}^M \rho_l x^{l-1}, \quad (2)$$

where  $\lambda_l$  represents the probability that an edge is connected to a UN with node degree  $l$  and  $\rho_l$  represents the probability that an edge is connected to a SN with node degree  $l$ . Then,

the following relations always hold by definition [10]:

$$\lambda(x) = \Lambda'(x)/\Lambda'(1), \quad \rho(x) = \Psi'(x)/\Psi'(1). \quad (3)$$

Recall that UEs randomly select slots to send packets. The probability that a specific UE sends packet in a slot of interest is  $\Lambda'(1)/N$ . Thus, it has

$$\Psi_l = \binom{M}{l} \left( \frac{\Lambda'(1)}{N} \right)^l \left( 1 - \frac{\Lambda'(1)}{N} \right)^{M-l}. \quad (4)$$

Substituting Eq. (4) into Eq. (1), the degree distribution of SN admits that

$$\Psi(x) = \left( 1 - \frac{\Lambda'(1)}{N} (1-x) \right)^M. \quad (5)$$

In particular, for large enough  $M$ , the number of UEs sending packets in each slot tends to be Poisson distributed. Hence, one has

$$\Psi(x) = \exp(-G\Lambda'(1)(1-x)), \quad (6)$$

$$\rho(x) = \Psi(x)/\Psi'(1) = \exp(-G\Lambda'(1)(1-x)), \quad (7)$$

where

$$G := \frac{M}{N} \quad (8)$$

represents the link load of the system, i.e., the average number of UEs choosing to transmit in each slot.

### III. DECODING FAILURE PROBABILITY ANALYSIS FOR PACKETS RECOVERY

Due to the existence of channel erasure, using SIC and multi-packet reception can not guarantee that all the packets transmitted to be recovered in IRSA scheme since it is possible that all the encoded packets are missing. The decoding failure probability of a packets becomes one of the key performance of the CSA scheme. In this section, we analyze the decoding failure probability of the multi-packet-reception-based IRSA scheme under packet erasure channel.

Under the packet erasure channel, some of the packets would be missing at the AP end or equivalently, some of the edges would be erased in  $\mathcal{G}$ . We consider an unerased packet transmitted in a slot with degree  $l$  in  $\mathcal{G}$ . After the packet/edge erasure, the probability that there remains  $v - 1$

interfering packets, or equivalently, observing from the corresponding unerased edge, the SN degree reduced from  $l$  to  $v$  is  $\binom{l-1}{v-1}(1-\epsilon)^{v-1}\epsilon^{l-v}$ . Accordingly, the SN degree distribution observed from an unerased edge in  $\mathcal{G}$  is denoted as

$$\begin{aligned}\tilde{\rho}(x) &:= \sum_{l=1}^M \rho_l \sum_{v=1}^l \binom{l-1}{v-1} (1-\epsilon)^{v-1} \epsilon^{l-v} x^{v-1} \\ &= \rho((1-\epsilon)x + \epsilon).\end{aligned}\quad (9)$$

Recall that in each round of SIC, the iteration starts from the SNs by checking whether packets enclosed in the corresponding interference-cancelled signals can be recovered using multi-packet reception. We say a SN is recovered when the unerased packets transmitted in that slot are recovered, and vice versa. Similarly, we call a UN is recovered if all of its packets are recovered at the AP side or not yet. Randomly selecting an edge in  $\mathcal{G}$ , there exists a probability that the connected SN is unrecovered. Let us denote this probability at the end of the  $i$ th round of iteration by  $p_i$ . Likewise, we denote the probability that the connected UN is unrecovered after the  $i$ th round of iteration by  $q_i$ . Along with the iteration of SIC, both probabilities would be updated. For example, before the first round of iteration,  $q_0 = 1$  while  $p_0 < 1$  if there exists some SNs that less than or equal to  $K$  packets are transmitted in that slot. Using multi-packet reception, some of the packets would probably be recovered and  $q_1 < 1$ . Those recovered packets would contribute to the signal cancellation at the SN side and reduce the number of unknown packets in a slot, yielding a  $p_1$  less than  $p_0$ .

Based on the IRSA scheme, unrecovered probability of a SN  $p_i$  can be expressed as the following lemma.

**Lemma 1.** *Consider an AP with multi-packet reception capability  $K$  and packet erasure probability  $\epsilon$ . Given the  $i$ th round of iteration of unrecovered probability  $q_i$  for the UNs in the IRSA scheme, the unrecovered probability of an SN at the end of the  $i$ th iteration is characterised as*

$$\begin{aligned}f(q_i) = p_i &:= 1 - \sum_{k=0}^{K-1} \frac{q_i^k}{k!} \left( -G\Lambda'(1)(1-\epsilon) \right)^k \\ &\times \exp \left( -G\Lambda'(1)(1-\epsilon)q_i \right).\end{aligned}\quad (10)$$

*Proof:* First, let us analyze the probability that an edge connected to a degree- $v$  SN in  $\mathcal{G}$  can not be recovered after the  $i$ th iteration of SIC, which we denote it by  $p_i^{(v)}$ . Specifically, with multi-packet reception capability  $K$ , if  $v - K$  packets transmitted in a slot have been successfully recovered, the remaining  $K$  packets can be recovered. Hence, we have that

$$p_i^{(v)} = 1 - \sum_{k=0}^{K-1} \binom{v-1}{k} (1-q_i)^{v-k-1} q_i^k. \quad (11)$$

Accordingly, the probability of interest  $p_i$  can be expressed as

$$p_i = \sum_{l=1}^M \rho_l \sum_{v=1}^l \binom{l-1}{v-1} (1-\epsilon)^{v-1} \epsilon^{l-v} p_i^{(v)}$$

$$\begin{aligned}&= \sum_{l=1}^M \sum_{v=1}^l \binom{l-1}{v-1} (1-\epsilon)^{v-1} \epsilon^{l-v} \rho_l \\ &\times \left( 1 - \sum_{k=0}^{K-1} \binom{v-1}{k} (1-q_i)^{v-k-1} q_i^k \right) \\ &= 1 - \sum_{k=0}^{K-1} q_i^k \sum_{l=1}^M \rho_l \sum_{v=1}^l \binom{l-1}{v-1} (1-\epsilon)^{v-1} \epsilon^{l-v} \\ &\times \binom{v-1}{k} (1-q_i)^{v-k-1} \\ &\stackrel{(a)}{=} 1 - \sum_{k=0}^{K-1} \frac{q_i^k}{k!} \tilde{\rho}^{(k)}(1-q_i),\end{aligned}\quad (12)$$

where (a) follows from the first equality of Eq. (9).

On the other hand, it is valuable to note from Eq. (7) and the last equality of Eq. (9) that

$$\tilde{\rho}(x) = \rho((1-\epsilon)x + \epsilon) = \exp \left( -G(1-\epsilon)\Lambda'(1)(1-x) \right). \quad (13)$$

Taking the  $k$ -th order derivative of  $\tilde{\rho}(x)$ , it has that

$$\tilde{\rho}^{(k)}(x) = (G(1-\epsilon)\Lambda'(1))^k \exp \left( -G(1-\epsilon)\Lambda'(1)(1-x) \right). \quad (14)$$

Substituting Eq. (14) into Eq. (12), we can obtain Eq. (10). ■

Eq. (10) manifests the updating process from  $q_i$  to  $p_i$  in the SIC. According to Lemma 1, it can be seen that the link load  $G$ , erasure rate  $\epsilon$  and multi-packets reception capability  $K$  would significantly affect the efficiency of SIC.

On the other hand, unrecovered probability  $q_i$  can also be computed as a function of  $p_{i-1}$ . In particular,  $q_i$  is closely related to the UEs degree distribution. By combining the functions  $p_i = f(q_i)$  and  $q_i = g(p_{i-1})$ , one can characterize the recursion function of unresolved probability of packet erasure channel during the SIC iteration process.

**Theorem 1.** *The updating process from  $p_{i-1}$  to  $p_i$  is*

$$\begin{aligned}p_i &= 1 - \sum_{k=0}^{K-1} \frac{[\lambda((1-\epsilon)p_{i-1} + \epsilon)]^k}{k!} \left( G(1-\epsilon)\Lambda'(1) \right)^k \\ &\times \exp \left( -G(1-\epsilon)\Lambda'(1)\lambda((1-\epsilon)p_{i-1} + \epsilon) \right).\end{aligned}\quad (15)$$

*Proof:* The probability that the desired packet cannot be recovered is equal to the probability that none of the remaining packets sent by the same UE can be recovered. Therefore, the probability that the packets cannot be decoded in the  $i$ th iteration can be expressed as

$$\begin{aligned}q_i &:= \sum_{l=1}^N \lambda_l \sum_{j=1}^l \binom{l-1}{j-1} (1-\epsilon)^{j-1} \epsilon^{l-j} p_{i-1}^{j-1} \\ &= \lambda((1-\epsilon)p_{i-1} + \epsilon) = g(p_{i-1}).\end{aligned}\quad (16)$$

By plugging  $q_i$  into Eq. (10), the recursion function Eq. (15) follows. ■

Theorem 1 reveals how the distribution  $\{\Lambda_l\}$  affects the updating process from  $p_{i-1}$  to  $p_i$  associated with the effect

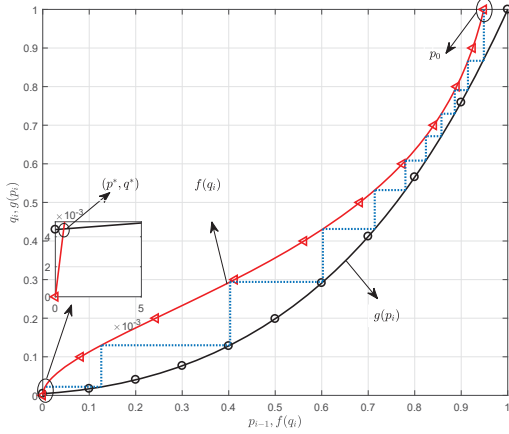


Fig. 3. Iteration of decoding failure probability  $p_i$  and  $q_i$ . In this case,  $\Lambda(x) = 0.5162x^3 + 0.2978x^4 + 0.1287x^5 + 0.0445x^6 + 0.0128x^7$ ,  $\epsilon = 0.1$ , and  $K = 2$ .

of parameters  $G$  and  $K$ . One can optimize  $\{\Lambda_l\}$  and these parameters to reduce the unrecovered probability and improve efficiency of the IRSA system in the packet erasure channel.

With specific parameters, the updating process between  $p_i$  and  $q_i$  can be visualized based on Eq. (15). For example, let us consider  $\Lambda(x) = 0.5162x^3 + 0.2978x^4 + 0.1287x^5 + 0.0445x^6 + 0.0128x^7$  and an AP with multi-packet capability of  $K = 2$ . We can obtain that the average degree distribution is 3.7399. According to Eq. (8),  $G = 1.4$ , and  $\lambda(x) = 0.4141x^2 + 0.31855x^3 + 0.1721x^4 + 0.0714x^5 + 0.0239x^6$ . In this case, we can see that

$$p_i = 1 - \exp(-4.7123\lambda(0.9p_{i-1} + 0.1)) \times (1 + 4.7123\lambda(0.9p_{i-1} + 0.1)). \quad (17)$$

The corresponding iteration process of the decoding failure probability is shown in Fig. 3 as the dotted line while  $\epsilon$  takes value 0.1. As shown in Fig. 3, for the first round of iterative decoding,  $p_0 = 0.9487$ ,  $p_1 = 0.8669$ ,  $p_2 = 0.7909$ . When  $p_i = p_{i-1} = p^*$ , the iteration stops, as marked in Fig. 3. It is noticed that  $(p^*, q^*)$  is just the intersection of  $f(q_i)$  and  $g(p_i)$ .

#### IV. THE PACKET LOSS RATIO AND THE THROUGHPUT

The unrecovered packets contribute to the packet loss of each frame of packet transmission in IRSA scheme. From the Eq. (15), when  $p_i = p_{i-1}$ , the iteration stops. We define this stopping unrecovered probability as  $p^*$ . Note that the unrecovered probability  $p^*$  represents the packet unrecovered probability observed from an edge in the graph  $\mathcal{G}$  and it is different from the packet loss ratio observed from UN. Let us denote the probability that the AP can not recover all the packets transmitted from a UN by  $P_{\text{err}}$ . Recall that packet loss occurs in packets transmitted through the packet erasure channel. We consider the UN with  $l$  connections without packet erasure in the bipartite graph. Due to the existence of packet erasure channel, the degree of UN would reduce from  $l$  to  $v$ . Hence, we can obtain the probability  $p^{*v}$  which means

the packet loss ratio with the UN of degree  $v$  under packet erasure channel. Then it has

$$P_{\text{err}} = \sum_{l=1}^M \Lambda_l \sum_{v=1}^l \binom{l}{v} (1 - \epsilon)^v \epsilon^{l-v} p^{*v}. \quad (18)$$

Based on the derived packet loss ratio  $P_{\text{err}}$ , one can further analyze the throughput of the CSA system, which is defined as the average number of successfully transmitted packets per slot. Denote the throughput by  $\Gamma$ . Recall that there are  $M$  UEs in the considered system. Hence, the number of packets successfully transmitted is  $sM(1 - P_{\text{err}})$ . Further dividing the slot number  $N$  of each frame, one can express the throughput of system as

$$\Gamma = \frac{M(1 - P_{\text{err}})}{N}. \quad (19)$$

From Eq. (19), one can find that the system throughput  $\Gamma$  is strongly related to the UE degree distribution  $\Lambda(x)$ , erasure probability  $\epsilon$  and the demodulation capability of the AP, i.e.,  $K$ .

#### V. SIMULATION RESULTS

Let us present some simulation results to investigate the packet loss ratio and the throughput of the considered system.

Fig. 4 depicts how the packet loss ratio  $P_{\text{err}}$  varies with the link load  $G$  under the packet erasure channel. We set the UEs degree distribution as  $\Lambda_1(x) = 0.5162x^2 + 0.2978x^3 + 0.1287x^4 + 0.0445x^5 + 0.0128x^6$  which is suitable for  $\epsilon = 0$  and  $K = 2$  [17],  $\Lambda_2(x) = x\Lambda_1(x)$  and  $K = 2$ . It can be seen from Fig. 4 that  $P_{\text{err}}$  increases with the growth of  $G$  for all cases where a critical  $G$  exists, beyond which  $P_{\text{err}}$  jumps to approach 1 and the system performance deteriorates dramatically. This is because with the increase of link load, the mixed signals within the system are more difficult to demodulate. For all considered  $\Lambda(x)$  and  $\epsilon$ , there exists a packet loss ratio floor caused by packet erasure. Note that when the link load  $G < 1.7$ ,  $\epsilon = 0.1$  with  $\Lambda_1(x)$ , the packet loss ratio  $P_{\text{err}}$  is about  $10^{-2}$ . When the link load  $G < 1.5$ ,  $\epsilon = 0.1$  with  $\Lambda_2(x)$ , the packet loss ratio  $P_{\text{err}}$  is about  $10^{-3}$ . Particularly, for  $\Lambda_1(x)$ , the error floor decreases while the critical  $G$  increases when  $\epsilon$  increases from 0.1 to 0.2. This implies that for different  $\epsilon$ , the UE degree distribution should be optimized such that a better error floor and throughput can be achieved. Comparing the packet loss ratio in theory and the simulation results, one can observe that the analytical results well approximate the simulated results for different erasure probability cases. More importantly, one can find from Fig. 4 that when there is no quadratic term of  $x$  in the distribution function, i.e., using  $\Lambda_1(x)$  and  $\Lambda_2(x)$ , the error floor caused by packet erasure can be obviously reduced. Hence, by adjusting the degree distribution function, we can alleviate the effect of erasure on  $P_{\text{err}}$ .

To intuitively observe the effect of different  $K$  for different link load  $G$  under packet erasure channel, we focus on the curves of throughput which varies with the system parameters such as link load  $G$  and  $K$  as shown in Fig. 5. In particular, we

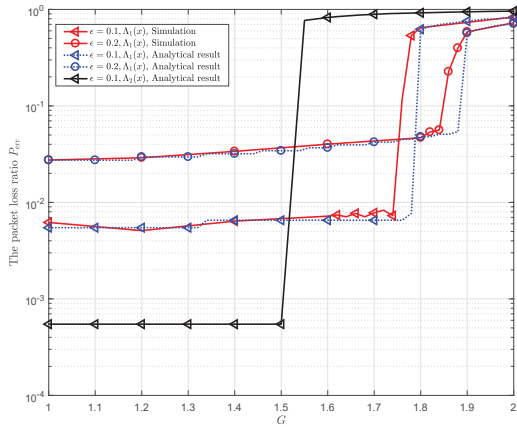


Fig. 4. The packet loss ratio  $P_{\text{err}}$  v.s. the link load  $G$  under packet erasure channel,  $K = 2$ ,  $N = 4000$ .

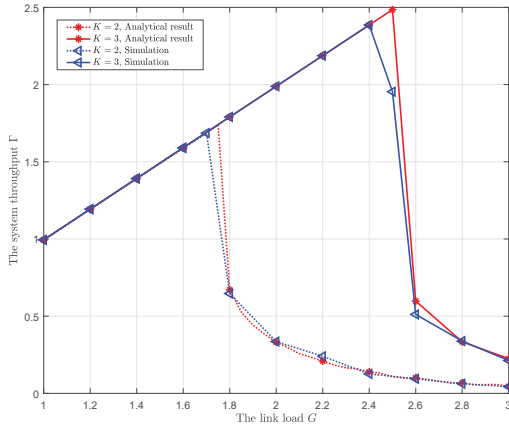


Fig. 5. The system throughput  $\Gamma$  v.s. the link load  $G$  under packet erasure channel.  $\epsilon = 0.1$ , UE degree distribution is  $\Lambda_1(x)$ ,  $N = 4000$ .

set  $\epsilon = 0.1$ . One can find that with the increase of link load  $G$ , the system throughput would increase from 1. Moreover, the throughput can achieve 2.4 when  $K = 3$  and the throughput can only achieve 1.7 when  $K = 2$  under packet erasure channel. This shows that throughput can increase as  $K$  increases. We can conclude that the capability of AP has a critical impact on the throughput. The better the performance of the AP, the higher throughput of the system can achieve. Moreover, the simulation of system throughput is expressed and we can compare the simulated throughput with the analytical throughput. The simulated throughput well approximates the analytical results under packet erasure channel which validates the correctness of the theoretical analyses.

## VI. CONCLUSION

In this paper, we establish a multiple packets reception system model based on IRSA scheme under packet erasure channel. The analysis framework of the IRSA scheme with

the APs multiple packets reception capability is formulated. Then, implicit conditions of the decoding failure probabilities during each round of packet recovery of the SIC process are derived explicitly. Based on the established convergence equation, we characterise the packet loss ratio and throughput in detail. Afterwards, we present numerical results about how the performance varies with the parameters such as  $\epsilon$ ,  $G$ ,  $K$  under packet erasure channel. It is concluded that throughput can be improved by optimizing the link load  $G$ . Moreover, some particular UE degree distributions can decrease error floor under erasure channels.

## REFERENCES

- [1] J. Yu, P. Zhang, L. Chen, J. Liu, R. Zhang, K. Wang, and J. An, "Stabilizing frame slotted ALOHA-based IoT systems: A geometric ergodicity perspective," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 3, pp. 714–725, 2021.
- [2] M. Shirvanimoghaddam, M. Dohler, and S. J. Johnson, "Massive non-orthogonal multiple access for cellular IoT: Potentials and limitations," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 55–61, 2017.
- [3] C. Bockelmann, N. K. Pratas, G. Wunder, S. Saur, M. Navarro, D. Gregoratti, G. Vivier, E. De Carvalho, Y. Ji, Čedomir Stefanović, P. Popovski, Q. Wang, M. Schellmann, E. Kosmatos, P. Demestichas, M. Raceala-Motoc, P. Jung, S. Stanczak, and A. Dekorsy, "Towards massive connectivity support for scalable mMTC communications in 5G networks," *IEEE Access*, vol. 6, pp. 28 969–28 992, 2018.
- [4] M. B. Shahab, R. Abbas, M. Shirvanimoghaddam, and S. J. Johnson, "Grant-free non-orthogonal multiple access for IoT: A survey," *IEEE Commun. Surv. Tutor.*, vol. 22, no. 3, pp. 1805–1838, 2020.
- [5] M. Hasan, E. Hossain, and D. Niyato, "Random access for machine-to-machine communication in LTE-advanced networks: issues and approaches," *IEEE Commun. Mag.*, vol. 51, no. 6, pp. 86–93, 2013.
- [6] L. Liang, E. G. Larsson, Y. Wei, P. Petar, S. Cedomir, and D. C. Elisabeth, "Sparse signal processing for grant-free massive connectivity: A future paradigm for random access protocols in the Internet of Things," *IEEE Signal Process. Mag.*, vol. 35, no. 5, pp. 88–99, 2018.
- [7] Roberts and G. Lawrence, "ALOHA packet system with and without slots and capture," *ACM Sigcomm Computer Communication Review*, vol. 5, no. 2, pp. 28–42, 1975.
- [8] S. Böcker, C. Arendt, P. Jörke, and C. Wietfeld, "LPWAN in the context of 5G: Capability of LoRaWAN to contribute to mMTC," in *Proc. IEEE 5th World Forum Internet Things (WF-IoT)*, 2019, pp. 737–742.
- [9] E. Casini, R. De Gaudenzi, and O. Del Rio Herrero, "Contention resolution diversity slotted ALOHA (CRDSA): An enhanced random access scheme for satellite access packet networks," *IEEE Trans. Wirel. Commun.*, vol. 6, no. 4, pp. 1408–1419, 2007.
- [10] G. Liva, "Graph-based analysis and optimization of contention resolution diversity slotted ALOHA," *IEEE Trans. Commun.*, vol. 59, no. 2, pp. 477–487, 2011.
- [11] E. Paolini, G. Liva, and M. Chiani, "High throughput random access via codes on graphs: Coded slotted ALOHA," in *2011 IEEE Int. Conf. Commun. (ICC)*, 2011, pp. 1–6.
- [12] —, "Coded slotted ALOHA: A graph-based method for uncoordinated multiple access," *IEEE Trans. Inf. Theory*, vol. 61, no. 12, pp. 6815–6832, 2015.
- [13] J. L. Massey, "Collision-resolution algorithms and random-access communications," 1981.
- [14] M. Ghanbarinejad and C. Schlegel, "Irregular repetition slotted ALOHA with multiuser detection," in *2013 10th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, 2013, pp. 201–205.
- [15] A. Munari, "Modern random access: An age of information perspective on irregular repetition slotted ALOHA," *IEEE Trans. Commun.*, vol. 69, no. 6, pp. 3572–3585, 2021.
- [16] Z. Sun, Y. Xie, J. Yuan, and T. Yang, "Coded slotted ALOHA for erasure channels: Design and throughput analysis," *IEEE Trans. Commun.*, vol. 65, no. 11, pp. 4817–4830, 2017.
- [17] Z. Chen, Y. Feng, C. Feng, L. Liang, Y. Jia, and T. Q. S. Quek, "Optimal distribution design for irregular repetition slotted ALOHA with multi-packet reception," *arXiv e-prints*, p. arXiv:2110.08166, Oct. 2021.

# Two-Policy Cooperative Transfer for Alleviation of Sim-to-Real Gap

Liangdong Wu  
School of Artificial Intelligence  
University of Chinese Academy of Sciences  
Beijing, China  
wuliangdong2018@ia.ac.cn

Fangzhou Xiong  
Meituan  
Beijing, China  
xiongfanzhou@meituan.com

Zhiyong Liu  
Institute of Automation  
Chinese Academy of Sciences  
Beijing, China  
zhiyong.liu@ia.ac.cn

**Abstract**—The main difficulty of sim-to-real is the reality gap between the source domain and the target domain. In order to solve it, various methods where domain randomization is the mainstream have been emerged, whose essence is to make the single policy more robust. In contrast, we propose a novel transfer method, namely two-policy cooperative transfer, whose core is that one policy (task policy) is used to complete the task and another policy (gap policy) aims to assist the former to cover the gap, hence we can focus on the training of task and the overcoming of gap respectively. Based on this method, the setting of the learning objective of gap policy depends on the transfer situation of deploying task policy into real system, besides how to conduct the cooperation of the both lies in the threshold reflecting gap and the coupling of output actions of two policies. For the typical contact-rich gap in the dynamics field, we design an adaptive object pushing experiment based on UR3 robot, and verify the effectiveness of the proposed method.

**Keywords**—Sim-to-real, Reinforcement learning, Robot control

## I. INTRODUCTION

In recent years, robot control based on reinforcement learning has gradually become an important direction in artificial intelligence field [1]. Its essence lies in training a policy achieving the maximum expected reward to complete the task through trial and error of large data samples [2]. Up to now, almost of all policy training is completed in virtual simulation, while direct training in the real robot system is faced with a series of problems of high cost, high risk and low efficiency [3]. Therefore, how to guarantee the effect of deploying the policy obtained via simulation training to the real system (sim-to-real transfer), specifically eliminating the impact of reality gap [4] between simulation and reality, has increasingly become a hot topic attracting wide attentions.

The difference between the simulated source domain and the real target domain leads to the gap [4, 5], which further render the application effect of the policy transferred into the real system is far less likely than that of the simulation. For this issue, some research advances have also emerged gradually,

This work is supported by Science and Technology Innovation 2030–“New Generation of Artificial Intelligence” Major Projects (2020AAA0108902), Strategic Priority Science and Technology Program of Chinese Academy of Sciences(Category B)(XDB32050100) and Dongguan City Core Technology Cutting-edge Project (2019622101001)

among which the more representative one is domain randomization (DR) [5], which was initially applied to cover sim-to-real gap in visual images, and later related studies employed this idea to narrow the gap in the field of dynamics [3, 6]. In this paper, our research focuses on the gap in dynamics, and a typical application scenario is the contact-rich [7] transfer experiment, in which the properties of the experimental object are difficult to accurately simulate.

In the initial phase of this research, we evaluated the performance of DR for gap bridging in dynamics, and found that it was effective but still seemed to fall short of the desired results. Additionally, some recent literature [8, 9] have raised doubts about DR. After more investigations, such as [3, 6, 10–12], we find that these methods including DR are basically single policy transfer, hence, the policy inevitably requires the two abilities at the same time, that is, completing the task and overcoming the gap. For now, there is no theoretical support for the coupling training of the two abilities to ensure that each ability can meet the established requirements. Meanwhile, the training complexity will also increase significantly [12]. Therefore, we tentatively propose a two-policy training mode, one policy is focused on the acquisition of task skills (task policy), another policy is focused on how to make up the gap (gap policy). The two policies will be deployed to the real system simultaneously, if there is no gap effect temporarily, the task policy will be executed; otherwise, the coupling action of the output actions of the two policies will be executed to eliminate the gap and continue the task at the same time. We define it as two-policy cooperative transfer.

For the proposed method, the setting of learning objectives of the gap policy depends on the transfer situation of deploying task policy into real system, that is, the pre-transfer of task policy. Although this method needs to train two policies, it can make more timely and reasonable adjustments to the impact of gap, so as to distinctly improve the effectiveness and success rate of sim-to-real transfer.

The main contributions in this paper are as follows: (1) We propose two-policy cooperative transfer to make up for the impact of gap. (2) Based on our method, we can focus on the training of task and the overcoming of gap respectively. (3) Experimental results show that the proposed method has a satisfactory covering effect for the contact-rich dynamic

gap, and is significantly better than the domain randomization method.

## II. RELATED WORK

Sim2real transfer has gradually become a topic of widespread concern, and the main difficulty is how to bridge the gap. For this issue, there have been a lot of research progress, which can be roughly divided into three categories [13]: system identification, domain adaptation, and domain randomization (DR).

The purpose of system identification is to accurately obtain the corresponding simulation model through the identification of real system and objects, so as to minimize the differences between simulation and reality. Hwangbo et al [14] trained the deep network model to conduct the mapping from motor instructions to torque based on the data of the actual system, and the transfer of walking and running on the quadruped robot is realized.

Domain adaptation is originated from computer vision, which aims to study how the visual-based model trained by the source domain adapts to the target domain that has never been encountered. The idea is also used to solve reality gap. Christiano et al [10] trained the inverse dynamics model through the data of the actual system to make the simulation model closer to the actual system, thus improving the transfer effect of the policy.

DR aims to experience various simulation environments as much as possible in the simulation training process (such as all kinds of simulation images, simulation objects with various dynamic characteristics in a certain range, etc), and try to include the approximate actual situation in them, so as to achieve a strong robust policy. Sadeghi et al [11] trained the visual-based quadrotor control policy by using random composite rendering scenes. In the paper [5], the authors used various images to train the policy for realizing the grasping task of the robot.

In order to solve the gap of dynamics, Andrychowicz et al [12] trained the policy of controlling object rotation by randomly setting physical parameters such as friction and delay, and transfer it to the Baxter robot without additional fine-tuning. In the paper [6], the authors explored how to reduce the parameter space of DR to improve training efficiency. In view of how to solve the dynamics gap effectively, Valassakis et al [9] comprehensively evaluated several main methods at the present stage, expressing that the result of DR is not ideal enough. The paper [8] also raised doubts about domain randomization.

## III. METHOD

For now, the policy transfer from simulation to reality is basically the transfer of a single policy. In view of the gap reflected in the transfer procedure, most of scientists are devoted to the research on how to make the single policy more robust and more generalized, so as to cover the gap. In contrary to this idea, we propose a concept which is the two-policy cooperative transfer, specifically meaning the cooperation of

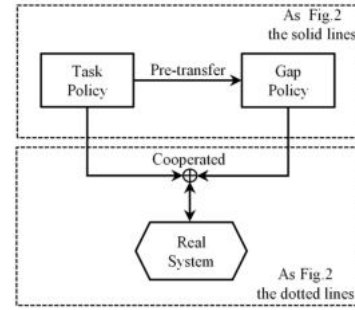


Fig. 1. The main logical framework of the two-policy cooperative transfer

task policy and gap policy, with the former focusing on task skills and the latter focusing on overcoming gap. Further, we clarify the method about how to train the two policies and transfer them to the real system cooperatively. Figure 1 shows the main logical framework of our proposed method, and more details are illustrated in Figure 2.

### A. Task Policy

In order to make the agent possess a certain skill or be able to complete a certain task, the relevant algorithm of reinforcement learning is used to train a policy that is inputted environmental states and output the actions, so as to realize the maximum reward return  $J(\theta) = E_{\theta}[\sum_{t=1}^{\infty} \gamma^t r_t]$ . We define this kind of policy as task policy  $\pi_{T\theta}(a_{Tt} | s_t)$ , and its action is further defined as  $a_{Tt}$ .  $T$  is the symbol representing task. The training of task policy is the same as the policy training of reinforcement learning in general sense. It is not be expanded here. For more details, refer to the relevant literature [3, 5].

For the training of task policy, we do not take consideration into narrowing the gap. Therefore, in the training process, we only focus on whether the policy can acquire the required task capability.

### B. Gap Policy

In this section, we illustrate how to train a specific policy for the gap, namely gap policy  $\pi_{G\theta}(a_{Gt} | s_t)$ . First, we hold the opinion that it is sufficient for task policy if its test results perform well in simulation. Further, we regard the state of task policy at each step in the simulation test as the reference state, and the deviation state obtained when the task policy is deployed into the actual system as the gap state. Based on these two different classes of states and possible modification actions, new learning objective are designed to train gap policy.  $G$  is the symbol representing gap.

After the task policy  $\pi_{T\theta}(a_{Tt} | s_t)$  is obtained through training under simulation, it is first tested in simulation, and the state  $s_t$  of the experimental object in each step is recorded and collected as a state reference set  $s_{tra_j}$ :

$$s_{tra_j} = (s_1, s_2, s_3, \dots, s_k, \dots, s_n) \quad (1)$$

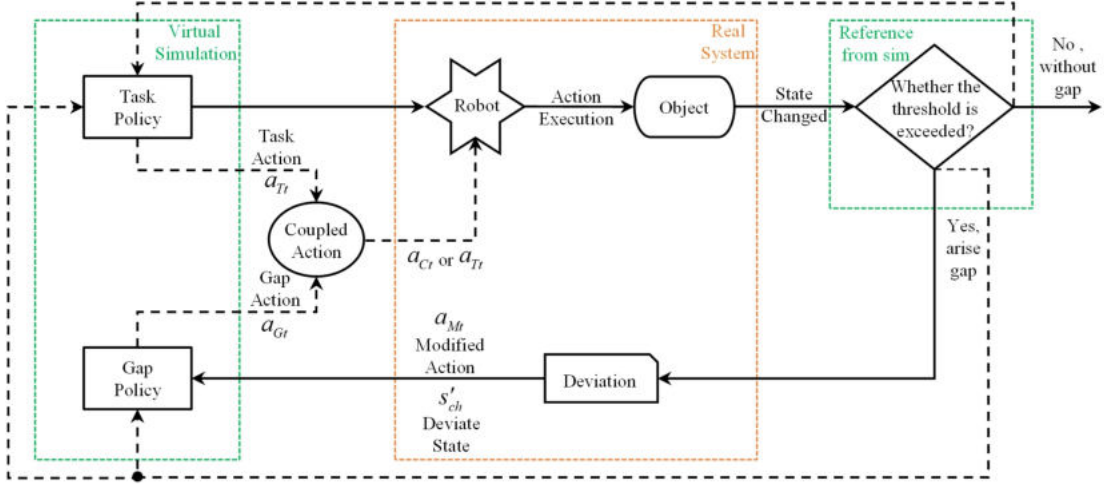


Fig. 2. The solid lines represent how to acquire the prior knowledge about designing the learning objective of gap policy through the task policy pre-transfer, and then train it in simulation. The dotted lines represent the operation logic and action selection of the two-policy cooperative transfer. The solid lines of robot and object are collinear with the dotted lines

Then the task policy is transferred to the actual system, and the state  $s'_t$  of the actual object is recorded at each step to obtain the actual state set  $s'_{traj}$ :

$$s'_{traj} = (s'_1, s'_2, s'_3, \dots, s'_k, \dots, s'_n) \quad (2)$$

And we can get the set of deviations  $e_{traj}$  between  $s_{traj}$  and  $s'_{traj}$ , as:

$$e_{traj} = (e_1, e_2, e_3, \dots, e_k, \dots, e_n) \quad (3)$$

Each deviation  $e_t$  is compared with the threshold  $c$  in chronological order. Based on experience, we set the threshold  $c$  at  $\pm 5\%$  of the corresponding reference state. The first deviation exceeding the threshold is set as  $e_c$ . And then the simulation state and the actual state at the corresponding time is respectively set as  $s_c, s'_c$ . The reason for adopting the first deviation is that it is most timely to make adjustments to overcome the impact of gap at this moment. Otherwise, the accumulation of errors will become larger, and the difficulty of adjustment will increase significantly. Due to the influence of the gap, each actual trajectory is likely to be different, assuming that  $k$  practical experiments are carried out, the state-deviation trajectory  $\tau_{es}$  can be obtained, as:

$$\tau_{es} = (e_{c1}, s_{c1}, s'_{c1}, \dots, e_{ck}, s_{ck}, s'_{ck}) \quad (4)$$

The three elements with the same subscript in  $\tau_{es}$  are classified as one class, and the class  $h$  is arbitrarily taken out of it, with time  $t$ . Suppose there is an action  $a_{Mt}$ , render that:

$$\hat{s}'_{ch} \sim P_R(\hat{s}'_{ch} | s'_{ch}, a_{Mt}) \quad (5)$$

$$0 \leq \hat{s}'_{ch} - s_{ch} < e_{ch} \quad (6)$$

After the execution of the action  $a_{Mt}$ , the updated state  $\hat{s}'_{ch}$  is obtained from the actual system transition probability distribution  $P_R$ , which is closer to the simulation state  $s_{ch}$ , so as to realize the state modification of the object. Actions  $a_{Mt}$  can be designed with reference to specific system and

task, and symbol  $R$  and  $M$  represents reality and modification respectively.

Thus, we get two important prior knowledge of the training setting of gap policy  $\pi_{G\theta}(a_{Gt} | s_t)$ : preliminarily reflecting various states  $s'_{ch}$  of the gap, and each action  $a_{Mt}$  to modify these states. Further exclude possibly similar states and actions, retain representative ones ( $s'_{ch}, a_{Mt}$ ), and construct corresponding learning objective based on them. After training, the gap action  $a_{Gt}$  should be equivalent to the modified action  $a_{Mt}$ . The corresponding logic is shown as the solid lines in Figure 2, and the setting of simulation training environment refers to the training situation of task policy.

### C. Two-Policy Cooperative Transfer

Task policy  $\pi_{T\theta}(a_{Tt} | s_t)$  and gap policy  $\pi_{G\theta}(a_{Gt} | s_t)$  are transferred into the real system to overcome the gap and complete the task. In this process, when the state of the experimental object does not reach the threshold  $c$ , we deem that the gap does not appear at the moment, and the task action  $a_{Tt}$  given by the task policy can be performed continually. When the threshold  $c$  is reached, the gap appears, then the gap action  $a_{Gt}$  given by the gap policy is coupled with the task action  $a_{Tt}$  to obtain the coupled action  $a_{Ct}$ , which is executed to modify the object state and make it drop below the threshold  $c$ .

The coupling of actions can be simply expressed as:

$$a_{Ct} = a_{Tt} \oplus a_{Gt} \quad (7)$$

In certain situation it might be adding vectors. The logic of coupled actions execution is shown in Figure 2 with the dotted lines.

## IV. EXPERIMENTS

We set up a simulated and realistic UR3 robot object pushing experimental platform respectively, as figure 3 (a)(b), then conduct policy training through the former and policy transfer

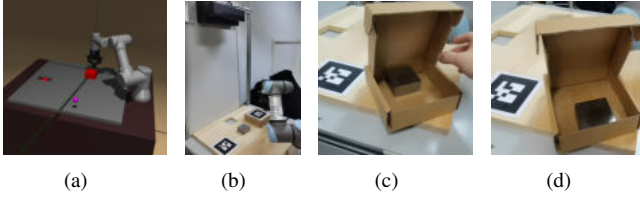


Fig. 3. The simulation (a) and real experiment platform (b) for UR3 robot object pushing, besides the box loaded 1000g iron block into the upper part of the center (c) and the lower part of the center (d) respectively

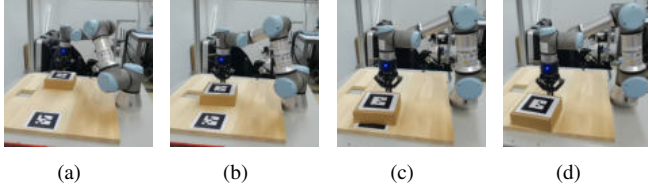


Fig. 4. The experiment process of object pushing by UR3

through the latter. Furthermore, we illustrate the experimental process based on the proposed method, also compare and evaluate it with domain randomization.

#### A. Experimental Setup

**Virtual simulation:** Our simulation environment is composed of Mujoco physics engine [15] and OpenAI Gym library [16]. We use a UR3 robot equipped with two-finger gripper at the end, which will be closed during the experiment to push the experimental object. The simulation step size is 0.002s, and each episode contains 50 steps. The output of the policy is the Cartesian coordinate at which the end of the robot gripper should reach. The input of the policy includes the Cartesian position coordinates of the end of the robot gripper, target position and the position attitude of the object.

**Real system:** The system communication is set up through the robot operating system (ROS). The Kinect2 camera is to get the target pose and position attitude of the experimental object, and UR3 robot is equipped Robotiq two-finger gripper with closed state. The end of the robot gripper is set perpendicular to the motion plane and the height is fixed to ensure that the end can avoid bumping.

**The experimental object and process in reality:** The object used in our experiment is a box with QR code, being  $(0.15m * 0.15m * 0.15m)$  and  $60g$ . For comparison, we further load 1000g iron block into the non-geometric center position of the box, such as the upper part of the center and the lower part of the center, shown as Figure 3 (c)(d), to significantly increase and change the whole box mass, friction force and center of gravity position. We set the mark of the success of the experiment as that the distance between the box and the target pose is less than  $0.02m$ , meanwhile the deflection angle is not more than  $5^\circ$ . The initial position of object is  $(x, y) = (0.36m, 0.15m)$ , and the target point is  $(x, y) = (0.11m, 0.40m)$ . The object pushing experiment process is shown as Figure 4.

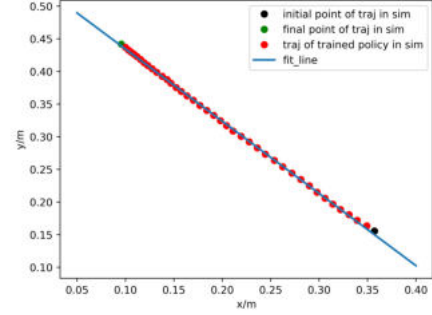


Fig. 5. The reference trajectory and fitting line of the pushed object in the simulation test



Fig. 6. (a) and (b) illustrate the two types of deflection

#### B. The Application of Proposed Method

**Task policy:** For the task policy training of UR3 robot object pushing experiment, the corresponding simulation environment is designed as described in 4.1 section. The reinforcement learning algorithm used in training is SAC [17], combined with the use of HER [18]. The neural network Settings and hyperparameters of the algorithm program are set using the corresponding default Settings in the Stable-baselines library [19]. For the design of the reward function, we take the distance function between the object and the target position, as:

$$r(s_t, a_t) = -d_t/d_0 \quad (8)$$

Where,  $d_t$  and  $d_0$  represents the distance of the current moment and the initial moment respectively. Additionally, we fixed the initial position and target position of the experiment to reduce the training difficulty and generalization ability of the task policy, so as to highlight the impact of gap and the effectiveness of the proposed method.

After training, we test the task policy in simulation, and record the states of each step of the object moving on the flat surface location, then gather into a reference trajectory and infer the fitting line, as shown in figure 5.

**Gap policy:** The task policy is deployed into the UR3 robot system, due to the uncertainty about the physical properties of the actual object, reality gap appears. Specifically, the position and deflection of the actual object at each moment will gradually deviate from the corresponding state of the reference trajectory, namely the deviation  $e_t$ . From analysis, there are actually no more than two types of deflection at the initial phase of arising deviation, that is, deflection along clockwise



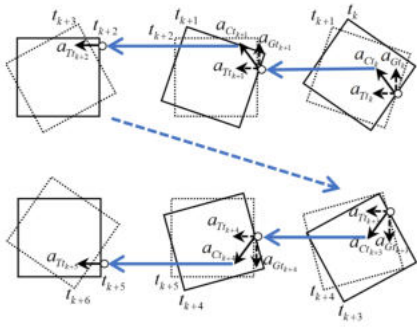


Fig. 7. the actions of the two policies are coupled and executed

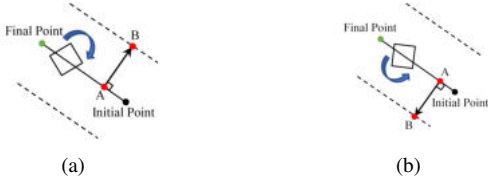


Fig. 8. Schematic diagram of the learning objective of gap policy

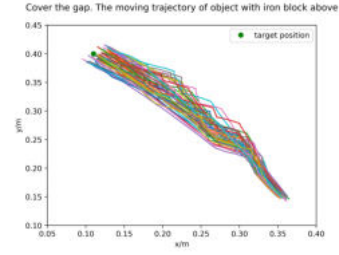
or counterclockwise direction of original object plane, with different specific degrees, as shown in Figure 6 (a)(b).

For the two direction deflections, obviously, the corresponding modified adjustment is the action from the oblique direction, defining it as coupled action  $a_{Ct}$ . While the action  $a_{Tt}$  given by task policy is along the direction of motion and is continuous, we can reverse infer that the action  $a_{Gt}$  given by gap policy should be perpendicular to the direction of motion up or down, as figure 7. Our output action is the coordinate of the gripper end, hence the coupling of actions in this experiment refers to the addition of coordinate vectors.

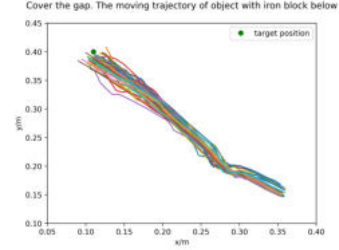
Furthermore, based on the deviation states and the modified actions inferred from the analysis, we can design new learning objective and obtain the gap policy through simulation training. The vertical view of learning objective is shown in Figure 8. When clockwise deflection or counterclockwise deflection occurs as shown in Figure 8 (a) or Figure 8 (b), the learning objective is how to move the end from the starting point A to the target point B. The AB distance in this experiment is set as 0.10m.

For simulation training of gap policy, its relevant Settings, such as reinforcement learning algorithm, hyperparameters, network setting and reward function, are the same as those of task policy, only the learning objective is different.

**Two-policy cooperative transfer:** The task policy and the gap policy are transferred to the actual system together. After each execution step, the state is compared with the threshold. If the threshold is not exceeded, only execute the task action at the next step; otherwise, execute the coupled action. The entire process can also be referenced in Figure 7 and the dotted lines of Figure 2.



(a)



(b)

Fig. 9. The motion trajectories of the box loaded iron block through our method

TABLE I  
DYNAMIC PARAMETERS AND THEIR RANGES IN SIMULATION

Parameters	Range
Mass	[0.05, 1.10] kg
Sliding Friction Coefficient	[0.2, 2.0]
Torsion Friction Coefficient	[0.01, 0.1]
Rolling Friction Coefficient	[0.005, 0.05]

### C. Results of Comparisons

We use the single transfer of task policy as the baseline and domain randomization [5, 9] as the comparison method to conduct a comprehensive evaluation of the object pushing experiment with our proposed method.

We employ domain randomization for object's parameters to train a policy with some generalization. Details are shown as Table I.

Each group of experiments is conducted for 50 times, and the corresponding success rate is recorded. The specific results are shown in Table II.

The results show that the baseline method has certain task completion ability only for the empty box, but it can't complete the experiment after loading iron block. For domain randomization, although it can enhance the generalization and improve the success rate partly, the overall effect is not ideal.

TABLE II  
THE SUCCESS RATE OF THREE TRANSFER METHODS FOR THREE TYPES OF BOXES

Three Types of Boxes	The Baseline	Domain Randomization	Ours
Empty Box	56%	68%	<b>92%</b>
Iron Block into The Upper	0%	10%	<b>84%</b>
Iron Block into The Lower	0%	8%	<b>82%</b>

In contrast, our proposed method is robust to the changes of physical properties of box and achieves relatively high success rate. As Figure 9 (a) and (b), the motion trajectories of the box loaded iron block reflect that based on our method, there are adaptive real-time adjustments in the pushing process to eliminate the influence caused by gap.

It should be pointed out that for the task policy of single transfer and domain randomization, we did not retrain targeted, but carried out experiments on three types of boxes with the same policy and set of procedures. In the experiment based on our method, we did not make any secondary adjustment to the task policy and the gap policy, and just ran the same set of programs based on the same policy, which also reflected the strong robustness of the proposed method, and there was no need to retrain the policy for the change of the physical properties of the experimental object.

## V. CONCLUSION

In order to solve the gap between simulation and reality, specifically dynamics field, we propose two-policy cooperative transfer, which is different from the previous other methods around single policy transfer. The basic intention of the proposed method is that the task policy is responsible for the acquisition of task skills and the gap policy is used to assist the former to cover reality gap, with cooperation of the two policies to achieve strong robust transfer. Consider that the setting of the learning objective of gap policy depends on the transfer situation of task policy, it embodies the characteristics of single policy pre-transfer. This method provides a new way of thinking for alleviating sim-to-real gap. The adaptive object pushing experiment based on UR3 robot verifies the effectiveness of the proposed method. In the future, we will research how to apply this method to more complex tasks and higher dimensional robots.

## REFERENCES

- [1] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. A brief survey of deep reinforcement learning. *arXiv preprint arXiv:1708.05866*, 2017.
- [2] Thanh Thi Nguyen, Ngoc Duy Nguyen, and Saeid Nahavandi. Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications. *IEEE transactions on cybernetics*, 50(9):3826–3839, 2020.
- [3] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 3803–3810. IEEE, 2018.
- [4] Nick Jakobi, Phil Husbands, and Inman Harvey. Noise and the reality gap: The use of simulation in evolutionary robotics. In *European Conference on Artificial Life*, pages 704–720. Springer, 1995.
- [5] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017.
- [6] Yevgen Chebotar, Ankur Handa, Viktor Makovychuk, Miles Macklin, Jan Issac, Nathan Ratliff, and Dieter Fox. Closing the sim-to-real loop: Adapting simulation randomization with real

- world experience. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8973–8979. IEEE, 2019.
- [7] Sebastian Höfer, Kostas Bekris, Ankur Handa, Juan Camilo Gamboa, Florian Golemo, Melissa Mozifian, Chris Atkeson, Dieter Fox, Ken Goldberg, John Leonard, et al. Perspectives on sim2real transfer for robotics: A summary of the r: Ss 2020 workshop. *arXiv preprint arXiv:2012.03806*, 2020.
- [8] Zhaoming Xie, Xingye Da, Michiel van de Panne, Buck Babich, and Animesh Garg. Dynamics randomization revisited: A case study for quadrupedal locomotion. *arXiv preprint arXiv:2011.02404*, 2020.
- [9] Eugene Valassakis, Zihan Ding, and Edward Johns. Crossing the gap: A deep dive into zero-shot sim-to-real transfer for dynamics. *arXiv preprint arXiv:2008.06686*, 2020.
- [10] Paul Christiano, Zain Shah, Igor Mordatch, Jonas Schneider, Trevor Blackwell, Joshua Tobin, Pieter Abbeel, and Wojciech Zaremba. Transfer from simulation to real world through learning deep inverse dynamics model. *arXiv preprint arXiv:1610.03518*, 2016.
- [11] Fereshteh Sadeghi and Sergey Levine. Cad2rl: Real single-image flight without a single real image. *arXiv preprint arXiv:1611.04201*, 2016.
- [12] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.
- [13] Wenshuai Zhao, Jorge Peña Queraltá, and Tomi Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 737–744. IEEE, 2020.
- [14] Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26), 2019.
- [15] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.
- [16] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.
- [17] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1861–1870. PMLR, 2018.
- [18] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *arXiv preprint arXiv:1707.01495*, 2017.
- [19] Ashley Hill, Antonin Raffin, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, Rene Traore, Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, and Yuhuai Wu. Stable baselines. <https://github.com/hill-a/stable-baselines>, 2018.

# Graph Neural Network-based Clustering Enhancement in VANET for Cooperative Driving

Hang Hu, Myung J. Lee  
Department of Electrical Engineering  
City College, City University of New York  
New York, NY, USA, 10031

Emails: hhu002@citymail.cuny.edu, mlee@ccny.cuny.edu

**Abstract**—The significantly increasing number of vehicles brings convenience to daily life while also introducing significant challenges to the transportation network and air pollution. It has been proved that platooning/clustering-based driving can significantly reduce road congestion and exhaust emissions and improve road capacity and energy efficiency. This paper aims to improve the stability of vehicle clustering to enhance the lifetime of cooperative driving. Specifically, we use a Graph Neural Network (GNN) model to learn effective node representations, which can help aggregate vehicles with similar patterns into stable clusters. To the best of our knowledge, this is the first generalized learnable GNN-based model for vehicular ad hoc network clustering. In addition, our centralized approach makes full use of the ubiquitous presence of the base stations and edge clouds. It is noted that a base station has a vantage view of the vehicle distribution within the coverage area as compared to distributed clustering approaches. Specifically, eNodeB-assisted clustering can greatly reduce the control message overhead during the cluster formation and offload to eNodeB the complex computations required for machine learning algorithms. We evaluated the performance of the proposed clustering algorithms on the open-source highD dataset. The experiment results demonstrate that the average cluster lifetime and cluster efficiency of our GNN-based clustering algorithm outperforms state-of-the-art baselines.

**Index Terms**—vehicular ad hoc network (VANET), graph neural networks (GNNs), clustering algorithm, stability, cooperative driving

## I. INTRODUCTION

With the rapid development of the Automobile Industry and Urbanization, there are more and more vehicles on the roads. It is well-established that more than one billion vehicles have been registered globally, expected to grow in the following decades. Consequently, the problems associated with the increased number of vehicles have become more severe, including traffic congestion, traffic accidents, energy waste, and air pollution. In the United States, traffic congestion costs drivers more than \$100 billion annually due to wasted fuel and lost time [1]. In addition, exhaust emissions caused by traffic congestion are considered a key contributor to air pollution and a major haze component in many cities. For instance, the most significant source of greenhouse gases in the USA comes from the transportation sector, which accounts for 29% of total greenhouse gas emissions [2].

While the construction of roads can increase traffic capacity and reduce traffic congestion to some extent, it is unsustainable due to the enormous construction costs and limited land availability, especially in urban areas. An effective way to solve these problems is to change the driving pattern from individual driving to platoon driving [3] [4]. In general, a platoon-based driving pattern is a cooperative driving pattern of a group of vehicles with common interests, where one vehicle follows another and keeps a small and almost constant distance from the preceding vehicle to form a platoon.

The cooperative platoon-based driving pattern can significantly enhance road capacity, safety, energy efficiency, and collaborative environment. However, establishing and maintaining stable clusters or platoons in Connected Vehicles Networks are challenging because of the heterogeneous and drastically changing traffic scenarios. Moreover, in order to maintain multicast group communication in cooperative platoon-based driving among cluster members (CMs), the stable clustering algorithm is crucial. As all future vehicles can access base stations (BS or eNodeB), more efficient clustering is possible but yet to be prevalent with the help of cellular infrastructure, i.e., BS.

Most vehicular clustering algorithms have taken a distributed approach based on Dedicated Short-Range Communications (DSRC) without infrastructure support. Thus, they rely mainly on periodic HELLO messages exchanged among vehicles. Moreover, the most vital part of the clustering algorithm is Cluster Head (CH) selection [5] [6] [7], which is essential for the stability of the cluster lifetime and the control message overhead involved in forming and maintaining these clusters. Weight-based algorithms are widely used for CH selection [8] [9]. Each vehicle calculates a metric according to messages received from its neighbors. The metric represents the fitness to serve as a CH and broadcast to each vehicle's neighbors. The vehicle with the highest metric weight will act as a CH among nearby vehicles. The metric can generally be related to network metrics, such as degree of connectivity, link stability, and density, and mobility metrics, such as position, velocity, acceleration, and destination.

This distributed clustering strategy tends to increase the control message overhead compared with centralized strategy. Additionally, the weight-based algorithm of CH selection needs to manually adjust the combination of hyper-parameters among multiple metrics. Last but not least, existing vehicular clustering algorithms are not intelligent and learnable to adapt to different traffic scenarios. As a result, they are not easy to satisfy the requirement of the evolving Intelligent Transportation System (ITS).

In this paper, we propose to use Graph Neural Network (GNN) [10] [11], which fits naturally to solve clustering type of graph problem. To the best of our knowledge, applying GNN to solve the clustering problem in Vehicular Ad hoc Network (VANET) is the very first attempt. GNN uses both feature and graph information and usually achieves better performance than methods leveraging single feature or graph structure such as  $k$ -means [17] or Spectral Clustering [12]. Our proposed algorithm is a centralized approach and offloads the computation of GNN to BS instead of executing it at individual vehicles, alleviating the computational burden from vehicular nodes. We note that a base station has a vantage view

of the vehicle distribution within the coverage area compared to distributed approaches. Specifically, eNodeB-assisted clustering can greatly reduce the control message overhead during cluster formation. In practice, a trained GNN model located at BS or edge cloud utilizes the collected vehicle information to perform clustering and informs CHs to formulate cooperative driving patterns to improve traffic efficiency. Fig. 1 illustrates the framework of vehicular network clustering.

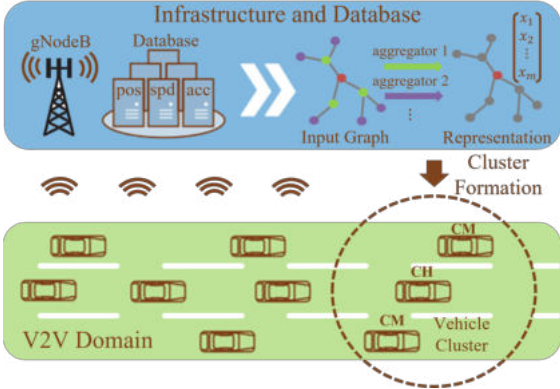


Fig. 1. Architecture of Vehicular Network Clustering.

## II. RELATED WORKS

Clustering is introduced to improve routing scalability and reliability and enhance the stability of the collaborative environment by exploiting the formation of hierarchical network structures. By grouping vehicles together in the consideration of correlated spatial distribution and relative velocity, these cluster groups can serve as the foundation for accident or congestion detection, information dissemination, and entertainment in the applications of ITS.

### A. Distributed Clustering Approaches

The earliest VANET clustering methods are derived from Mobile Ad hoc Network (MANET) clustering approaches to facilitate the distribution of network resources. A majority of them have taken a distributed approach based on DSRC without infrastructure support. Many existing VANET clustering algorithms focus on optimally selecting CHs because the stability of a cluster depends primarily on the selection of the CH. Weight-based metrics are the key to the most clustering strategies: position [13], speed [14], destination [15], and multiple metrics [16]. Most conventional distributed approaches still incur high communication overhead and prove inefficient in a highly dense and dynamic environment.

### B. Machine Learning based Clustering Approaches

Clustering algorithms began adopting machine learning to overcome the required complex computation in distributed clustering.  $K$ -means algorithm [17] is the most frequently used machine learning algorithm in VANET. Some  $k$ -means variant algorithms [18] are proposed to enhance initial centroid selection to boost performance. Often, the fuzzy logic inference is integrated with a machine learning algorithm to enhance the stability of a cluster [19] by predicting the future speed and the positions of CMs. Nevertheless, the design of fuzzy rules needs much domain knowledge, and fuzzy logic does not have the learning ability as is well-known. Some researchers recently proposed using Spectral Clustering, which is related to Eigenvalue Decomposition (EVD) on the normalized graph Laplacian matrix, to enhance clustering stability in VANET [20].

### C. GNN based Clustering Approaches

Recently, research on analyzing graphs with machine learning has been receiving more attention because of the great expressive power of graph data, which contains rich relation information among elements. Hence, GNNs have been proposed to solve the non-Euclidean domain problem. In GNNs, node clustering divides the nodes into several disjoint groups where similar nodes should be in the same group. [21] has applied graph autoencoder (GAE) to node clustering (citation network) by an unsupervised learning framework. Even though GNNs have many applications across different tasks and domains, applying GNN to solve the clustering problem in VANET is the very first attempt.

In this paper, our goal is to enhance the vehicle system's stability and optimize the average lifetime of all clusters. The problem of clustering is innovatively transformed into aggregating vehicles with similar node representations (embeddings) in the same cluster as learned by the GNN model. Specifically, a partitioning method optimally divides the vehicle nodes into groups with a minimum intra-cluster dissimilarity. Our proposed algorithm coincides with the goal of VANET clustering, which is to encourage vehicles with similar motion patterns, such as similar position, velocity, acceleration, etc., to form a cluster.

## III. GNN-BASED CLUSTERING IN VANET

In this section, we demonstrate how to develop a GNN-based clustering scheme that collects vehicle feature as its input and node representation as its output.

### A. Graph Construction

The interconnections among vehicles driving on the road can be formulated as an undirected homogeneous dynamic graph. To this end, we propose to use GNN, which fits naturally to solve clustering type of graph problem and uses both feature and graph information. We use the raw vehicle feature as the node feature of the graph. A vehicle feature of vehicle node  $v_i$  at time  $t$  is  $x_i(t) = \{s_i, p_i, a_i, l_i, w_i\}$ , where  $s_i$  is the speed,  $p_i$  is the position,  $a_i$  is the acceleration,  $l_i$  and  $w_i$  are the length and width of vehicle  $v_i$ . In the following, we use the subscript  $i$  to denote vehicle node  $v_i$ .

The vehicle interconnection metric is designed to weigh the similarity between the movement patterns of two vehicles. In our model, the vehicle interconnection metric is calculated by the improved force-directed algorithm designed based on virtual forces [22], which is inspired by Coulomb's Law to select CH and create stable clusters. The force-directed algorithm assigns the forces on the edges in the VANET graph. Note that the force also represents the weight between any two connecting vehicle nodes. The most straightforward way is to assign force as if the edges were springs and the nodes were electrically charged particles. The entire network is modeled as a physical system. The forces are applied to the vehicle nodes, pulling them closer together or pushing them further away. Every vehicle node exerts a force  $F$  on its neighbors according to their distance and relative velocities. Fig. 2 shows the neighbored vehicle nodes apply relative forces to the vehicle  $v_i$ .

A positive force between two nodes indicates that the pair of nodes are moving in the same direction. In contrast, a negative force between two nodes means that the vehicles are moving in opposite directions. In our model, the relative force is always positive since we only consider the vehicles are moving in the

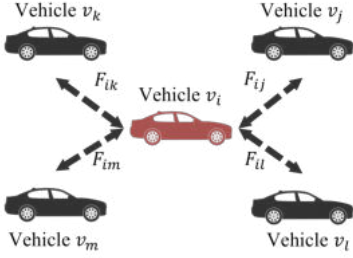


Fig. 2. Neighbored forces applied to vehicle  $v_i$ .

same direction, which facilitates the stability of VANET. The greater the positive forces among nodes are, the more similar the moving pattern is.

Here we explain how to calculate the pairwise relative force  $F_{ij}$  for every neighbor applied. Naturally, the relative force is decomposed along the  $x$ -axis and the  $y$ -axis. We can obtain relative force  $F_{ij}$  according to equations defined as:

$$F_{ijx} = k_{ijx} \frac{q_i q_j}{D_{ij}^2}; F_{ijy} = k_{ijy} \frac{q_i q_j}{D_{ij}^2} \quad (1)$$

$$D_{ijx}(t) = x_i - x_j; D_{ijx}(t + dt) = x_i + dx_i - x_j - dx_j \quad (2)$$

$$D_{ijy}(t) = y_i - y_j; D_{ijy}(t + dt) = y_i + dy_i - y_j - dy_j \quad (3)$$

$$k_{ijx} = \frac{1}{1 + |D_{ijx}(t + dt) - D_{ijx}(t)|dt} \quad (4)$$

$$k_{ijy} = \frac{1}{1 + |D_{ijy}(t + dt) - D_{ijy}(t)|dt} \quad (5)$$

$$q_i = q_j = \begin{cases} R - D_{ijx}(t), & \text{if } D_{ijx}(t) \leq D_{ijx}(t + dt) \\ R + D_{ijx}(t), & \text{if } D_{ijx}(t) > D_{ijx}(t + dt) \end{cases} \quad (6)$$

$$\|F_{ij}\|_2 = \sqrt{F_{ijx}^2 + F_{ijy}^2} \quad (7)$$

Where  $F_{ijx}$  and  $F_{ijy}$  are the relative forces along the  $x$ -axis and  $y$ -axis.  $k_{ijx}$  and  $k_{ijy}$  are the relative mobility parameters along the  $x$ -axis and  $y$ -axis, respectively.  $q_i$  and  $q_j$  represent relative maintenance parameters indicating that how far they are beyond communication distance.  $D_{ij}$  is the current distance among the nodes.  $D_{ijx}(t)$  and  $D_{ijy}(t)$  are the distance between two nodes along the  $x$ -axis and  $y$ -axis at time  $t$ . Similarly,  $D_{ijx}(t + dt)$  and  $D_{ijy}(t + dt)$  are the distance between two nodes along the  $x$ -axis and  $y$ -axis at time  $t + dt$ .  $x_i$  and  $y_i$  represent the  $x$ -axis and  $y$ -axis position of node  $v_i$ .  $dx_i$  and  $dy_i$  are the position increment in time  $dt$  on the  $x$ -axis and  $y$ -axis of node  $v_i$ .  $R$  is the transmission range. This force  $F_{ij}$  will be used as weight to propagate information in Eq. (8).

### B. Design of GNN Clustering Algorithm

Having obtained a customized graph dataset in the last section, we present our GNN-based clustering algorithm here. In general, our GNN model comprises four layers, including an input layer, two SAGE (SAmple and aggreGatE) Convolutional layers (SAGEConv), and an output layer. The dimension of the input layer is the vehicle feature, and the output dimension is predefined (e.g., 4 in our experiment). The core layer of our GNN is inductive SAGE Convolutional layers.

SAGEConv layer is derived from graphSAGE [23], a general inductive framework that leverages node feature information to generate node embeddings for each node efficiently. This inductive capability can generalize to operate on evolving graphs and unseen nodes. Specifically, the SAGEConv layer

generates node embeddings by aggregating information from their local neighbors. The detailed visual illustration of the SAGEConv layer is shown in Fig. 3. The aggregation of SAGEConv is formulated as:

$$h_i^k = \sigma \left( W^k \cdot \frac{1}{|N_i|} \sum_{j \in N_i} (h_j^{k-1} \cdot F_{ij}) \right) \quad (8)$$

Where  $h_i^k$  is the embedding of node  $v_i$  in the  $k$ th layer.  $N_i$  is the neighborhood set connected to node  $v_i$ .  $W^k$  is the learnable weight parameters of fully connected layer  $k$ .  $F_{ij}$  is the weight to aggregate message.  $\sigma(\cdot)$  is the activation function ReLU. The number of GNN layers, namely search depth, is also the number of neighbors of hops aggregated information by the target node. In a  $k$ -layer GNN, nodes are able to collect information from the neighbors of the  $k$ -hops.

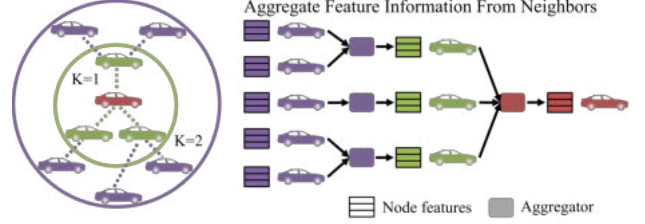


Fig. 3. Visual illustration of the SAGEConv layer.

On the other hand, early node embedding approaches are inherently transductive and directly optimize the embedding for each node using matrix-factorization-based objectives [25]. Consequently, they do not naturally generalize to unseen data since they predict nodes in a single, fixed graph. Combined with our application, the characteristic of the SAGEConv satisfies the dynamic traffic scenario. Our proposed GNN-based clustering algorithm is shown in Algorithm 1, where  $J_G(z_i)$  is the objective function discussed in section III part C. Due to the deeper layers (2nd loop), this process is iterative, and the nodes gradually acquire more and more information from further away from the graph. We use 2 SAGEConv layers (i.e., search depth  $K = 2$ ). There are several choices of aggregator architectures, such as Mean aggregator, Long Short-Term Memory (LSTM) aggregator, and Pooling aggregator. Here we choose Mean aggregator, which is a simple but with significant gain in performance to compute the embedding. The output is a low-dimensional vector embedding of the nodes. It already proves to be extremely useful for feature input for various downstream tasks such as classification, prediction, and clustering.

### C. Model Training

To learn effective and useful representations in a completely unsupervised learning fashion, a graph-based loss function  $J_G(z_i)$  is applied to the output representations  $z_i$ , and the weight matrices  $W^k$  is tuned via backward propagation. This loss function is defined as:

$$J_G(z_i) = - \sum_{i,j \in V} (y_{ij} \log(\hat{y}_{ij}) + (1 - y_{ij}) \log(1 - \hat{y}_{ij})) \quad (9)$$

$$\hat{y}_{ij} = \sigma(z_j^T z_i), \quad (10)$$

where  $j$  is a node  $v_j$  which is within the transmission range to node  $v_i$ .  $\hat{y}_{ij}$  denotes the probability of an edge with logits between the node  $i$  and  $j$ .

Specifically, we sample the edges in a graph as positive examples and non-existent edges (i.e., node pairs with no

edges between them) as negative examples. Positive and negative examples have the same number. Then, the positive and negative examples form positive and negative graphs, respectively. Along with the forward propagation, we can calculate node representations via the GNN model and apply them to the positive and the negative graphs for computing pairwise probability among nodes. We can calculate the loss and update model parameters via stochastic gradient descent along with the backward propagation.

---

**Algorithm 1:** GNN-based Clustering Algorithm

---

**Input :** Graph  $G(V, E)$ , edge weight  $F_{ij}$   
 Vehicle feature  $x_i, \forall i \in V$   
 Search depth  $k, \forall k \in \{1, \dots, K\}$   
 Weight matrices  $W^k$   
 Neighborhood set  $N_i$   
 Number of iterations  $T$   
 Graph-based loss function  $J_G$

**Output:** Clustering assignments

```

1  $h_i^0 \leftarrow x_i;$ 
2 for  $t = 1$  to  $T$  do
3   for  $k = 1$  to  $K$  do
4     for  $i \in V$  do
5        $h_i^k \leftarrow \sigma\left(W^k \cdot \frac{1}{|N_i|} \sum_{j \in N_i} (h_j^{k-1} \cdot F_{ij})\right);$ 
6        $z_i^k \leftarrow h_i^k / \|h_i^k\|_2;$ 
7     end
8   end
9   Calculate  $J_G(z_i)$  and update via stochastic gradient descent
10 end
11 Run k-means on output embeddings  $z_i$  to obtain final clustering results

```

---

#### IV. PERFORMANCE EVALUATION

In this experimental section, we first introduce the benchmark datasets and experimental parameter settings used in the experiments. After that, we evaluate the training of the proposed GNN model to validate that our model can learn useful and effective node representations. In addition, we show the baseline algorithms used in the results. Finally, we evaluate the metric performance used for VANET clustering between our algorithm and the baseline algorithms.

We implement our simulation platform in Python framework with PyTorch [26] and Deep Graph Library (DGL), which is a Python package built for easy implementation of graph neural network model family [24]. In addition, we conduct the following experiments on a computer with a 2.21GHz Intel Core i7-8750H CPU, 16GB Memory. Our proposed scheme is used for highway scenarios. Furthermore, the traffic model and the evaluations are based on real traffic data.

##### A. Datasets and Parameter Settings

1) *Datasets:* We construct a customized graph dataset for model training on the open-source highD dataset [27]. The highD dataset is new naturalistic vehicle trajectory recordings on German highways. A camera-equipped drone recorded the traffic with a 25fps frame rate at six different locations, and it covered a road segment of about 420 m length. The locations vary by the number of lanes, speed limits, and traffic density. Using state-of-the-art computer vision algorithms for semantic segmentation, the authors have estimated every pixel of each

frame, whether it belongs to a vehicle or the background. The positioning error is typically less than ten centimeters. It is convenient to obtain traffic information, including vehicle trajectory, vehicle type, size, and maneuvers. We extract vehicle feature  $x_i$  including speed  $s_i$ , position  $p_i$ , acceleration  $a_i$ , length  $l_i$  and width  $w_i$  of vehicles and standardize features by removing the mean and scaling to unit variance.

We choose sequence 13 recording to build the graph dataset, which involves the largest number of vehicles, since machine learning algorithms generally have a distinct advantage when dealing with a large amount of data. With the graph construction algorithm in section III part A, we generated 1000 training graphs and 210 testing graphs where we only consider cars instead of trucks since cars and trucks might have different driving patterns. For simplicity, we consider the same type of cars. The graph construction of data frame 32 is shown in Fig. 4. The green triangle stands for the vehicles. The brown dashed lines represent the edges among the vehicles. The blue dashed lines indicate lanes.

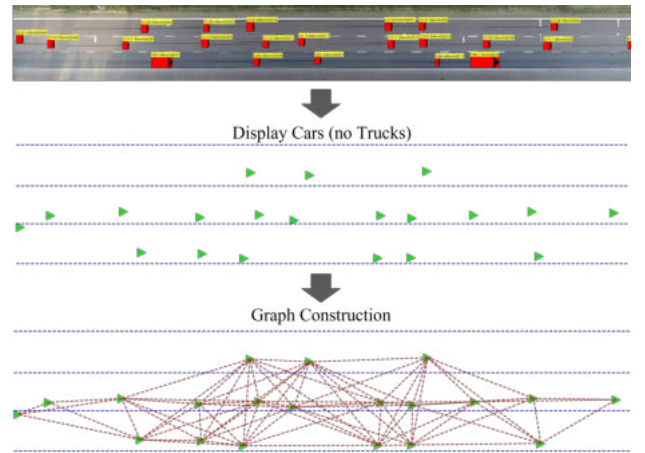


Fig. 4. Visual illustration of Graph Construction.

2) *Parameter Settings:* The dimension of the input layer is 8 (i.e., vehicle feature dimension). We set the dimension of the hidden layer and output layer to low-dimensional as 4. The maximum epoch for training is 400. To avoid overfitting, we apply early stopping. If the number of times that the validation loss is greater than the minimum loss exceeds a threshold (e.g., 150 in our experiment), the training will stop. We randomly sample the edges on each graph to form the training and validation sets. The edge ratio in the training set to the validation set is 9 to 1. We select ADAM with a learning rate of 0.003 as the optimization strategy. In addition, for reproducibility, we set a random seed (42069).

##### B. Model Training and Clustering Results

In order to evaluate the performance of node representations of our GNN model, we employ three metrics: Binary Cross Entropy Loss, Accuracy, and Area Under Curve (AUC). Binary Cross Entropy Loss is the objective function (i.e., Eq. (9)). AUC stands for the area under Receiver Operating Characteristic (ROC) curve, and the higher value indicates better performance. Accuracy is the ratio of the correctly classified elements to the total number of elements.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \quad (11)$$

where  $TP$ ,  $TN$ ,  $FP$  and  $FN$  are true positive, true negative, false positive and false negative, respectively.

After 1.5 hours of computation, the training of our GNN model is completed. The results show that the training and validation losses are 0.041 and 0.084, respectively. The training and validation accuracy are 0.986 and 0.969, respectively. In addition, the loss and accuracy on the testing graphs are 0.063 and 0.978, respectively. Here validation set is used to check the model convergence during training and avoid overfitting. Moreover, the testing set is applied to evaluate model generalization capability, and the AUC on testing graphs gets a good score of 0.998. Thus, we conclude that our GNN model can learn useful and predictive node representations. The training loss and accuracy are shown in Fig. 5. We have obtained a trained GNN model that can be used to learn useful node representations. Furthermore, we apply the trained GNN model on a graph and then obtain the clustering results by using  $k$ -means on node representations of the graph.

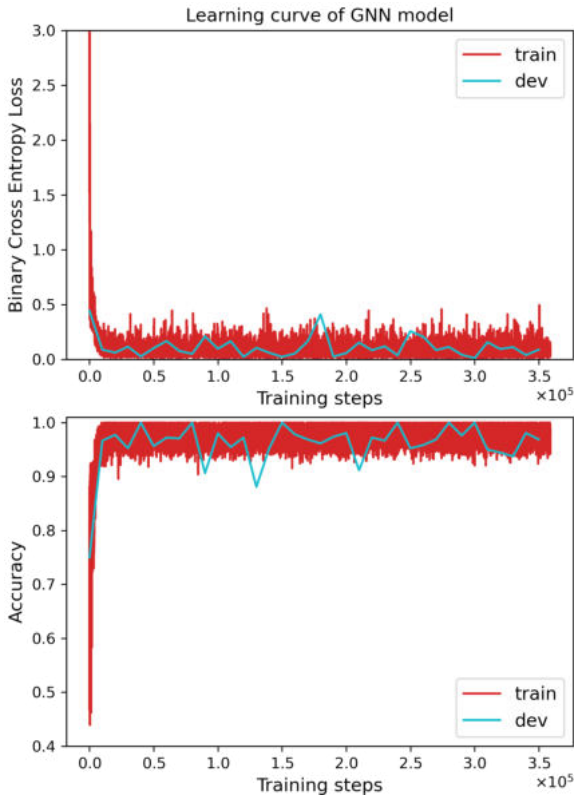


Fig. 5. Loss and accuracy curve during training.

In order to see the clustering results, we refer to [18] to select the number of clusters:

$$n = \left\lceil \frac{L}{2R} \right\rceil, \quad (12)$$

where  $L$  denotes the length of the road.  $R$  denotes the transmission radius of the vehicle. The length of the road segment  $L$  is about 420 m, and the transmission range  $R$  is defined as 100 m. Therefore, the number of clusters is  $n = 3$ . The clustering result on testing data frame 66 is shown in Fig. 6. The colored triangles represent the 3 clusters formed by our proposed clustering algorithm. The red dashed circles mark the CHs.

### C. Baseline Algorithms

We used the following clustering methods as the baseline algorithms in our comparisons.

1) *Method using features only*:  $k$ -means is traditional clustering algorithms [17]. Here we run  $k$ -means on our original vehicle feature as a benchmark.

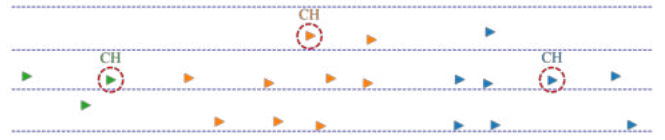


Fig. 6. GNN-based clustering results.

2) *Method using graph structure only*: Spectral Clustering [20] uses the adjacency matrix as the input similarity matrix to perform dimensionality reduction before clustering and is widely used in graph clustering.

3) *Method using both features and graph*: We also compare our proposed algorithm with the Graph Autoencoder (GAE) based clustering algorithm [21], which is an unsupervised learning network embeddings by encoding nodes/graphs into a latent vector space and reconstructing graph data from the encoded information. Since GAE is a matrix-factorization-based method, it can only be used for fixed graphs. Thus, we trained every graph before evaluating them.

### D. VANET Performance Evaluation and Results

We evaluate the clustering performance in VANET based on the highD dataset by our GNN-based algorithm and baseline algorithms. Since the highD dataset covers a road segment, we can only obtain a limited period of tracking time and distance. The coverage of the road segment is about 420 m length. Each vehicle is visible for a median duration of 13.6 s.

We run our algorithm on 210 testing graphs. Then, we take each testing graph as the initial frame and track each initial frame. We read the data frames backward until all vehicles on the initial frame disappear in the road segment. In this finite process, we record the number of vehicles out of the transmission range of corresponding CHs and leaving the initial clusters. In the whole process, we tracked 67,119 frames, and it involved 4558 vehicles. The number of vehicles breaking the initial clusters is shown in Fig. 7. In the above and following process, we both run ten times to eliminate randomness and then calculate the mean and standard deviation. The results show that our algorithm corresponds to the minimum number of vehicles breaking the initial clusters.

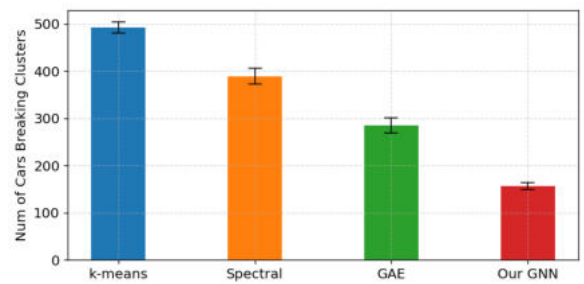


Fig. 7. Number of vehicles breaking the initial clusters.

On this basis, we evaluate the average cluster lifetime on the testing graphs, which indicates the time span that the CMs keep unchanged. If a cluster whose CMs remain unchanged during our whole trace on the highD dataset, its average cluster lifetime is this whole trace time. As shown in Fig. 8, the average cluster lifetimes are  $11.039 \pm 0.038$  s,  $11.231 \pm 0.099$  s,  $11.837 \pm 0.110$  s,  $12.069 \pm 0.037$  s with confidence 95% for  $k$ -means, Spectral Clustering, GAE, and our GNN. Clearly, the longer the average cluster lifetime is, the more stable the cluster is. Compared with baseline algorithms, our GNN-based clustering algorithm has the longest average cluster

lifetime, reaching 12 s, which is very close to the median duration of 13.6 s. Furthermore, methods using both features and graph structure, i.e., GAE and our GNN, are more stable than methods using either features or graph structure only.

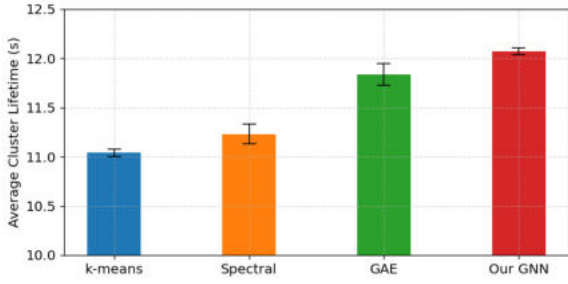


Fig. 8. Average cluster lifetime.

We also performed a quantitative analysis of coverage percentage (CP). In this paper, the CP is defined as:

$$CP = \frac{N - N_{Iso}}{N}, \quad (13)$$

where  $N_{Iso}$  is the number of isolated vehicles which do not belong to any cluster. As shown in Fig. 9, the coverage percentage of four algorithms are  $86.062 \pm 0.455\%$ ,  $98.383 \pm 0.173\%$ ,  $97.734 \pm 0.517\%$ ,  $98.927 \pm 0.111\%$  with confidence 95%, respectively. The results indicate our GNN-based algorithm's cluster efficiency outperforms baselines.

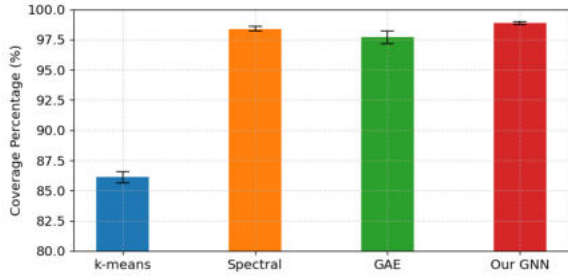


Fig. 9. Coverage percentage.

## V. CONCLUSION

In this paper, our research aimed to establish and maintain stable clusters for cooperative driving in VANET. Based on quantitative and qualitative analysis on the open-source highD traffic dataset, it can be concluded that our GNN-based clustering algorithm using both features and graph structure outperforms the baseline algorithms. Moreover, this is the very first attempt to solve the VANET clustering problem by applying GNN. Our intelligent and learnable GNN model gives the possibility to adapt to different traffic scenarios.

As future works, we plan to study other traffic scenarios like urban environment and Simulation of Urban MObility (SUMO) for long-term performance since the highD dataset is limited in total tracking time.

## ACKNOWLEDGMENT

This research is supported by NSF IRNC Grant No. 2029295.

## REFERENCES

[1] Transport Topics. "Traffic congestion costs billions in wasted fuel, time, report says." Mar. 28, 2014. [Online]. Available: <http://www.ttnews.com/articles/basetemplate.aspx?storyid=29007>.

[2] EPA (United States Environmental Protection Agency). Sources of Green-house Gas Emissions. Greenh. Gas Emiss. Accessed 3 Mar 2020. <https://www.epa.gov/ghgemissions/sources-greenhouse-gas-emissions>.

[3] P. Kavathekar and Y. Chen, "Vehicle platooning: A brief survey and categorization," in Proc. 7th ASME/IEEE Int. Conf. MESA/ASME DETC/CIE, 2011, pp. 1-17.

[4] D. Jia, K. Lu, J. Wang, X. Zhang, and X. Shen, "A Survey on Platoon-Based Vehicular Cyber-Physical Systems," IEEE Communications Surveys & Tutorials, 18(1), 2016, pp.263-284.

[5] C. Cooper, D. Franklin, M. Ros, F. Safaei, and M. Abolhasan, "A comparative survey of VANET clustering techniques," IEEE Communications Surveys & Tutorials, 19(1), 2016, 657-681.

[6] A. Katiyar, D. Singh, and R. S. Yadav, "State-of-the-art approach to clustering protocols in vanet: A survey," Wireless Networks, 26(7), 2020, 5307-5336.

[7] M. Ren, J. Zhang, L. Khoukhi, H. Labiod, and V. Vèque, "A review of clustering algorithms in VANETs," Annals of Telecommunications, 1-23, 2021.

[8] A. Daeinabi, A. G. P. Rahba, and A. Khademzadeh, "VWCA: An efficient clustering algorithm in vehicular ad hoc networks," J. Netw. Comput. Appl., vol. 34, no. 1, 2011, pp. 207-222.

[9] R. Chai, B. Yang, L. Li, X. Sun and Q. Chen, "Clustering-based data transmission algorithms for VANET," International Conference on Wireless Communications and Signal Processing, Hangzhou, 2013, pp. 1-6.

[10] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," IEEE transactions on neural networks and learning systems, 32(1), 2020, 4-24.

[11] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, M. Sun, and et al., "Graph neural networks: A review of methods and applications," AI Open, 1, 2020, 57-81.

[12] U. Von Luxburg, "A tutorial on spectral clustering," Statistics and computing, 17(4), 2007, 395-416.

[13] X. Bao, H. Li, G. Zhao, L. Chang, J. Zhou, and Y. Li, "Efficient clustering V2V routing based on PSO in VANETs," Measurement, 152, 2020, 107306.

[14] M. Ren, L. Khoukhi, H. Labiod, J. Zhang, and V. Veque, "A mobility- based scheme for dynamic clustering in vehicular ad-hoc networks(VANETs)," Vehicular Communications, 9, 2017, 233-241. Communications and Information Technologies (ISCIT), 2014, pp.233-237

[15] A. Bello Tambawal, R. Md Noor, R. Salleh, C. Chembe, and M. Oche, "Enhanced weight-based clustering algorithm to provide reliable delivery for VANET safety applications," PloS one, 14(4), 2019, e0214664.

[16] B. Azat, B and T. Hong, "Destination based stable clustering algorithm and routing for vanet," Journal of Computer and Communications, 8(01), 2020, 28.

[17] N. Taherkhani and S. Pierre, "Centralized and localized data congestion control strategy for vehicular ad hoc networks using a machine learning clustering algorithm. IEEE Transactions on Intelligent Transportation Systems," 17(11), 2016, 3275-3285.

[18] R. Chai, X. Ge, and Q. Chen, "Adaptive K-harmonic means clustering algorithm for VANETs," International Symposium on computing, networking and communications (WiMob), 2012, pp. 593-599.

[19] M. A. Saleem, S. Zhou, A. Sharif, T. Saba, M. A. Zia, A. Javed, M. Mittal, and et al., "Expansion of cluster head stability using fuzzy in cognitive radio CR-VANET," IEEE Access, 7, 2019, 173185-173195.

[20] G. Liu, N. Qi, J. Chen, C. Dong, and Z. Huang, "Enhancing clustering stability in VANET: A spectral clustering based approach. China Communications," 17(4), 2020, pp. 140-151.

[21] T. N. Kipf, and M. Welling, "Variational graph auto-encoders," arXiv preprint arXiv:1611.07308, 2016.

[22] L. A. Maglaras, and D. Katsaros, "Distributed clustering in vehicular networks," IEEE 8th international conference on wireless and mobile computing, networking and communications (WiMob), 2012, pp. 593-599.

[23] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," In Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 1025-1035.

[24] M. Wang, L. Yu, D. Zheng, Q. Gan, Y. Gai, Z. Ye, Z. Zhang, and et al., "Deep Graph Library: Towards Efficient and Scalable Deep Learning on Graphs," arXiv preprint arXiv:1909.01315, 2019.

[25] S. Cao, W. Lu, and Q. Xu. Grarep: Learning graph representations with global structural information. In KDD, 2015.

[26] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, S. Chintala, and et al., "Pytorch: An imperative style, high-performance deep learning library," Advances in neural information processing systems, 32, 2019, 8026-8037.

[27] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems," International Conference on Intelligent Transportation Systems (ITSC), 2018, pp. 2118-2125.



# Machine Learning-Based Power Loading for Massive Parallel Gaussian Channels

Min Jeong Kang and Jung Hoon Lee

Department of Electronics Engineering and Applied Communications Research Center,  
Hankuk University of Foreign Studies, Yongin, Korea  
{love\_minmin926, tantheta}@hufs.ac.kr

**Abstract**—In parallel channels, it is well known that the waterfilling is optimal power allocation that maximizes the sum achievable rate. However, the waterfilling requires iterative calculations, so may not be suitable especially when the number of total channels is very large or the delay constraint is very tight. In this paper, we propose machine learning-based power loading to reduce the computational complexity of power allocation in massive Gaussian parallel channels, which emulates on-off power allocation, where some of the channels equally share total transmit power. Our proposed scheme adopts a deep neural network structure that takes channel gains with total transmit power and returns an on-off power allocation strategy. The numerical results show that our proposed scheme achieves almost the same performance with the on-off power allocation with reduced complexity.

**Index Terms**—Machine learning (ML), waterfilling, on-off power allocation, parallel channels, deep neural network (DNN).

## I. INTRODUCTION

With the development of machine learning technology, there have been many attempts to use machine learning for wireless communications. Machine learning can be applied for many purposes; one purpose is to solve difficult problems such as joint optimization, and another purpose is to reduce the computational complexity when the optimal solution can be found, but its complexity is prohibitive. One popular machine learning technology is deep neural network (DNN), which accelerated the development of machine learning in many research areas [1].

In parallel channels, it is well known that the optimal power allocation is waterfilling. However, the waterfilling iteratively finds optimal powers, so may not be suitable when the number of total subchannels is very large or the delay constraint is very tight. One suboptimal power allocation with reduced complexity is on-off power configuration [2], where some of channels (“on” channels) equally share total power budget, while the others (“off” channels) are turned off. Also, the authors of [3] proposed multi-level power loading that generalizes on-off power configuration. However, these power allocation schemes still requires high complexity when the number of total channels are very large. The authors of [4] proposed the waterfilling with reduced complexity, where the

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2021R1H1A1010858) and by Hankuk University of Foreign Studies Research Fund of 2021.

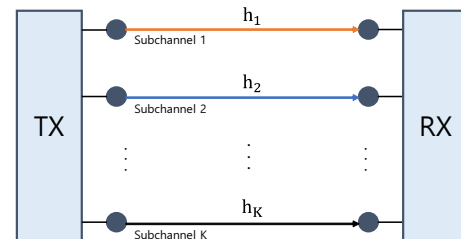


Fig. 1. System model.

waterfilling is over the subchannel groups, and subchannels in the same group use the equal power.

In this paper, we propose machine learning-based power loading to reduce the computational complexity of power allocation in massive Gaussian parallel channels. Our proposed scheme adopts a deep neural network (DNN) and emulates on-off power allocation. Thus, once trained in offline, our proposed scheme (with the DNN structure) can find power loading sequences after finite clocks. We evaluate our proposed scheme and show that our proposed scheme achieves almost the same performance with the on-off power allocation with reduced complexity.

## II. PROBLEM FORMULATION

### A. System Model

Our system model is illustrated in Fig.1. We consider a  $K$ -parallel Gaussian channel when the number of total subchannels (i.e.,  $K$ ) is very large. The received signal at the  $k$ th subchannel is modeled by

$$y_k = h_k \sqrt{p_k} s_k + n_k, \quad k \in \{1, \dots, K\}, \quad (1)$$

where  $h_k \in \mathbb{C}^{1 \times 1}$  is the channel at the  $k$ th subchannel, which is a circularly symmetric complex Gaussian random variables with zero mean and unit variance, i.e.,  $h_k \sim \mathcal{CN}(0, 1)$ . In this paper, without loss of generality, we assume that the subchannel gains are sorted in descending order such that

$$|h_1|^2 \geq \dots \geq |h_K|^2. \quad (2)$$

The variable  $s_k \in \mathbb{C}^{1 \times 1}$  is the  $k$ th transmit symbol such that  $|s_k|^2 = 1$ , and  $p_k$  is the transmit power for the  $k$ th symbol. Also,  $n_k$  is a complex Gaussian noise with zero mean and

unit variance, i.e.,  $n_k \sim \mathcal{CN}(0, 1)$ . When total transmit power budget is  $P$ , it should be satisfied that

$$\sum_{k=1}^K p_k = P. \quad (3)$$

The achievable rate at the  $k$ th subchannel is

$$\mathcal{R}_k = \log_2(1 + p_k |h_k|^2), \quad (4)$$

so the sum achievable rate becomes  $\sum_{k=1}^K \mathcal{R}_k$ .

**B. Waterfilling power allocation and on-off power configuration**

In parallel Gaussian channels, it is well known that the *waterfilling* is optimal power allocation for sum rate maximization. With the waterfilling, the optimal power for the subchannel  $k$  is given by

$$p_k^* = \left[ \frac{1}{\mu} - \frac{1}{|h_k|^2} \right]^+, \quad (5)$$

where  $\mu$  is a constant satisfying the total power constraint given in (3).

Although the waterfilling power allocation given in (5) maximizes the sum achievable rate, the value of  $\mu$  should be calculated with an iterative manner, so the computational complexity becomes huge burden when the number of total subchannels (i.e.,  $K$ ) is very large or the delay constraint is very tight.

One suboptimal power allocation is on-off configuration [2], where only some of subchannels (“on” subchannels) equally share total power budget, while the others (“off” subchannels) are turned off. In this case, power allocation sequence  $(p_1, \dots, p_K)$  is one among  $K$  power allocation sequences given by

$$\left\{ ([P/n]_n, [0]_{K-n}) \mid n = 1, \dots, K \right\}, \quad (6)$$

where  $[m]_n$  denotes the sequence of consecutive ‘ $m$ ’s repeated  $n$  times. For example, the power allocation  $([P]_1, [0]_{K-1})$  indicates the total power allocation to the first subchannel, i.e.,  $([P]_1, [0]_{K-1}) = (P, 0, \dots, 0)$ , while  $([P/N]_N, [0]_0)$  indicates equal power allocation over all subchannels, i.e.,  $([P/K]_K, [0]_0) = (P/K, \dots, P/K)$ .

When  $n(\leq K)$  subchannels shares total powers, the achievable rate with on-off configuration, the achievable rate is given by

$$\mathcal{R}_{\text{sum}}^{\text{on-off}}(n) = \sum_{k=1}^n \log_2 \left( 1 + \frac{P}{n} |h_k|^2 \right). \quad (7)$$

Thus, for the optimal on-off configuration, the transmitter need to compare total  $K$  power allocation sequences given in (6). The transmitter first solve the following problem

$$n^* = \arg \max_{n \in \{1, \dots, K\}} \mathcal{R}_{\text{sum}}^{\text{on-off}}(n), \quad (8)$$

and then obtain the optimal power allocation sequence as follows:

$$([P/n^*]_{n^*}, [0]_{K-n^*}). \quad (9)$$

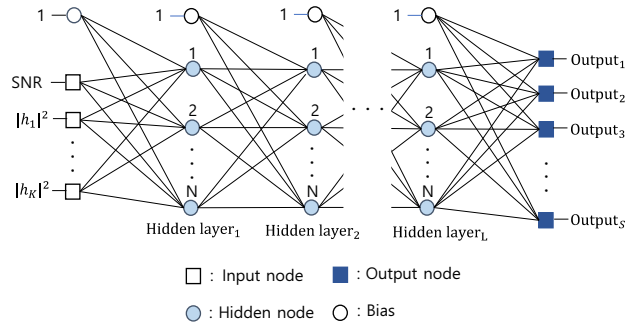


Fig. 2. Deep neural network (DNN) model for on-off power allocation.

### III. PROPOSED MACHINE LEARNING-BASED POWER LOADING

In this section, we explain our machine learning-based power loading scheme.

#### A. Basic idea

As we stated earlier, when waterfilling may not be suitable when the number of total subchannels is very large or the delay constraint is very tight. Although the suboptimal on-off configuration reduces the computational complexity of waterfilling, but still requires high complexity when the number of total subchannels are very large. Thus, to circumvent this complexity issue, we consider machine learning-based power loading that emulates the on-off power allocation (i.e., solves the problem in (8)). Note that once trained in offline, a DNN structure can yield a solution after finite clocks.

#### B. Structure of our proposed machine learning-based power loading

Fig. 2 shows the structure of our machine learning model. We consider a deep neural network (DNN) structure that takes the channel gains of  $K$  subchannels (i.e.,  $|h_1|^2, \dots, |h_K|^2$ ) with total power budget (i.e.,  $P$ ) and returns the optimal number of turned-on subchannels (i.e.,  $n^*$  in (8)). Thus, our machine learning structure has  $K + 1$  nodes in the input layer and  $K$  nodes in the output layer. Also, we consider  $L$  hidden layers, each of which is comprise of  $N$  nodes.

For activation functions, we employ the Rectified linear unit (ReLU) function and the Softmax function at each hidden node and each output node, respectively. Also, we consider the cross entropy for the loss function, which is widely used for weight correction in machine learning models defined as follows:

$$e = - \sum_{i=1}^S l_i \log_2 o_i + (1 - l_i) \log_2 (1 - o_i), \quad (10)$$

where  $l_i$  is the  $i$ th label. Since only one of the  $s$  data labels is the correct answer, only the correct label has the value of one, and the rest ones have the value of zero. In (10),  $o_i$  is the  $i$  output data of the  $i$  output node. Also, for optimization, we employ the adaptive momentum (AdaM) algorithm. Overfitting may occur if the initial learning rate

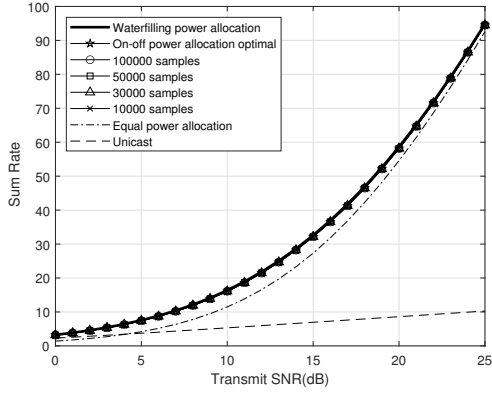


Fig. 3. The achievable sum rates of various schemes when  $K = 32$ .

is low, so the initial learning rate is set to 0.001. To prevent overfitting, we employed an early stop method, where training is stop when the loss value is not decreasing more than consecutive 20 times.

#### IV. NUMERICAL RESULTS

In this section, we evaluate our machine learning model. We assume that there are total 32 subchannels (i.e.,  $K = 32$ ). Thus, our machine learning model has 33 input nodes at the input layer and 32 output nodes at the output layer. Also, our machine learning model has five hidden layers, each of which has 600 nodes.

For evaluation, we vary the number of data samples when training our machine learning model. We consider four cases of  $(1, 3, 5, 10) \times 10^4$  data samples and for each case, 80% of data samples is used for training and the remaining 20% is used for validation. Then, we measure the performances of the machine learning models with the same 5000 test data samples. As reference schemes, we consider the waterfilling power allocation, the on-off configuration, equal power allocation, and unicast to the best subchannel (i.e., whole power to the best subchannel).

Fig. 3 shows the achievable sum rates of various schemes, while Fig. 4 shows the normalized ones with the waterfilling. As we can see in Fig. 4, on-off power allocation achieves about or more than 98% performance compared to the waterfilling. Also, our proposed machine learning models achieve almost the same performance with on-off power allocation.

Fig.5 shows how accurate our machine learning models emulate the on-off power allocation. As we can see, the accuracy tends to increases as the number of sample data increases in four cases.

#### V. CONCLUSION

In this paper, we proposed machine learning-based power loading to reduce the computational complexity of power allocation in massive Gaussian parallel channels, where the waterfilling may not be suitable because the number of total subchannels is very large or the delay constraint is very tight. Our proposed scheme emulates on-off power allocation,

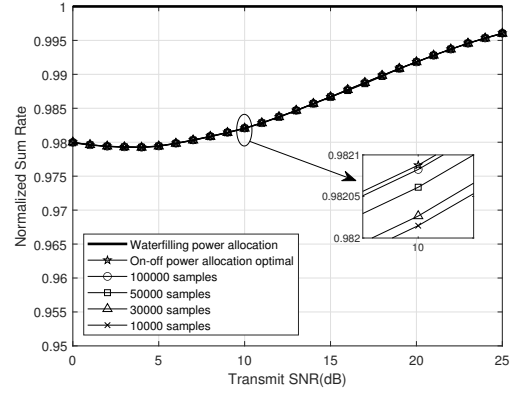


Fig. 4. The achievable sum rates of various schemes normalized with the achievable sum rate of the on-off configuration when  $K = 32$ .

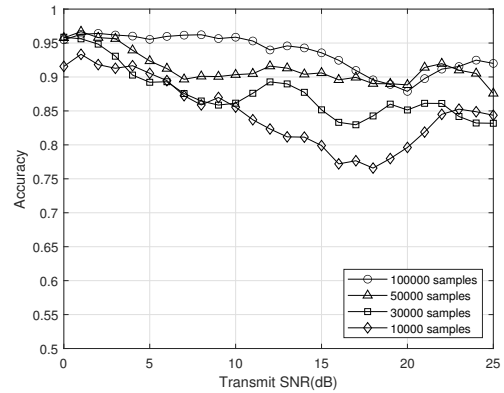


Fig. 5. The accuracy of our machine learning model when emulating the optimal on-off configuration.

where some of subchannels equally share total transmit power. In numerical results, we showed that our proposed machine learning structure achieves almost the same performance with the on-off power allocation with reduced complexity.

#### ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2021R1H1A1010858) and by Hankuk University of Foreign Studies Research Fund of 2021.

#### REFERENCES

- [1] J. Kaur, M. A. Khan, M. Iftikhar, M. Imran, and Q. E. U. Haq, "Machine learning techniques for 5G and beyond," *IEEE Access*, vol. 9, pp. 23472–23488, 2021.
- [2] A. Leke and J. M. Cioffi, "A maximum rate loading algorithm for discrete multitone modulation systems," in *Proc. 1997 IEEE Global Telecommunications Conf.*, vol. 3, pp. 1514–1518.
- [3] J. H. Lee and W. Choi, "Multi-level power loading using limited feedback," *IEEE Commun. Lett.*, vol. 16, no. 12, pp. 2024–2027, Dec. 2012.
- [4] Q. Qi, A. Minturn, and Y. Yang, "An efficient water-filling algorithm for power allocation in OFDM-based cognitive radio systems," in *Proc. 2012 International Conference on Systems and Informatics*, pp. 2069–2073.

# Enhanced Semi-persistent scheduling (e-SPS) for Aperiodic Traffic in NR-V2X

Malik Muhammad Saad, Muhammad Ashar Tariq, Md. Mahmudul Islam, Muhammad Toaha Raza Khan, Junho Seo, Dongkyun Kim

*School of Computer Science and Engineering, Kyungpook National University, Daegu, Republic of Korea*  
{maliksaad,tariqashar,mislam,toaha,jhseo,dongkyun}@knu.ac.kr

**Abstract**—In cellular vehicle-to-everything (C-V2X) mode 4 and New Radio V2X (NR-V2X) mode 2 based on local observations resources are scheduled by the vehicles themselves. For resource scheduling operation third generation partnership project (3GPP) defined semi-persistent scheduling (SPS). Vehicles rely on the sensing information received in sidelink control information (SCI) over physical sidelink control channel (PSCCH). Based on the sensing information vehicle select the resources for its transmission and reserve the resources for its successive future transmissions. For periodic transmission, SPS works fine comparatively to aperiodic messages. Because aperiodic messages compelled the vehicle to select new resources for its transmission based on the latency associated with the generated packet. In turn, it results in unutilized resources which were reserved before. This would also increase resource contention. To overcome this, we have proposed the enhanced semi-persistent scheduling (e-SPS) method for resource reservation for aperiodic traffic. The proposed scheme utilizes the reinforcement learning mechanism where each vehicle act as an agent. Based on the traffic density and speed of the vehicle, the size of the sensing window is dynamically adjusted and re-evaluation mechanism is also introduced to confirm the available resources by performing the sensing again while selecting the resources. The performance of the proposed scheme is evaluated in ns-3 and compared with the naïve sensing mechanism. Results show that the e-SPS scheme outperforms the others.

**Index Terms**—C-V2X, NR-V2X, Vehicular Communications, Aperiodic traffic, Semi-Persistent Scheduling (SPS), Cooperative Awareness Messages (CAMs)

## I. INTRODUCTION

The next-generation safety and management applications in vehicular environments are enabled by Long Term Evolution-Vehicle to Everything (LTE-V2X) communications. The awareness range of the autonomous and connected vehicle is extended in LTE-V2X utilizing the information received from neighboring vehicles, the infrastructure, and other users in the vicinity [1]. The 802.11p protocol is replaced by the Cellular-V2X (C-V2X) to enable enhanced applications owing to its scalability and flexibility. The C-V2X, proposed in Release 14 by the 3rd generation partnership project (3GPP), includes two transmission modes for direct communication, i.e., mode 3 over the Uu interface and mode 4 over the PC5 interface. In mode 3, the resources are allocated by the eNodeB (eNB) under its coverage region whereas, in mode 4, the vehicles autonomously reserve the resources in out of coverage regions [2], [3]. To evaluate the performance of C-V2X mode 4 in

the worst-case scenario, [4] presents an open-source simulator based on the network simulator ns-3.

These standardization efforts were continued by 3GPP for V2X communications in Release 16 and 17 towards new radio (NR) access. NR-V2X holds several enhancements to enable a wide range of V2X applications in order to increase the data rate, minimize the latency, and enhance the spectral efficiency of V2X communications [5]. Similar to C-V2X, NR-V2X includes two transmission modes: mode 1 (centralized) and mode 2 (decentralized). In mode 1, the resources are scheduled by the eNB when the vehicles are within the coverage region, whereas, in mode 2, the vehicles carry out the resource reservation by themselves using semi-persistence scheduling (SPS) algorithm in out of coverage region. The sensing-based SPS or SB-SPS assumes the exchange of periodic messages on the physical layer. These messages are called cooperative awareness messages (CAMs) in Europe, defined by the ETSI, and basic safety messages (BSMs) in the US, defined by the Society of Automotive Engineers (SAE). Using these awareness messages, the occupied resources in the last time interval are estimated and then the future usage of resources is predicted. Finally, the reservation of identified resources is carried out using the semi-persistent method for a given period of time [6]. In addition, although, according to the current 3GPP standard, the awareness messages (CAM and BSM) are not periodic, and several studies have shown that the interval between these messages as well as their size has a significant effect on the performance of medium access control (MAC) in distributed environments such as CV2X mode 4 or NR-V2X mode 2 [1].

Machine learning has opened a significant number of ways to find solutions in all domains. Similarly, in vehicular communications, the advanced tools of machine learning have been extremely beneficial in finding optimal solutions. Numerous studies have been carried out to tackle the problems in V2X communications using the advanced machine learning tools and algorithms such as the issues in resource allocation method in C-V2X mode 4 where vehicles are vulnerable to make self-ish decisions based on the limited available knowledge leading to contention in the network or the problem of excessive signaling overhead in C-V2X mode3 and so on. In this paper, we propose a solution for the distributed resource scheduling. The proposed scheme utilizes reinforcement learning and

reevaluation mechanism (also a potential technique drafted in 3GPP Rel.17 meetings) to reduce the resource contention in NR-V2X mode 2.

In the remainder of this paper section II presents the naive SPS method. Section III and section IV present the European Telecommunication Standard Institute (ETSI) standards related to CAM generation process and impact of aperiodic traffic generation on resource reservations respectively. Section V introduces the enhanced semi-persistent scheduling (e-SPS) that is proposed to address resource contention in NR-V2X. Performance results are evaluated in in Section VI. Finally the conclusion is drawn in section VII.

## II. SEMI-PERSISTENT SCHEDULING

In this section a short overview of naive scheduling meachism in NR-V2X mode 2 and C-V2X mode 4 is discussed.

In C-V2X mode 4 and NR-V2X mode 2, resources are scheduled by the vehicles using SPS. Based on SPS sensing vehicle user equipment (V-UE) reserved the number of subchannels for periodic and aperiodic transmission. In C-V2X vehicles sense for 1 sec, which is beneficial for periodic message transmission. Fig. 1. shows the illustration of sensing based SPS mechanism. The transmit V-UE will sense for 100 ms equivalent to 1000 subframes and then after 1 sec at time  $n$  select the resources in a resource selection window.

From Fig. 1 the lower bound of selection window  $t1$  is in between (1, 4) and upper bound  $t2$  is in between (20, 100).  $t1$  is defined by V-UE configuration, whereas  $t2$  depends upon the maximum inter reception time interval of packets. Based on probability of resource reselection, the resources are scheduled for the UE. If in a case, all the resources are occupied then vehicles will defer the channel access or transmit on occupied resources of low received power.

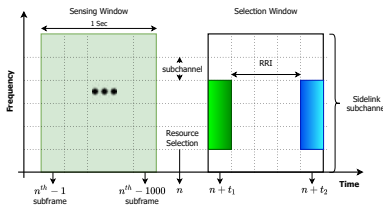


Fig. 1. Semi-Persistent Scheduling based Sensing Mechanism

## III. ETSI STANDARD FOR CAM GENERATION

The CAM generation depends on the following conditions as follows.

- **Speed:** A change in position by more than 4 m.
- **Heading:** A change of direction of  $\geq +/ - 4^\circ$
- **Change of speed:** A change of speed equal to or larger than  $0.5 \text{ msec}^{-1}$ .

In general the time of generation between CAM is not fixed and it varies [7]. In particular, the time between CAMs depends on the mobility of vehicles and the vehicles will generate more CAMs per second when their acceleration is higher. Moreover,

the size of CAM is also not fixed and it depends on the intelligent transportation system (ITS) packet datagram unit header (PDU), basic container, low frequency container and special vehicle container. Fig. 2 shows the CAM PDU. Basic container and high frequency containers are mandatory fields where former includes the position information of the vehicle and latter includes the velocity information of the vehicle. However, the size of CAM depends on the optional containers which include the low frequency container and special vehicle container. Optional containers include the information regarding the change of lane and to inform any accident to the neighboring vehicles.

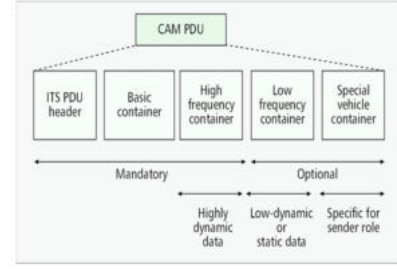


Fig. 2. Cooperative awareness message (CAM) packet format

## IV. IMPACT OF APERIODIC MESSAGE GENERATION ON THE RESOURCE RESERVATION

According to the ETSI standards, the generation of CAMs is no more periodic that can adversely affect the performance of resource scheduling in V2X. Since SPS based mechanism relies on the local observations, the vehicles made the announcement of their reserved resources in its SCI over PSCCH. The SCI includes RRI and RC value. The vehicle that wants to transmit its packet at time  $t$ , makes the reservation for its successive transmissions at  $(t + RRI)$  for RC times as shown in Fig. 1. The vehicle transmits the RRI and RC information in its SCI, so the neighboring vehicles get information about the already reserved resources. However, this might work in the case of ideally periodic traffic generation, wherein in the case of aperiodic traffic, this will result in unutilized resources. The following three cases are discussed which would result in unutilized resources and adversely affect the performance of resource scheduling in V2X.

### A. Resource reselection due to variation in message size

As shown in Fig. 3, the vehicle has selected the resource for its packet transmission of bytes  $N$  at time  $t$  and reserves its resources for future transmissions at a time  $(t + RRI)$ . If the next packet of size  $N' > N$  is generated at  $T_{gen2}$  that does not fit into the already reserved resource at  $(t + RRI)$ , then the vehicle should reserve the new resource for its transmission. In turn, the already reserved resources would result in unutilized resources.

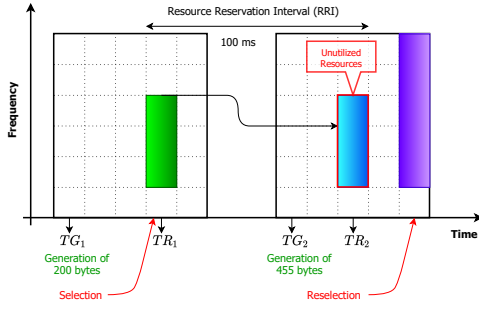


Fig. 3. Resource reselection due to Variation in Message Size

### B. Resource reselection due to the latency associated with the generated packet

Similarly, in this case, as shown in Fig. 4, if the next packet is generated at  $T_{gen_2}$  and the latency associated with the generated packet is let suppose 100 ms whereas the reserved resource is at  $(t + RRI) \geq 200$  ms. The vehicle should reserve the new resource at time  $t'$  for its transmission which should meet the latency of the generated packet. This would also result in unutilized resources which were already reserved at  $(t + RRI)$ .

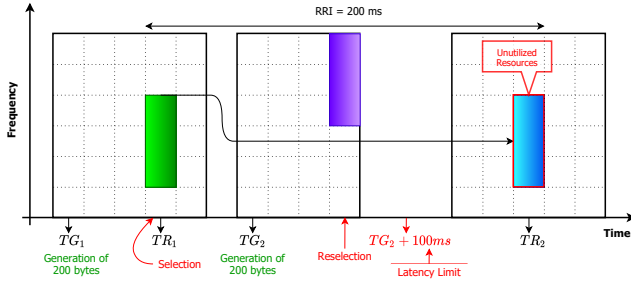


Fig. 4. Resource reselection due to latency associated with packet

### C. Resource reselection due to time interval between generations of packets

Likewise, in this case, as shown in Fig. 5, if the next packet is generated at  $T_{gen_2}$  such that  $T_{gen_2} > (t + RRI)$ , this would also result in unutilized resources at the time  $(t + RRI)$ .

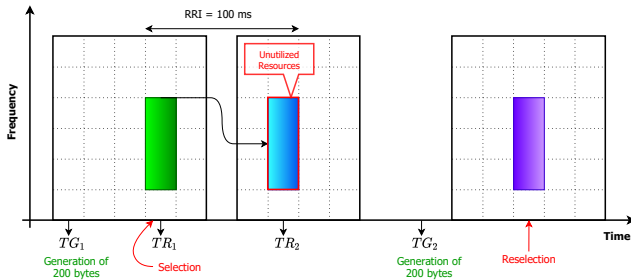


Fig. 5. Resource reselection due to variation in Message interval time

In order to address the resource scheduling for the aperiodic generation of messages, we proposed the e-SPS based scheme as discussed in Section V.

## V. ENHANCED SEMI-PERSISTENT SCHEDULLING (E-SPS)

In C-V2X mode 4, 3GPP defined a fixed sensing window size of 1 sec also known as Long-term sensing (LTS). The sensing window with a 1 sec duration includes 1000 subframes and 1000 slots with subcarrier spacing of 15 kHz. Where, NR-V2X with subcarrier spacing of 15 kHz, 30 kHz, 60 kHz includes 1000, 2000 and 4000 slots respectively. However, for both NR-V2X mode 2 and C-V2X mode 4 SPS is defined by the 3GPP and the sensing window size of 1 sec is considered. The current LTS mechanism can add delay in communication, in order to fulfil the ultra-low latency for future applications short-term sensing (STS) window is a potential solution proposed by the 3GPP in unlicensed channels. However, in LTS the sensing results become quickly obsolete due to the highly dynamic vehicular environment, wherein STS can add delay and increases resource contention if the vehicles could not find the available resources in a short duration. To complement the mode 2 operation of NR-V2X we proposed the enhanced-SPS (e-SPS) mechanism as explained following.

The sensing window size is adjusted dynamically based on the machine learning outcome. Deep reinforcement Q-learning is proposed to predict the sensing window size between 0.1 second to 1 second i.e.,  $[0.1, 0.2, 0.3, 0.8, \dots, 1]$  sec. The subframes that the vehicle needs to observe include the last  $[100, 200, 300, 800, \dots, 1000]$  subframes respectively corresponding to the sensing window size. Each vehicle act as an agent, that observes the environment. The state-space includes the density of the vehicles, the current vehicle speed, the last 20 temporal sequences of CAM generation intervals. The state space is shown in equation 1.

$$S_t = \{V, d, [i_{t-20}, i_{t-19}, \dots, i_{t-1}]\} \quad (1)$$

The action space set consists of sensing window size duration between (0.1, 1) seconds, resource reservation interval (RRI) between  $[0 - 99, 100, 200, \dots, 1000]$  ms, and resource reselection counter (RC) between (1 - 15). In the proposed e-SPS mechanism the RC and RRI are selected based on the vehicle current speed and temporal sequence of CAM generation intervals. This would assist in scheduling resources for aperiodic traffic.

The reward is designed based on the packet delivery ratio (PDR) as the goal is to reduce the resource contention and fair resource scheduling in V2X that lead to an increased in overall PDR and improved network performance.

Based on the machine learning outcome the e-SPS mechanism is summarized in the following three steps. In step 1, the vehicle observes the last subframes from the sensing window as selected based on the machine learning outcome. The vehicle based on the SCI information selects the sidelink resources for its transmission from the selection window. The length of the selection window is from  $[n + T_1, n + T_2]$ , where  $T_1$  is between (0ms - 4ms),  $n$  is the time at which the packet

is generated and T2 depends on the latency of the generated packet.

In step 2, the vehicle identifies the available resources that can be selected from the list of available resources ( $L_A$ ). The list is identified based on the sensing and the information received by the neighboring vehicles in SCI over PSCCH. The resources from the list  $L_A$  will be excluded based on the following conditions.

- The resources would be excluded if the RSSI received is higher than the threshold.
- Those resources would also be excluded based on the RRI received in SCI indicating other vehicles already reserved those slots for successive transmissions in future.

One more process is added in step 2 which is called a re-evaluation mechanism. In this process, while identifying the  $L_A$  the UE can again execute the sensing mechanism to make sure that the resources that have been added in the  $L_A$  are still available or not. This can also address the resource scheduling for the aperiodic generation of traffic. The length of the selection window is set between  $[(n' + T'_1), (n' + T'_2)]$ . Where  $n'$  is the slot at which the UE again executes the sensing mechanism and  $T'_2 = (n' - n)$ .  $T'_2$  shows the time elapsed since the generation of the initial sensing window at time  $n$ . During the re-evaluation mechanism if the resources that were identified before if still available the UE will not execute the step 2. If not then the UE will again execute the step 2 to identifies the  $L_A$  and then select the resource as explained in step 3.

In step 3, a vehicle will select the resources randomly from the candidate subframe resources list ( $L_C$ ).  $L_C$  includes the least RSSI which constitutes 20% of  $L_A$ . If the 20% target is not met the RSSI threshold is iteratively increased by 3 dB. For successive transmissions, the value of the RC depends on the output of machine learning and semi persistently resources are scheduled for future transmissions for RC times after each RRI interval.

## VI. SIMULATION RESULTS

### A. WINNER+ B1 Channel

The WINNER + B1 channel is considered as used by 3GPP [8]. The B1 channel model is considered for 5.9 GHz band and the antenna height set for vehicles is 1.5m. The path loss model is calculated for non -line of sight (NLOS) and line of sight model from WINNER+ B1 model [9]. Table I shows the simulation parameters.

TABLE I  
SIMULATION PARAMETERS

Vehicular Speed	20-130 kmh <sup>-1</sup>
Number of Vehicles	(100...300)
$T1, T2$	4 ms, [100, 200, ..., 1000] ms
Resource Blocks per Subchannels	10
Channel Model	WINNER+ B1
Transmission Power	23 dBm
Antenna Height	1.5 m

### B. Simulation

We consider a Manhattan grid scenario of  $500 \times 500$  m<sup>2</sup>. The proposed scheme is evaluated in sparse and dense traffic conditions. Vehicles that are considered are from 100-to-300 in numbers. The simulator is built in compliance with the 3GPP standards defined for C-V2X and NR-V2X. The enhanced semi-persistent scheduler is implemented to complement C-V2X mode 4 and NR-V2X mode 2.

Fig. 6, shows the impact of the increase in the number of vehicles on the PDR. The proposed scheme is compared with the naïve SPS mechanism. From the Fig. 6, with the increase in the number of vehicles the PDR degrades. The PDR degrades because of resource contention. However, the performance is better in the case of NR-V2X 30 kHz and 60 kHz subcarrier spacing as compared to C-V2X 15 kHz subcarrier spacing. This is due to the increase in number of slots as 2000 and 400 in NR-V2X with 30 kHz and 60 kHz subcarrier spacing. The subcarrier spacing with 30 kHz and 60 kHz can accommodate more number of vehicles in terms of resource assignment. This in turn reduces the probability of resource collisions such as by  $\lambda/2000$  and  $\lambda/4000$ . In order to improve the reliability and to reduce the packet dropped ratio the e-SPS scheme works effectively. Because of the re-evaluation mechanism, the e-SPS outperforms the others in each case. The e-SPS scheme helps vehicles to identify the available resources and in turn reduces the resource contention.

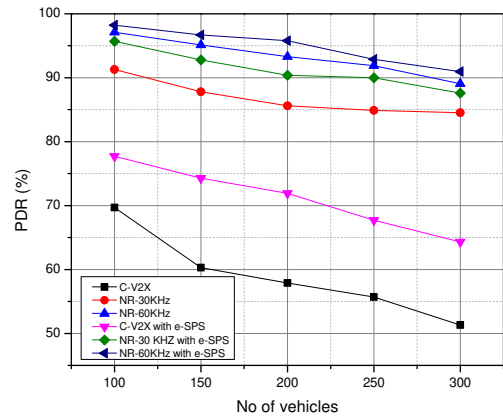


Fig. 6. Impact of number of vehicles

Fig. 7, shows the impact of the number of available sidelink subchannels on PDR. With the increase in the sidelink subchannels, the PDR gets better. This is because the increase in the number of sidelink channels results in more resources that vehicles can select for their transmission. From Fig. 7, it is shown even with less number of available sidelink subchannels the PDR is better with e-SPS based resource selection as compared to naïve SPS based scheme.

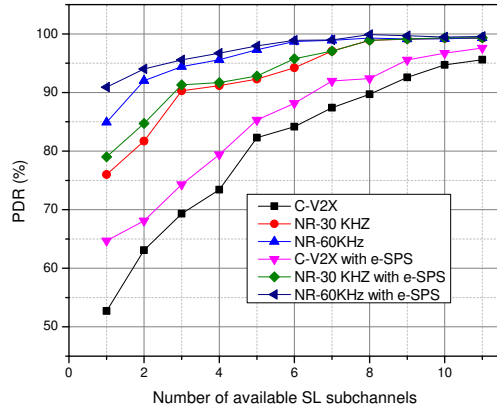


Fig. 7. Impact of number of available sidelink (SL) subchannels

## VII. CONCLUSIONS

In this paper, we have proposed the e-SPS method to complement NR-V2X mode 2 in order to schedule resources for aperiodic CAMs. If the resources are scheduled using the naive mechanism lead to the unutilized resources because of aperiodic CAMs. This also increased resource contention and degrades the overall network performance. The re-evaluation mechanism introduced in the e-SPS assists in the resource scheduling for aperiodic traffic. Also, each vehicle is modeled as an agent, and based on machine learning outcome the size of the sensing window is dynamically adjusted and the other parameters of the SPS are adjusted. This reduces resource contention and assists in conflict-free resource assignment for aperiodic message transmission. The performance results show the overall increase in network performance in terms of PDR.

## VIII. ACKNOWLEDGMENT

This study was supported by the BK21 Plus project (SW Human Resource Development Program for Supporting Smart Life) funded by the Ministry of Education, School of Computer Science and Engineering, Kyungpook National University, Korea (21A20131600005). In addition, this research was supported by the 2021 Kyungpook National University BK21 FOUR Graduate Innovation Project (International Joint Research Project for Graduate Students).

## REFERENCES

- [1] R. Molina-Masegosa, J. Gozalvez, and M. Sepulcre, "Comparison of IEEE 802.11p and LTE-V2X: An evaluation with periodic and aperiodic messages of constant and variable size," *IEEE Access*, vol. 8, pp. 121526–121548, 2020.
- [2] M. M. Saad, M. T. R. Khan, S. H. A. Shah, and D. Kim, "Advancements in vehicular communication technologies: C-V2X and NR-V2X comparison," *IEEE Communications Magazine*, vol. 59, no. 8, pp. 107–113, 2021.
- [3] R. Molina-Masegosa and J. Gozalvez, "LTE-V for sidelink 5G V2X vehicular communications: A new 5G technology for short-range vehicle-to-everything communications," *IEEE Vehicular Technology Magazine*, vol. 12, no. 4, pp. 30–39, 2017.

- [4] A. Nabil, K. Kaur, C. Dietrich, and V. Marojevic, "Performance analysis of sensing-based semi-persistent scheduling in c-v2x networks," in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, pp. 1–5, IEEE, 2018.
- [5] Z. Ali, S. Lagén, and L. Giupponi, "On the impact of numerology in NR V2X mode 2 with sensing and no-sensing resource selection," *arXiv preprint arXiv:2106.15303*, 2021.
- [6] S. Bartoletti, B. M. Masini, V. Martinez, I. Sarris, and A. Bazzi, "Impact of the generation interval on the performance of sidelink c-v2x autonomous mode," *IEEE Access*, vol. 9, pp. 35121–35135, 2021.
- [7] 3GPP, "Technical Specification Group Services and System Aspects; Study on Architecture Enhancements for the Evolved Packet System (EPS) and the 5G System (5GS) to Support Advanced V2X services," Tech. Rep. TR 23.786, 3GPP Rel. 16, 16, June 2019.
- [8] 3GPP, "Technical Specification Group Services and System Aspects," Tech. Rep. 21.914, 3GPP Rel.14, March 2017.
- [9] F. Eckermann, M. Kahlert, and C. Wietfeld, "Performance analysis of c-v2x mode 4 communication introducing an open-source c-v2x simulator," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, pp. 1–5, IEEE, 2019.



# Target Detection using U-Net for a DTV-based Passive Bistatic Radar System

Ji-Hun Park  
Department of Electrical and  
Electronics Engineering  
Pusan National University  
Busan, Republic of Korea  
pos02112@pusan.ac.kr

Do-Hyun Park  
Department of Electrical and  
Electronics Engineering  
Pusan National University  
Busan, Republic of Korea  
dohpark@pusan.ac.kr

Hyoung-Nam Kim  
Department of Electronics Engineering  
Pusan National University  
Busan, Republic of Korea  
hnkim@pusan.ac.kr

**Abstract**—Digital Television (DTV) has a wider bandwidth than a radio signal, and the signal power is stronger than Wi-Fi or LTE, so it is advantageous in exploiting a Passive Bistatic Radar (PBR) signal for drone detection. Most of the PBR systems detect a target using a Constant False Alarm Rate (CFAR) detector. However, CFAR detection suffers from multi-clutters, noise, and sidelobes. To overcome the limitation of CFAR, we propose a target detector exploiting semantic segmentation. The proposed detector is based on U-Net, a model for semantic segmentation. Training datasets for the proposed detector are generated by synthesized DTV signals. The performances of the CFAR and the proposed detector were compared using actual drone measurement data. The proposed detector detects drones better than the CFAR one while reducing the number of false alarms.

**Keywords**—DTV, passive bistatic radar, drone, target detector, semantic segmentation

## I. INTRODUCTION

For the past few years, different types of drones have been commonly used in many ways because of their excellent accessibility and low cost. However, despite using their advantages, there are many problems such as invasion of privacy and attacks on national facilities, and these cases can weaken the security of society and the national defense [1]. Therefore, it is essential to build a system for detecting and tracking drones.

Recently, many drone detection technologies using a Passive Bistatic Radar (PBR) have been presented. PBR has a structure in which a transmitter and a receiver are separated from each other, and illuminators of opportunity for PBR are broadcasting signals such as Frequency Modulation (FM), Digital Television (DTV), Wi-Fi, and so on [2]-[4]. Because of the structure, the detection area is more expansive than monostatic, and by using a high-power illuminator like DTV, it is easier to detect small objects [5]. Besides, since it is not necessary to set a transmitter separately, the cost of installation is reduced.

PBR receives a line of sight signal and target reflection signal to calculate the Cross Ambiguity Function (CAF) and estimates the target's bistatic distance and Doppler frequency from CAF. However, in the received signal, not only the target signal but also multi-clutters and noise can appear [3]. Moreover, sidelobes may exist in CAF due to signal characteristics [6]. Since these factors cause false alarm in target detection, it is essential to determine whether the target is within the Range Doppler (RD) map, which is the 2D matrix with bistatic range axis and Doppler frequency axis. Usually, PBR determines targets by applying the Constant False Alarm Rate (CFAR) detector to the CAF [7].

However, the CFAR detector needs to apply a separate algorithm that calculates the threshold value for a particular environment, and performance depends on the structure and parameters. In this paper, a new strategy for target detection is proposed by exploiting semantic segmentation. The proposed target detector detects a target by imaging and learning various synthetic data of CAF for drone detection. Unlike the CFAR detector, which calculates a threshold according to a noise figure within a specific range, the proposed method can use the more broad values of CAF to detect a target. Learning the synthetic data has the advantage of reducing false alarms and the accuracy of target detection.

## II. FUNDAMENTALS

### A. Passive Bistatic Radar

Fig. 1 is a simplified structure of PBR. In the signal reception, there are two channels, the reference channel and surveillance channel, and they receive a reference signal and target signal, respectively. Commonly, a reference antenna is used to obtain a direct path signal from a transmitter. A surveillance antenna is used to receive target or multi-clutter signals which are delayed signals of the reference signal.

In this process, mitigating the multi-clutter signals is essential because the target signal usually has lower power than the other signals. Typically, we can use the Extensive Cancellation Algorithm (ECA) to remove the clutters in the surveillance signal, making the target results more evident in the RD map [8].

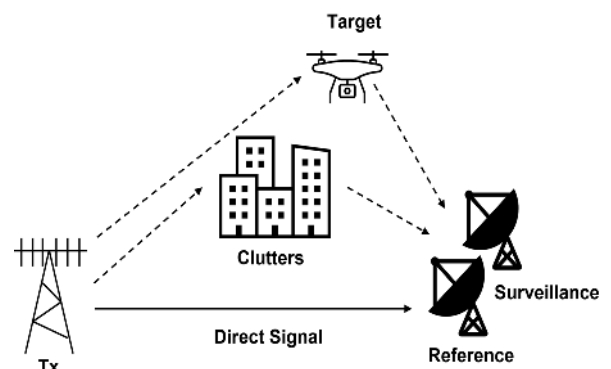


Fig. 1. Sketch of passive bistatic radar

CAF can be calculated by cross-correlation with a reference signal and surveillance signal. Bistatic range and Doppler frequency of target can be estimated to find a peak of

CAF. Target signal has delay and Doppler frequency shift, so when the RD map is drawn, information of target's bistatic range and velocity can be extracted. The CAF equation can be derived from time delay  $\tau$  and Doppler frequency  $f_d$ . The equation is represented by

$$C(\tau, f_d) = \int_{-\infty}^{\infty} x(t) s_{rf}^*(t - \tau) e^{-j2\pi f_d t} dt \quad (1)$$

where  $x(t)$  is result of ECA, and  $s_{rf}(t)$  is reference signal.

### B. Constant False Alarm Rate Detector

The structure of the CFAR detector is shown in Fig 2. The CFAR detector consists of a test cell, guard cell, and reference cell. It estimates the average power of noise to get the threshold value from the surrounding reference cells and compares the value with the test cell to determine the presence of a target [7]. At this time, the area around the target peak has relatively high power, so setting a guard cell can exclude them when calculating the average power. The threshold is adaptively determined according to the noise measurement environment to keep the false alarm probability of the test cell constant. Therefore, the type of calculating algorithm can affect the performance of detection. In this paper, under the assumption of a uniform noise figure, 1-D Cell-Average (CA) CFAR detector along the Doppler axis was used in drone detection. Equation (2) is the threshold calculated with average noise power  $n_{CA}$ , number of the reference cell  $N$ , and false alarm probability  $P_{FA}$ .

$$T_{CA} = n_{CA} (P_{FA}^{-1/N} - 1) \quad (2)$$

### C. Proposed U-Net-based detector

Semantic segmentation is a pixelwise classification method that predicts and labels whether an image has a specific class for each pixel to sort the objects in the image from one another [9]. In this paper, we exploited semantic segmentation method to target detection since the RD map displayed with single image as a result of CAF. Therefore, the most representative U-Net model in semantic segmentation was used [10]. The structure of U-Net has an encoding part for extracting the features of the input image and a decoding part for creating the desired result by using the convolutional neural network. The development of the input data coming out through the decoder shows probabilities of the classes for each pixel, and it is called a score map. Finally, U-Net classify the target by setting a threshold on the score map.

## III. EXPERIMENT AND RESULT

In this chapter, we will compare the results of applying the U-Net-based target detector and CFAR Detector to the actual data of the drone detection. About CFAR Detector, there were 8 guard cells, 16 reference cells. We has tests to find suitable false alarm probability in practical drone detection data and

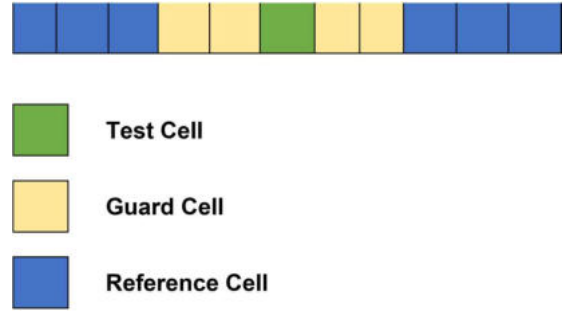


Fig 2. Simplified structure of 1-D CFAR detector

selected as  $10^{-4}$  for superior performance. Less than  $10^{-4}$ , the number of false alarm had increased, and in opposite case, the CFAR detector couldn't find the target.

When it comes to the U-Net-based target detector, we collected DTV signals and added a random target signal to the collected data to make synthetic training data. A total of 12,000 pieces of training data were generated through simulation with 10 signal-to-noise ratio (SNR) ranging uniformly from -44 to -35 dB, and the target positions were randomly placed on the RD map. In addition, the number of targets was set to be uniformly selected in range of 1 to 3 for each training data. We set the maximum value of target peaks on the RD map to label of training data. Fig 3. is an example pair of training and label data. In the Label, only the peak point is the target class, and others are noise. We exploited training optimizer of the U-Net-based target detector as Adam algorithm, and the first learning rate was  $10^{-4}$ , which reduced by 2% every 2 epochs [11]. The batch size was 32, and when the 50 epochs were completed, training was finished.

Fig. 4 shows the steering direction of the reference and surveillance channels and the path the drone moves. The reference channel was steered in the direction of Hwangyeongsan, where the DTV transmitting station is

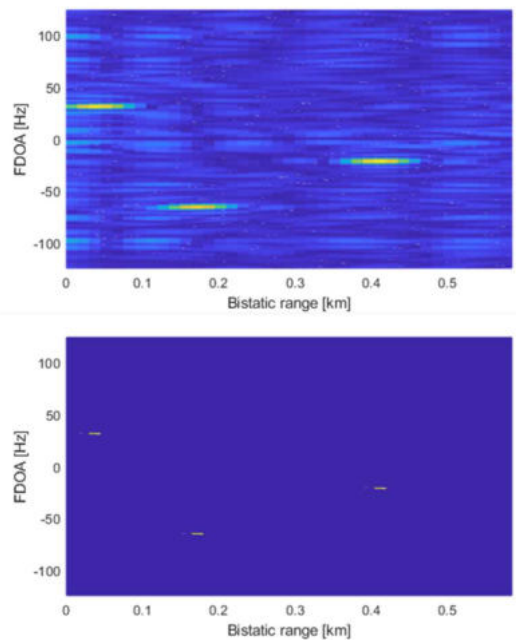


Fig 3. Example of synthetic training data and label

located, and the surveillance channel rotates about 90 degrees from the reference channel. A 701 MHz DTV signal was used to detect the drone, and the receiver was located on the roof of the Pusan National University building. The drone flew in parallel to the directivity of the surveillance channel for 1 minute while passing above the receiver.

In Fig 5. (a) shows a CAF with drone detection that bistatic range is about 90m, and Doppler frequency is about 38Hz, from one of the experiment data. It also can be shown that the noise and sidelobes are displayed on the RD map. Fig 5 (b) and (c) are the results of applying the CFAR detector and the trained U-Net-based target detector to Fig. 5 (a), respectively. Both the CFAR detector and the U-Net-based detector detected the drone well. However, (b) has more false alarms than (c). In addition, U-Net based detector's range resolution for target detection is more sensitive than the CFAR detector's, so it can estimate a more accurate bistatic range.

#### IV. CONCLUSION

Since the CFAR detector has problems of false alarm occurrence and inferior target resolution in the range axis, a U-Net-based target detector was proposed in this paper. To



Fig 4. Geometric environment of drone detection experiment

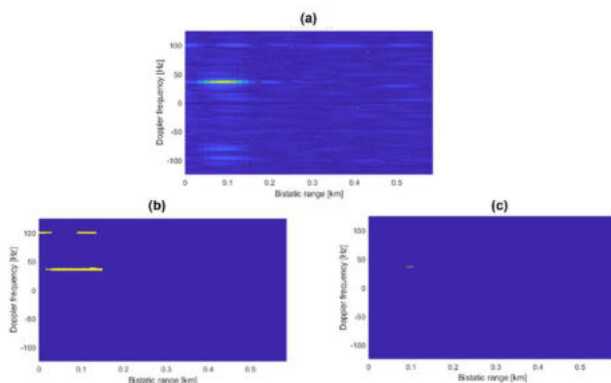


Fig 5. (a) CAF for drone detection, (b) Detection result via CFAR detector, (c) Detection result via proposed detector

learn the statistical characteristics of DTV-based CAF for the proposed detector, synthetic data by adding an arbitrary target to the actual data was exploited. In addition, the performance of the proposed detector was compared and analyzed with the conventional detector using the CFAR algorithm. As a result, we can see that the ability to eliminate the false alarm is better than that of the CFAR detector while the detection performance is not deteriorated. Moreover, the proposed detector has a more precise range resolution than the CFAR detector for target detection. Therefore, it can be regarded as a suitable PBR system target detector in various drone measurement environments.

#### ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2021R1F1A1060025).

#### REFERENCE

- [1] X. Zhang and K. Chandramouli, "Critical Infrastructure Security Against Drone Attacks Using Visual Analytics", *International Conference on Computer Vision Systems*, Thessaloniki, pp. 713-722, Nov 2019.
- [2] Geun-Ho Park, Dong-Gyu Kim, Ho Jae Kim, and Hyung-Nam Kim, "Maximum-likelihood angle estimator for multi-channel FM-radio-based passive coherent location," *IET Radar, Sonar & Navigation* vol. 12, no. 6, pp. 617-625, May 2018.
- [3] J. E. Palmer, H. A. Harms, S. J. Searle and L. Davis, "DVB-T Passive Radar Signal Processing," *IEEE Transactions on Signal Processing*, vol. 61, no. 8, pp. 2116-2126, April 2013.
- [4] W. Li, R. J. Piechocki, K. Woodbridge, C. Tang and K. Chetty, "Passive WiFi Radar for Human Sensing Using a Stand-Alone Access Point," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 3, pp. 1986-1998, March 2021.
- [5] G. Fang, J. Yi, X. Wan, Y. Liu and H. Ke, "Experimental Research of Multistatic Passive Radar With a Single Antenna for Drone Detection," *IEEE Access*, vol. 6, pp. 33542-33551, June 2018.
- [6] G. Bournaka, M. Ummerhofer, D. Cristallini, J. Palmer and A. Summers, "Experimental Study for Transmitter Imperfections in DVB-T Based Passive Radar," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 3, pp. 1341-1354, June 2018.
- [7] W. A. Holm, *Principles of Modern Radarm*, Raleigh, NC, USA: Scitech Publishing, Inc., 2007.
- [8] F. Colone, D. W. O'Hagan, P. Lombardo and C. J. Baker, "A Multistage Processing Algorithm for Disturbance Removal and Target Detection in Passive Bistatic Radar," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 45, no. 2, pp. 698-722, April 2009.
- [9] E. Shelhamer, J. Long and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640-651, April 2017.
- [10] O. Ronnerberger, P. Fischer and T. Brox, "*U-Net: Convolutional Networks for Biomedical Image Segmentation*", *Medical Image Computing and Computer-Assisted Intervention*, vol 9351. Springer, May 2015.
- [11] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Jan. 2017, *e-print arXiv:1412.6980v9*.

# Privacy-preserving collaborative machine learning in biomedical applications

Wonsuk Kim  
School of Electrical Engineering,  
Korea University,  
Seoul, South Korea  
Email: won425@korea.ac.kr

Junhee Seok  
School of Electrical Engineering,  
Korea University,  
Seoul, South Korea  
Email: jseok14@korea.ac.kr

**Abstract**—Machine learning (ML) algorithms are now widely used to tackle computational problems in diverse domains. In biomedicine, the rapidly growing amounts of experimental data increasingly necessitate the use of ML to discern complex data patterns. However, biomedical data is often considered sensitive, and the privacy of individuals behind the data is increasingly put at risk as a result. Traditional methods such as anonymization and pseudonymization are not always applicable and have limited effectiveness with respect to risk mitigation. Privacy researchers are actively developing alternative approaches to privacy protection, including strategies based on cryptography, such as homomorphic encryption and secure multiparty computation. This paper discusses recent advances in biomedical applications of these privacy techniques. We first review the key privacy techniques, then provide an overview of their applications in biomedical machine learning. Finally, we highlight the remaining challenges of current approaches and suggest directions for future work.

**Index Terms**—Privacy-Preserving Machine Learning, Collaborative Learning, Federated Learning, Secure Multi-party Computation

## I. INTRODUCTION

Machine learning algorithms have revolutionized the way we solve problems in many domains, including computer vision, natural language processing, physical simulations, stock and housing market predictions, and biomedicine [1], [2], [3], [4], [5]. Since machine learning is driven by data, the quantity and quality of the data determine the performance of these algorithms. However, especially in biomedicine, it is often difficult to gather large amounts of data due to privacy and intellectual property issues associated with data sharing.

A traditional approach to protecting the privacy of biomedical data is to apply de-identification techniques [6]. "Anonymization" and "pseudonymization" are the most widely used methods for de-identification of private data. The possibility of re-identification can be reduced by removing identifying data features (anonymization) or replacing them with a random identifier for each subject (pseudonymization). However, these techniques are limited because they are still vulnerable to re-identification attacks, e.g. when linked with additional datasets [7].

Recent advances in cryptographic frameworks to overcome traditional limitations enabled new approaches to protecting

privacy. In this paper, we present a review of emerging techniques for enhancing privacy in machine learning workflows in biomedicine based on a survey of recent literature. We discuss the promises and challenges of these techniques, and conclude with an outlook on future developments.

## II. PRIVACY-PRESERVING COLLABORATIVE MACHINE LEARNING

Privacy-preserving technologies are primarily used to allow multiple input parties to collaboratively train ML models without revealing sensitive data, and it mainly includes technologies such as differential privacy, homomorphic encryption, trusted execution environment, and secure multiparty computation.

### A. Differential privacy

Differential privacy (DP) [8] is a theoretical framework for releasing statistics from a dataset while limiting the leakage of information associated with each individual by adding a controlled amount of noise to the released data. Owing to these advantages, DP is adopted by the US Census Bureau [9] and several major technology organizations, including Google [10], Apple [11] and Microsoft [12]. These companies adopted DP to get insight from user behavior without revealing individual users' browsing habits. DP is also used in research dealing with genetic data and clinical data [13], [14]. DP can be applied not only to the dataset, but also to the parameters of algorithms or algorithm updates during training [15], [16]. However, there are limitations in that it reduces the accuracy of the model and makes it ambiguous to define an appropriate noise level.

### B. Homomorphic encryption

Homomorphic encryption (HE) [17] is a cryptographic technique that allows computations on encrypted data without first decrypting it. By ensuring that no one can read or modify the data, HE can keep data safe, even in untrusted environments such as public clouds or external parties. Owing to this advantage, HE has numerous applications in genomics and biomedicine, where data is mainly spread across multiple institutions. For example, secure logistic regression and secure statistical tests in genome-wide association studies (GWAS) have

been proposed using HE [18], [19]. However, since HE mainly supports only addition and multiplication operations [20], the major limitation is that it is difficult to develop complex AI models with non-linear operations such as deep neural networks (DNNs) using HE. CryptoNets [21], training neural networks using HE, approximated non-polynomial functions such as sigmoid and rectified linear units and adopted mean-pooling layers instead of max-pooling layers. Also, HE is computationally expensive because it operates with encrypted data [22].

### C. Trusted execution environment

Trusted execution environment (TEE) [23], also known as secure enclaves, provides hardware-level isolation and memory encryption on every server or device, which keeps application code and the data hidden from end users, credentialed insiders and third parties. For example, Intel Software Guard Extensions (SGX), ARM TrustZone and AMD Secure Processor allows the execution of the applications or code inside the enclaves that claims to be secure. With hardware-level isolated execution using TEE, secure implementation of genetic analytics has been applied [24], [25], [26], [27]. The major challenges for TEE are limited scalability of enclave memory and the need for additional development using software development kits (SDKs) from vendors providing TEE. Side-channel attacks are also a threat, such as timing attack, which is based on measuring the time taken to perform various computations [28]. To overcome these challenges, recent studies proposed efficient scalable frameworks based on TEE [29] and software timer manipulation to prevent side-channel attacks [30].

### D. Federated Learning

Federated learning (FL) [31] is a machine learning technique that trains an algorithm across multiple decentralized edge devices or servers holding local data samples without exchanging them. This approach stands in contrast to traditional centralized machine learning techniques where all the local datasets are uploaded to one server. In FL, a server coordinates a network of nodes as demonstrated in Figure 1, each of which has training data that it cannot or will not share directly. The nodes each train a model, and they share the model or the updates of the model with the server. By not transferring the data itself, federated learning helps to ensure privacy and minimizes communication costs of sharing data.

Since the final model in FL is aggregated using models trained with local data, one might think that the model is biased or the performance is poor, but it can be overcome by regularization and frequent synchronizations of the models. Therefore, FL is mainly used for applications using clinical data as summarized in Table I. Secure implementation for survival analysis [32] and secure prediction of in-hospital mortality [33] using electrical health records (EHR) was proposed. Privacy-preserving structure for epileptic seizure detection [34] and arrhythmia detection [33] using electroencephalography (EEG) and electrocardiography (ECG) were also studied.

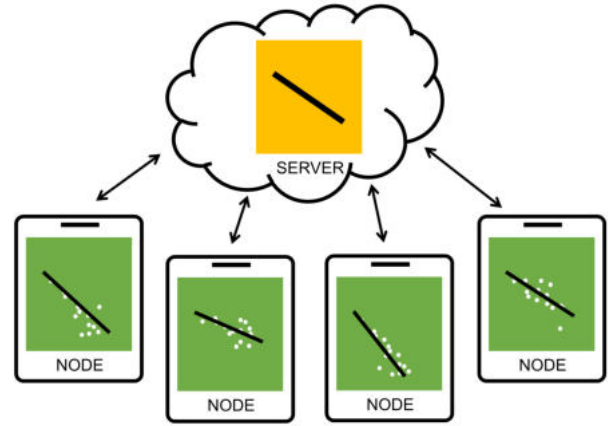


Fig. 1. The architecture of FL framework.

Secure diabetic retinal detection using retinal image [35], and prostate cancer diagnosis model using MRI data [36] were also proposed. The challenges in FL is that the data owners have to perform computations on the device or server that holds data and the performance is worse than centralized training. So, performance issues can occur if the data owners have limited computational capacity or small amount of local data. Since local data is not encrypted and the private data can be leaked only by the gradients of the model [37], [38], privacy is not fully guaranteed.

### E. Secure multiparty computation

Secure multiparty computation (SMPC) [39] is a cryptographic protocol that distributes computations across multiple parties, where individual parties cannot see the data of others. In other words, SMPC allows joint analysis of data without sharing the raw data. The architecture of SMPC is shown as Figure 2. SMPC protocol utilizes a well-established cryptographic concept called additive secret sharing. Each input party sends a different secret to each computation nodes. Each computation node computes the result on the secret shares from the input parties and shares the results with other computation nodes. Each computation node computes the final result by aggregating the results of all nodes. The SMPC-based privacy-preserving algorithm has been applied to medical diagnosing pneumonia and hepatitis [40], survival analysis [41], drug-target interaction using genetic data [42], [43] and quality control and population stratification for large-scale GWAS [44]. Although SMPC has a wider range of available operations than HE and is more efficient than HE in terms of computational cost, there are limitations in that it is a highly interactive computation and requires a large amount of communication between parties. The models implemented in SMPC are relatively simple compared to FL, as shown in Table I, because they have bottlenecks in performing non-linear operations such as ReLU, which are essential for implementing complex AI models. However, more privacy is

guaranteed because only unidentifiable secret shares remain on each node.

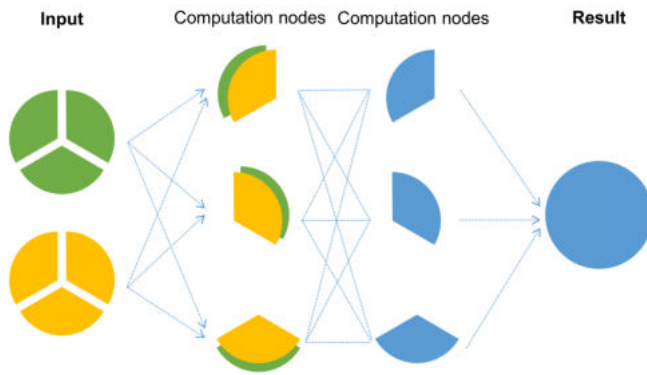


Fig. 2. The architecture of SMPC framework.

### III. OUTLOOK

In this paper, we investigated use cases applied to biomedical data, focusing on FL, which is flexible in terms of types of operations when training predictive models while keeping the data local and private, and SMPC, which encrypts both data and models as shown in Table I. Since SMPC still suffers from high overhead of cryptographic operations, it has been applied to EHR and genomic data, which are relatively low-dimensional among biomedical data, and implemented less complex models with fewer parameters compared to FL. To enhance the feasibility of SMPC, cryptography-side research has been conducted such as improving the encryption protocol [45] or using GPUs [46]. On the other hand, there have been ML-side approaches to reduce these bottlenecks. It is possible to speed up the runtime while maintaining the performance by reducing the number of ReLUs [47], [48] and using learnable scalars instead of the batch normalization layer [49].

Beyond the aforementioned issues and challenges, there are a number of practical concerns when applying FL and SMPC in production. Both technologies require a system design that considers the heterogeneity of the hardware level because computations are performed on multiple servers or devices. Since various environments such as hardware capacity, network connectivity, power, and physical location exist for each device, asynchronous communication [50] or fault-tolerant training methods [51] are necessary when applying FL, and not only semi-honest models but also malicious models [52] are required when applying SMPC.

Beyond the application of privacy-preserving techniques to supervised learning using biomedical data, there are many other problems that can be applied. For example, it can be applied to clustering for data analysis [53], reinforcement learning to infer treatment policies for patients [54], generative learning to impute missing genomic data [55] or to generate fake data to anonymize healthcare data [56]. Therefore, improving the runtime of these privacy-preserving techniques,

solving the challenges that arise in production, and broadening the scope of application will be the future direction.

### IV. CONCLUSION

With the advances in machine learning, various AI applications are emerging in biomedicine. However, the more AI models are trained on private biomedical data, the more the importance of the privacy and intellectual property issues increases. Thus, privacy-preserving techniques, such as DP, HE, TEE, FL and SMPC, are critical to making biomedical data more available and accessible. Among them, SMPC and FL were the most accurate methods, followed by HE and DP. A hybrid approach [57] (e.g., applying both FL and MPC) can be applied, and speeding up the operation in privacy technique to achieve a better-performing privacy-preserving model can be a future work.

### ACKNOWLEDGMENT

This research was supported by the MOTIE (Ministry of Trade, Industry, and Energy) in Korea, under the Fostering Global Talents for Innovative Growth Program (P0008749) supervised by the Korea Institute for Advancement of Technology (KIAT) and National Research Foundation of Korea (NRF-2019R1A2C1084778).

### REFERENCES

- [1] W. Kim and J. Seok, "Indoor semantic segmentation for robot navigating on mobile," in *2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*, 2018, pp. 22–25.
- [2] E. Alpaydin, *Introduction to machine learning*. MIT press, 2020.
- [3] W. Kim and J. Seok, "Simulation acceleration for transmittance of electromagnetic waves in 2D slit arrays using deep learning," *Scientific reports*, vol. 10, no. 1, pp. 1–8, 2020.
- [4] H. Cho, B. Berger, and J. Peng, "Compact integration of multi-network topology for functional analysis of genes," *Cell systems*, vol. 3, no. 6, pp. 540–548, 2016.
- [5] H. Cho, D. Ippolito, and Y. W. Yu, "Contact tracing mobile apps for COVID-19: Privacy considerations and related trade-offs," *arXiv preprint arXiv:2003.11511*, 2020.
- [6] B. Berger and H. Cho, "Emerging technologies towards enhancing privacy in genomic data sharing," 2019.
- [7] G. A. Kassis, M. R. Makowski, D. Rückert, and R. F. Braren, "Secure, privacy-preserving and federated machine learning in medical imaging," *Nature Machine Intelligence*, vol. 2, no. 6, pp. 305–311, 2020.
- [8] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3-4, pp. 211–407, 2014.
- [9] J. M. Abowd, "The US Census Bureau adopts differential privacy," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2867–2867.
- [10] Úlfar Erlingsson, V. Pihur, and A. Korolova, "Rappor: Randomized aggregatable privacy-preserving ordinal response," in *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, 2014, pp. 1054–1067.
- [11] A. G. Thakurta, A. H. Vyrros, U. S. Vaishampayan, G. Kapoor, J. Freudiger, V. R. Sridhar, and D. Davidson, "Learning new words," 2017.
- [12] B. Ding, J. Kulkarni, and S. Yekhanin, "Collecting telemetry data privately," *arXiv preprint arXiv:1712.01524*, 2017.
- [13] H. Cho, S. Simmons, R. Kim, and B. Berger, "Privacy-preserving biomedical database queries with optimal privacy-utility trade-offs," *Cell systems*, vol. 10, no. 5, pp. 408–416, 2020.
- [14] M. Winslett, Y. Yang, and Z. Zhang, "Demonstration of damson: Differential privacy for analysis of large data," in *2012 IEEE 18th International Conference on Parallel and Distributed Systems*, 2012, pp. 840–844.

TABLE I  
SUMMARY OF RECENT LITERATURE ON FEDERATED LEARNING AND SECURE MULTIPARTY COMPUTATION APPROACHES IN GENOMICS AND BIOMEDICINE. (FL: FEDERATED LEARNING, SMPC: SECURE MULTIPARTY COMPUTATION, CNN: CONVOLUTIONAL NEURAL NETWORK, MLP: MULTI-LAYER PERCEPTRON, LSTM: LONG SHORT-TERM MEMORY)

Privacy Technique	Model	Authors	Year	Application
FL	MLP, CNN	Baghersalimi et al. [34]	2021	epileptic seizure detection
FL	CNN	Karthik et al. [36]	2021	prostate segmentation, MRI diagnosis of cancer
FL	LSTM	Lee et al. [33]	2020	in-hospital mortality prediction
FL	CNN	Lee et al. [33]	2020	arrythmia detection
FL	CNN	Balachandar et al. [35]	2020	diabetic retinopathy detection
FL	cox regression	Dai et al. [32]	2020	survival analysis
SMPC	logistic regression	Li et al. [40]	2021	medical diagnosis - pneumonia, hepatitis
SMPC	log-rank test, Kaplan-Meier estimator	von Maltitz et al. [41]	2021	survival analysis
SMPC	MLP	Ma et al. [42]	2020	drug-target interaction
SMPC	MLP	Hie et al. [43]	2018	drug-target interaction
SMPC	quality control, population stratification	Cho et al. [44]	2018	genetic associations

- [15] J. Dong, A. Roth, and W. J. Su, "Gaussian differential privacy," *arXiv preprint arXiv:1905.02383*, 2019.
- [16] N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, and Úlfar Erlingsson, "Scalable private learning with pate," *arXiv preprint arXiv:1802.08908*, 2018.
- [17] C. Gentry, *A fully homomorphic encryption scheme*. Stanford university, 2009.
- [18] M. Kim, Y. Song, S. Wang, Y. Xia, and X. Jiang, "Secure logistic regression based on homomorphic encryption: Design and evaluation," *JMIR medical informatics*, vol. 6, no. 2, p. e8805, 2018.
- [19] T. Morshed, D. Alhadidi, and N. Mohammed, "Parallel linear regression on encrypted data," in *2018 16th Annual Conference on Privacy, Security and Trust (PST)*, 2018, pp. 1–5.
- [20] L. Morris, "Analysis of partially and fully homomorphic encryption," *Rochester Institute of Technology*, pp. 1–5, 2013.
- [21] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy," in *International conference on machine learning*, 2016, pp. 201–210.
- [22] C. Moore, M. O'Neill, E. O'Sullivan, Y. Doröz, and B. Sunar, "Practical homomorphic encryption: A survey," in *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2014, pp. 2792–2795.
- [23] M. Sabt, M. Achemlal, and A. Bouabdallah, "Trusted execution environment: what it is, and what it is not," in *2015 IEEE Trust-com/BigDataSE/ISPA*, vol. 1, 2015, pp. 57–64.
- [24] N. Dokmai, C. Kockan, K. Zhu, X. Wang, S. C. Sahinalp, and H. Cho, "Privacy-preserving genotype imputation in a trusted execution environment," *Cell Systems*, 2021. [Online]. Available: <https://doi.org/10.1016/j.cels.2021.08.001>
- [25] F. Chen, S. Wang, X. Jiang, S. Ding, Y. Lu, J. Kim, S. C. Sahinalp, C. Shimizu, J. C. Burns, and V. J. Wright, "Princess: Privacy-protecting rare disease international network collaboration via encryption through software guard extensions," *Bioinformatics*, vol. 33, no. 6, pp. 871–878, 2017.
- [26] F. Chen, C. Wang, W. Dai, X. Jiang, N. Mohammed, M. M. A. Aziz, M. N. Sadat, C. Sahinalp, K. Lauter, and S. Wang, "Presage: privacy-preserving genetic testing via software guard extension," *BMC medical genomics*, vol. 10, no. 2, pp. 77–85, 2017.
- [27] C. Kockan, K. Zhu, N. Dokmai, N. Karpov, M. O. Kulekci, D. P. Woodruff, and S. C. Sahinalp, "Sketching algorithms for genomic data analysis and querying in a secure enclave," *Nature methods*, vol. 17, no. 3, pp. 295–301, 2020.
- [28] W. Wang, G. Chen, X. Pan, Y. Zhang, X. Wang, V. Bindschaedler, H. Tang, and C. A. Gunter, "Leaky cauldron on the dark land: Understanding memory side-channel hazards in SGX," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 2421–2434.
- [29] J.-B. Truong, W. Gallagher, T. Guo, and R. J. Walls, "Memory-Efficient Deep Learning Inference in Trusted Execution Environments," *arXiv preprint arXiv:2104.15109*, 2021.
- [30] W. Huang, S. Xu, Y. Cheng, and D. Lie, "Aion Attacks: Manipulating Software Timers in Trusted Execution Environment."
- [31] J. Koney, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.
- [32] W. Dai, X. Jiang, L. Bonomi, Y. Li, H. Xiong, and L. Ohno-Machado, "VERTICOX: Vertically Distributed Cox Proportional Hazards Model Using the Alternating Direction Method of Multipliers," *IEEE Transactions on Knowledge and Data Engineering*, 2020.
- [33] G. H. Lee and S.-Y. Shin, "Federated learning on clinical benchmark data: Performance assessment," *Journal of medical Internet research*, vol. 22, no. 10, p. e20891, 2020.
- [34] S. Baghersalimi, T. Teijeiro, D. Atienza, and A. Aminifar, "Personalized Real-Time Federated Learning for Epileptic Seizure Detection," *IEEE Journal of Biomedical and Health Informatics*, 2021.
- [35] N. Balachandar, K. Chang, J. Kalpathy-Cramer, and D. L. Rubin, "Accounting for data variability in multi-institutional distributed deep learning for medical imaging," *Journal of the American Medical Informatics Association*, vol. 27, no. 5, pp. 700–708, 2020.
- [36] K. V. Sarma, S. Harmon, T. Sanford, H. R. Roth, Z. Xu, J. Tetreault, D. Xu, M. G. Flores, A. G. Raman, and R. Kulkarni, "Federated learning improves site performance in multicenter deep learning without data sharing," *Journal of the American Medical Informatics Association*, vol. 28, no. 6, pp. 1259–1264, 2021.
- [37] A. Wainakh, F. Ventola, T. Mü, J. Keim, C. G. Cordero, E. Zimmer, T. Grube, K. Kersting, and M. Mühlhäuser, "User Label Leakage from Gradients in Federated Learning," *arXiv preprint arXiv:2105.09369*, 2021.
- [38] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, 2019, pp. 2512–2520.
- [39] Y. Lindell, "Secure multiparty computation for privacy preserving data mining," *Encyclopedia of Data Warehousing and Mining*, pp. 1005–1009, 2005.
- [40] D. Li, X. Liao, T. Xiang, J. Wu, and J. Le, "Privacy-preserving self-serviced medical diagnosis scheme based on secure multi-party computation," *Computers & Security*, vol. 90, p. 101701, 2020.
- [41] M. von Maltitz, H. Ballhausen, D. Kaul, D. F. Fleischmann, M. Niyazi, C. Belka, and G. Carle, "A Privacy-Preserving Log-Rank Test for the Kaplan-Meier Estimator With Secure Multiparty Computation: Algorithm Development and Validation," *JMIR medical informatics*, vol. 9, no. 1, p. e22158, 2021.
- [42] R. Ma, Y. Li, C. Li, F. Wan, H. Hu, W. Xu, and J. Zeng, "Secure multiparty computation for privacy-preserving drug discovery," *Bioinformatics*, vol. 36, no. 9, pp. 2872–2880, 2020.
- [43] B. Hie, H. Cho, and B. Berger, "Realizing private and practical pharmacological collaboration," *Science*, vol. 362, no. 6412, pp. 347–350, 2018.
- [44] H. Cho, D. J. Wu, and B. Berger, "Secure genome-wide association analysis using multiparty computation," *Nature biotechnology*, vol. 36, no. 6, pp. 547–551, 2018.
- [45] D. Escudero, S. Ghosh, M. Keller, R. Rachuri, and P. Scholl, "Improved primitives for MPC over mixed arithmetic-binary circuits," in *Annual International Cryptology Conference*, 2020, pp. 823–852.

- [46] S. Tan, B. Knott, Y. Tian, and D. J. Wu, "CRYPTGPU: Fast Privacy-Preserving Machine Learning on the GPU," *arXiv preprint arXiv:2104.10949*, 2021.
- [47] N. K. Jha, Z. Ghodsi, S. Garg, and B. Reagen, "DeepReDuce: Relu reduction for fast private inference," *arXiv preprint arXiv:2103.01396*, 2021.
- [48] I. Helbitz and S. Avidan, "Reducing ReLU Count for Privacy-Preserving CNN Speedup," *arXiv preprint arXiv:2101.11835*, 2021.
- [49] S. De and S. Smith, "Batch Normalization Biases Residual Blocks Towards the Identity Function in Deep Networks," *Advances in Neural Information Processing Systems*, vol. 33, 2020.
- [50] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.
- [51] C. Xie, O. Koyejo, and I. Gupta, "Generalized byzantine-tolerant sgd," *arXiv preprint arXiv:1802.10116*, 2018.
- [52] J. B. Nielsen, P. S. Nordholt, C. Orlandi, and S. S. Burra, "A new approach to practical active-secure two-party computation," in *Annual Cryptology Conference*, 2012, pp. 681–700.
- [53] A. Hegde, H. Möllering, T. Schneider, and H. Yalame, "SoK: Efficient Privacy-preserving Clustering." PETS, 2021.
- [54] A. Raghu, M. Komorowski, I. Ahmed, L. Celi, P. Szolovits, and M. Ghassemi, "Deep reinforcement learning for sepsis treatment," *arXiv preprint arXiv:1711.09602*, 2017.
- [55] R. Viñas, T. Azevedo, E. R. Gamazon, and P. Liò, "Gene expression imputation with generative adversarial imputation nets," *bioRxiv*, 2020.
- [56] E. Piacentino and C. Angulo, "Generating fake data using gans for anonymizing healthcare data," in *International Work-Conference on Bioinformatics and Biomedical Engineering*, 2020, pp. 406–417.
- [57] D. Froelicher, J. R. Troncoso-Pastoriza, J. L. Raisaro, M. Cuendet, J. S. Sousa, J. Fellay, and J.-P. Hubaux, "Truly Privacy-Preserving Federated Analytics for Precision Medicine with Multiparty Homomorphic Encryption," *bioRxiv*, 2021.



# Computer Code Representation through Natural Language Processing for fMRI Data Analysis

Jaeyoon Kim  
Department of Electrical and  
Computer Engineering  
Korea University  
Seoul, South Korea  
jyoonkim@korea.ac.kr

Una-May O'Reilly  
Computer Science and Artificial  
Intelligence Laboratory  
Massachusetts Institute of Technology  
Cambridge, United States  
unamay@csail.mit.edu

Junhee Seok  
Department of Electrical and  
Computer Engineering  
Korea University  
Seoul, South Korea  
jseok14@korea.ac.kr

**Abstract**— There are many attempts to analyze the relationship between functional magnetic resonance imaging (fMRI) data and text stimuli representation in cognitive neuroscience research. Because programming codes are exemplary text stimuli, appropriate code representation for neuroscience research has been actively studied. In this paper, we focus on representing python code for fMRI research through natural language processing (NLP) techniques. We collect 7,893 python codes of 23 question types from a code competition website and build three different models based on sequence-to-sequence, bag-of-words, and bigram representation. The model is evaluated to classify the types of questions. Finally, the model is applied to classify 108 python codes which were used for a cognitive neuroscience study of fMRI. We are looking forward to analyzing fMRI data with the proposed code representation for understanding how the human brain is active.

**Keywords**— natural language processing, computer code representation, sequence-to-sequence, bag-of-words, bigram, Functional magnetic resonance imaging, cognitive neuroscience.

## I. INTRODUCTION

A large amount of data and high-performance hardware have accelerated the development of deep learning in various research areas. Image and text data are mainly used in deep learning research. For instance, image generation [1], super-resolution [2], missing data imputation [3], and object detection [4] research have been actively studied. In the case of text data analysis, several NLP models are used to do keyword extraction [5], sentimental analysis [6], document classification, and summarization [7-8].

Recently, deep learning has also been used in cognitive neuroscience to find out the relationship between fMRI data and stimuli representation, called brain decoding. fMRI can measure different brain activity levels in different brain regions while people perform diverse cognitive tasks [9-12]. We can predict that the highly activated brain region might differ when solving logical math problems and reading sentences. Some kinds of research suggest methods to match fMRI activities with the meanings of stimuli. Stimuli can be diverse. It can be pictures, videos, natural language sentences, or computer codes. Vodrahalli et al. (2018) collected fMRI data from subjects while watching an episode of the BBCs

Sherlock and mappings between fMRI brain activities and natural language representation [13]. Floyd, Santander, & Weimer (2017) examined code comprehension. They found out that the representation of the program and natural language are different [14]. Liu et al. (2020) looked more deeply at program structure and logic presentation [15].

In this paper, we are interested in representing python codes that can help analyze how the human brain is active while they read python codes. Furthermore, this research would be helpful to do brain decoding by investigating the relationship between fMRI data and code representation. We are inspired by the research done by Ivanova et al., 2020 and used the same 108 python code stimuli, which can be obtained from <https://github.com/ALFA-group/neural-program-comprehension> [16].

Our purpose is to train NLP models to represent 108 python codes. We have simple python codes, which have two types of problem (math and string manipulation), and three types of problem structure (sequential statements, for loops, and if statements). However, 108 data are not enough to train models. We downloaded 23 question types of 8305 python codes from the code competition website 'CodeChef' through web crawling to handle this problem. With this data, we trained three models. Seq2Seq, BOW, and Bigram model. We measured the performance of the models through the classification problem, which predicts the question types. Seq2Seq test accuracy is 0.64, BOW is 0.68, and Bigram is 0.71. Considering there are 23 question types, the accuracy score is reasonable. After that, we infer those models with 108 python codes to get representations. At this point, we trained logistic regression (LR), support vector classifier (SVC), and random forest (RF) to classify the type of the code (Math vs. String and Seq vs. For vs. If) using representation which came from Seq2Seq, BOW, and Bigram. In the case of math and string classification, Seq2Seq has the highest accuracy score in LR, SVC, and RF. The accuracy was 0.88, 0.89, and 0.84, respectively. For problem structure classification, Seq2Seq has the highest accuracy in LR and SVC. The accuracy was 0.58 and 0.61, respectively. Bigram shows the highest accuracy, 0.57 in RF.

## II. METHODS

### A. Data collection through web crawling

108 python codes are simple, so we collected the most uncomplicated codes on the Codechef website. However, it was still complicated than the 108 stimuli. Table 1 and Table 2 show the statistics of the data. On average, CodeChef data is twice as long as 108 python codes. The number of tokens, loops, and operators is also almost double. Therefore, we excluded python codes that have more than 25 lines. As a result, we only used 95.02% of the data (7893 codes).

TABLE I. STATISTICS OF 108 PYTHON CODES

Data ( Python 3.6)	Total # : 108
# line	Mean : 5.42 Std : 1.79
# token	Mean : 23.08 Std: 9.18
#loop(for,while)	Mean : 0.33 Std : 0.47
# operator	Mean : 5.37 Std : 2.29

TABLE II. STATISTICS OF CODECHEF DATA

Data ( Python 3.6)	Total # : 8305
# line	Mean : 12.00 Std : 7.08
# token	Mean : 58.85 Std: 32.50
#loop(for,while)	Mean : 1.63 Std : 1.14
# operator	Mean : 12.03 Std : 8.33

### B. Model training and inference

Figure 1 is a diagram that shows the process of model training and inference. First, we did web crawling to collect the data from the CodeChef, tokenize python codes and make a token dictionary for each model. Then, after training the models, we also tokenized the original dataset (108 python codes) and use the model to get representations. As a result, we could obtain 108 representations as a vector.

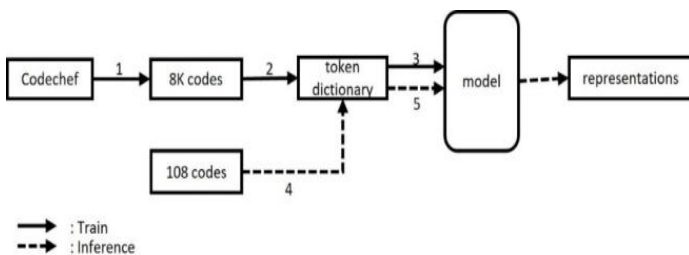


FIGURE I. A DIAGRAM OF MODEL TRAINING AND INFERENCE PROCESS

When we do text analysis, the first thing we need to do is tokenize. We used the python library to tokenize codes. After tokenizing, we replace every numeric value with 'NUM,' variable with 'VAR,' and string with 'String.' This is because we thought that specific numeric value, variable name, and string do not significantly affect extracting code features. Figure 2 is an example of how the Python code is tokenized. The example code's problem type is 'string,' and the problem structure type is an 'if' statement.

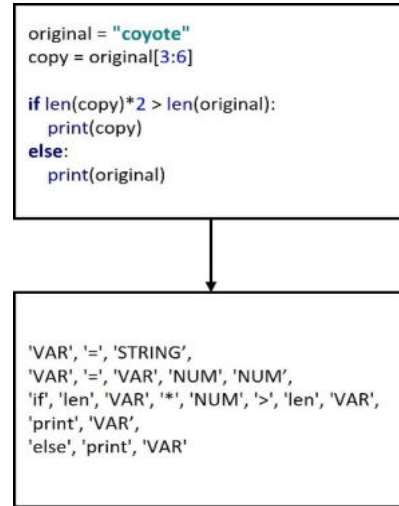


FIGURE II. TOKENIZING PYTHON CODE

After tokenizing, we made a token dictionary for each model. We removed some tokens which frequency is belonged to the lower 5%. In addition, unknown token 'UNK' was added to the token dictionary. The unknown token is used when we cannot match tokens in the token dictionary during the model inference. For the Seq2Seq model, we added '<eos>' and '<eos>' tokens. '<eos>' token is a starting token, and '<eos>' token is an end token.

There are three modules in our Seq2Seq model. Encoder, Attention, and Decoder. The input is word embedding vectors. Before getting word embedding vectors, we matched the sequence length in the batch. Every python code has a different number of tokens. Thus, we need to set the sequence length into the maximum tokens for each program. The codes with fewer tokens than the sequence length would be filled with padding tokens as they are insufficient. As a result, in the Seq2Seq model, 'UNK,' '<eos>,' '<eos>,' and padding tokens were added.

We used bi-directional gated recurrent units (GRUs) in the encoder and decoder module. GRUs is a gating mechanism in recurrent neural networks (RNNs) and have fewer parameters than RNNs due to lack of output gate. The purpose of the Attention module is masking. The masking matrix is filled with zero if a token is a padding token and filled with one otherwise. The multiplication between the encoder's output and the attention mask is the program representation that we want.

Bag-of-word (BOW) and Bigram are simple models. For the BOW model, it used the frequency of each token. The Bigram model is almost similar to BOW, but it uses a sequence of two adjacent elements.

In this section, we showed how do we train the models. Next section, we explain how we assessed the models' performance and how well our code representation works for classification problems.

### III. EXPERIMENTS AND RESULTS

To evaluate our three models, we made a simple feed-forward neural network to classify 23 question types. Table 3 shows the accuracy of the classification problem.

TABLE III. THE ACCURACY OF 23 QUESTION TYPES CLASSIFICATION

Model	Dataset	# Data	Accuracy
BOW	Train	60%	0.72
	Validation	20%	0.68
	Test	20%	<b>0.68</b>
Bigram	Train	60%	0.79
	Validation	20%	0.70
	Test	20%	<b>0.71</b>
Seq2Seq	Train	60%	0.70
	Validation	20%	0.63
	Test	20%	<b>0.64</b>

We split the dataset into a train, validation, and test set (60%/20%/20%). The BOW accuracy is 0.68, Bigram is 0.71, and Seq2Seq is 0.64. We suspected that the Seq2Seq model has the lowest test accuracy because we do not have enough data to train the complicated deep learning model. However, considering there are 23 question types, the accuracy score is reasonable.

By using those three models, we get 108 python code representations. To estimate code representations, we classify problem type and problem structure type. There are two types of problem (math and string) and three types of problem structure (sequence, for, and if). We reduced the representation dimension using principal component analysis (accounted for about 90%) and then trained LR, SVC, and RF for each classification task. Table 4 shows the accuracy of math and string classification. As we can see, the Seq2Seq representation performs the best. Its accuracy score is 0.88, 0.89, and 0.84 in LR, SVC, and RF. Bigram accuracy is also 0.84, which is the same as Seq2Seq. Table 5 shows the accuracy of sequential statements, for loops, and if statements classification. In this case, Seq2Seq has the highest accuracy in both LR and SVC. The accuracy is 0.58 and 0.61, respectively. However, Bigram works the best in RF.

TABLE IV. THE ACCURACY OF MATH AND STRING CLASSIFICATION

	LR	SVC	RF
BOW	0.80	0.80	0.82
Bigram	0.79	0.82	<b>0.84</b>
Seq2Seq	<b>0.88</b>	<b>0.89</b>	<b>0.84</b>

TABLE V. THE ACCURACY OF SEQUENTIAL, FOR AND IF CLASSIFICATION

	LR	SVC	RF
BOW	0.50	0.53	0.55
Bigram	0.54	0.58	<b>0.57</b>
Seq2Seq	<b>0.58</b>	<b>0.61</b>	0.55

The random chance for the two-class classification is 0.5, and the three-class is 0.33. Compared to this, our highest accuracy is 0.89 and 0.61. It means our representation works well with the test dataset (108 python codes).

### IV. CONCLUSION

In this paper, we focused on representing python codes. We trained three NLP models. Seq2Seq, BOW, and bigram. We evaluate models' classification performance, and the highest accuracy is 0.71. Considering random chance for 23 class classification is 0.04, 0.71 is a relatively high score. In addition, predicting problem and structure type works well with our model's representation. Even we did not use the original test dataset (108 python codes) while training models, the models could extract features well with the test dataset. Thus, we are convinced that our research would contribute to cognitive neuroscience studies, such as analyzing Functional magnetic resonance imaging (fMRI) data with code representation to understand how the human brain is active.

### ACKNOWLEDGEMENT

This research was supported by the MOTIE (Ministry of Trade, Industry, and Energy) in Korea, under the Fostering Global Talents for Innovative Growth Program (P0008749) supervised by the Korea Institute for Advancement of Technology (KIAT) and National Research Foundation of Korea (NRF-2019R1A2C1084778).

### REFERENCES

- [1] I. Goodfellow et al. "Generative adversarial networks", *Communications of the ACM*, vol. 63, no. 11, pp. 139-144, Nov. 2020
- [2] C. Dong, CC. Loy, K. He and X Tang, "Image Super-Resolution Using Deep Convolutional Networks", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295-307, Feb. 2016
- [3] J. Kim, D. Tae and J. Seok, "A Survey of Missing Data Imputation Using Generative Adversarial Networks", *IEEE International Conference on Artificial Intelligence in Information and Communication (ICAIIIC)*, Fukuoka, Japan, Feb. 2020

- [4] ZQ. Zhao, P. Zheng, S. Xu and X. Wu, "Object Detection With Deep Learning: A Review", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, Nov. 2019
- [5] S. Kim, S. Choi and J. Seok, "Keyword Extraction in Economics Literatures using Natural Language Processing", *Twelfth International Conference on Ubiquitous and Future Networks (ICUFN)*, Jeju Island, Korea, Aug. 2021
- [6] J. Kim, J. Seo, M. Lee and J. Seok, "Stock Price Prediction Through the Sentimental Analysis of News Articles", *Eleventh International Conference on Ubiquitous and Future Networks (ICUFN)*, Zagreb, Croatia, July. 2019
- [7] M. Afzal et al. "Deepdocclassifier: Document classification with deep Convolutional Neural Network", *International Conference on Document Analysis and Recognition (ICDAR)*, Tunis, Tunisia, Aug. 2015
- [8] M. Yousefi-Azar and Len-Hamey, "Text summarization using unsupervised deep learning", *Expert Systems with Applications*, vol. 68, pp. 99-105, Feb. 2017
- [9] R. Poldrack, "The role of fMRI in Cognitive Neuroscience: where do I stand?", *Current Opinion in Neurobiology*, vol.18, no. 2, pp. 223-227, April. 2008
- [10] R.Henson, "Forward inference using functional neuroimaging: dissociations versus associations", *Trends in Cognitive Sciences*, vol. 10, no. 2, pp. 64-69, Feb. 2006
- [11] T.Mitchell, R. Hutchinson, M. Just, R. Niculescu, F. Pereira & X. Wang, "Classifying Instantaneous Cognitive States from fMRI Data", *AMIA Annual Symposium Proceedings Archive*, pp. 465-469, 2003
- [12] J. Detre and T. Floyd, "Functional MRI and Its Applications to the Clinical Neurosciences", Feb. 2001
- [13] K. Vodrahalli et al. , "Mapping between fMRI responses to movies and their natural language annotations", *NeuroImage*, vol. 180, pp. 223-231, Oct. 2018
- [14] B. Floyd, T. Santander & W. Weimer, "Decoding the Representation of Code in the Brain: An fMRI Study of Code Review and Expertise", *IEEE/ACM 39th International Conference on Software Engineering (ICSE)*, Buenos Aires, Argentina, July. 2017
- [15] Y. Liu, J. Kim, C. Wilson & M. Bedny, "Computer code comprehension shares neural resources with formal logical inference in the frontoparietal network", *eLife*, Dec. 2020
- [16] A. Ivanova et al. "Comprehension of computer code relies primarily on domain-general executive brain regions", *eLife*, Dec. 2020

# A Machine Learning Approach in Evaluating Symptom Screening in Predicting COVID-19

John Althom A. Mendoza<sup>1</sup>, Geoffrey A. Solano<sup>2</sup>, Marc Jermaine Pontiveros<sup>1,2</sup>, Jaime DL Caro<sup>1</sup>, Peter Martin D. Gomez<sup>4</sup>, Conner G. Manuel<sup>4,5</sup>, Paulyn Jean Buenaflor Rosell-Ubial<sup>5</sup>, Michael Tee<sup>3,5</sup>

<sup>1</sup>Department of Computer Science, College of Engineering  
University of the Philippines Diliman, Philippines

<sup>2</sup>Department of Physical Sciences and Mathematics, College of Arts and Sciences  
University of the Philippines Manila, Philippines

<sup>3</sup>Department of Physiology, College of Medicine  
University of the Philippines Manila, Philippines

<sup>4</sup>Dashlabs.ai, Manila, Philippines

<sup>5</sup>Philippine Red Cross, Manila, Philippines

**Abstract**—COVID-19 is a disease caused by the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) that, to date, has over 245 million confirmed cases and claimed almost 5 million lives. This disease attacks the respiratory system and comes with a number of symptoms. The US Center for Disease Control and Prevention presents a set of symptoms. However, these symptoms only begin to manifest after a number of days, which prevents early detection of this disease. This absence of symptoms during the early stages is what is considered by many to be the very factor that caused the virus into becoming a pandemic. Nonetheless, symptoms checking has been used in practice by commercial and business establishments as an initial screening for COVID-19. The bothersome process of symptom checking are still in place at the entrances of malls and airports. In this study, we determine whether or not symptom screening is an effective system to be employed to assess individuals for COVID-19. Specifically, it aims to determine whether or not one or a set of symptoms are effective predictors of the RT-PCR test results, the gold standard in Covid-19 testing, using machine learning. Using data from the Philippine Red Cross, classification models are developed using LightGBM, AdaBoost, Gaussian Naïve-Bayes, MultiLayer Perceptron, Quadratic Discriminant Analysis and Decision Tree. These models were evaluated using the following metrics: precision, sensitivity, specificity and the type II error rate. Furthermore, for explainability, symptoms are analyzed as to whether or not they are relatively influential on the predicting whether or not a patient has COVID-19. The high type II error rate, low sensitivity and low relative predictor scores of the most significant predictor symptoms clearly show that symptoms do not correlate with the RT-PCR testing results. Thus, we conclude that symptom screening is not a medically suitable process for determining whether an individual has COVID-19. In fact, it even exposes us to the risk of viral transmission as people congregate at the entrances and lobbies of establishments.

**Index Terms**—symptom screening, machine learning, COVID-19

## I. INTRODUCTION

It was in December 2019 when the coronavirus disease 2019 (COVID-19), a condition caused by the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), was first identified in the capital of China's Hubei province of Wuhan. In just a few weeks it has swept across the globe. On the

30th of January, 2020, the World Health Organization (WHO) declared the outbreak to be a Public Health Emergency of International Concern. By March 11, 2020, it was declared by WHO as a pandemic [1], [2]. As of the 29th of October, 2021, there are over 245 million confirmed cases and almost 5 million deaths worldwide spanning 222 countries and territories [3], with the Philippines having a total of 2.7 million confirmed cases and over 42 thousand deaths [4].

Various testing types were used to detect COVID-19, the most reliable of which, is the reverse transcriptase-polymerase reaction (RT-PCR) nasopharyngeal (NP) swab testing. It works by identifying viral RNA and is currently the gold standard in detecting whether an individual has COVID-19. One major challenge however is the cost of the procedure and the accessibility of testing centers, especially in less-developed countries. Furthermore, this type of testing has the biggest drawback of having the results available at least a day after the samples were collected. Thus, several faster tests are being administered despite having low confidence on the results. Such tests include lung function testing (LFT) [5], saliva testing [6], and blood testing [7].

Since this disease attacks the respiratory system and comes with a number of symptoms, many have resorted to symptom checking has been used as an initial screening for COVID-19. The US Center for Disease Control and Prevention presents the following, among others, as symptoms which may appear 2-14 days after exposure to the virus, ranging from mild to severe cases [8]:

- Fever or chills
- Cough
- Shortness of breath or difficulty breathing
- Fatigue
- Muscle or body aches
- Headache
- New loss of taste or smell
- Sore throat
- Congestion or runny nose

- Nausea or vomiting
- Diarrhea

All over the globe, airlines have required Health Declaration Forms (HDFs) to be filled out by those who will be taking flights. This is essentially declaring whether or not one is manifesting any of the above-mentioned symptoms. Companies have also employed the use of online symptom self-checking systems for their employees. Commercial and business establishments have required individuals to fill out an online health declaration form prior to entering the building premises. The bothersome process of symptom checking are still in place at the entrances of malls and airports. These have resulted in queues at the building entrance. Aside from the hassle and delays, this brings the risk of exposure to the virus due to human aggregation. Furthermore, these symptoms only begin to manifest 2-14 days after exposure, which prevents early detection of this disease [2], [9], [10]. In fact, this absence of symptoms during the early stages is what is considered by many to be a major factor that caused the virus into becoming a pandemic [7].

All these bring about the question of whether symptoms checking is indeed an effective means for detecting COVID-19. This question is what is investigated in this paper. This study aims to determine whether or not one or a set of symptoms are effective predictors of the RT-PCR test results using machine learning. Using data from the Philippine Red Cross, classification models are developed and symptoms will be analyzed as to whether or not they are relatively influential on the predicting whether a patient has COVID-19. The paper is organized as follows: On the next section we explore related work. This is followed by the Methodology where we present the series of steps were taken in order to accomplish the objectives of this study. In Section IV and V, we present the results of this study and the discussion of these results, respectively. Finally, the conclusion and suggestions for future work are in Section VI.

## II. RELATED WORK

Many supervised learning and feature extraction approaches have been used focusing on COVID-19 with different objectives. In a review by Alyasseri, et al., some of the objectives are: (1) to determine how the COVID-19 pandemic will end, (2) to predict how the coronavirus gets transmitted over regions, (3) to correlate the effect of weather conditions on coronavirus and (4) to diagnose COVID-19 based on symptoms and various X-ray and CT scan images [11].

Among the reviewed approaches in [11], one study conducted by Fayyoubi et. al have similar focus on diagnosing COVID-19 status based on signs and symptoms using 64 positive and 41 negative PCR tests of patients in Jordan, and reported an accuracy of 91.67% using Multilayer Perceptron (MLP). [12]. In their work, several attributes have been used to build the statistical models, namely age, smoker status, positive x-ray chest, fever, sore throat, aches and pain, dry cough, nasal congestion, absence of smell, diarrhea, vomiting, and breathing difficulty. The data was collected by answering questionnaires.

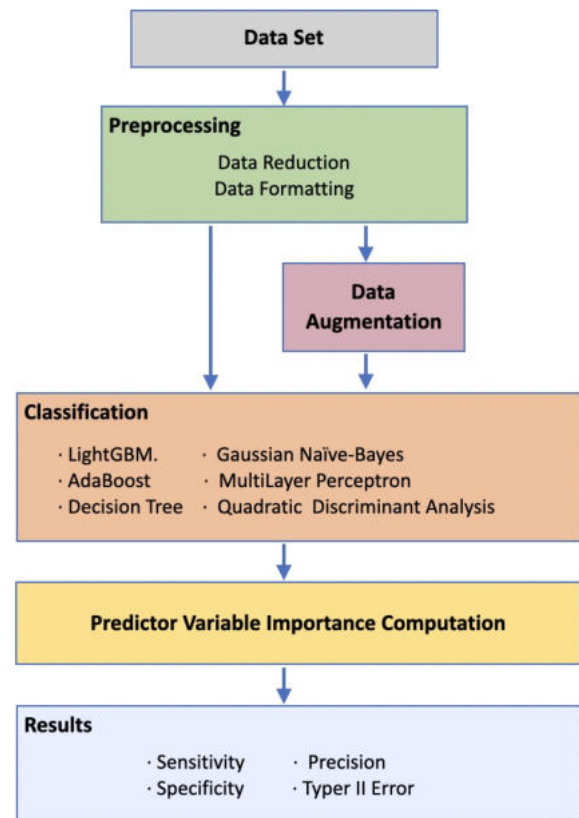


Fig. 1. The general workflow of the study

Among the attributes, only age were considered numeric and the rest are binary categorical variable. The authors mentioned that applying the technique in much larger dataset must be done in future studies.

Another related work by Zoabi et. al utilized a total of 99,232 COVID-19 test results, with a focus of diagnosing COVID-19 based on symptoms [13]. Their model utilizes the following attributes: sex, binarized age (greater than or equal 60 years), known contact with an infected individual, and appearance of five clinical symptoms (cough, fever, sore throat, shortness of breath, and headache). Their data is based on published data by Israel Ministry of Health of individuals who were tested for SARS-CoV-2 via RT-PCR assay of a nasopharyngeal swab. It contains initial records of all the residents who were tested for COVID-19 nationwide on daily basis. The study mentioned some shortcomings, and one would be missing information in some of the features, another would be the lack of records in reported symptoms by the Ministry of Health, such as the loss of smell and loss of taste.

Both studies are affirming the use of signs and symptoms in prioritizing testing and triaging for COVID-19, especially when the resources are limited.

## III. METHODOLOGY

A series of steps were taken in order to accomplish the objectives of this study. These steps are illustrated in a

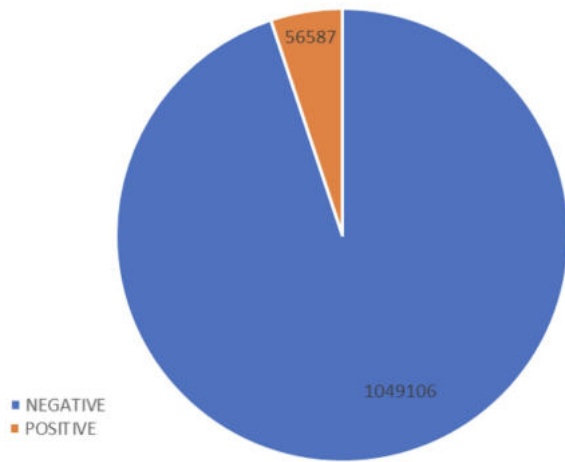


Fig. 2. The ratio of the positive to the negative cases prior to augmentation

flowchart in Figure 1.

### A. The Data Set

The data used in the study was collected by the Philippine National Red Cross between June 2020 – January 2021 totaling 1,434,868 records.

### B. Preprocessing

1) *Reduction*: This included observations which were still “in-process”, and thus, whose positivity to COVID-19 were not yet identified. These observations were therefore removed and those that remained totaled to 1,105,693 records.

2) *Formatting*: The dataset was originally in text with each row representing an observation. Part of each row is the positivity value, along with a list of symptoms separated by commas. These were all processed by transforming positivity along with the most frequent symptoms into columns or features with boolean values. Positivity became the target variable with boolean values indicating whether the patient was positive or otherwise. Similarly each of the most frequent symptoms all became features (column) whose boolean values indicated whether that particular patient (row) had that symptom. All the other symptoms that were present in less than ten observations were fused into one feature/column “others”, and so if at least one of these were present in an observation, that the value of the column for the said observation is 1 or “true”. The summary of frequencies of these features are seen in Table I. Clearly, the fused symptom “others” was the most frequent one, being present in 21,827 cases, whereas cough was present in 17,421 cases. Only 18 cases experienced appetite loss. On the other hand in Table II we see the symptom count on the observations. Around 1,046,932 cases were asymptomatic. There were 46,411 cases who had exactly one symptom and there was one case that had 10 symptoms.

TABLE I  
SUMMARY OF FEATURES (SYMPTOMS) AND THEIR RESPECTIVE FREQUENCIES.

Symptom	Count
others	21,827
cough	17,421
colds	15,789
sore throat	6,989
fever	6,741
difficulty breathing	3,651
headache	1,163
smell loss	1,073
taste loss	1,066
body malaise	1,066
diarrhea	644
body pain	539
anosmia	367
sense loss	271
ageusia	172
agnosia	29
apetite loss	18

TABLE II  
SUMMARY OF SYMPTOM COUNTS.

Asymptomatic	1,046,932
1	46,411
2	7,884
3	2,514
4	1,057
5	627
6	174
7	63
8	23
9	7
10	1

### C. Data Augmentation

It was noticed that with respect to the target variable, the data set has a class imbalance, with the ratio of those positive to those negative being approximately 1:18.54. This disparity can be seen in Figure 2. Thus, for this step, each of the positive observations was copied 18 times in order for the data set to be balanced. Thus, after augmentation, there were 1,018,566 positive cases.

Furthermore, there are therefore two sets of data used in the succeeding sections of this study : one with data augmentation and one without data augmentation.

TABLE III

PERFORMANCE OF CLASSIFICATION MODELS (WITHOUT AUGMENTATION)

Classifier	Precision	Type II Error	Sensitivity	Specificity
LightGBM	0.5862	0.4138	0.4215	0.6949
AdaBoost	0.5417	0.4583	0.3223	0.7203
GaussianNaiveBayes	0.5000	0.5000	0.9339	0.0424
MultiLayerPerceptron	0.4861	0.5139	0.2893	0.6864
QuadraticDiscriminant	0.5063	0.4937	1.0000	0.0000
DecisionTree	0.1818	0.8182	0.1322	0.3898

#### D. Development of Classification Models

There were six machine learning models developed for both working data sets. These are LightGBM, AdaBoost, Gaussian Naïve-Bayes, MultiLayer Perceptron, Quadratic Discriminant Analysis and Decision Tree.

#### E. Computation of Predictor Variable Importance Scores

One important task in interpreting classification models is understanding which predictor variables are relatively influential on the predicted outcome. Thus, aside from measuring evaluation metrics, variable importance was determined for both data sets. This variable importance measure was computed via permutation which included the following steps:

```

For any given loss function do
1: compute loss function for full model (denote _full_model_)
2: randomize response variable, apply given ML, and compute loss function
3: for variable j
   | randomize values
   | apply given ML model
   | compute & record loss function
end

```

This model agnostic variable importance measure computed via permutation [14] is essential in the explainability of the classification models developed.

#### F. Evaluation of Classification Models

For both the augmented and non-augmented data sets, the six models were evaluated using the following metrics : precision (positive predictive value), sensitivity (true positive rate or probability of detection), specificity (true negative rate) and the type II error rate (false negative rate or error of omission). Type II error rate is also chosen as the main metric since this is a health science study.

### IV. RESULTS

In Figure 3 we see the importance of the variables and how they fare with each other with respect to determining the target variable for the data set where augmentation was not applied. *Smell loss* outperforms the rest at 3.7%, followed by *fever* (1.4%) and *colds* (1.1%). Also, the features *body malaise*, *ageusia* and *others* do not seem to be contributing as good predictors of COVID-19 positivity. In Table III we see the comparison of the performance of the models on

TABLE IV

PERFORMANCE OF CLASSIFICATION MODELS (WITH AUGMENTATION)

Classifier	Precision	Type II Error	Sensitivity	Specificity
LightGBM	0.7521	0.2479	0.0921	0.9697
AdaBoost	0.7521	0.2479	0.0921	0.9697
GaussianNaiveBayes	0.7521	0.2479	0.0921	0.9697
MultiLayerPerceptron	0.7572	0.2428	0.0902	0.9711
QuadraticDiscriminant	0.7521	0.2479	0.0921	0.9697
DecisionTree	0.7536	0.2464	0.0921	0.9699

the non-augmented data set based on the metrics that were used in this study. LightGBM performed best with respect to Precision and type II error. The quadratic discriminant provides 100% sensitivity while Gaussian Naive Bayes provides 93.4%. However, their specificity results are 0% and 4.24%, respectively. Adabost, on the other hand, provides the highest specificity at 72.03%. However, all the six models for the non-augmented data have a very high type II error at  $\geq 41.38\%$ .

On the other hand, for the augmented data set, we can see the variable importance in Figure 4. Clearly, all of the variables performed better with *smell loss* still leading at 13.7%, followed by *fever* (13%) and *colds* (12.4%). *Sense loss*, *agnosia* and *others* appeared to be the least important variables. In Table IV we see the comparison of the performance of the models on the augmented data set based on the metrics that mentioned. Multilayer perceptrons performed better than the rest with respect to precision, type II error and specificity at 75.72%, 24.28% and 97.11% respectively. However, it dipped a bit compared to the other models with sensitivity. The Decision Tree model, however, was just second to Multilayer perceptrons with respect to precision, type II error and specificity at at 75.36%, 24.64% and 96.99% respectively, but it performed slightly better with sensitivity.

### V. DISCUSSION

It can be observed that augmenting the data generally helps in achieving better models. One possible reason is that training biases over the uneven distribution of the observations are solved. In non-augmented data, none of the symptoms has an importance of more than 5%. However, after augmentation, 11 symptoms obtained variable importance which were higher than 5%, with *smell loss* leading with an improvement from 3.7% to 13.7%. Furthermore, all the models improved after data augmentation with three of the four metrics. However, the sensitivity scores in the augmented data are quite curious as all the models scored very low at sensitivity, with Multilayer perceptrons scoring only 9.02% and the rest of the models scoring 9.21%.

The most significant symptoms are *smell loss*, *fever* and *colds*. However, their scores as relative predictors for the target variable are still low at at 13.7%, 13% and 12.4%, respectively.

It can be observed based on the results that COVID-19 cannot be predicted accurately with just the symptoms. The



two better models are Multilayer perceptrons and Decision Trees. Multilayer perceptrons had precision, type II error and specificity at 75.72%, 24.28% and 97.11% respectively. However, the sensitivity score is at 9.02% only. Decision Trees on the other hand had precision, type II error and specificity scores of 75.36%, 24.64% and 96.99% respectively, but the sensitivity score is at 9.21% only. Furthermore, all the models still had a high Type II error, which is a big issue for health prediction classifier models.

Given the hightype II error, low sensitivity and low relative predictor scores of the most significant predictor symptoms, thus there are no symptoms, whether one or a set, that is an effective predictor of the RT-PCR test results based on the data set. This study finds that symptom screening is not an effective system to be employed to monitor and assess individuals for COVID-19.

### VI. CONCLUSION

In this study, data from the Philippine Red Cross was used to determine whether or not symptom screening is an effective system to be employed to assess individuals for COVID-19. Classification models were developed using LightGBM, AdaBoost, Gaussian Naïve-Bayes, MultiLayer Perceptron, Quadratic Discriminant Analysis and Decision Tree and were evaluated using the following metrics: precision, sensitivity, specificity and the type II error rate. Furthermore, for explainability, symptoms were analyzed as to whether or not they are relatively influential on the predicting whether or not a patient has COVID-19. Across all models, the high type II error rate ( $\geq 24.28\%$ ), low sensitivity ( $\leq 9.21\%$ ) and low relative predictor scores of the most significant predictor symptoms ( $\leq 13.7\%$ ) clearly there are no symptoms, whether

one or a group of symptoms, that is an effective predictor of the RT-PCR test results based on the Red Cross data set. Thus, we conclude that symptom screening is not a medically suitable process for determining whether an individual has COVID-19. In fact, it even exposes us to the risk of viral transmission as people congregate at the entrances and lobbies of establishments.

### ACKNOWLEDGEMENT

The authors would like to thank the Philippine Red Cross, particularly Senator Richard J. Gordon and Ms. Elizabeth Zavalla.

### REFERENCES

- [1] C. Wang, P. W. Horby, F. G. Hayden, and G. F. Gao, "A novel coronavirus outbreak of global health concern," *The Lancet*, vol. 395, pp. 470–473, 2020. [Online]. Available: [https://doi.org/10.1016/S0140-6736\(20\)30185-9](https://doi.org/10.1016/S0140-6736(20)30185-9)
- [2] S. Chauhan, "Comprehensive review of coronavirus disease 2019 (covid-19)," *Biomedical Journal*, vol. 43, no. 4, pp. 334–340, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2319417020300871>
- [3] "Who coronavirus (covid-19) dashboard." [Online]. Available: <https://covid19.who.int/>
- [4] "Covid-19 tracker: Department of health website." [Online]. Available: <https://doh.gov.ph/covid19tracker>
- [5] J. Hull, J. Lloyd, and B. Cooper, "Lung function testing in the covid-19 endemic," *The Lancet Respiratory Medicine*, vol. 8, 05 2020.
- [6] L. M. Czumbel, S. Kiss, N. Farkas, I. Mandel, A. Hegyi, A. Nagy, Z. Lohinai, Z. Szakacs, P. Hegyi, M. C. Steward, and G. Varga, "Saliva as a candidate for covid-19 diagnostic testing: A meta-analysis," *Frontiers in Medicine*, vol. 7, p. 465, 2020. [Online]. Available: <https://www.frontiersin.org/article/10.3389/fmed.2020.00465>
- [7] J. Giesecke, "The invisible pandemic," *The Lancet*, vol. 395, no. 10238, May 2020.
- [8] "Center for disease control and prevention: Symptoms of covid-19." [Online]. Available: <https://www.cdc.gov/coronavirus/2019-ncov/symptoms-testing/symptoms.html>

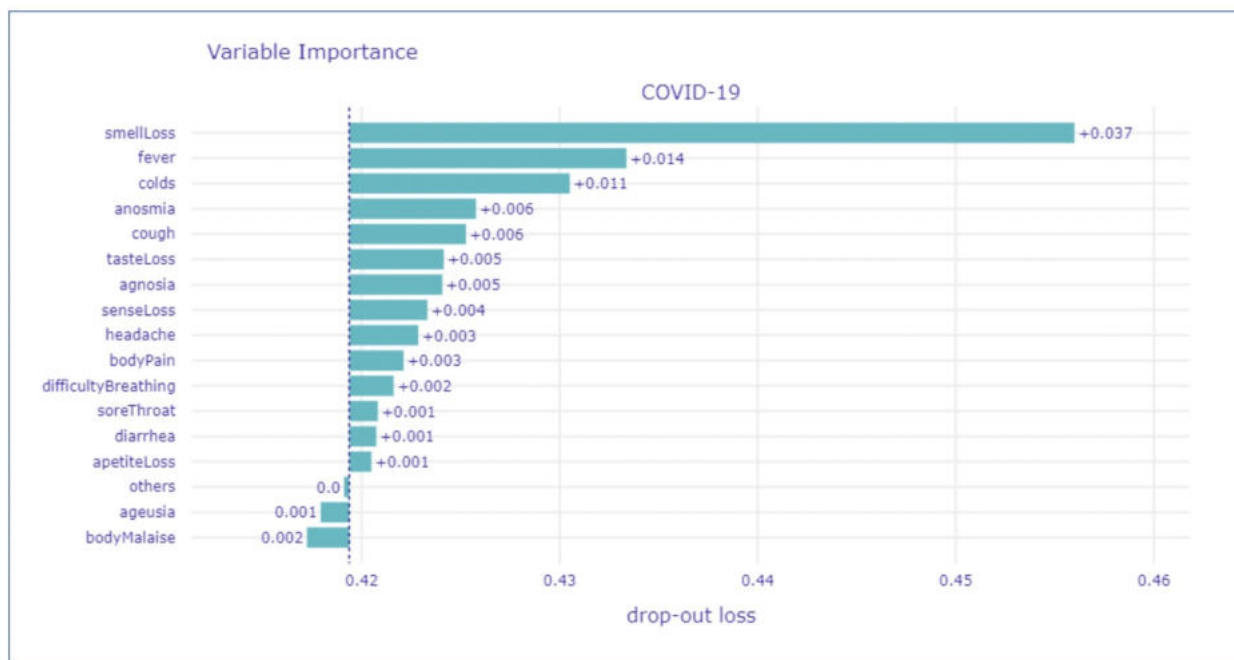


Fig. 3. Variable importance (without augmentation)

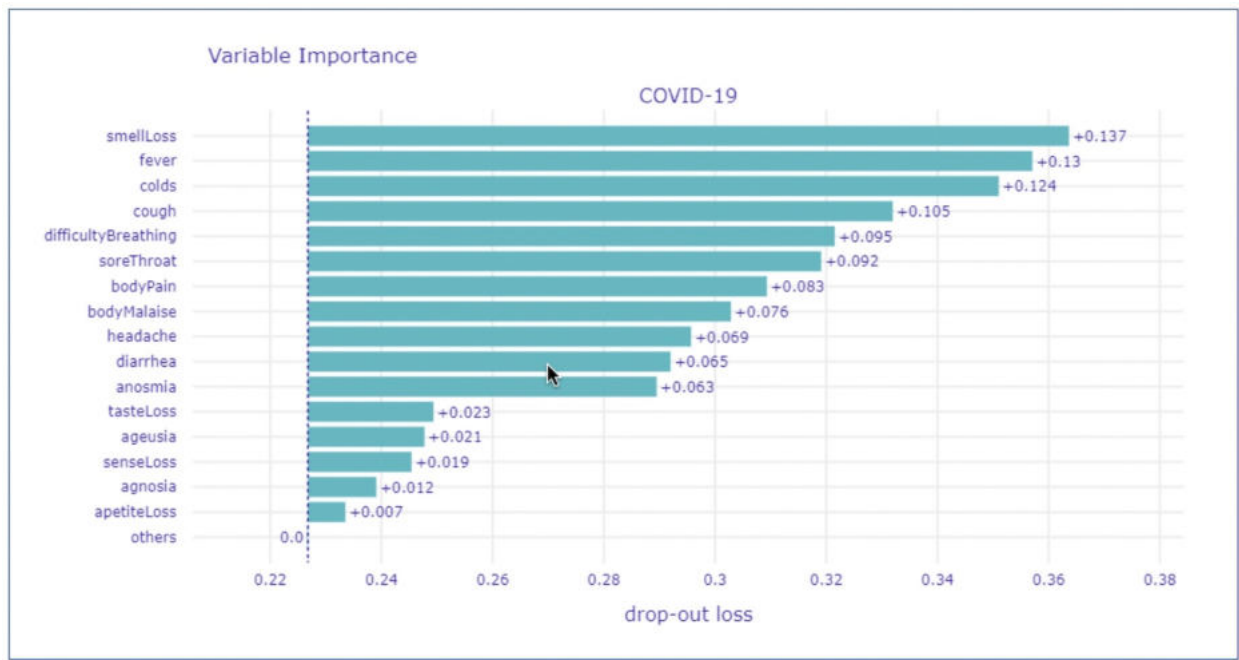


Fig. 4. Variable importance (with augmentation)

- [9] L. Wang, Y. Wang, D. Ye, and Q. Liu, "Review of the 2019 novel coronavirus (sars-cov-2) based on current evidence," *International Journal of Antimicrobial Agents*, vol. 55, no. 6, p. 105948, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0924857920300984>
- [10] R. Wang, M. Pan, X. Zhang, M. Han, X. Fan, F. Zhao, M. Miao, J. Xu, M. Guan, X. Deng, X. Chen, and L. Shen, "Epidemiological and clinical features of 125 hospitalized patients with covid-19 in fuyang, anhui, china," *International Journal of Infectious Diseases*, vol. 95, pp. 421–428, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1201971220302034>
- [11] Z. A. A. Alyasseri, M. A. Al-Betar, I. A. Doush, M. A. Awadallah, A. K. Abasi, S. N. Makhadmeh, O. A. Alomari, K. H. Abdulkareem, A. Adam, R. Damasevicius, M. A. Mohammed, and R. A. Zitar, "Review on COVID -19 diagnosis models based on machine learning and deep learning approaches," Jul. 2021. [Online]. Available: <https://doi.org/10.1111/exsy.12759>
- [12] E. Fayyumi, S. Idwan, and H. AboShindi, "Machine learning and statistical modelling for prediction of novel COVID-19 patients case study: Jordan," vol. 11, no. 5, 2020. [Online]. Available: <https://doi.org/10.14569/ijacsa.2020.0110518>
- [13] Y. Zoabi, S. Deri-Rozov, and N. Shomron, "Machine learning-based prediction of COVID-19 diagnosis based on symptoms," vol. 4, no. 1, Jan. 2021. [Online]. Available: <https://doi.org/10.1038/s41746-020-00372-6>
- [14] "Model interpretability with dalex." [Online]. Available: <https://uc-r.github.io/dalex?fbclid=IwAR2vtKGe6Eht5sDgkHHtiLkXVR88K3OgN0Adbk1cWDLhfGEYEOPGeEf02bovi>
- [15] T. Zitek, "The appropriate use of testing for covid-19," *Western Journal of Emergency Medicine*, vol. 21, 04 2020.
- [16] F. Zeng, L. Li, J. Zeng, Y. Deng, H. Huang, B. Chen, and G. Deng, "Can we predict the severity of coronavirus disease 2019 with a routine blood test?" *Polish archives of internal medicine*, vol. 130, no. 5, May 2020.
- [17] Y. Zoabi, S. Deri-Rozov, and N. Shomron, "Machine learning-based prediction of covid-19 diagnosis based on symptoms," *npj Digital Medicine*, vol. 4, 01 2021.
- [18] A. Callahan, E. Steinberg, J. Fries, S. Gombar, B. Patel, C. Corbin, and N. Shah, "Estimating the efficacy of symptom-based screening for covid-19," *npj Digital Medicine*, vol. 3, 12 2020.

# A Study on the Clinical Effectiveness of Deep Learning CAD Technology

Han Ju-Hyuck  
Department of Medical Engineering  
Konyang University  
Daejeon, South Korea  
dnfwlq203@gmail.com

Oh Hyun-Woo  
Bio Convergence Technology Training  
Project Group, Konyang University  
Daejeon, South Korea  
osj0805@naver.com

Kim Woong-Sik\*  
Department of Medical A.I.  
Konyang University  
Daejeon, South Korea  
wskim@konyang.ac.kr

**Abstract**—Chest radiography is the most common method of examining chest disease. However, interpretation of chest X-rays is difficult, and the diagnosis may vary depending on the doctor's proficiency. In order to solve this problem, additional diagnosis using a computer is attracting attention in the medical imaging field. In addition, the recently developed artificial intelligence technology has been applied to the analysis of chest X-rays, and commercialization has entered the stage as a computer-aided diagnostic tool. However, the reading model based on artificial intelligence has different performance depending on the type of data. In addition, current medical data is a weak standardization stage and the data form varies from institution to institution. Therefore, the performance of the model may not be guaranteed if the data for training artificial intelligence and the data from the real institution are different. The purpose of this study is to verify the clinical effectiveness of a computer-aided diagnostic tool based on chest X-rays. To this end, data from a different source than the training data were applied to the reading model. In addition, for validation, we prepared a doctor's lung lesion labeling findings for clinical validation. In this study, OPT (Observer Performance Test) was conducted by clinical experience level to evaluate the reading model.

**Keywords**—Chest Radiography, Deep learning Algorithm, Observer Performance Test, CAD

## I. INTRODUCTION

Chest radiography is the most common method for examining chest diseases and monitoring chest abnormalities such as lung cancer[1]. However, interpretation of chest X-Ray (CXR) is difficult and misreadable, requiring a lot of image analysis experience[2]. In other words, the reading of chest radiographs depends on the physician's clinical experience. Recently, in order to solve this problem, the development of computer aided diagnosis (CAD) is growing. In addition, advanced artificial intelligence technology is being applied to the medical imaging field. The purpose of this paper is to verify the clinical effectiveness of artificial intelligence models that support the interpretation of chest radiographs. In addition, three cohorts were organized and studied as methods for verification.

## II. METHODE

### A. CAD and Data Configuration

In this paper, Lunit INSIGHT for Chest Radiography certified by the Ministry of Food and Drug Safety in Korea, was used to clinically evaluate CAD performance based on deep learning. The performance evaluation was applied by dividing a cohort of patients who visited respiratory outpatient hospitals in three institutions in 2018 and underwent chest radiography. All data used in this study are retrospective data approved by the instrumental review boards, and the requirements for patient consent have been omitted. Data screening targeted 26,988 patients.

There are two criteria for not selecting data. First, the unexamined case of chest CT (n=17,871), secondly, the interval between chest CT and chest X-Ray is more than 1 month (n=3,165). Therefore, as shown in Table 1, the final data for performance comparison is 6,006 people's data. Data collection is based on CXR, and meaningful patient information (age, gender, chest CT, smoking history, past history) was collected from electronic medical records. In addition, in the case of images, they were collected by the picture archiving and communication system (PACS) and all of them were de-identified for research.

TABLE I. DEMOGRAPHIC DATA OF RESPIRATORY PATIENTS

Category	Institutions			Total	Dataset for OPT	P value
	B	G	K			
No. of Patients	2536	1470	2000	6006	230	-
Female	1166	643	798	2607	107	0.53
Male	1370	827	1202	3398	123	0.50
Age	61 ±16	61 ±14	61 ±16	61 ±16	60 ±16	0.21
Interval CXR & CT	3 ±9	3 ±11	1 ±7	2 ±9	2 ±9	0.42
No. of PA image	2536	1421	1952	5908	229	0.15

Table 1 compares the notations of radiologists and CAD at three institutions. CAD marks Activation Map at the location of the abnormal lesion. This was considered positive when there was a coincidence rate of more than 15% at the center of the lesion based on pixels. Previous studies showed that the AUC of Test1 was 0.814 and the AUC of Test2 was 0.904 [3]. It has a test set number of 230, and constitutes a dataset of positive 0.4, negative 0.6. In this study, 230 random sampling data were composed of OPT Data Set to apply the same criteria as in previous studies. In addition, CT images are used to compare with CXR. The CT image was selected based on the closest shooting date from the CXR shooting date. The criteria for chest abnormalities were selected by radiologists. They analyzed the dataset to define reference settings for chest abnormalities. In this study, when disagreement occurs in the diagnosis, an investigation was conducted to agree on it. The CXR lesion in this study consisted of nodules/species, integration, and pneumothorax as target lesions. In addition, lethargy or fibrosis, bronchiectasis, heart lesions, diffuse lung genes, longitudinal lesions, and pleural effusion were composed of comparative lesions for patient comparison. This classification referred to the labeling standard of the ChestX-ray14 dataset or MIMIC-CXR database[4][5][6]. This study was conducted with the approval of the Institutional Review Committee[7].

TABLE II. LESION TYPES OF CHEST RADIOGRAPHY IN PATIENTS

Lesion	Institutions			Total	Random sample for OPT	P value
	B	G	K			
<b>Target Lesions</b>						
Nodule/mass	446 (18)	259 (18)	468 (23)	1173 (20)	41 (18)	0.79
Consolidation	341 (13)	212 (14)	366 (18)	919 (15)	35 (15)	0.99
Pneumothorax	5 (0.3)	2 (0.1)	8 (0.4)	15 (0.2)	2 (0.9)	0.87
<b>Non-target Lesions</b>						
Atelectasis or fibrosis	93 (4)	62 (4)	185 (9)	340 (6)	15 (7)	0.90
Bronchiectasis	217 (9)	286 (20)	107 (5)	610 (10)	27 (12)	0.80
Cardiomegaly	21 (0.8)	48 (3)	67 (3)	136 (2)	4 (2)	0.94
Diffuse interstitial lung opacities	115 (5)	73 (5)	65 (3)	253 (4)	10 (4)	0.99
Mediastinal lesion	11 (0.4)	27 (2)	36 (2)	74 (1)	4 (2)	0.93
Pleural effusion	81 (3)	29 (2)	76 (4)	186 (3)	7 (3)	0.99
Other	188 (7)	172 (12)	198 (10)	558 (9)	28 (12)	0.61
<b>Total</b>						
Sum of target or non-target lesions	1518	1170	1576	4264	173	N/A
Participants with any types of lesions	1317 (52)	889 (61)	113 (57)	3337 (56)	137 (60)	0.36
No. of lesion type per patient	1.2 (1-3)	1.3 (1-4)	1.4 (1-5)	1.3 (1-5)	1.3 (1-4)	0.83

Table 2 shows the configuration of the data set used in this study. According to this, out of 4,274 reference chest abnormal lesions of 6,006 CXR, pulmonary nodules/tumor, aggregation, pneumothorax, and other reference chest abnormalities were found in 1,173 (20%), 919 (15%), 15 (0.2%), and 2,157 (51%) CXRs, respectively. Among the 26 classified final diagnoses, pneumonia was the most common diagnosis (n = 696, 12%). Pulmonary tuberculosis and malignant neoplasm (neoplasm) of the bronchial tubes or lungs were found in CXRs of 550 (9%) and 355 (6%), respectively.

### B. Verification Method

In this study, OPT was constructed to evaluate CAD. The OPT of this study was conducted by dividing the number of data collection days to prevent data bias. In OPT Test 1, observers conducted CXR evaluation alone without CAD help. It provided observers with CXR and CT, and patient information (gender, age, etc.). The observers consisted of 12 doctors, including 3 chest radiologists, 3 board-certified radiologists, 3 radiologists, and 3 lung specialists. They constructed the same form as the result of CAD by marking chest anomalies on the image. In OPT test 2, observers were assisted by CAD. This is based on the patient's CXR image and patient information, indicating chest abnormalities in the image. In addition, if CAD's Activation Map matches the observer's lesion indication, it was treated as true positive. On the other hand, if the CAD's Activation Map and the observer's lesion indication did not match, they were marked as false positive and false negative. In the case of false positive, CAD was marked positive, but there was no indication from the observer. In the case of false negative, the observer's notation exists, but there is no CAD notation. We confirmed the effect of CAD by doctors' clinical experience through OPT.

### C. CAD Model

In this study, Lunit Insight-CXR was used as CAD[8]. According to this, the model was used by extracting data sets for model training from six multinational multi-centers. In addition, a technique to mark chest abnormalities in CXR was used. The model consists of 27 layers and 12 residual connections based on convolutional neural networks (CNNs). This model uses a semisupervised localization approach with partial data annotation.

### III. RESULT

In this study, receiver operating characteristics (ROC) curves and jackknife free-response receiver operating characteristic (JAFROC) curves are used for outcome indicators. The receiver operating characteristic curve is calculated as a true-positive rate and a false-positive rate. In addition, the jackknife replacement free response ROC curve is calculated with the local fraction of the lesion to the probability of false positive (FP) per normal CXR. The number of false positive marks per image is defined as the value obtained by dividing the number of false positive marks by the total number of radiographs. Statistical analysis was performed using Medcalc version 19.5.1 or R version 3.5.3. All statistical analyses were performed using R software version 3.6.1.

TABLE III. PERFORMANCE OF OBSERVER GROUP IN THE RANDOMLY SAMPLED DATASET (N=230)

Observer Group		Test1	Test2	*P value	†P value
AUC	Thoracic radiologist(n=3)	0.89 (0.84,0.93)	0.89 (0.84,0.95)	0.21	0.58
	Board-certified radiologist(n=3)	0.87 (0.83,0.91)	0.89 (0.83,0.95)	0.14	0.12
	Radiology residents(n=3)	0.85 (0.80,0.89)	0.88 (0.85,0.91)	0.07	0.03
	Pulmonologist (n=3)	0.84 (0.80,0.85)	0.88 (0.85,0.92)	0.03	0.01
JAFROC	Thoracic radiologist(n=3)	0.82 (0.75,0.89)	0.84 (0.76,0.91)	0.03	0.60
	Board-certified radiologist(n=3)	0.80 (0.76,0.85)	0.82 (0.76,0.88)	0.29	0.12
	Radiology residents(n=3)	0.79 (0.72,0.85)	0.83 (0.79,0.87)	0.05	0.10
	Pulmonologist (n=3)	0.78 (0.73,0.83)	0.81 (0.77,0.75)	0.07	0.04

Table 3 shows the OPT results. This was constructed to confirm the effect of CAD according to the clinical experience of doctors. According to Table 3, Test 1 and Test 2 to view the influence of CAD, the average AUC was 0.86 (95% CI: 0.82, 0.90) to 0.89 (95% CI: 0.85, 0.92). the average JAFROC was 0.92 at 0.80 (95% CI: 0.76, 0.84). First, for Test 1 without CAD, a thoracic radiologist showed AUC: 0.87 and JAFROC: 0.80 for radiation resistors, AUC: 0.85 and JAFROC: 0.79, pulmonary tuberculosis specialist showed AUC: 0.89 and JAFROC: 0.82 for radiation residents. On the other hand, Test 2 assisted by CAD showed AUC: 0.89, JAFROC: 0.82 for chest radiologists, AUC: 0.88, JAFROC: 0.83 for radiation resistors, and AUC: 0.89 for pulmonary tuberculosis specialists showed AUC: 0.89 and JAFROC: 0.84. In OPT to verify the clinical effectiveness of CAD, Test 2 showed better performance than Test 1. In particular, the less experienced doctors, the more significantly their diagnostic ability in the

assisted state of CAD. In addition, it can be seen that JAFROC, which quantifies the local features of the lesion, increased by about 2%–3% in all observer groups.

### IV. CONCLUSION

This study was conducted to verify the clinical validity of CAD. This confirmed the results by directly applying data from the learned data set and data from a different real-world medical environment to CAD. The effectiveness of CDA was discovered by using CAD in OPT, which consisted of groups by clinical experience. The clinical data application performance of CAD used in this study showed an AUC of 0.87 at 0.86 and an JAFROC of 0.87 at 0.86 as a sole test. Also, the performance of OPT is from AUC 0.814 to 0.932, 0.904 to 0.958 and JAFROC from 0.781 to 0.907, 0.873 to 0.938. The results of the observer performance test show that CAD improved the ability of the observer (chest radiologist, radiologist, lung cancer specialist) to detect chest abnormalities. This means that CAD has a clinical effect on locating lesions. Therefore, it can be said that the accuracy of CAD is guaranteed even if it is clinical data from a source (hospital) other than the learned data. In addition, in the case of doctors with long clinical experience, it can be effective as CAD.

### ACKNOWLEDGMENT

This thesis was conducted with the support of the Korea Industrial Technology Promotion Agency's "Bio Convergence Technology Professional Training Project" with the funding of the government (Ministry of Trade, Industry and Energy) in 2021 (No. P0017805).

### REFERENCES

- [1] B.P. Little, M.D. Gilman, K.L. Humphrey, T.K. Alkasab, F.K. Gibbons, J.O. Shepard, and C.C. Wu, "Outcome of recommendations for radiographic follow-up of pneumonia on outpatient chest radiography," *American Journal of Roentgenology*, vol. 202, pp. 54-59, 2014.
- [2] H.B. Harvey, M.D. Gilman, C.C. Wu, M.S. Cushing, E.F. Halpern, J. Zhao, P.V. Pandharipande, J.O. Shepard, and T.K. Alkasab, "Diagnostic yield of recommendations for chest CT examination prompted by outpatient chest radiographic findings," *Radiology*, vol. 275, pp. 262-271, 2015.
- [3] E.J. Hwang, S. Park, K-N. Jin et al, "Development and validation of a deep learning-based automated detection algorithm for major thoracic diseases on chest radiographs," *JAMA network open* 2.3, e191095-e191095, 2019.
- [4] P. Rajpurkar, J. Irvin, K. Zhu, et al, "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning," *arXiv preprint, arXiv:1711.05225*, 2017.
- [5] D.M. Hansell, A.A. Bankier, H. MacMahon, T.C. McLoud, N.L. Müller, and J. Remy, et al, "Fleischner Society: glossary of terms for thoracic imaging," *Radiology*, vol. 246, pp. 697-722, 2008.
- [6] A.E.W. Johnson, T.J. Pollard, N.R. Greenbaum, et al, "MIMIC-CXR-JPG, a large publicly available database of labeled chest radiographs," *arXiv preprint, arXiv:1901.07042*, 2019.
- [7] World Health Organization, ICD-10 Version: 2016, [apps.who.int/classifications/icd10/browse/2016/en.F00-F09](https://apps.who.int/classifications/icd10/browse/2016/en.F00-F09), February 2016.
- [8] E.J. Hwang, S. Park, K.N. Jin, J.I. Kim, S.Y. Choi, J.H. Lee, J.M. Goo, J. Aum, J.J. Yim, C.M. Park, and Deep Learning-Based Automatic Detection Algorithm Development and Evaluation Group, "Development and validation of a deep learning-based automatic detection algorithm for active pulmonary tuberculosis on chest radiographs," *Clinical Infectious Diseases*, vol. 69, pp. 739-747, 2019.

# Fake Data Generation for Medical Image Augmentation using GANs

Donghwan Kim  
Dept. Electronic Engineering  
Pusan National University  
Pusan, Korea  
dongh@pusan.ac.kr

Jaehan Joo  
Dept. Electronic Engineering  
Pusan National University  
Pusan, Korea  
jhjoo2018@pusan.ac.kr

\*Suk Chan Kim  
Dept. Electronic Engineering  
Pusan National University  
Pusan, Korea  
sckim@pusan.ac.kr

**Abstract**—This paper uses WGAN-GP to generate fake data that can be used as augmented data for strabismus classification and analyze the results. In the introduction of this paper, the general diagnostic technique for strabismus disease is described and the diagnostic technique using deep learning is described. And the reason for generating fake data is described. Main subject describes the WGAN-GP, data set used for data generation and evaluation metrics of GAN. In the experimental result, the data generated by the GAN is visually checked, and the performance of the fake data is evaluated with the FID that is one of the evaluation metrics of the GAN. And in the conclusion, evaluation of the proposed GAN and future work are described.

**Index Terms**—WGAN-GP, strabismus, diagnosis, deep learning

## I. INTRODUCTION

Recently, deep learning has been widely used in many fields. Among them, a diagnostic technique using deep learning based on medical images will present a new paradigm in disease diagnosis. Strabismus, which is treated in this study, is an ophthalmic disease in which the two eyes are not aligned. Strabismus is a disease with a good prognosis and a high cure rate when it is detected early and treated at an early age. However, because there is a risk of blindness when strabismus treatment is neglected at early age. So early detection of strabismus is very important.

Diagnosis of a disease usually involves two steps. The first step in diagnosis is to determine the presence or absence of the disease, and the second step is to determine the severity of the disease. Strabismus is also diagnosed with the same procedure. Generally, diagnosis of strabismus is to determine the presence or absence of strabismus by conducting a cover test. The cover test is a diagnostic technique that covers one of the two eyes and determines the movement of the remaining eye to determine strabismus. In patients with mild strabismus, MRI (Magnetic Resonance Imaging) is sometimes used to determine the presence or absence of strabismus. However, in the case of infants and young children, it may be uncooperative during the cover test and MRI scan. Therefore, deep learning-based diagnostic techniques can be effective when diagnosing uncooperative patients.

Prior to this study, in order to determine the presence or absence of strabismus, the first stage of diagnosis, a CNN-based classifier was designed to determine the presence or

absence of strabismus [1]. Finally, the accuracy of the test set was 66.7%. However, it cannot be said to be accurate because the number of data in the data set is imbalanced. Due to the characteristics of medical data, data imbalance may occur. Various techniques are used when using imbalanced data for deep learning. To overcome the limitations of imbalanced data, this research applies a data augmentation technique using WGAN-GP.

## II. MAIN SUBJECT

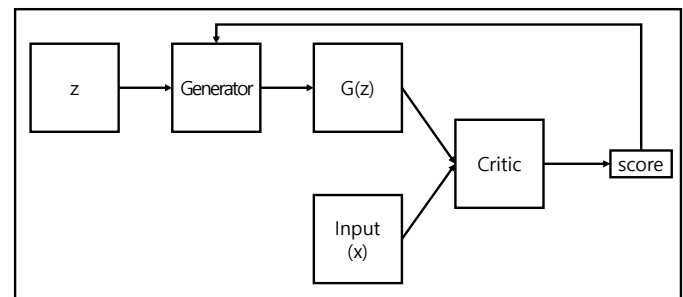


Fig. 1. Architecture of WGAN-GP.

WGAN-GP (Wasserstein Generative Adversarial Networks - Gradient Penalty) is an upgraded version of GAN (Generative Adversarial Networks)'s loss [2]. And the WGAN-GP is more usable than the original GAN. Because WGAN-GP is stronger than the original GAN at the mode collapse problem which is the serious issue of the GAN. The GAN has two networks called a generator and a discriminator. The two networks minimize and maximize the same objective function respectively, and continue learning while maintaining an adversarial relationship [3]. As a result, the generator of the GAN that has finished learning generates fake data with a distribution similar to the distribution of the training set by using the noise vector as an input. WGAN-GP has the same architecture and operation principle as GAN. Its architecture is shown in Fig. 1. Also the task of the WGAN-GP is the same as the GAN. It is generating fake data. As mentioned above, the objective function of WGAN-GP is an upgraded version of the objective function of GAN and is the same as Eq. (1)

$$L = \mathbb{E}_{\hat{x} \sim \mathbb{P}_g} [D(\hat{x})] - \mathbb{E}_{\hat{x} \sim \mathbb{P}_r} [D(x)] + \lambda \mathbb{E}_{\hat{x} \sim \mathbb{P}_g} [(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2] \quad (1)$$

WGAN-GP uses Eq. (1) as an objective function and limits the maximum value of the weight by giving a penalty to the gradient. By limiting the maximum value of the weight, it somewhat solves the instability of the training process and mode collapse, which are chronic issues of GAN, although not completely. And in the WGAN-based GAN, the discriminator is called a critic because it scores the input rather than calculating the probability that the input is real or fake [4].

The architecture of the original WGAN-GP is based on DC-GAN(Deep Convolutional Generative Adversarial Networks) [5]. The generator and critic of the model proposed in this paper have a RESNET(RE)architecture by adding a residual block while having a CNN-based architecture [6]. And the objective function uses the same Eq. (1) as that of the original WGAN-GP.

### B. Dataset

The dataset used in this study consists of a photograph which only the eye part of the patient's frontal photograph. The dataset consists of three classes: exotropia, esotropia, normal eyes. The label for each photo is the result of a cover test mentioned in the introduction. To validate the data and the trained model, 10 Pusan National University Hospital ophthalmologists re-diagnosed the patients with naked eyes and the photograph which has the correct alignment by cover test. As a result, the correct rate of 10 doctors for each data is defined as the Selection Rate(SR) in this paper.

TABLE I  
THE NUMBER OF DATA EACH SELECTION RATE

		Class		
		Exotropia(XT)	Esotropia(ET)	Normal(NO)
SR	None	1250	451	999
	60	1175	412	959
	70	1067	367	886
	80	938	316	784

As shown in Table. I, the dataset used in this study is reconstructed into a total of four datasets: a dataset with selection rates of 60, 70, 80, and 0. After that, each of the four datasets is trained with the proposed model mentioned above.

### C. Evaluation Metrics of GAN

The representative evaluation metrics of GAN are Inception Score(IS) and FID(Fr chet Inception Distance). Both IS and FID use pre-trained weights of a CNN, called a network called Inception v3. And both metrics use the distribution of the dataset. In particular, IS indicates how well the generated samples match the real data, and the higher the score, the better the quality of data. And FID indicates the distribution distance of the generated samples with the real

data based on KL Divergence. The closer the distance, that is, the lower the FID score, the better the data.

In the case of IS, since the diversity of the generated samples cannot be considered, even if mode collapse, which is a chronic issue of GAN, occurs, it tends to maintain a high value as long as the quality of fake data is good. On the other hand, FID considers the diversity of fake data and real data. Therefore, the generated samples in this study are evaluated by FID.

## III. EXPERIMENTAL RESULTS

In this section, as in the main subject of the above, WGAN-GP of the RESNET architecture is trained on four datasets divided by selection rate. And the experimental results are discussed. 100 samples were generated for each dataset, and the top 10 and bottom 10 were selected visually. And we discuss the results by comparing the FID that compares the samples generated for each dataset with the distribution of the training set.

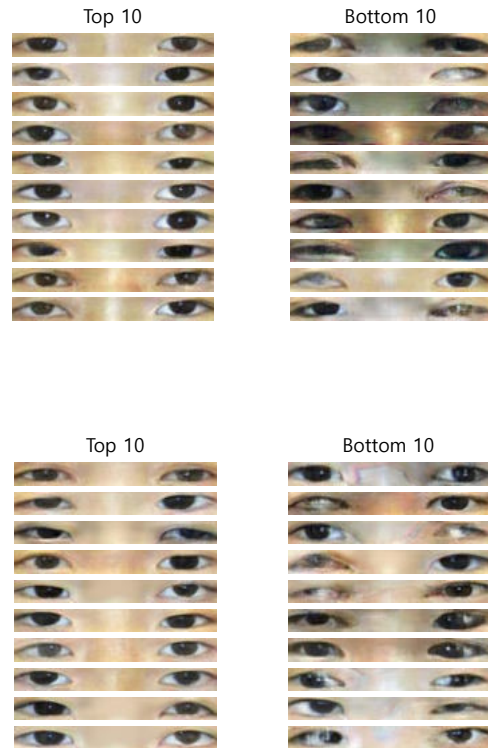


Fig. 3. Generated Samples with selection rate that is 60.

Fig. 2, Fig. 3, Fig. 4 and Fig. 5 are the top 10 and bottom 10 of the generated samples from the dataset composed by selection rate, respectively. Although more than half of the dataset are photos of strabismus patients, most of the top 10 do not show strabismus with the naked eyes. Also, there is no big difference in the image quality of the top 10 by selection rate. However, in the case of bottom 10, it can be seen that the color tends to be slightly better as the selection rate increases.

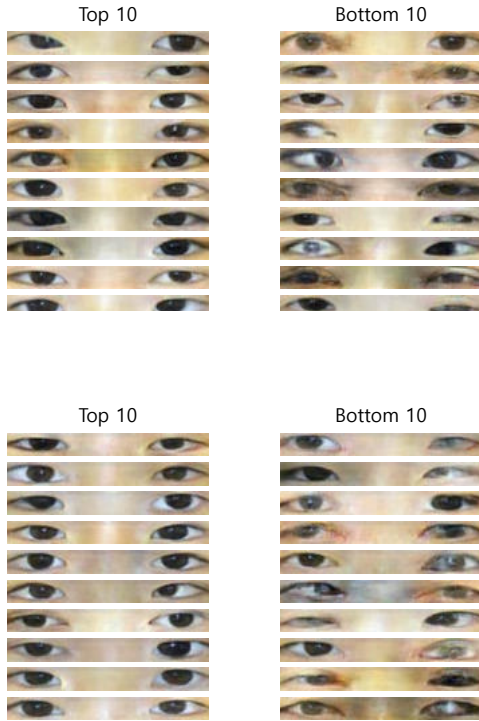


Fig. 5. Generated Samples with selection rate that is 80.

These generated samples are compared by the FIDs in the next section.

### B. FID of Generated Samples

TABLE II  
FIDS EACH SELECTION RATE

	All(SR:0)	SR:60	SR:70	SR:80
All	2.185	2.297	1.971	1.630
Top 10	8.575	8.377	10.734	10.716
Bot 10	9.785	8.606	8.542	8.167

The FIDs shown in Table. II came out as a result. As expected, the FID tends to decrease as the selection rate increases. This is because the higher the selection rate, the higher the probability that the dataset has information about strabismus and better quality of samples. Also, in the case of bottom 10, as visually confirmed in Fig. 2, Fig. 3, Fig. 4 and Fig. 5, the higher the selection rate, the lower the FID. However, in the case of top 10, the higher the selection rate, the higher the FID tends to be. FID considers the diversity of samples as mentioned above. However, 10 photos are not enough to consider diversity. Therefore, an exceptional result would have been obtained.

## IV. CONCLUSIONS & FUTURE WORKS

At the section of FID of Generated Samples, you can see the results by the FIDs. The larger SR, the smaller FID. By [8], Augmenting data at the only discriminator, it can lower FID. In this paper, augmenting data was used at the only

discriminator. But you can see the FID score is good at high SR. So consisting useful data is important at the GAN also.

As can be seen from the experimental results, the generated samples by the proposed methods were confirmed. In both top 10 and bottom 10, the quality did not change significantly according to the selection rate. In this study, data augmentation using GAN was performed because of the lack of ET data. However, most of the generated samples are difficult to observe strabismus with the naked eyes. Therefore, in future work, data with strabismus will be generated based on Conditional GAN. In addition, a CNN-based classifier will be used to verify the augmentation effect of the generated samples.

## ACKNOWLEDGMENT

This work is financially supported by Korea Ministry of Land, Infrastructure and Transportation(MOLIT) as [Innovative Talent Education Program for Smart City].

This work was supported by the Human Resources Development program (No. 20204030200030) of the Korea Institute of Energy Technology Evaluation and Planning (KETEP) grant funded by the Korea government Ministry of Trade, Industry and Energy.

## REFERENCES

- [1] Kim, D., Joo, J., Zhu, G., Seo, J., Ha, J., & Kim, S. C. (2021, April). Strabismus Classification using Convolutional Neural Networks. In 2021 International Conference on Artificial Intelligence in Information and Communication (ICAIC) (pp. 216-218). IEEE.
- [2] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. (2017). Improved training of wasserstein gans. arXiv preprint arXiv:1704.00028.
- [3] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. Advances in neural information processing systems, 27.
- [4] Arjovsky, M., Chintala, S., & Bottou, L. (2017, July). Wasserstein generative adversarial networks. In International conference on machine learning (pp. 214-223). PMLR.
- [5] Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.
- [6] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
- [7] Pan, Z., Yu, W., Yi, X., Khan, A., Yuan, F., & Zheng, Y. (2019). Recent progress on generative adversarial networks (GANs): A survey. IEEE Access, 7, 36322-36333.
- [8] Zhao, Z., Zhang, Z., Chen, T., Singh, S., & Zhang, H. (2020). Image augmentations for GAN training. arXiv preprint arXiv:2006.02595.



# Vision Anomaly Detection Using Self-Gated Rectified Linear Unit

Israt Jahan<sup>1,2</sup>, Md. Osman Ali<sup>1,3</sup>, Md. Habibur Rahman<sup>1</sup>, ByungDeok Chung<sup>4</sup>, and Yeong Min Jang<sup>1</sup>

<sup>1</sup>Department of Electronics Engineering, Kookmin University, Seoul 02707, South Korea

<sup>2</sup>Department of Electrical and Electronic Engineering, Daffodil International University, Dhaka 1341, Bangladesh

<sup>3</sup>Department of Electrical and Electronic Engineering, Noakhali Science and Technology University, Noakhali 3814, Bangladesh

<sup>4</sup>ENS. Co. Ltd., Ansan 15655, South Korea

Email: israt@kookmin.ac.kr; osman@kookmin.ac.kr; rahman.habibur@ieee.org; bdchung@ens-km.co.kr; yjang@kookmin.ac.kr

**Abstract**—In the area of image processing and computer vision, visual anomaly detection is a critical and difficult task. For anomaly detection in surface image data, a customized neural network incorporating self-gated rectified linear unit (SGReLU) was designed, and the SGReLU-based model excelled other activation function-based models with a top-20 average test accuracy of 99.84%. The computational time needed for the operation is 10533 s for 20 epochs and the top-20 average test loss is 0.0125 using SGReLU, both of them were comparatively less than other activation functions.

**Index Terms**—vision anomaly detection, self-gated rectified linear unit, computer vision.

## I. INTRODUCTION

Visual anomaly detection, often known as anomaly detection in images is essential in terms of both theoretical and empirical work [1], [2]. When it comes to recognizing visual anomalies, deep learning networks outperform machine learning algorithms. To detect surface irregularities or fissures, the Keras functional API is utilized to generate sophisticated models in a flexible pattern.

Activation function, an essential part of the neural network, has a vital role in image processing. Different activation functions such as rectified linear unit (ReLU) [3], [4], Leaky ReLU (LReLU) [5], swish [6], scaled exponential linear unit (SELU) [7], exponential linear unit (ELU) [8] and self-gated rectified linear unit (SGReLU) [9] are mainly used in the dense layers of neural networks. Previously SGReLU was used mainly in the case of image classification. In this paper, SGReLU is used for vision anomaly detection and the comparison with other activation functions in the case of surface anomaly detection has been also investigated.

The contributions of this paper are as follows:

- To design a customized neural network with convolution layers, maxpooling layers, dense layers, dropout layers, and activation layers.
- To compare the performance of different activation functions used in the activation layer of the customized neural network.

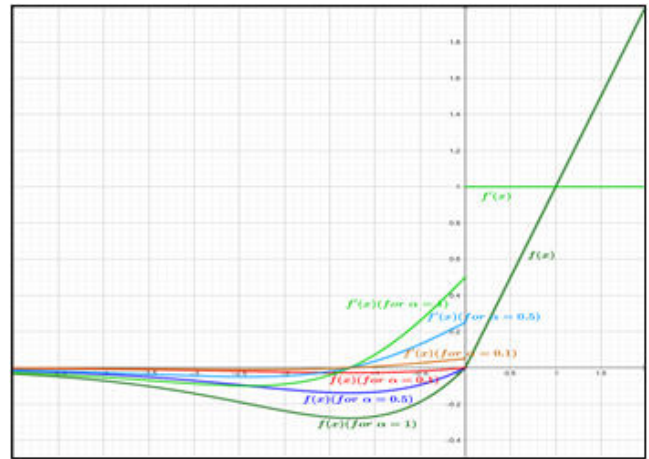


Fig. 1: SGReLU function and its first derivative for different values of  $\alpha$ .

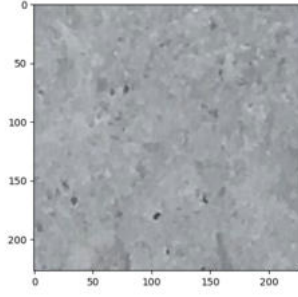
- The SGReLU function outperforms other activation functions in the case of top-20 average test accuracy.
- The SGReLU function has also achieved better performance in terms of test loss and computational time in comparison to other activation functions.

The following sections of this paper are arranged as follows. Section II presents the methodology including the SGReLU function, dataset preparation, and model architecture. In Section III, the obtained results are discussed. Finally, the conclusion, as well as future work, are described in Section IV.

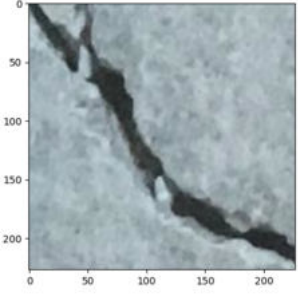
## II. METHODOLOGY

### A. SGReLU Function

The generic nature of activation functions can be expressed as a bivariate function,  $b(x, g(x))$ , where  $x$  is the unfiltered preactivation, that is the input to the final bivariate function. SGReLU function is a non-monotonic continuous function that



(a)



(b)

Fig. 2: Surface crack detection (a) normal condition and (b) anomaly condition.

maintains the identical binary format. It can be expressed as follows:

$$f(x) = \begin{cases} x, & x \geq 0 \\ \alpha \cdot x \cdot \sigma(x), & x < 0 \end{cases} \quad (1)$$

where  $\sigma(x)$  is sigmoid function, which equals to  $\frac{1}{1+e^{-x}}$ , and  $\alpha$  is a hyper-parameter that ranges from 0.1 to 1 in magnitude. In the positive zone, SGRReLU behaves like ReLU in terms of unbounded linear nature. As a result, it solves the issue of saturation as well as has a non-zero gradient, which speeds up the learning process. The absence of the sigmoid function, in contrast to swish, reduces the need for power operations, leading to a faster processing period. The self-gating approach, conversely, is used to make the negative area of the function adaptable with a single input. Consequently, with a huge value of preactivation,  $x$ , a little negative bump develops at the start of the negative area that tends to zero. SGRReLU provides a negative bump for small negative inputs, which eliminates the neuron death issue. It has negative parts, however, unlike LReLU and PReLU, it is confined below since it tends to zero for huge values of preactivation, providing sparsity in the structure and lowering computational time. As a result, the overfitting issue is eliminated, and the network achieves noise-resilience at the same time. As a result, the network achieves not only non-monotonicity but also self-regularization, making it immune to neuronal death as well as data compatibility. The

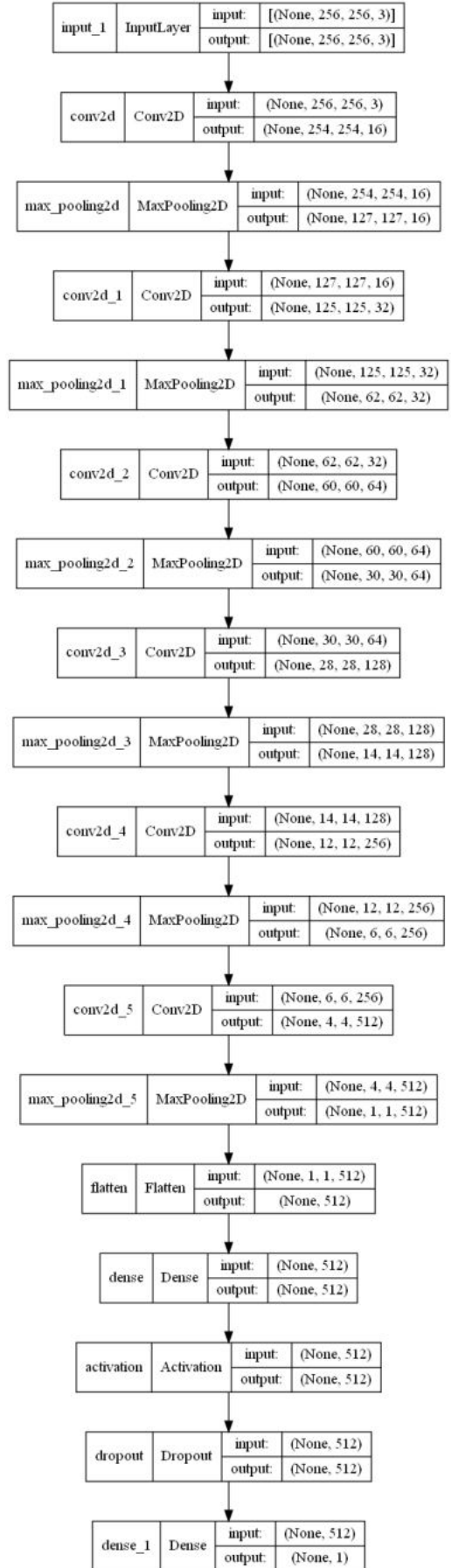


Fig. 3: Overall model architecture for vision anomaly detection.

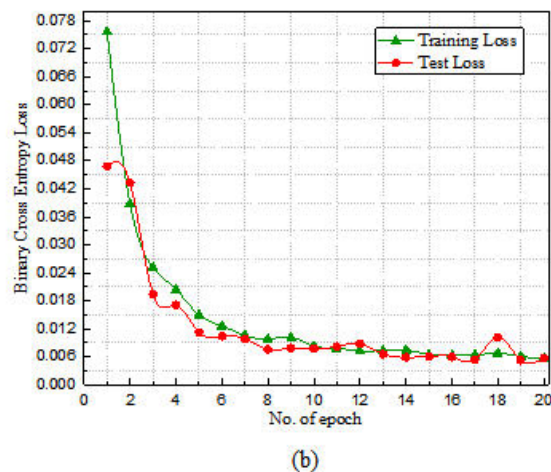
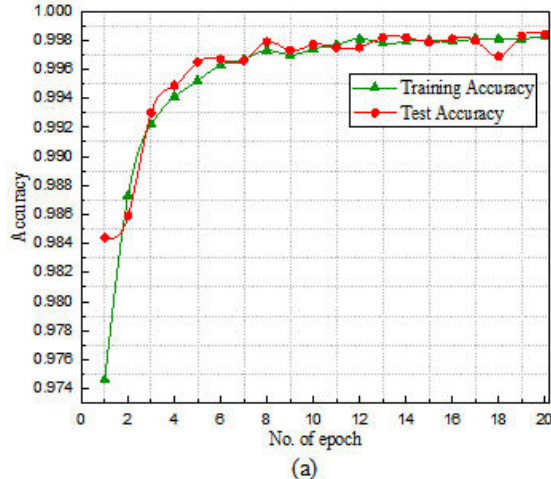


Fig. 4: (a) Training accuracy and test accuracy, and (b) training loss and test loss, curve of the surface crack detection data set using SGRReLU.

first derivative of SGRReLU function is as follows:

$$f'(x) = \begin{cases} 1, & x \geq 0 \\ \alpha \left[ \frac{1}{1+e^{-x}} + \frac{xe^{-x}}{(1+e^{-x})^2} \right], & x < 0 \end{cases} \quad (2)$$

Fig. 1 shows SGRReLU function and its derivative for different values of  $\alpha$ .

### B. Dataset Preparation

The surface crack detection dataset is used for vision anomaly detection [10]. It contains two types of data, both normal condition and anomaly condition. Normal condition dataset has no cracks in the concrete surface but anomaly condition dataset has cracks in the concrete surface. Both types of datasets contain 20,000 images each with RGB channels and a resolution of  $227 \times 227$  pixels. Fig. 2 shows samples for both normal and anomalous data. Image data generator is used to process the dataset. The rotational range, horizontal flip class mode and color mode are selected 30°C, True, Binary

TABLE I: Performance comparison among different activation functions in terms of test accuracy, test loss and computational time.

Activation function	ReLU	LReLU	Swish	ELU	SELU	<b>SGReLU</b>
Top- 20 average test accuracy (%)	99.4	99.57	99.52	98.93	99.25	<b>99.6</b>
Top- 20 average test loss	0.017	0.014	0.015	0.033	0.025	<b>0.0125</b>
Time complexity (s)	10955	11109	11109	11259	11326	<b>10533</b>

and rgb respectively. Width shift range, zoom range and height shift range, all three are selected as 0.2. The target size of the dataset is (256,256) and  $\frac{1}{255.0}$  is used for data scaling.

### C. Model Architecture

Fig. 3 shows the overall model architecture for surface anomaly detection. Six conv2D layers and six maxpooling2D layers are used consecutively, then flatten layer, dense layer, activation layer, dropout layer, and another dense layer is used to get the final output result. In the activation layer, six different types of activation functions including SGRReLU are used and the results are compared in the next section.

## III. RESULT AND DISCUSSION

In the instance of anomaly detection for the surface crack detection dataset, both training and test accuracy using SGRReLU have obtained satisfactory values, as shown in Fig. 4(a). After the 20<sup>th</sup> epoch, the test accuracy is 99.84%. In the case of the designed SGRReLU-based deep learning network, both training and validation losses are presented in Fig. 4(b).

In Table I, SGRReLU not only surpasses ReLU in terms of accuracy, but it also significantly reduces binary cross-entropy loss. The total time required by the SGRReLU function-based approach is likewise less than that required by ReLU. The value of  $\alpha$  was chosen 0.5 for the operation.

The performance comparison of different activation functions in the surface crack detection dataset in terms of test accuracy, test loss and time complexity are provided in Table I. SGRReLU has gained 99.6% top-20 average test accuracy and outperformed other activation functions, such as ReLU, LReLU, swish, ELU, and SELU, by 0.2%, 0.03%, 0.08%, 0.67%, and 0.35%, respectively. Furthermore, the top-20 average test loss is 0.0125 s in case of SGRReLU which is less than others. The computation complexity in case of SGRReLU is 10533 s which is also comparatively less than others.

## IV. CONCLUSION AND FUTURE WORK

From the experimental results, it can be seen that SGRReLU function with a fixed hyper-parameter value of  $\alpha = 0.5$  has outperformed different activation functions in case of vision anomaly detection. SGRReLU function has not only achieved

the highest accuracy but also has maintained the lowest loss as well as operational time in compared to others. In future, SGRReLU with various values of  $\alpha$  using hyper-parameter tuning or trainable parameter can be experimented for vision anomaly detection to boost performance.

#### ACKNOWLEDGMENT

This work was supported by the Technology Development Program (S3098815) funded by the Ministry of SMEs and Startups (MSS, Korea).

#### REFERENCES

- [1] J. Yang, R. Xu, Z. Qi, and Y. Shi, "Visual anomaly detection for images: A survey," *arXiv [cs.CV]*, 2021.
- [2] X. Xie and M. Mirmehdi, "TEXEMS: texture exemplars for defect detection on random textured surfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 8, pp. 1454–1464, 2007.
- [3] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. International Conference on Machine Learning*, Haifa, Israel, 2010, pp. 807-814.
- [4] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks." in *Proc. International Conference on Artificial Intelligence and Statistics Conference*, Ft. Lauderdale, FL, USA, 2011.
- [5] A. Apicella, F. Donnarumma, F. Isgrò, and R. Prevete, "A survey on modern trainable activation functions," *Neural Network*, vol. 138, pp. 14–32, 2021.
- [6] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for Activation Functions," *arXiv [cs.NE]*, 2017.
- [7] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-Normalizing Neural Networks," *arXiv [cs.LG]*, 2017.
- [8] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," *arXiv [cs.LG]*, 2015.
- [9] I. Jahan, M. F. Ahmed, M. O. Ali, and Y. M. Jang, "Self-gated rectified linear unit for performance improvement of deep neural networks," *ICT Express*, 2022.
- [10] trinadhbavisetti, "Surface crack detection," *Kaggle.com*, 31-Oct-2021. [Online]. Available: <https://www.kaggle.com/trinadhbavisetti/surface-crack-detection/data>. [Accessed: 28-Jan-2022].

# A Comparison of YOLO and Mask-RCNN for Detecting Cells from Microfluidic Images

1<sup>st</sup> Mehran Ghafari  
*Dept. of Computer Science  
& Engineering*  
*U. of Tennessee at Chattanooga*  
Chattanooga, TN, U.S.A.  
ryg668@mocs.utc.edu

2<sup>nd</sup> Daniel Mailman  
*Dept. of Computer Science  
& Engineering*  
*U. of Tennessee at Chattanooga*  
Chattanooga, TN, U.S.A.  
daniel-mailman@utc.edu

3<sup>rd</sup> Parisa Hatami  
*Dept. of Computer Science  
& Engineering*  
*U. of Tennessee at Chattanooga*  
Chattanooga, TN, U.S.A.  
qxy699@mocs.utc.edu

4<sup>th</sup> Trevor Peyton  
*Dept. of Computer Science  
& Engineering*  
*U. of Tennessee at Chattanooga*  
Chattanooga, TN, U.S.A.  
qtx464@mocs.utc.edu

5<sup>th</sup> Li Yang  
*Dept. of Computer Science  
& Engineering*  
*U. of Tennessee at Chattanooga*  
Chattanooga, TN, U.S.A.  
li-yang@utc.edu

6<sup>th</sup> Weiwei Dang  
*Dept. Molecular & Human Genetics  
Huffington Ctr. on Aging  
Baylor Coll. of Medicine  
Houston, U.S.A.*  
weiwei.dang@bcm.edu

7<sup>th</sup> Hong Qin  
*SimCenter, Dept. of Computer Science & Engineering*  
*U. of Tennessee at Chattanooga*  
Chattanooga, TN, U.S.A.  
hong-qin@utc.edu

**Abstract**—As an effective model to study aging, the budding yeast *Saccharomyces cerevisiae* has revealed aging mechanisms that are shared with human aging. Yeast cell lifespan can be measured in replicative lifespans (RLS) - the number of cell divisions from a single mother cell before dying. However, counting yeast cell divisions from microscopic images is a tedious task. Here, we address this challenge with computer vision object detection. We compared two deep learning methods, YOLO and MASK R-CNN to detect cells from microfluidic images. We concluded that YOLO is more sensitive at detecting cells, whereas MASK-RCNN is more informative on cell sizes. Therefore, both methods are useful for automatic microfluidic image analysis.

**Index Terms**—machine learning, instance segmentation, cell detection, cellular aging.

## I. INTRODUCTION

Computer Vision (CV) approaches in recent years have led to advancements in many fields including medical, civil, surveillance, auto, etc [1]. There is a tremendous demand for CV in healthcare as many diagnoses and disease treatments rely on medical imaging [2]. Objects' appearances in images are associated with many features, most notably volume, dimensionality, color, resolution, and moving object demeanor. [3], [4].

This study analyses the effectiveness of CV techniques for microfluidic cell detection (MFCDD). In MFCDD images, cells are: visually extremely similar, extremely close together (often sharing boundaries), and often overlap due to the image being a map of 3 dimensions to 2 dimensions. These factors make cell detection a challenging task. Other factors which

contribute to the difficulty of MFCDD are uneven illumination, low contrast, low resolution, out-of-focus images, and varying foreground/background intensities [5]. The core task of object detection in general and MFCDD in particular is segmentation - distinguishing object borders - into local and global regions [6], [7]. MFCDD images contain hundreds of cells to distinguish using CV segmentation methods. Precision is required, especially in the identification of overlaps [8]. Segmentation methods and models rely on image pixel characteristics as well as sub-sectioning. Various methods and approaches have been implemented to improve segmentation efficiency.

Many segmentation models are based on a convolutional neural networks (CNNs). Based on preliminary literature examination, we chose two CNN-based models - You Only Look Once (YOLO) [9] and Mask R-CNN [10] - to evaluate for the task of MFCDD.

YOLO uses a single CNN, predicts multiple bounding boxes, and determines class probability for each available image bounding box. YOLO is very fast and does not need a complex pipeline, since it relies on regression analysis. The model potentially runs at 45 frames per second (FPS) without batch processing requirements - meaning it is also capable of processing stream video in near-real-time. The model uses a simple down-sampling method which has the advantage of learning complex depth features of images using residual blocks. [11] used YOLO-based system to achieve 99.7% accuracy detecting mass located in the breast. YOLO's main drawback is using bounding boxes (rather than extracting

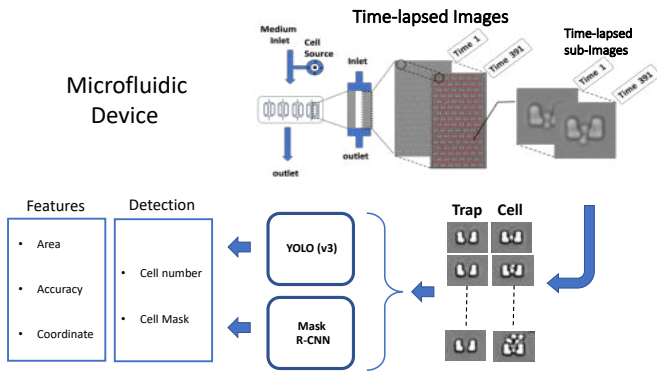


Fig. 1. Microfluidic device and image pre-processing steps. YOLO and Mask R-CNN are applied to partitioned microfluidic images. Each model’s output yields object detection and feature extraction.

shape/contour details) in approximating target positions.

Mask R-CNN is an instance-segmentation method. It is a regional CNN model that generates detected object masks, increasing accuracy in contour detection and determining shape information [12]. However, some problems were reported that Mask R-CNN has a Resnet-101 [13] as its backbone which makes it a deep neural network. Hence, it requires more computational space for each training dataset. To address this problem, modified Mask R-CNN (Resnet-86) uses fewer backbone layers for vehicle and pedestrian detection [14]. Furthermore, [15] demonstrates that Mask R-CNN has poor performance for segmentation in comparison to U-net.

Details of the comparison of YOLO to Mask R-CNN follow in the remainder of this study. Section II addresses methods. Section III compares and discusses the two methods. Section IV summarizes the research.

## II. METHOD

We used an Ubuntu 18.04.4 with Intel Xeon processor with 10 cores, 64GB of RAM, and nVidia RTX 2080 Ti GPU.

### A. Dataset

The dataset is experimental results obtained from microfluidic HYAA chips [16]. Grayscale images were acquired by a microscope (Olympus IX-81) equipped with a camera (Olympus DP72 CCD). The temperature was set at 86°F. 391 time-lapse microfluidic images were taken at 10-minute intervals over a 96-hour period. On average, each image contains 104 silicon-made traps with rows of 6 or 7 traps. (Fig.1). Since the direct-object detection methods performed poorly on cell detection due to microfluidic low image resolution (grayscale 1280x960), we cropped traps by partitioning images into sub-images based on the number of available traps on each image. This approach is an effective technique for improving accuracy as well as generating more datasets without data augmentation.

### B. Annotation process

We used 2 datasets. The first dataset (used for cell detection) contained 100 training sub-images and 30 test sub-images.

The second dataset (used for feature extraction) contained 100 training sub-images and 40664 test sub-images. Sub-images for the first dataset were randomly selected from a batch containing a maximum of 5 cells per image.

We used "Microsoft VoTT Tool" and "Image-J" for image annotation. Mask R-CNN annotation is polygon-based. YOLO (bounding box) annotation format is  $[x,y,w,h]$ , where  $(x,y)$  is the bounding box centroid,  $w$  is the width, and  $h$  is the height. Training-set sub-image dimension is 60X60. We used 60X60 and 512X512 sub-image size for the cell detection test dataset. The larger images were made using cubic interpolation.

### C. YOLO

YOLO takes an image and estimates a confidence level for each detected object. YOLO’ strategy is to reframe object detection as a single regression problem from image pixels to bounding box and classification probabilities. Fig 2a shows YOLO network architecture where the input image is 60x60, scaled up to 448x448x1. The next section is the DarkNet Architecture which is a CNN based on GoogleNet architecture [17]. DarkNet transforms image dimensions from 448x448x1 to 7x7x1024. Further, 2 full-connected neural networks are applied to the model with 2 outputs (2b): object bounding box (including object score) and class probability. In the entire YOLO network, the down-sampling of the network is based on setting the convolution stride hyperparameter to 2 without applying the pooling layer. The loss function consists of classification loss for the class probability and localization loss for the confidence level and bounding box which are both based on the squared error (sum).

[ht]

The improved version of this model is YOLOv2 [18], YOLOv3 [19], and YOLOv4 [20]. This work mainly focuses on YOLOv3, and all results are based on version 3 of this model.

### D. Mask R-CNN

Region-Based CNN (R-CNN) is used for semantic segmentation and object detection and builds on other CNN models. The baseline models - Fast R-CNN [21], Faster R-CNN [22], and Fully Connected Network (FCN) [23] - are robust, pliable, fast-training, and conceptually intuitive. Mask R-CNN is based on Faster R-CNN. Mask R-CNN outperforms traditional semantic segmentation models by offering instance segmentation, including object mask. Fig 3 illustrates the varieties of R-CNN architecture. The salient differences among the models is summarized as follows.

In Fig 3a, multiple region features (size, shape, texture, color) are determined via multiple deep CNNs (e.g., AlexNet [24]) and fed separately to the bounding box offset regressor and the support vector machine (SVM) object classifier. In Fig 3b, the CNN region output is consolidated with a Region-Of-Interest (ROI) pooling layer. The consolidated data is fed to the regressor and the classifier enabling association of class labels to ROIs. In Fig 3c, multiple region proposals are eliminated in favor of using the CNN output as input to a Region Proposal

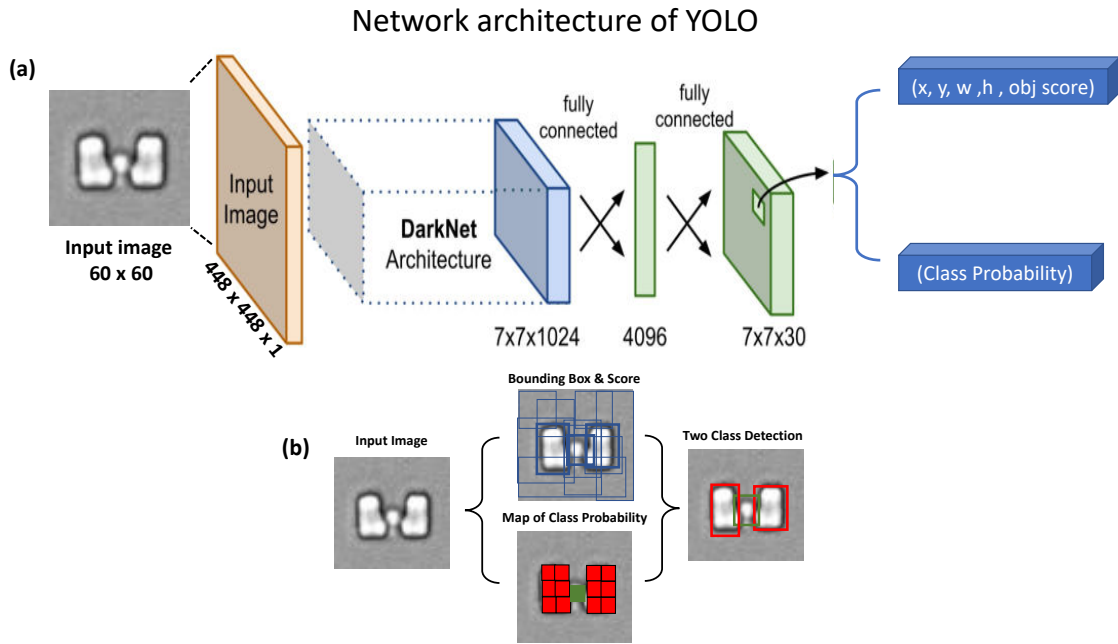


Fig. 2. YOLO architecture. (a) YOLO architecture with 60x60 image dimensions which scaled up to 448x448x1. The output contains bounding box information, object score, and object class. (b) Minimizing bounding box error with the map of class probability.

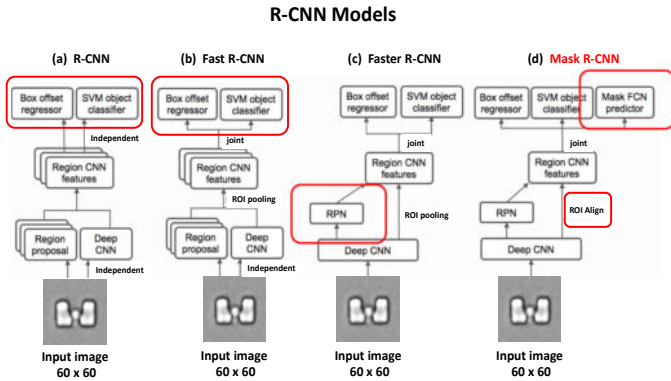


Fig. 3. Development of Region-Based Convolutional Neural Network architectures including (a) R-CNN , (b) Fast R-CNN , (c) Faster R-CNN and (d) Mask R-CNN.

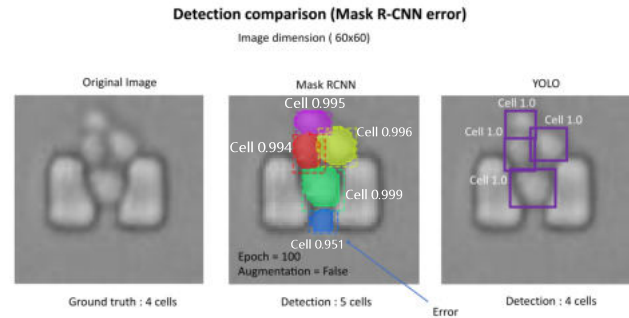


Fig. 4. Model detection comparison for trap with 4 cells. Mask R-CNN detected an extra cell. YOLO detection matched ground truth.

Network (RPN). Fig 3d illustrates Mask R-CNN modifications to Faster R-CNN: ROI pooling is replaced with ROI alignment and a fully convolutional network (FCN) is added to feature analysis for determining object masks.

### III. RESULTS AND DISCUSSION

#### A. Cell detection

This study assessed YOLO and Mask R-CNN object detection performance with 60x60 and augmented 512x512 test datasets. We trained YOLO for 200 epochs and Mask R-CNN for 100 and 400 epochs. YOLO performance was evaluated only with dataset augmentation; Mask R-CNN was evaluated both with and without dataset augmentation.

Fig 4 shows detection results for both models. Ground truth was 4 cells, Mask R-CNN detected an extra cell at the trap

outlet (blue cell). This illustration is based on 60x60 image dimensions, 200 epochs for YOLO, and 100 epochs for Mask R-CNN without dataset augmentation.

Fig 5 shows Mask R-CNN detecting the correct number of cells, but overestimating cell size. Dataset augmentation enables Mask R-CNN to better estimate cell size.

Fig 6 illustrates dataset augmentation and 400 epochs improving Mask R-CNN cell detection. The detected cell and mask image counts are similar to the source image. In contrast, YOLO detected 1 less cell than ground truth (purple cell).

Fig 7 illustrates the benefit of using larger images created with cubic interpolation. Since YOLO and Mask R-CNN are designed to detect objects at higher image resolution, we supplemented the study with scaled-up images. Fig 7 represents detection accuracy differences due to image size.

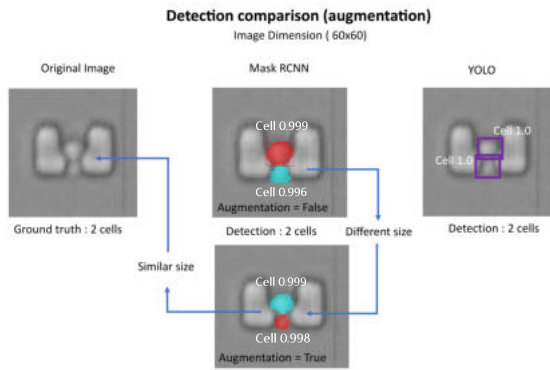


Fig. 5. Detection with dataset augmentation. YOLO and Mask R-CNN detection matched the ground truth. Dataset augmentation improved the accuracy of mask area for Mask R-CNN.

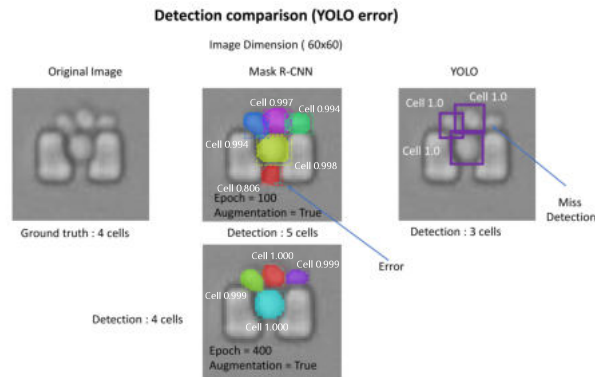


Fig. 6. Modification of Mask R-CNN including YOLO detection error. Mask R-CNN with augmentation and 400 epochs detected 4 cells (matched the ground truth), and YOLO detected 3 cells.

The top-row images show similar results for Mask R-CNN and YOLO with dataset augmentation and 400 epochs applied to Mask R-CNN. The bottom-row shows YOLO detecting 2 cells with an inaccurate bounding box (covering only half the cell area). In this example, Mask R-CNN with data augmentation and 400 epochs was more accurate than YOLO..

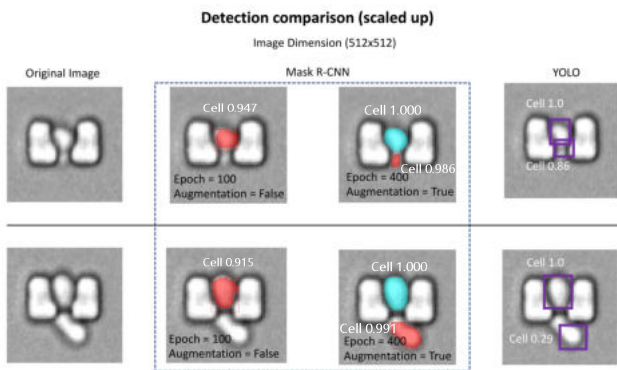


Fig. 7. Comparison of YOLO and Mask R-CNN with higher image resolution. In the first row, modified Mask R-CNN and YOLO matched the ground truth (2 cells). In the second row, YOLO detected a small portion of the cell below the trap.

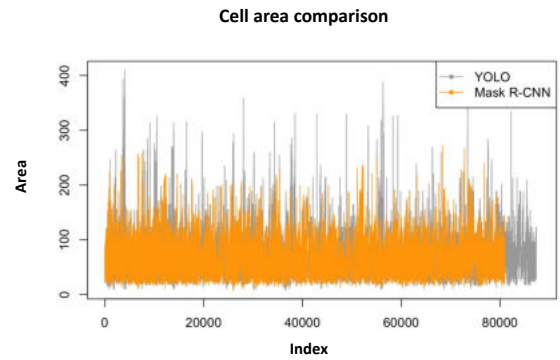


Fig. 8. Cell area comparison for YOLO and Mask R-CNN. Orange/Gray discrepancy illustrates Mask R-CNN detecting fewer cells.

### B. Feature extraction

In this section, we evaluate the performance of YOLO and Mask R-CNN on a dataset that contains 100 training images and 40,664 test images. YOLO trained for 200 epochs and Mask R-CNN trained for 400 epochs. Our dataset augmentation was used for both models. Features for both models are 'area', 'total objects', 'confidence', and 'coordinates'.

Fig 8 shows cell size comparison using both models. YOLO results are in gray, Mask R-CNN results are in orange. Yolo's average cell area is larger Mask R-CNN's. YOLO's average cell size ranges from 80 to 100 pixels with confidence rate from 10% to 100%. In contrast, Mask R-CNN's average detected cell size ranges from 50 to 80 pixels, and its detection rate confidence ranges from 90% to 100%.

Fig 9 plots cell size variation versus detection counts for sample traps 01, 20, and 60. for both models. YOLO results show many same-size cells (represented as a row) which indicates that YOLO is less accurate predicting cell size. More variation with Mask R-CNN indicates greater accuracy determining cell size.

Fig 10 shows total counts: 87,908 (YOLO) and 81,842 (Mask R-CNN).

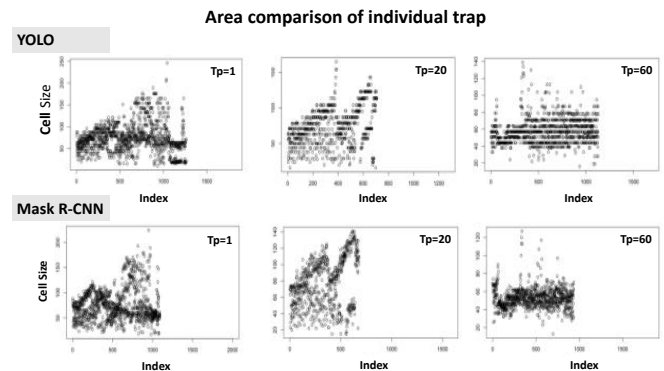


Fig. 9. Area variation for sample traps. YOLO is more accurate for larger cell sizes and Mask R-CNN is more accurate for smaller cell sizes.



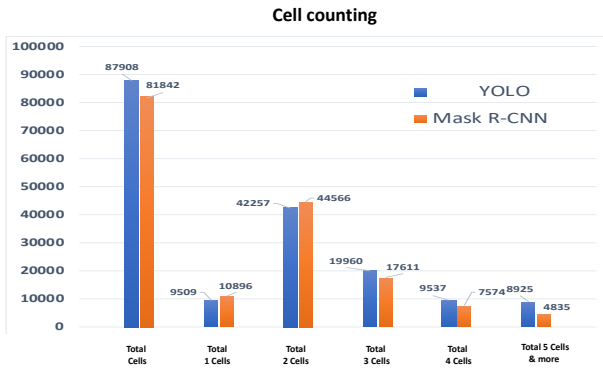


Fig. 10. Cell counting comparison for the individual model. YOLO had better cell detection when the number of cells inside a trap was more the 2 cells. Mask R-CNN performed better when the number of cells was in the range of 1 to 2 cells.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

### C. Models comparison

Performance metrics are calculated using equations 1, 2 and 3 where TP is true positives, TN is true negatives, FP is false positives, and FN is false negatives.

Table I compares simple metrics for both methods. The metrics for TP and FP indicate YOLO is more accurate for cell detection with mAPs of 90.6% (YOLO) and 73% (Mask R-CNN). Total cell detections indicate that YOLO is more sensitive for object detection and has less variation in the cell area.

Fig 11 compares mean average precisions (mAPs) for the dataset comprising the first 30 images, indicating YOLO fluctuates less than Mask R-CNN. YOLO cell area calculation uses bounding boxes, decreasing accuracy.

In this work, we modelled cell area as ellipses and calculated it using bounding box information. Both models had the highest performance when there were 2 cells inside traps and had poor performance when there were more than 3 cells inside traps. Mask R-CNN performed much better than YOLO when the number of cells inside the trap is less than 3 cells. Although Mask R-CNN has a lower mAP, its cell area detection is more accurate compared to ground truth. Since Mask R-CNN generates masks, cell area accuracy is much higher than YOLO.

TABLE I  
MODELS PERFORMANCE COMPARISON

Metric	YOLO	Mask R-CNN
	Cell	
TP	74	53
FP	7	11
Precision	95.03%	85%
Recall	92%	82%
Total Detection	81	64
Total Image	30	30
<b>mAP</b>	<b>90.6 %</b>	<b>73 %</b>

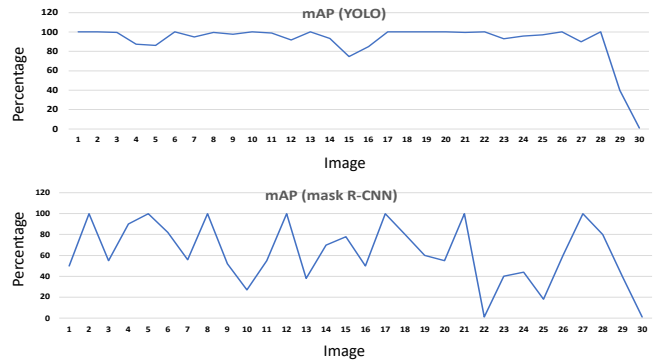


Fig. 11. mAP comparison for YOLO and Mask R-CNN.

## IV. CONCLUSION

We evaluated two CNN models for detecting cells in microfluidic images. YOLO and Mask R-CNN were trained with 100 yeast microfluidic images, tested for object detection on 30 images, and feature extraction on 40,664 images. The results indicate that YOLO was more accurate for object detection but was very sensitive to noise. Yolo also was less accurate for area estimation.

To both generalize and summarize: YOLO appears useful for feature extraction and object detection, but less-so for cell area determination and produces extra unnecessary details (noise). Mask R-CNN produces better estimates of area due to its use of masking and can be improved with data augmentation and increasing epoch count, which increases already computationally expensive training.

This comparison implies that YOLO and Mask R-CNN are both useful for automatic small object detection from medical images. However, we emphasize the present study highlights the need for further development of deep learning methods to facilitate the analysis of time-lapse microscopic images generated by microfluidic devices.

## V. ACKNOWLEDGEMENT

This work was partially supported by NSF CAREER award #1453078 (transferred to #1720215), NSF award #1761839, a start-up fund, internal awards from the University of Tennessee at Chattanooga. We acknowledge the support of the College of Engineering and Computer Science for a Lambda Quad GPU workstation. We thank the Research as a Service (RaaS) cluster at the SimCenter that supported the prototyping of

this work. We also acknowledge the support of NIH grants #R01AG052507 and #R42AG058368.

## REFERENCES

- [1] M. McAuliffe, F. Lalonde, D. McGarry, W. Gandler, K. Csaky, and B. Trus, "Medical image processing, analysis and visualization in clinical research," in *Proceedings 14th IEEE Symposium on Computer-Based Medical Systems. CBMS 2001*, 2001.
- [2] F. Ritter, T. Boskamp, A. Homeyer, H. Laue, M. Schwier, F. Link, and H.-O. Peitgen, "Medical image analysis," *IEEE Pulse*, vol. 2, no. 6, pp. 60–70, 2011.
- [3] M. McAuliffe, F. Lalonde, D. McGarry, W. Gandler, K. Csaky, and B. Trus, "Medical image processing, analysis and visualization in clinical research," in *Proceedings 14th IEEE Symposium on Computer-Based Medical Systems. CBMS 2001*, 2001, pp. 381–386.
- [4] M. Ghafari, D. Mailman, and H. Qin, "Application note: polar - an interactive 2d visualization tool for time-series," 2021, pp. 1–6. [Online]. Available: <https://ssrn.com/abstract=3827406>
- [5] R. L. Brocca, F. Menolascina, D. di Bernardo, and C. Sansone., "A novel graphical model approach to segmenting cell images," pp. 131–139, 2012.
- [6] M. Kvarnström, K. Logg, A. Diez, K. Bodvard, and M. Käll, "Image analysis algorithms for cell contour recognition in budding yeast," *Opt. Express*, vol. 16, no. 17, pp. 12943–12957, Aug 2008. [Online]. Available: <http://www.opticsexpress.org/abstract.cfm?URI=oe-16-17-12943>
- [7] D. Rea, G. Perrino, D. di Bernardo, L. Marcellino, and D. Romano, "A gpu algorithm for tracking yeast cells in phase-contrast microscopy images," *The International Journal of High Performance Computing Applications*, vol. 33, no. 4, pp. 651–659, 2019. [Online]. Available: <https://doi.org/10.1177/1094342018801482>
- [8] S.-C. Chen, T. Zhao, G. J. Gordon, and R. F. Murphy, "A novel graphical model approach to segmenting cell images," in *2006 IEEE Symposium on Computational Intelligence and Bioinformatics and Computational Biology*, 2006, pp. 1–8.
- [9] S.-C. Chen, G. J. Gordon, and R. F. Murphy, "Graphical models for structured classification, with an application to interpreting images of protein subcellular location patterns," *J. Mach. Learn. Res.*, vol. 9, p. 651–682, Jun. 2008.
- [10] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 386–397, 2020.
- [11] M. A. Al-masni, M. A. Al-antari, J.-M. Park, G. Gi, T.-Y. Kim, P. Rivera, E. Valarezo, M.-T. Choi, S.-M. Han, and T.-S. Kim, "Simultaneous detection and classification of breast masses in digital mammograms via a deep learning yolo-based cad system," *Computer Methods and Programs in Biomedicine*, vol. 157, pp. 85–94, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0169260717314980>
- [12] H. Wu and J. P. Siebert, "Fully convolutional networks for automatically generating image masks to train mask R-CNN," *CoRR*, vol. abs/2003.01383, 2020. [Online]. Available: <https://arxiv.org/abs/2003.01383>
- [13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
- [14] S. Y. J. Y. B. Z. S. D. Y. L. L. X. Chenchen Xu, Guili Wang, "Fast vehicle and pedestrian detection using improved mask r-cnn," *Mathematical Problems in Engineering.*, vol. 15, 2020.
- [15] A. O. Vuola, S. U. Akram, and J. Kannala, "Mask-rcnn and u-net ensembled for nuclei segmentation," in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, 2019, pp. 208–212.
- [16] M. C. Jo, W. Liu, L. Gu, W. Dang, and L. Qin, "High-throughput analysis of yeast replicative aging using a microfluidic system," *Proceedings of the National Academy of Sciences*, vol. 112, no. 30, pp. 9364–9369, 2015. [Online]. Available: <https://www.pnas.org/content/112/30/9364>
- [17] P. Salavati and H. M. Mohammadi, "Obstacle detection using googlenet," in *2018 8th International Conference on Computer and Knowledge Engineering (ICCKE)*, 2018, pp. 326–332.
- [18] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6517–6525.
- [19] —, "Yolov3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018. [Online]. Available: <http://arxiv.org/abs/1804.02767>
- [20] W. Boyuan and W. Muqing, "Study on pedestrian detection based on an improved yolov4 algorithm," in *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*, 2020, pp. 1198–1202.
- [21] R. B. Girshick, "Fast R-CNN," *CoRR*, vol. abs/1504.08083, 2015. [Online]. Available: <http://arxiv.org/abs/1504.08083>
- [22] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015. [Online]. Available: <http://arxiv.org/abs/1506.01497>
- [23] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.
- [24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'12. Red Hook, NY, USA: Curran Associates Inc., 2012, p. 1097–1105.

# Multiview Attention for 3D Object Detection in Lidar Point Cloud

Kevin Tirta Wijaya\*

*The Robotics Program  
KAIST*

Daejeon, Republic of Korea  
kevin.tirta@kaist.ac.kr

Donghee Paek\*

*The Cho Chun Shik*

*Graduate School of Green Transportation  
KAIST*

Daejeon, Republic of Korea  
donghee.paek@kaist.ac.kr

Seung-Hyun Kong†

*The Cho Chun Shik*

*Graduate School of Green Transportation  
KAIST*

Daejeon, Republic of Korea  
skong@kaist.ac.kr

**Abstract**—Prior works in voxel-based Lidar 3D object detection have demonstrated promising results in detecting a variety of road objects such as cars, pedestrians, and cyclists. However, these works generally reduce the feature space from a 3D volume into a 2D bird eye view (BEV) map before generating object proposals to speed up the inference runtime. As a result, the resolution of information in the z-axis is reduced significantly. In this work, we hypothesize that augmenting the BEV features with features obtained from a front view (FV) map may provide a way for the network to partially recover the high-resolution z-axis information. The augmentation allows object proposals to be inferred in the BEV, maintaining the fast runtime, and simultaneously improving the 3D detection performance. To support our hypothesis, we design a multi-view attention module that augments the BEV features with the FV features and conduct extensive experiments on the widely used KITTI dataset. Based on the experimental results, our method successfully improves various existing voxel-based 3D object detection networks by a significant margin.

**Index Terms**—3D object detection, Lidar point cloud, multi-view attention

## I. INTRODUCTION

Object detection is one of the most researched topics in the field of computer vision. Various 2D object detection networks with remarkable performance have been introduced, owing to the advancements in deep learning, convolutional neural networks (CNN), and transformers. However, 2D object detection alone is often not sufficient for a machine, such as an autonomous car, to operate in a real-world scenario. Objects in the real world exist in the three-dimensional space, therefore, 3D information is required to plan safe maneuvers. Hence, a robust 3D object detection technique is a crucial function for an autonomous car.

3D object detection is often performed in a point cloud obtained from a Lidar sensor. Recent deep learning-based 3D object detection networks for Lidar point cloud [13]–[17] often consist of a preprocessing module, a feature extraction backbone, a detection head, and in the case of two-stage detectors, an additional refinement head.

\* co-authors, † corresponding author

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (2021R1A2C300837011)

As Lidar measurements are often stored in an unordered list, the preprocessing module needs to introduce spatial structure to the data so that convolution operations can be applied. The most commonly used preprocessing method is voxelization, where the 3D point cloud space is discretized into non-overlapping cuboids of identical sizes. The attributes of points that lie in the same voxel are used as the feature vectors of the voxel. A series of 3D convolutions are then applied to the voxels, decreasing the spatial dimension of the voxel volume from  $\mathbb{R}^{D \times W \times H}$  to  $\mathbb{R}^{D' \times W' \times 1}$  where D, W, and H are the voxel volume's depth (x-axis), width (y-axis), and height (z-axis), respectively. This feature extraction process can be seen as a dimensionality reduction process from a 3D feature volume to a 2D bird eye view (BEV) feature map.

There are at least two compelling reasons to choose a 2D BEV feature map as the final feature map. Firstly, object proposals are predicted by applying a shared multilayer perceptron (shared-MLP) to every feature vector in the final feature space, either a 2D feature map or a 3D feature volume. Consequently, performing such a process in the 3D space requires a lot of computations, resulting in prohibitively slow inferences. Secondly, the 2D BEV feature map is an excellent choice of representation since objects of interest on the road (i.e., Cars, Pedestrians, and Cyclists) do not overlap in the BEV. This is not the case for the front-view map, where objects may fall in a line such that farther objects are occluded by nearer objects.

Although the 2D BEV feature map has desirable properties in terms of inference speed and objects separations, the resolution of information in the z-axis is significantly reduced. We hypothesize that the accuracy of 3D bounding box proposals generated from features with low-resolution z-axis information should be lower compared to when high-resolution z-axis information is available.

In order to provide high-resolution information z-axis information to the detection head while maintaining the 2D BEV feature map shape, we propose a new front-view (FV) to BEV feature augmentation via the multiview attention (MVA) module. The MVA module consists of learnable functions that robustly augment the BEV feature vectors with the FV feature vectors that have high-resolution z-axis information at similar

locations. The augmented BEV features enable the detection head to access high-resolution z-axis information, resulting in the improvement of the 3D detection performance of the network.

Our contributions can be summarized as follows:

- We propose a new framework for 3D object detection where the BEV feature map is augmented with its FV counterpart, thus enriching the BEV features with high-resolution z-axis information.
- We introduce a new module termed multiview attention (MVA) to augment BEV features with FV features. The MVA module is designed to be compatible with any 3D object detection networks that use the popular 3D voxel volume to 2D BEV feature map feature extraction process. Therefore, any system that uses such networks can experience improvement in its 3D detection performance with only small updates in its code, limiting the possibility of bug introduction on already-deployed systems.
- We conduct evaluations on four popular existing 3D object detection networks augmented with our MVA modules on the widely used KITTI dataset. Evaluation results show that our module successfully improves those networks. In addition, we perform an ablation study to figure out the effects of different architectures and hyperparameters in the MVA module.

The remaining of this paper is structured as follows: we discuss relevant prior works in Section II, introduce the structure of 3D object detection with our MVA module in Section III, discuss our experimental results in Section IV, and conclude the paper in Section V.

## II. RELATED WORKS

Various 3D object detection networks have been introduced in recent years. In general, the existing networks can be classified into three different categories based on their feature extraction approach, i.e., voxel-based, point-based, or hybrid approach.

### A. Point-based

In the point-based approaches, the feature extractor of the network is conditioned to learn directly from a set of raw points without discretizing the point cloud as in the voxel-based approach. To achieve this objective, networks such as Point-RCNN [2] utilizes PointNet++ [3] to learn the point-wise features. One advantage of using point-based feature extractors is that they preserve high-resolution spatial structure information of the points as there is no discretization process. However, point-based approaches often suffer from expensive computations as the number of points in a point cloud is often enormous. Therefore, point-based methods are often slower than their voxel-based counterparts.

### B. Voxel-based

In the voxel-based approaches, points in a point cloud are first discretized into non-overlapping cuboids of identical

sizes. The feature extractor is then conditioned to learn features for each voxel according to the points that lie inside. The voxel-based approach was first popularized by VoxelNet [4], where the features of each voxel are learned by a voxel feature encoder module. Meanwhile, SECOND [16] introduced the use of sparse 3D convolution to replace the computationally expensive 3D convolution operations, leading to a significant improvement in the inference speed.

Recent voxel-based methods try to improve the training scheme or the second stage network. For example, SE-SSD [5] introduced the teacher-student learning scheme to train the networks with soft targets, while Voxel-RCNN [13] proposed a fast query technique to obtain voxel features to be used in the second stage refinement.

### C. Hybrid-based

Hybrid approaches use both the point-based and voxel-based features to create rich feature vectors. PV-RCNN [14] first extract the voxel features and BEV features by utilizing the sparse 3D convolutions and 2D feature pyramid networks (FPN) [6], respectively. In the second stage, the network queries raw point features, voxel features, and BEV features of the object proposals to refine the bounding box predictions. Similarly, PVGNet [7] stacks together point-based features from the raw points, voxel-based features from the voxel feature volume, and grid features from the BEV feature map to create enriched feature vectors. BtcDet [17] on the other hand augments the raw point features and voxel features with occupancy features predicted by an auxiliary network.

## III. METHODOLOGY

In this section, the problem of 3D object detection in a point cloud will first be defined. Following that, we explain the common structure of voxel-based 3D object detection networks that we used in our experiments. Last, we describe in detail how our MSA module works.

### A. Problem Definition

A Lidar point cloud  $\mathbf{P}$  is defined as a set of  $N^p$  points,  $\mathbf{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_{N^p}\}$ . A point  $\mathbf{p}_i$  can be described as a feature vector  $\mathbf{p}_i = [x_i, y_i, z_i, \mathbf{f}_i^{raw}] \in \mathbb{R}^{3+C^{raw}}$  with  $(x_i, y_i, z_i)$  as the 3D coordinates of the point and  $\mathbf{f}_i^{raw}$  is an additional feature vector of  $C^{raw}$  dimension that describes the point, such as intensity, reflectivity, or ring information. An object  $m$  in a point cloud can be described by its class  $s_m$  and bounding box  $\mathbf{b}_m$ . In most dataset,  $\mathbf{b}_m$  is constructed by using the center coordinates of the object  $(x_m, y_m, z_m)$ , its dimension  $(d_m, w_m, h_m)$ , and its yaw angle  $\theta_m$ .

Using previously described notations, the basic objective of the 3D object detection with a deep neural network is to find the best parameters of a learnable function  $\Phi$  that is conditioned on the point cloud  $\mathbf{P}$  to predict a set of classes  $\mathbf{S}$  and bounding boxes  $\mathbf{B}$  of objects that are present in the scene. The optimization objective becomes,

$$\Theta_{MLE} = \underset{\Theta}{\operatorname{argmax}}(\mathcal{P}(\mathbf{S}, \mathbf{B}|\mathbf{P})), \quad (1)$$

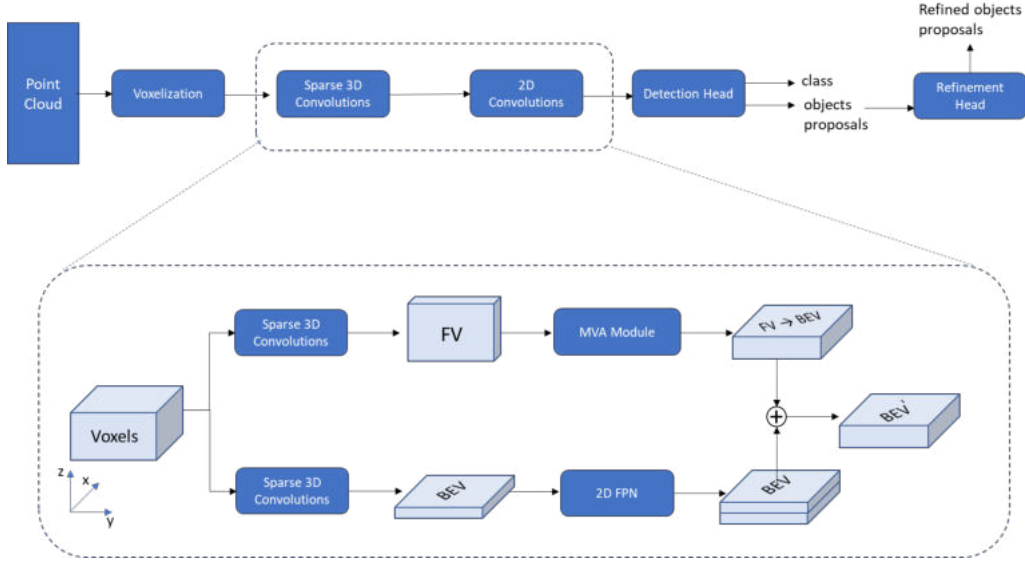


Fig. 1. An overview of the general voxel-based 3D object detection networks with our proposed MVA module. We modify the common sequence of sparse 3D convolutions and 2D convolutions into a two-branch sparse 3D convolutions, 2D convolutions (FPN), and an MVA module.

and during inference, the learnable function  $\Phi$  will produce outputs,

$$\Phi(\mathcal{P}) = \{(\hat{s}_1, \hat{b}_1), \dots, (\hat{s}_{M^P}, \hat{b}_{M^P})\}. \quad (2)$$

Note that the number of predicted objects  $M^P$  is not necessarily equal to the number of actual objects (ground truths)  $M$  that are present in the scene.

### B. Voxel-based Framework for 3D Object Detection

In this work, we design our MVA module to be compatible with any 3D object detection network that uses the popular voxel-based feature extractor. Figure 1 shows the overall structure of a general 3D object detection network with the addition of our MVA module. Traditional voxel-based 3D object detection networks generally start with a voxelization module, followed by 3D sparse convolutions and 2D convolutions, and finally a detection head that produces object proposals. In the case of two-stage detection network, object proposals from the first stage are used by the refinement head to refine the bounding box proposals.

**Voxelization** As shown in Figure 1, a 3D object detection network that uses voxel-based framework will first discretize the point cloud  $\mathcal{P}$  into non-overlapping voxels of equal sizes which will result in a set of non-empty voxels  $\mathcal{V}^{raw} = \{\mathbf{v}_1^{raw}, \dots, \mathbf{v}_{N^v}^{raw} | \mathbf{v}_i^{raw} \in \mathbb{R}^{K \times (3+C_{raw})}\}$ , where  $N^v$  is the number of non-empty voxels and  $K$  is the predetermined number of points in a voxel. As the original number of points in a voxel,  $K_0$ , varies, we apply zero-padding to create "fake" points if  $K_0 < K$  and random sampling if  $K_0 > K$ . All non-zero points that lie in the voxel  $\mathbf{v}_i^{raw}$  will be averaged to create a single feature vector so that the voxels become  $\mathcal{V}^{mean} = \{\mathbf{f}_1^{mean}, \dots, \mathbf{f}_{N^v}^{mean} | \mathbf{f}_i^{mean} \in \mathbb{R}^{3+C_{raw}}\}$ .

The non-zero averaging is used to reduce the effect of zero-padding towards feature vectors in the subsequent layers. If the

voxels contain feature vectors from "fake" points, then an all-zero feature vector from a "fake" point may be transformed into a feature vector with non-zero values by any affine transformation that we apply in subsequent layers. Such non-zero feature vectors coming from non-existent points may be considered as noises to our network. Performing non-zero averaging guarantees that the mean feature vectors always come from existing points. We can maintain the representativeness of the mean feature vectors by setting the voxel size small enough such that there are only small variances in the raw feature vectors of the points inside each voxel.

**Two-way Sparse 3D Backbone** Convolution in the 3D space is computationally expensive, especially when the operation is performed on a large volume. To speed up the process of 3D convolution, we leverage the fact that point cloud data are often sparse and apply the widely used sparse and submanifold 3D convolutions [11] that operate only on the non-zero elements of the feature volume.

We define the spatial shape of a voxel volume  $\mathbf{V}$  as the number of voxels in each axis of the 3D coordinates that are required to cover the region of interest in the point cloud. For input voxels  $\mathcal{V}^{mean}$  with a spatial shape of  $D^{in} \times W^{in} \times H^{in}$ , the sparse 3D convolutional backbone will produce two 2D feature maps, FV and BEV. In our experiments, the default spatial shape of the FV is  $1 \times \frac{W^{in}}{8} \times \frac{H^{in}}{2}$ , while the default spatial shape of the BEV is  $\frac{D^{in}}{8} \times \frac{W^{in}}{8} \times 1$ . To create the two maps, we apply two branches of sparse 3D convolutional blocks, as shown in Figure 1. The upper branch is responsible for the FV feature map, while the lower branch is responsible for the BEV feature map.

The FV feature map has high-resolution in the z-axis, the BEV feature map has high-resolution in the x-axis, and both feature maps have the same dimension in the y-axis. Therefore, combining FV features to the BEV features based on their

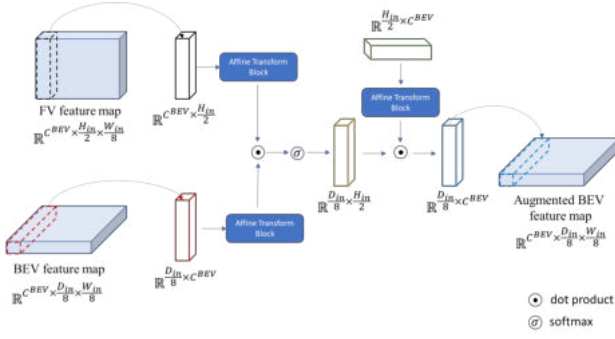


Fig. 2. MVA with Dot Product Attention

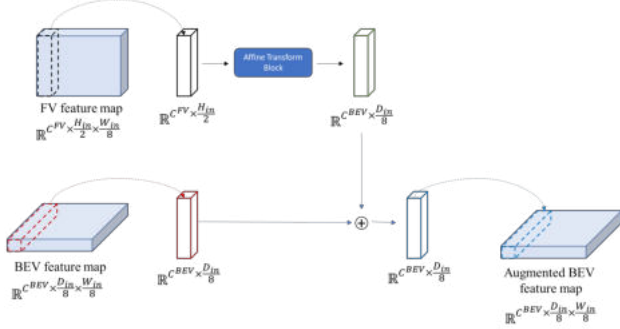


Fig. 3. MVA with Affine Transformation

location on the y-axis should yield meaningful features that contain high-resolution z-axis information while still maintaining the BEV shape. The combination procedure is explained in detail in the Multiview Attention Module subsection.

**2D Convolutional Backbone, Detection Head, and Refinement Head** After obtaining both FV feature map and BEV feature map via two-way sparse 3D convolutions, the BEV feature map is further processed with a 2D feature pyramid network (FPN) [6] as in recent 3D object detection networks [13]–[16]. The 2D FPN enables the network to capture global structural relationships between the BEV features. The FPN output of BEV feature map is subsequently augmented with the FV feature map by leveraging our proposed MVA module. The details of this combination process are explained in the next subsection. Given the augmented BEV feature map from the MVA output, the detection head performs  $1 \times 1$  convolutions on each feature vector  $f_i^{BEV}$  on the map to predict its class,  $s_i$  and bounding box proposal  $b_i$ . In the case of two stage detectors, the bounding box proposals are further refined by the refinement head by leveraging features from different sources such as raw points, voxels, and BEV map.

### C. Multiview Attention Module

An object in the 3D space should occupy the same y-axis coordinate in both FV and BEV. In other words, FV and BEV features that lie on the same y-axis coordinate should represent the same objects and environments. We leverage this fact to augment the BEV features with the FV features. Given an FV feature map  $f^{FV}$  of size  $H^{out} \times W^{out} \times C^{out}$  and a

BEV feature map  $f^{BEV}$  of size  $D^{out} \times W^{out} \times C^{out}$ , the MSA module takes the features residing on the  $i$ -th index of the y-axis,  $f_i^{FV} \in \mathbb{R}^{H^{out} \times C^{out}}$  and  $f_i^{BEV} \in \mathbb{R}^{D^{out} \times C^{out}}$ , and combine the two features together to make the augmented BEV features  $f_i^{BEV'} \in \mathbb{R}^{D^{out} \times C^{out}}$ . We provide two ways of combining the features, via dot-product attention mechanism or via a single affine transform block applied to the FV features.

**MVA with Dot Product Attention** The process of constructing the augmented BEV features  $f_i^{BEV'}$ , shown in Figure 2, can be described as transforming a sequence of feature vectors  $f_i^{FV}$  of length  $H^{out}$  into a new sequence  $f_i^{FV'}$  of length  $D^{out}$ . The transformed sequence is combined with another sequence  $f_i^{BEV}$  of length  $D^{out}$  via an aggregating function such as a summation. For the dot-product attention, we design the dimension size of  $f_i^{FV}$  channels to be the same as  $f_i^{BEV}$  channels,  $C^{BEV}$ .

The dot product attention mechanism has been widely used for sequence-to-sequence transformation. In recent years, the multihead variant of the dot product attention mechanism was popularized by [8] in the natural language processing domain and [9] in the computer vision domain. In this work, we utilize the multihead attention mechanism, defined as,

$$MHA(Q, K, V) = \text{Concat}(h_1, \dots, h_{N^h})W^{out}, \quad (3)$$

where,

$$h_k = \text{Att}(f_i^{BEV} W_k^Q, f_i^{FV} W_k^K, f_i^{FV} W_k^V), \quad (4)$$

$$\text{Att}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (5)$$

$d_k = C^{BEV}/N^h$ , and  $W$ s are learnable weight matrices.

Intuitively, the attention mechanism figures out the importance of each element in  $f_i^{FV}$  with regard to  $f_i^{BEV}$  in the form of a weight matrix via softmax and scaled-correlation of the two. The weight matrix is then used to construct a new sequence  $f_i^{FV'}$  from  $f_i^{FV}$  values but in the shape of  $f_i^{BEV}$ . Combining the sequence of feature vectors  $f_i^{FV'}$  with  $f_i^{BEV}$  via an aggregation function, in our case a summation, will result in an augmented BEV feature vectors  $f_i^{BEV'}$ . The augmented BEV feature map provides a way for the detection head to access high-resolution z-axis information that is not available with only a BEV feature map.

**MVA with Affine Transform** The dot-product attention mechanism requires at least three different affine transform blocks per head and requires large memory space as observed in many transformer-based architectures. We provide an alternative solution where we only use a single affine transform block without dot product operation between the sequence of feature vectors, as shown in Figure 3. In this case,  $f_i^{FV}$  is directly transformed into  $f_i^{FV'}$ , which has the same shape as  $f_i^{BEV}$ , by a single layer of affine transformation. This mechanism can be seen as a location-based attention introduced in [18]. This alternative is more efficient compared

TABLE I

EVALUATION RESULTS OF VARIOUS 3D OBJECT DETECTION NETWORKS WITH AND WITHOUT THE PROPOSED MVA MODULE ON THE KITTI *val* SET. THE VALUES ARE FOR %AP WITH 11 SAMPLING POINTS. FOR NETWORKS WITH MVA, THE SECOND ROW INDICATES THE PERFORMANCE DIFFERENCE COMPARED TO THE ORIGINAL NETWORK.

	Car			Pedestrian			Cyclist			Mean
	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard	
Voxel-RCNN [13]	89.21	83.41	78.60	67.19	60.65	55.16	85.73	72.21	68.46	73.40
Voxel-RCNN with MVA	89.77	84.06	78.99	67.02	61.25	55.41	86.28	72.42	68.51	73.74
	+0.56	+0.65	+0.39	-0.17	+0.60	+0.25	+0.55	+0.21	+0.05	+0.34
PV-RCNN [14]	89.33	83.61	78.71	63.10	54.82	51.77	86.06	69.48	64.54	71.27
PV-RCNN with MVA	89.17	84.58	78.66	65.44	57.97	53.80	86.36	72.68	69.25	73.10
	-0.16	+0.97	-0.05	+2.34	+3.15	+2.03	+0.30	+3.20	+4.71	+1.83
PV-RCNN++ [15]	88.88	79.04	78.26	64.00	59.40	54.49	86.76	66.94	65.69	71.50
PV-RCNN++ with MVA	89.06	83.56	78.35	65.14	61.60	55.96	86.84	67.31	65.50	72.59
	+0.18	+4.52	+0.09	+1.14	+2.20	+1.47	+0.08	+0.37	-0.19	+1.09
SECOND [16]	88.61	78.62	77.21	56.54	52.98	47.73	80.58	67.13	63.10	68.06
SECOND with MVA	88.93	79.11	77.88	61.75	55.92	49.99	85.57	70.44	64.86	70.49
	+0.32	+0.49	+0.67	5.21	+2.94	+2.26	+4.99	+3.31	+1.76	+2.43

TABLE II

MEAN OF THE %AP DIFFERENCE BETWEEN NETWORKS WITH MVA AND THEIR ORIGINAL NETWORK

Car			Pedestrian			Cyclist		
Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
+0.23	+1.65	+0.27	+2.13	+2.22	+1.50	+1.48	+1.77	+1.58
+0.72			+1.95			+1.61		

to the dot-product attention mechanism, however it has lower model capacity. We show the effects between utilizing dot-product attention and affine transform in the ablation study.

#### IV. EXPERIMENTAL SETUPS AND RESULTS

In this section, we will describe the dataset that we used to conduct our experiments and list all implementation details that are needed to reproduce the experimental results. We then show and discuss our evaluation results on the validation set.

##### A. Dataset

In this work, we utilize the widely used KITTI dataset [1] for 3D object detection with Lidar point cloud. The KITTI dataset is one of the earliest and most popular datasets for 3D object detection in Lidar point cloud. The dataset contains 7,481 Lidar frames for training and 7,518 Lidar frames for testing. There are three major classes in the dataset: Car, Pedestrian, and Cyclist. For every object sample of each class, we can assign a difficulty level, i.e. easy, moderate, or hard, depending on the level of occlusions of said object.

As the annotation for test set is not publicly available, we split the training data into *train* split with 3,712 frames and *val* split with 3,769 frames. We follow the commonly used protocol for splitting the KITTI training data [10] such that the frames in the *train* split and *val* split originate from different sequences. We perform extensive evaluations on the *val* set, not the *test* set, as per KITTI’s official rules for works that are a modification of existing techniques.

##### B. Implementation Details

We set the number of epochs as 80, batch size as 4, and use Adam [12] as the optimizer. The one cycle scheduler is

used to control the learning rate, where the maximum learning rate, 0.01, is set to be achieved at about halfway through the training. For all networks, we define the voxel size as  $5cm \times 5cm \times 10cm$  for the x, y, and z axis, respectively, and limit the detection range to be  $0 \sim 70m$  for the x-axis,  $-40 \sim 40m$  for the y-axis, and  $-3 \sim 1m$  for the z-axis relative to the position of the Lidar sensor on the ego car.

To make fair comparisons between networks with MVA and their original counterparts, we retrain the original networks using the publicly available code with the same hyperparameters as the ones with MVA modification. We also refer to the original publication of each network for the loss functions.

##### C. Main Results

Table I shows the evaluation results of various voxel-based 3D object detection networks with and without the proposed MVA module on the *val* set of the KITTI dataset. We choose to use the dot-product MVA module with 8 heads following the ablation results, which is explained in details in the next subsection. As shown on the table, our MVA module improves the 3D detection performance of voxel-based 3D object detection networks in general. More specifically, our module improve the detection performance of all four networks by 0.72% for car class, 1.95% for pedestrian class, and 1.61% for cyclist class, as shown in Table II.

It is interesting to note that the improvements in both pedestrian and cyclist classes are higher compared to the improvement in the car class. This phenomena can be explained by the bounding box dimensions of the objects on the road. Cars in general have similar shapes and sizes as car manufacturers have to follow existing regulations, therefore, their bounding box dimensions are similar for all samples in the dataset. On

TABLE III  
COMPARISON BETWEEN DOT-PRODUCT AND AFFINE TRANSFORMATION MVA MODULES WITH VOXEL-RCNN-MVA

	Car			Pedestrian			Cyclist			Overall Inference Time (ms)
	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard	
Dot Product	<b>89.77</b>	<b>84.06</b>	<b>78.99</b>	<b>67.02</b>	<b>61.25</b>	<b>55.41</b>	<b>86.28</b>	<b>72.42</b>	<b>68.51</b>	33.6
Affine Transform	89.60	83.97	78.88	65.56	60.05	54.97	84.95	72.25	68.22	<b>30.4</b>

TABLE IV  
EFFECTS OF THE NUMBER OF HEADS IN THE DOT-PRODUCT MVA MODULE WITH VOXEL-RCNN-MVA

Number of Heads	Car			Pedestrian			Cyclist			Overall Inference Time (ms)
	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard	
4	89.65	83.96	78.91	65.05	58.70	52.89	85.90	72.38	68.48	<b>33.1</b>
8	<b>89.77</b>	<b>84.06</b>	<b>78.99</b>	<b>67.02</b>	<b>61.25</b>	<b>55.41</b>	86.28	72.42	68.51	33.6
16	89.37	83.72	78.75	66.14	60.06	55.22	<b>92.07</b>	<b>73.25</b>	<b>69.30</b>	33.8

the contrary, pedestrians and cyclists have varying bounding box dimensions, particularly in the z-axis due to the variance in people’s heights. As such, the importance of obtaining high-resolution information in the z-axis from the FV features is far greater for predicting bounding boxes of pedestrians and cyclists compared to bounding boxes of cars.

#### D. Ablation Study

We conduct ablations on the Voxel-RCNN network for two aspects: the differences between the two MVA modules and the number of heads in the dot-product MVA module. As shown in Table III, the dot-product MVA module performs better compared to the affine transformation MVA module in terms of %AP. However, this method has a slower overall inference time of about 3.2ms (10.5%). This phenomena is expected, and has been previously explained in Subsection 3.C.

Another hyperparameters that can affect networks performance is the number of heads in the dot-product MVA module. Table IV shows the effects of number of heads on the network performance. As expected, the processing time gets slower when the number of heads is increased. However, higher number of heads does not necessarily means the network has a better performance in terms of %AP. We find that the network performs optimally when we set the number of heads as 8.

Note that the original Voxel-RCNN has an overall inference time of 29.6 ms, meaning that utilizing the two-way 3D sparse convolutions and either the affine transform MVA or dot-product MVA only adds 0.8 ms (2.7%) or 4 ms (13.5%) to the overall inference time, respectively.

#### V. CONCLUSIONS

We present the Multiview Attention Module (MVA) for 3D object detection that augments a bird-eye-view (BEV) feature map with features from a front-view (FV) feature map. The feature augmentation process enable the detection head to obtain high-resolution information in the z-axis while still allowing object predictions to be made on the BEV. As such, we successfully maintain reasonable inference speed while simultaneously improve the 3D detection performance. Based on the experimental results on the KITTI *val* set, our MVA module successfully improve the detection performance of

all four voxel-based networks that we evaluated, proving the effectiveness and the adaptability of the MVA module.

#### REFERENCES

- [1] A. Geiger, P. Lenz, R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," 2012 IEEE conference on computer vision and pattern recognition, 2012, pp. 3354-3361
- [2] S. Shi, X. Wang, H. Li, "Pointcnn: 3d object proposal generation and detection from point cloud," Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 770-779
- [3] C.R. Qi, L. Yi, H. Su, L.J. Guibas, "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space," Advances in Neural Information Processing Systems 30, 2017.
- [4] Y. Zhou, O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4490-4499
- [5] W. Zheng, W. Tang, L. Jiang, C.W. Fu, "SE-SSD: Self-Ensembling Single-Stage Object Detector From Point Cloud," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 14494-14503
- [6] T.Y. Lin, et al., "Feature pyramid networks for object detection," Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2117-2125
- [7] Z. Miao, et al., "PVGNet: A Bottom-Up One-Stage 3D Object Detector With Integrated Multi-Level Features," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 3279-3288
- [8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, "Attention is all you need," Advances in neural information processing systems, 2017, pp. 5998-6008
- [9] A. Dosovitskiy, et al., "An image is worth 16x16 words: Transformers for image recognition at scale," arXiv preprint arXiv:2010.11929, 2020
- [10] X. Chen, et al., "3d object proposals for accurate object class detection," Advances in Neural Information Processing Systems, 2015, pp. 424-432
- [11] B. Graham, L. van der Maaten, "Submanifold sparse convolutional networks," arXiv preprint arXiv:1706.01307, 2017
- [12] D. P. Kingma, J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014
- [13] J. Deng, et al., "Voxel R-CNN: Towards High Performance Voxel-based 3D Object Detection," arXiv preprint arXiv:2012.15712 (2020)
- [14] S. Shi et al., "PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 10526-10535
- [15] S. Shi, et al., "PV-RCNN++: Point-Voxel Feature Set Abstraction With Local Vector Representation for 3D Object Detection," arXiv preprint arXiv:2102.00463, 2021
- [16] Y. Yan, Y. Mao, B. Li, "Second: Sparsely embedded convolutional detection," Sensors 18, no. 10, 2018
- [17] Q. Xu, Y. Zhong, U. Neumann, "Behind the Curtain: Learning Occluded Shapes for 3D Object Detection," arXiv preprint arXiv:2112.02205, 2021
- [18] M. T. Luong, H. Pham, C. D. Manning, "Effective approaches to attention-based neural machine translation," arXiv preprint arXiv:1508.04025, 2015



# Multi-scale synergy approach for real-time semantic segmentation

Quyen Van Toan

*School of Electronic and Electrical Engineering  
Kyungpook National University  
Daegu, Korea  
yersin@knu.ac.kr*

Min Young Kim

*School of Electronic and Electrical Engineering  
Kyungpook National University  
Daegu, Korea  
minykim@knu.ac.kr*

**Abstract**—In deep convolution neural network based models for semantic segmentation, diverse receptive fields improve the performance by capturing disparate context information. Multi-scale inference is good for both thin and large objects. However, the final result is not optimal through averaging or Max pooling combination. In this paper, we propose an approach to take advantage of multi-scale predictions. Our uncertain-pixels part discovers the worse prediction of a low scale and chooses the complement from a high scale. The final output is effectively merged from two scales. We validate our proposed model with a series of experiments on different datasets. The results achieve the accuracy and speed for real-time semantic segmentation. On Cityscapes dataset, our network achieves 76.3 % mIoU at 50 FPS, and on Mapillary, 42.6 % mIoU.

**Index Terms**—Multi-scale, semantic segmentation, real time.

## I. INTRODUCTION

In the age of artificial intelligence (AI) advances and the high qualities of modern computers and cameras, several applications in the fields of robot vision and self-driving cars have remarkable developments. Semantic segmentation plays an important role in the input information of the autonomous system. It is utilized to recognize and understand what is in the image at pixel levels. This method assigns labels to specific regions of an image. The performance has been directly affected by accuracy segmentation and inference speed.

In the semantic segmentation revolution, we briefly review the proposed approaches after the deep learning emergence. The Fully Convolution Networks (FCNs) [1] pioneer the way to build the deeper structures of models. The method extracts image features through hierarchical convolution layers after utilizing the last one to predict the segmentation. At that time, the output results have reached an excellent benchmark and significantly impacted the field. Characteristics of the final layer are deep features and low resolutions. To overcome low resolution drawback, DeeplabV3 [2] enrich spatial contextual information by deploying dilated convolutions with flexible rates. Another problem of FCNs, the gradient values are gradually inclined to zeros, so the process suffers the loss of importation details. Skip connections are introduced and provide an alternative path for the gradient by adding information of lower layers to higher layers [3]. The combination of multiple skip connections and multi-dilated convolutions enhanced accuracy [4].

Throughout the improvement of semantic segmentation, novel methods concentrate on enhancing the accuracy and speed inference. Early, they address the problem of segmentation in class categories. They propose region-based object detectors with scanning-windows part models and global appearances to obtain quality object segmentation [5], a combination of regions and convolution neural network (CNN) to boost the performance [6], room-out the regions to obtain a higher resolution in [7], and exploit a multi-region to rich representation [8]. The accuracy is gradually improved over time. Later, innovative approaches focus on real-time semantic segmentation with high accuracy. To attain real-time, we need to deal with spatial information for high accuracy as well. The rich spatial information is captured by a new design with a small stride [9], an effective fast attention with cosine modification [10], and incorporation between high-resolution global edge and low resolution [11]. In these ways, we can achieve 74.4 % at 72 FPS on Cityscapes dataset or 68.4 % with 105 FPS on COCO datasets. When features have high resolution, multi-scale inputs are proposed to capture diverse context information [12], [13].

In this paper, we propose multi-scale synergy approach for real-time semantic segmentation. We use two different-size inputs. One is remaining as a high scale, and the other is downsampling by 2 as a low scale. In order to leverage advances from both scales, we deploy the uncertain-pixel determination that detects the bad prediction of the low scale and effectively fuses the predictions from two scales. We achieve high results on Cityscapes 76.3 mIoU and 50.1 FPS, and on Mapillary 42.6 mIoU.

## II. RELATED WORKS

Semantic segmentation demands spatial information to resolve fine detail. When the resolution is high, it causes the receptive field to be shrunken. In this case, the receptive field is small, it is hard to cover the whole context of large objects. It leads the bad prediction for large objects. Many approaches are proposed a contextual combination from multiple scales. Multi-scale context can be obtained from various levels of pyramid pooling [14] or different rates of dilation [15]. In [16], author builds the architecture with two different-size inputs. The small scale is used to get contextual information and the

others are used for generating spatial detail. An image cascade network with multi-resolution branches is proposed in [17]. The score maps of multiple scales are combined by averaging or max-pooling methods to generate the final output. With the average method, each pixel value is the result of an average combination between a pair scale or all scales. It leads to the problem of combining the best predictions with poorer ones. Instead of selecting only one of N scales like Max-pooling, We propose a novel method to combine two scale inputs via the uncertain-pixel determination part. The uncertain-pixel part keeps the best prediction of the low scale and improves the uncertain parts through the high scale complement.

### III. METHOD

In this section, we introduce three main parts. The uncertain-pixels determination for low scale segmentation is described in section III-A. An overview of the proposed architecture is shown in section III-B. Lastly, we will represent the optimization function III-C

#### A. Uncertain-pixels determination for low scale segmentation

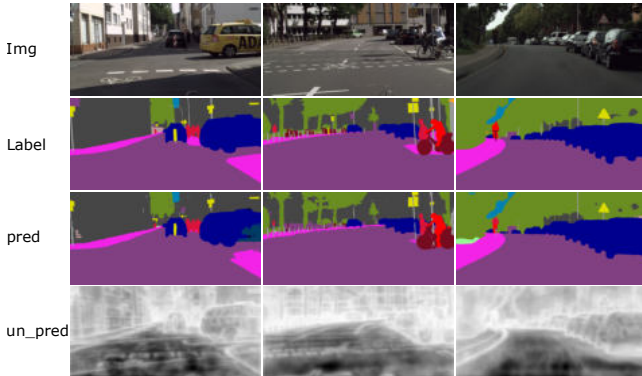


Fig. 1: Illustrate uncertain masks of the prediction. The rows are images, labels, prediction masks and uncertain-pixels masks of the prediction, top to bottom respectively. The uncertain pixels are pixels at prediction which have high probabilities of wrong labels. The white color indicate the uncertain areas.

In general, the semantic segmentation of large objects in the image such as buildings, sky, or trucks always have good predictions, contrast narrow objects like poles, fences, traffic lights, etc. will have low precision. Moreover, object’s boundaries are the most uncertain prediction. To achieve high accuracy, we intend to deal with the precision of narrow objects and boundaries.

In the networking, the trunk generates separately the final map for each scale, including  $N\_classes$  channels of the datasets. We subtract the chosen final segmentation by one to produce "1-segmentation layers" or uncertain layers. The pixel’s values of the uncertain layers are depended on those of the final segmentation layers. Pixels at the object’s body have one probability much higher than the others, but pixels at the boundaries will have at least two high probabilities

which are nearly equal. For clear understanding, we make a prediction step for an explanation. Fig. 1 includes images, labels, prediction masks, and uncertain-pixels masks. The final layers are predicted to generate prediction masks. The uncertain masks are utilized to determine which parts have of the prediction have high probabilities of wrong labels.

In our proposed method, the uncertain masks are generated from the lower scale. The low scale predicts good results with large objects, especially near the screen. The fourth row in Fig. 1 shows that the uncertain areas have a white color and locate at boundaries and far away from screens. The main function of this mask tweaks the high scale to more focus on the white color.

#### B. Proposed structure

Fig. 2 illustrates the proposed architecture, including three main components. Firstly, there are two scales as inputs. The low scale is 0.5x and the other is 1.0x. Second, we utilized the model Deeplabv3+ with ResNet-50 as trunk. The shared weighted model employs for generating score maps for all scales. Lastly, the combination of two scales is demonstrated in the dash-line box which is executed in pixel-wise levels.

We visualize the combination part for obviously understanding, depicted in the red box.  $N\_classes$  channels, generated by trunk, are passed through the Max function to predict the segmentation label for each location. As mentioned above, the segmentation of a low scale is focused on the near screen. It achieves good precision with large objects such as roads, cars, but it gets worse for small objects, shown in the red box of Fig. 2. Inversely, the segmentation label of a high scale has a better result at a far screen. The primary hinder of the high scale is not covering the whole necessary context of large objects, especially large objects near the screen. Our target is to take advantage of both scales. Instead of utilizing max pooling or average pooling methods for the combination, we deploy an uncertain-pixels determination for the network. The uncertain mask is produced from a low scale. This mask detects precarious labels of the low scale, assigned in white color in the mask. The white color appears at the far screen, small objects, and particularly at the object’s boundaries. The white color indicates attentive locations and the others are non-attentive. The uncertain mask will select which parts of the high scale mainly contribute to the final output. The contributions by multiplying between the uncertain mask and the high scale are object’s boundaries, small objects, and far screen objects. conclusively, the advance of the uncertain mask leverages all best predictions and almost neglects the worst prediction of the high scale.

In our proposal, the weight output is excellently merged by two scales. The objects on the far screen are mainly contributed from both scales. The final segmentation map is calculated as equation 1.

$$S = U(S_{lo}) + S_{hi} * U(1 - S_{lo}) \quad (1)$$

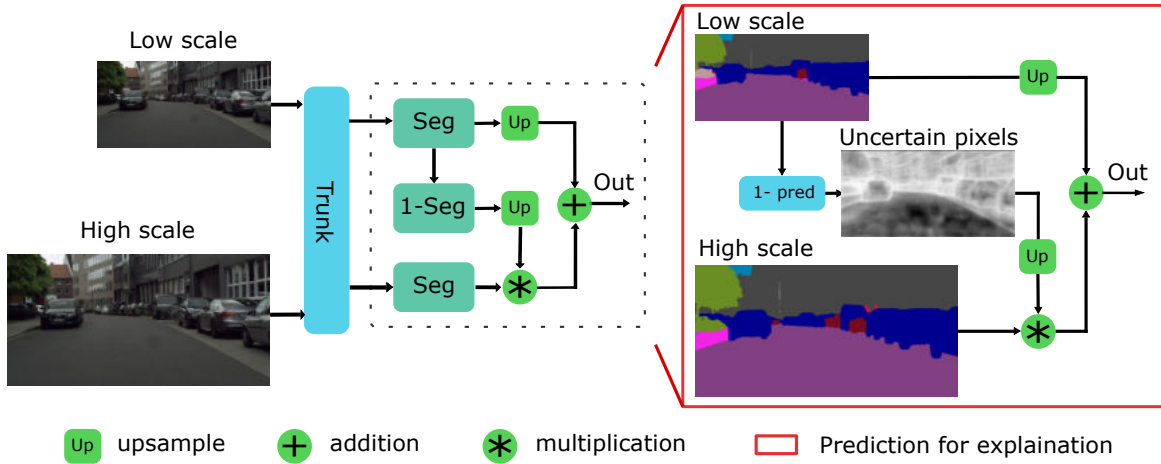


Fig. 2: Structural diagram of the proposed multi-scale synergy approach. The red box is prediction step for explanation. In the uncertain-pixel mask, the white color indicates a high value, and the dark color is small value.

where  $\mathcal{S}$  and  $\mathcal{U}$  are the semantic segmentation and the bilinear upsampling operation, respectively. Two input scales, "lo" denote as low and "hi" is high scale.

### C. Optimization

We use Stochastic Gradient Descent (SGD) for our optimizer. SGD performs frequent updates with a high variance that cause the objective function to fluctuate heavily. SGD in contrast performs a parameter update for each training example  $x_i$  and label  $y_i$ :

$$\theta = \theta - \eta \Delta_{\theta} \mathcal{J}(\theta; x_i; y_i) \quad (2)$$

where  $\theta$  is stochastic gradient decent,  $\Delta_{\theta}$  and  $\mathcal{J}(\theta)$  are gradient of the objective function and an objective function

For loss computation, we utilize a Cross-Entropy to calculate the losses. The Cross-Entropy is measured how accurate the model is in predicting the data shown in equation 3.

$$L = \frac{1}{N} \sum_{i=1}^N (-y_i \log(p_i)) \quad (3)$$

where  $L$  and  $N$  are the loss and the size of dataset, respectively.  $p_i$  denotes the segmentation predicted probability, and  $y_i$  is true labels.

The last formula, intersection over union (IoU) is an evaluation metric used to measure the accuracy of an object detector on a particular dataset shown in equation below.

$$IoU = \frac{|Target \cap Prediction|}{|Target \cup Prediction|} \quad (4)$$

## IV. EXPERIMENTS

In experiments, we use the following standard measures: mini-batch stochastic gradient descent (SGD) for the optimizer, cross-entropy loss function, intersection of union (IoU), and fame per second (FPS). We trained the model on an Nvidia Titan X with 12GB of GDDR5X memory for the Cityscapes dataset, and a GeForce RTX 3090 with 24GB of G6X memory

for Mapillary. We train for 150 epochs with a batch size of 2 per GPU, a momentum of 0.9, weight decay of  $5e-4$ , and a batch size of 2 per GPU. With an initial learning rate of 0.01, we use a polynomial learning rate.

### A. Cityscapes dataset

Cityscapes dataset consists of 5,000 images with 19 semantic classes captured from 50 different countries. The resolution is 2048 x 1024 pixels. The whole dataset is partitioned into three sets training, validation, and test. There are 2,979 images in the training set, 500 images in the validation set, and 1,525 images in the test set.

In Table I, we demonstrate the class-accuracy comparison. The performance displays a reasonable balance between thin objects and large objects compared to Previous SOTA methods. Our proposed model achieves 76.3 % mIoU. In Table II, the results show that our model outperforms existing approaches for segmentation accuracy while still achieving real-time implementing efficiency. Although FANet [10] has a faster speed than ours, our proposal gets 1.3 mIoU greater than them in an accuracy improvement. conclusively, Our approach performs effectively in terms of accuracy and speed on Cityscapes dataset. Qualitative results are visualized in Fig. 3a

### B. Mapillary Vista

Mapillary Vista is a large dataset collected from city streets around the world. It consists of 25,000 images with 66 object categories, and the images have various resolutions. Due to large number of classes and high resolution, images are cropped to 2177x1632 pixels to reduce the computation and memory requirement.

In this section, we evaluate and compare the segmentation accuracy with other methods such as AGLNet [25], DABNet [21], RGPNet [27]. With 66 classes, our approach still catches 42.6 mIoU for segmentation accuracy. Table III show that our method surpasses other approaches, particularly DABNet. Finally, Qualitative results are visualized in Fig. 3b

Method	Road	swalk	build	wall	fence	pole	tlight	tsign	veg.	terr	sky	pers	rider	car	truck	bus	train	mcle	bicle	mIoU
CGNet [18]	97.7	81.0	89.8	42.5	48.0	56.2	59.8	65.3	91.4	68.2	94.2	76.8	57.1	92.8	50.8	60.1	51.8	47.3	61.7	68.0
EDANet [19]	97.8	80.6	89.5	42.0	46.0	52.3	59.8	65.0	91.4	68.7	93.6	75.7	54.3	92.4	40.9	58.7	56.0	50.4	64.0	67.3
ESPNet [20]	97.3	78.6	88.8	43.5	42.1	49.3	52.6	60.0	90.5	66.8	93.3	72.9	53.1	91.8	53.0	65.9	53.2	44.2	59.9	66.2
DABNet [21]	97.8	80.7	90.2	47.9	48.1	56.4	61.8	67.0	92.0	69.5	94.3	80.3	59.2	93.7	46.0	57.1	35.0	50.4	66.8	68.1
CFPNet [22]	97.8	81.4	90.5	46.4	50.6	56.4	61.5	67.7	92.1	68.9	94.3	80.4	60.7	93.9	51.4	68.0	50.8	51.2	67.7	70.1
RELAXNet [23]	98.94	84.9	92.2	57.2	54.8	64.3	70.6	74.0	93.0	71.8	94.8	83.7	64.4	95.1	58.6	72.7	58.2	59.9	71.8	74.8
DSANet [24]	96.8	78.5	91.2	50.5	50.8	59.4	64.0	71.7	92.6	70.0	94.5	81.8	61.9	92.9	56.1	75.6	50.6	50.9	66.8	71.4
FANet [10]	97.9	83.3	91.6	55.5	55.1	60.3	66.2	74.9	91.7	61.8	94.7	78.5	58.1	94.1	76.8	85.1	74.5	50.7	73.9	75.0
AGLNe [25]	97.8	81.0	91.0	51.3	50.6	58.3	63.0	68.5	92.3	71.3	94.2	80.1	59.6	93.8	48.4	68.1	42.1	52.4	67.8	70.1
<b>Ours</b>	97.9	83.9	92.0	60.1	60.2	59.4	63.6	74.6	91.8	61.9	94.3	78.6	59.6	94.2	80.1	87.1	75.3	62.56	73.3	76.3

TABLE I: Class-accuracy comparison on the Cityscapes dataset. List of classes from left to right: road, side walk, building, wall, fence, pole, traffic light, traffic sign, vegetation, terrain, sky, person, rider, car, truck, bus, train, motorcycle, and bicycle.

Method	Resolution	mIoU	FPS
AGLNet [25]	512 × 1024	70.1	52
FANet [10]	1024×2048	75.0	72
ICNet [17]	1024×2048	67.7	38
BiseNet [9]	768×1536	74.8	47
SwiftNet [26]	1024×2048	75.4	40
<b>Ours</b>	1024×2048	76.3	50

TABLE II: Accuracy and speed comparison of proposed method against other SOTA methods on Cityscapes.

Method	Resolution	mIoU
AGLNet [25]	1024×2048	30.7
DABNet [21]	1024×2048	29.6
RGPNet [27]	1024×2048	41.7
<b>Ours</b>	2177×1632	42.6

TABLE III: Accuracy comparison of proposed method against other SOTA methods on Mapillary Vista.

## V. CONCLUSION

In this work, the multiple scales help capture the contextual information at different receptive fields. We propose an uncertain-pixels determination which brings an effective way to combine multiple scales at element-wise levels. Our approach shows the improvement in segmentation accuracy while still achieving real-time implementing efficiency. Due to the hardware limitation, we just use a lightweight network for our method. In the future, we will implement on a heavy network to enhance the accuracy.

## ACKNOWLEDGMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2021R1A6A1A03043144)

## REFERENCES

- [1] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.
- [2] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.
- [3] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1925–1934.
- [4] T. Yamashita, H. Furukawa, and H. Fujiyoshi, "Multiple skip connections of dilated convolution network for semantic segmentation," in *2018 25th IEEE international conference on image processing (ICIP)*. IEEE, 2018, pp. 1593–1597.
- [5] P. Arbeláez, B. Hariharan, C. Gu, S. Gupta, L. Bourdev, and J. Malik, "Semantic segmentation using regions and parts," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3378–3385.
- [6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.
- [7] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich, "Feedforward semantic segmentation with zoom-out features," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3376–3385.
- [8] S. Gidaris and N. Komodakis, "Object detection via a multi-region and semantic segmentation-aware cnn model," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1134–1142.
- [9] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 325–341.
- [10] P. Hu, F. Perazzi, F. C. Heilbron, O. Wang, Z. Lin, K. Saenko, and S. Sclaroff, "Real-time semantic segmentation with fast attention," *IEEE Robotics and Automation Letters*, vol. 6, no. 1, pp. 263–270, 2020.
- [11] H. Lyu, H. Fu, X. Hu, and L. Liu, "Esnet: Edge-based segmentation network for real-time semantic segmentation in traffic scenes," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 1855–1859.
- [12] W. Wang, S. Wang, Y. Li, and Y. Jin, "Adaptive multi-scale dual attention network for semantic segmentation," *Neurocomputing*, vol. 460, pp. 39–49, 2021.
- [13] A. Tao, K. Sapra, and B. Catanzaro, "Hierarchical multi-scale attention for semantic segmentation," *arXiv preprint arXiv:2005.10821*, 2020.
- [14] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890.
- [15] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.
- [16] R. P. Poudel, U. Bonde, S. Liwicki, and C. Zach, "Contextnet: Exploring context and detail for semantic segmentation in real-time," *arXiv preprint arXiv:1805.04554*, 2018.
- [17] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "Icnet for real-time semantic segmentation on high-resolution images," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 405–420.
- [18] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, "Erfnet: Efficient residual factorized convnet for real-time semantic segmentation,"

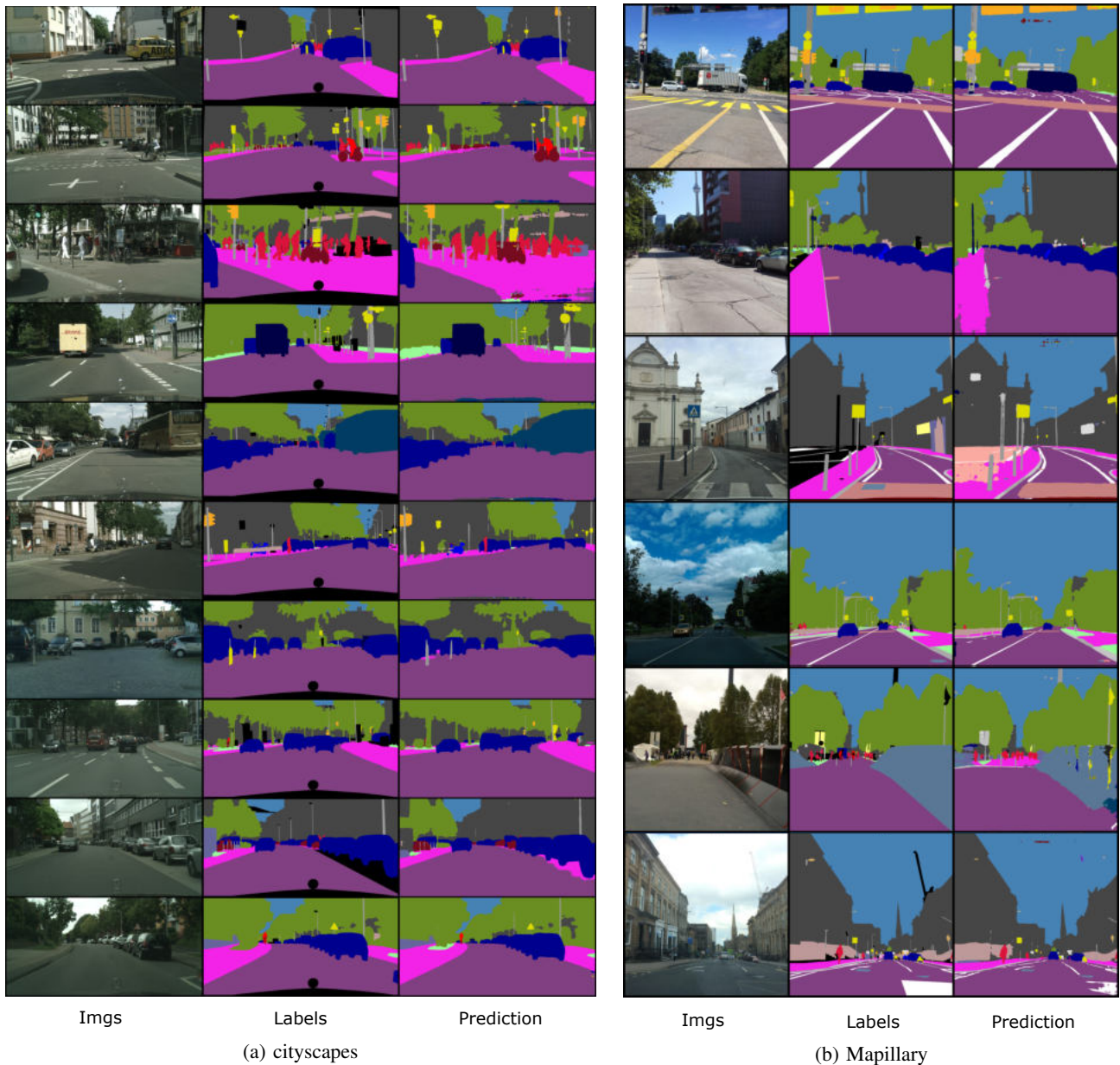


Fig. 3: Qualitative results of proposed approach on Cityscapes val set (left) and Mapillary Vista val set (right).

- IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 263–272, 2017.
- [19] S.-Y. Lo, H.-M. Hang, S.-W. Chan, and J.-J. Lin, “Efficient dense modules of asymmetric convolution for real-time semantic segmentation,” in *Proceedings of the ACM Multimedia Asia*, 2019, pp. 1–6.
- [20] S. Mehta, M. Rastegari, L. Shapiro, and H. Hajishirzi, “Espnetv2: A light-weight, power efficient, and general purpose convolutional neural network,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9190–9200.
- [21] G. Li, I. Yun, J. Kim, and J. Kim, “Dabnet: Depth-wise asymmetric bottleneck for real-time semantic segmentation,” *arXiv preprint arXiv:1907.11357*, 2019.
- [22] A. Lou and M. Loew, “Cfpnet: Channel-wise feature pyramid for real-time semantic segmentation,” *arXiv preprint arXiv:2103.12212*, 2021.
- [23] J. Liu, X. Xu, Y. Shi, C. Deng, and M. Shi, “Relaxnet: Residual efficient learning and attention expected fusion network for real-time semantic segmentation,” *Neurocomputing*, vol. 474, pp. 115–127, 2022.
- [24] M. A. Elhassan, C. Huang, C. Yang, and T. L. Munez, “Dsanet: Dilated spatial attention for real-time semantic segmentation in urban street scenes,” *Expert Systems with Applications*, vol. 183, p. 115090, 2021.
- [25] Q. Zhou, Y. Wang, Y. Fan, X. Wu, S. Zhang, B. Kang, and L. J. Latecki, “Aglnet: Towards real-time semantic segmentation of self-driving images via attention-guided lightweight network,” *Applied Soft Computing*, vol. 96, p. 106682, 2020.
- [26] M. Orsic, I. Kreso, P. Bevandic, and S. Segvic, “In defense of pre-trained imagenet architectures for real-time semantic segmentation of road-driving images,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 607–12 616.
- [27] E. Arani, S. Marzban, A. Pata, and B. Zonooz, “Rgpnnet: A real-time general purpose semantic segmentation,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 3009–3018.

# CIAFill: Lightweight and Fast Image Inpainting with Channel Independent Attention

1<sup>st</sup> Chung-Il Kim

Artificial Intelligence Research Center  
Korea Electronics Technology Institute  
Gyeonggi-do, Korea  
cilkim1@keti.re.kr

2<sup>nd</sup> Saim Shin

Artificial Intelligence Research Center  
Korea Electronics Technology Institute  
Gyeonggi-do, Korea  
sishin@keti.re.kr

3<sup>rd</sup> Han-Mu Park

Artificial Intelligence Research Center  
Korea Electronics Technology Institute  
Gyeonggi-do, Korea  
hanmu@keti.re.kr

**Abstract**—Image inpainting is a classic technique in computer vision research. The quality of image inpainting has improved significantly since the advent of convolutional neural networks. However, this approach generally results in blurry and semantically inconsistent reconstruction because of operating valid and invalid pixels with equal weight. The gated convolution computing feature attention is proposed to resolve this issue but this attention mechanism was less efficient and computationally expensive. This study proposed CIAFill that alleviates this problem using channel independent attention. This mechanism applied channel attention to each channel for activating valid channels and reduced the computational cost to the dimension of the channel. The proposed architecture included a channel attention generator and a channel attention projection PatchGAN that utilize the channel independent attention mechanism. This study proved that CIAFill could successfully train the inpainting model with 1/1600 smaller gating parameters than the earlier gated convolution-based study. CIAFill achieved comparable performance to other feature attention-based approaches in the experiments on CelebA-HQ and Places2 datasets.

**Index Terms**—inpainting, attention module, generative adversarial net

## I. INTRODUCTION

An image inpainting task involves reconstructing occluded or blank regions to make images plausible by referring to information in surrounding regions. This is a popular task in the field of computer vision and image processing [1]–[3], because corruptions by noise and occlusions frequently occur in real-world [4]–[7].

The performances of image inpainting have been significantly improved with the advent of deep learning technologies [8], so most recent image inpainting techniques are based on deep convolutional neural networks (CNN) [9]–[12]. CNN-based networks trained by massive data can reconstruct highly structured regions including complex semantics—faces, hands, and buildings—because the learning paradigm of CNN involves analyzing the pixel-wise data distribution from training data [13]–[15]. However, CNN-based inpaintings commonly generate implausible results, such as blurry texture, apparent color discrepancy, and abnormal edges around erased

This work was supported by Institute of Information and communications Technology Planning and evaluation (IITP) grant funded by the Korea government (MSIT) (2021-0-00537, Visual common sense through self-supervised learning for restoration of invisible parts in images).

regions [16]. These errors occur because the CNN filters indiscriminate to both valid and invalid regions [17].

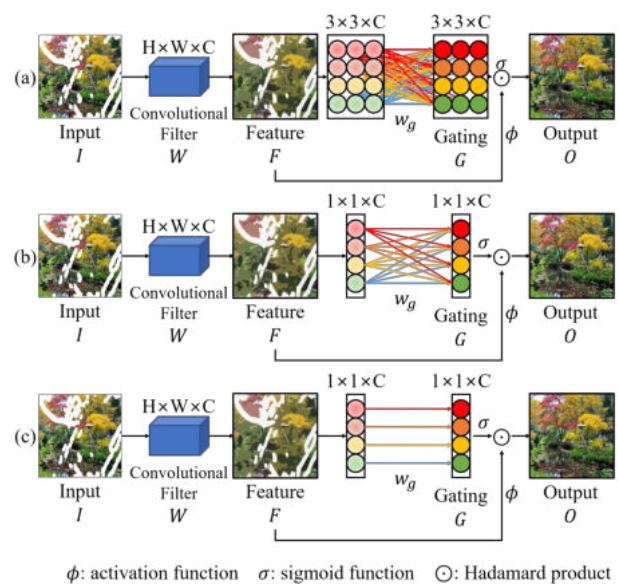


Fig. 1. Architecture of three attention mechanisms. (a): feature attention, (b): channel attention, (c): channel independent attention.

The gated convolution (GC) in *DeepFill* v2 adopts feature attention in each core block to address this problem and filters out pixels interfering with reconstruction [16]. GC-based architectures [16], [18]–[20] can generate more detailed results than earlier CNN-based approaches in irregular masks [9]–[11], but the performance improvement of feature attention has been negligible compared to the effect of other vision tasks such as image recognition [21]–[23].

The channel attention can improve its performance better than that of the feature attention for high-speed computation and lightweight [24], [25], as shown in Fig. 1(b) and (a), respectively. Although this technique used fewer parameters than the earlier feature attention technique, its performance improved significantly. However, channel attention for such inpainting is still inefficient because two things are not considered: 1) the destruction of direct correspondence between channels and weights owing to changes in the channel dimen-

sion and 2) computational amount owing to the total coupling between channels [20], [22], [25].

To address these issues, this paper proposes channel independent attention (CIA) in Fig. 1(c). The proposed CIA focuses on the channels of the current features, activates valid channels, and reduces the parameters required for computing. This paper also proposes a CIAFill architecture that includes a channel attention generator (CAG) and a channel attention projection PatchGAN (CAPP). CAG is a generator that adopts CIA blocks as core modules and requires fewer parameters to adopt the gating concept than the earlier approaches [16], [20]. CAPP is an attention-guided discriminator that also adopts CIA blocks as core modules to boost the quality of reconstructed results.

## II. RELATED WORK

### A. Convolution structure for inpainting

[9] presented deep learning-based inpainting utilizing a CNN with a generative adversarial network (GAN) [26]. Following this study, the use of CNN with GAN has become mainstream in inpainting research [10], [11], [16], [17]. However, because of the limitation of semantic understanding, CNN-based methods commonly yield blurry reconstruction results in complex scenes.

[17] investigated the reason for the blurry results from the CNN-based inpainting models. This study revealed that convolutional filters were spatially shared for all input pixels or features, leading to a blurry reconstruction. This implied that the invalid pixels or features in the region, such as holes, propagated the meaningless information to surrounding regions and generated blurry results. To address this problem, [17] proposed a partial convolution (PC) mechanism that considers only valid pixels for reconstruction by masking invalid pixels and renormalizing. However, this model was stacked by PC, layer to layer, and invalid pixels gradually converted to valid pixels through training; nevertheless, these valid pixels were neglected.

[16] suggested GC utilizing feature attention. The GC provided information for each layer from its previous layer using the feature-wise product. Because of its effectiveness in extracting valid features from its previous layer and the capability to utilize user sketch input to guide the results, GC has been widely used in recent inpainting [18]–[20].

[20] suggested three types of light weight gated convolution (LWGC) - depth-separable LWGC (LWGC<sup>ds</sup>), pixel-wise LWGC (LWGC<sup>pw</sup>), and single-channel LWGC (LWGC<sup>sc</sup>) - for lightweight inpainting and synthesizing high-resolution images. These models reduced the number of parameters but did not outperform GC.

### B. Channel attention mechanism

Channel attention has been proposed in image recognition tasks to alleviate the high computation of feature attention [22], [24], [25], [27]. The squeeze-excitation module improved the recognition performance compared to general convolution by compressing and re-activating the channel [24]. The

gather-excite module was proposed for a lightweight approach and better context exploitation in CNNs using strided depth-wise convolution [23], [28]. The bottleneck attention module (BAM) divided feature attention into channel attention and spatial attention, and then computed them in parallel [27]. The convolutional bottleneck attention module improved the performance over the BAM by replacing parallel operations with serial operations [22]. The channel attention module efficiently improved the image recognition performance with fewer channel interactions and a similar channel size [25].

### C. Generators inside GAN for inpainting

*DeepFill v1* [11] introduced two-stage inpainting models with a dilated convolution structure. Strided convolution [29] was used to reduce the resolution of the image by one-quarter for preventing the loss of pixel information and then added five extended layers to increase the receptive field of this structure. However, this module is slow and heavy load because it utilizes two stages and is calculated with high resolution.

U-net was proposed for inpainting structure [17], [30], but these structures utilized a relatively large number of learnable parameters than dilated convolution-base structure.

### D. Discriminators inside GAN for inpainting

The most commonly used discriminator in inpainting is PatchGAN [31]. Each dimension of the PatchGAN output receives only the patch region in the images and determines the region as real or fake. This model allows the generator to produce realistic images in image-to-image translation [31], [32], but commonly tends to undergo unstable training [33].

SN-PatchGAN [16] provided more stable training than the earlier work [11] by widening the receptive fields and adopting spectral normalization [33]. This discriminator eliminated the one-channel convolution in PatchGAN and adjusted the adversarial loss on each neuron of the output, leading to fast and stable training. However, this discriminator exhibited poor performance in inpainting largely masked images.

Boundless' discriminator [34] was a modified version of the conditional projection discriminator [35]. This discriminator replaced the classification label input with a pre-trained ImageNet model [36]. Because features from Inception v3 [37] contain more information from the conditioning vector, this structure improved the discriminator in photo-realistic and seamless synthesis. However, this model lost spatial conditions in the discriminating process.

## III. APPROACHES

Figure 2 demonstrates the overall architecture proposed in this study. CIA is an effective structure for fast and correct inpainting. To improve the attention-based GAN structure for the inpainting task, the channel attention generator and channel attention projection PatchGAN are introduced for the generator and discriminator in GAN, respectively.

The contributions of the proposed architecture are as follows. First, this model can achieve higher performance without increasing the number of channels, compared to the earlier

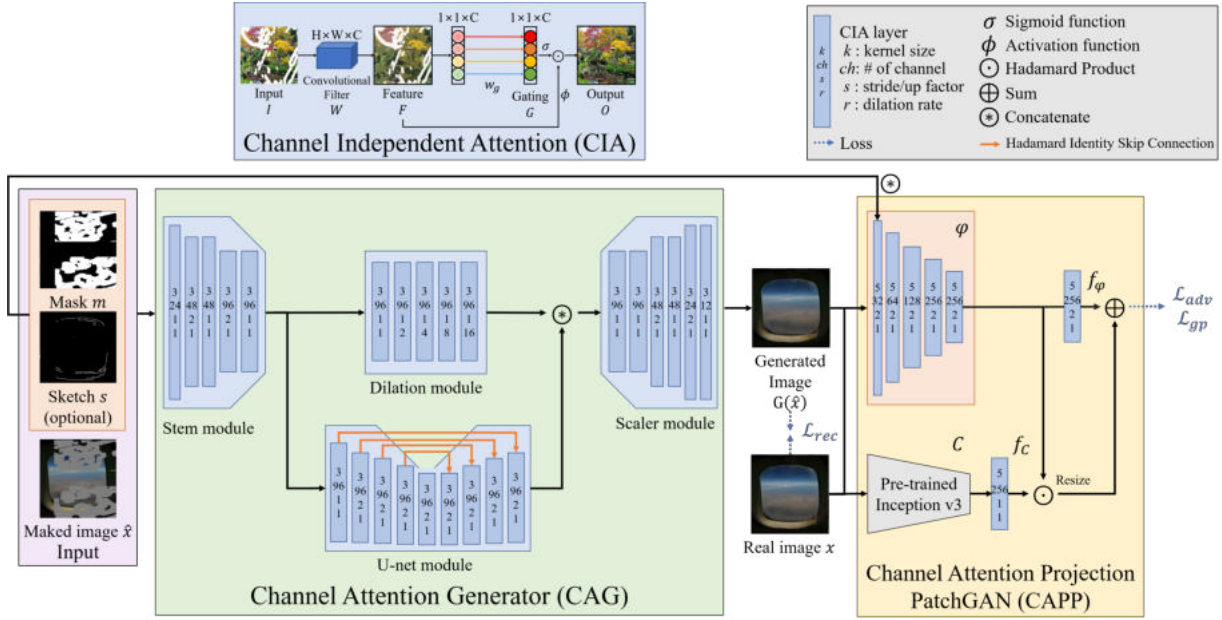


Fig. 2. Architecture of CIA, CAG, and CAPP for image inpainting.

attention-based models [16], [20]. This architecture introduces a serial operation mechanism for high-performance image inpainting. Earlier attention mechanisms for inpainting models such as GC and LWGC applied parallel operations. However, in the modeling process of repeated channel expansions and reductions, the semantic correspondences between channels and their weights with different channel sizes can be doubly destructive in parallel architectures, compared to the proposed serial mechanisms [25]. Next, the channel attentions in this architecture is calculated independently for each channel. This method can expect an effect similar to that of increasing the number of channel parameters and reduces the parameter computation by the number of channels compared to a typical channel attention [23]. Table I shows the number of training parameters required by each convolution mechanism used in inpainting.

TABLE I  
THE NUMBER OF PARAMETERS NEEDED TO COMPUTE GATING FOR EACH MECHANISM.

Mechanism	Parameter calculation	$k_h, k_w=3$ $C, C'=32$
GC	$k_h \times k_w \times C \times C'$	9216
LWGC <sup>ds</sup>	$k_h \times k_w \times C + C \times C'$	1312
LWGC <sup>pw</sup>	$C \times C'$	1024
LWGC <sup>sc</sup>	$k_h \times k_w \times C \times 1$	288
<b>CIA (Ours)</b>	$C$	<b>32</b>

### A. Channel Independent Attention

The proposed CIA introduces an improved channel attention mechanism for image inpainting. Figure 2 demonstrates the CIA architecture.  $w_g$  in this figure shows that each channel of gating  $G$  is calculated independently of the channel of feature

$F$  with the same index. The CIA formulation is as follows. Assume that input  $I$  is  $C$ -channel, each pixel located at  $y, x$  in  $C'$ -channel feature  $F_{y,x}$  is computed using Equation (1).

$$F_{y,x} = \sum_{i=-k'_h}^{k'_h} \sum_{j=-k'_w}^{k'_w} W_{k'_h+i, k'_w+j} \cdot I_{y+i, x+j} \quad (1)$$

Where  $x$  and  $y$  represents  $x$ -axis and  $y$ -axis of output map respectively,  $k_h$  and  $k_w$  denote the kernel size.  $k'_h = \frac{k_h-1}{2}$ ,  $k'_w = \frac{k_w-1}{2}$ ,  $W \in \mathbb{R}^{k_h \times k_w \times C \times C'}$ , and  $O_{y,x} \in \mathbb{R}^{C'}$  are inputs and outputs. This mechanism generates  $G$  by the given  $F$  for reflecting the characteristics of the features. Assume that  $G_{y,x}$  is the pixel located at  $(y, x)$  in  $G$ . The  $G_{y,x}$  is expressed as in Equation (2).

$$G_{y,x} = w_g \odot F_{y,x} \quad (2)$$

Where  $w_g \in \mathbb{R}^{C'}$  is a learnable parameter,  $\odot$  means Hadamard product [38]. The pixel of output at  $(y, x)$ ,  $O(y, x)$  is computed using Equation (3).  $\sigma$  represents a sigmoid function [39],  $\phi$  represents an activation function. The exponential linear unit is selected in this paper [40].

$$O = \phi(F_{y,x}) \odot \sigma(G_{y,x}) \quad (3)$$

Compared to earlier studies, the proposed CIA is different from spatial attention in vision understanding tasks. Because the input data in the image inpainting task already contain temporal validity for each pixel with masked images, spatial attention focused on the location patterns is unnecessary. In the case of LWGC<sup>sc</sup> using only spatial attention, poor performance was recorded compared to feature or channel attention-based models [20].



## B. Channel Attention Generator

For CAG module in Figure 2, let  $x$ ,  $\hat{x}$ ,  $m$ , and  $s$  represent samples from the original data, erased data, mask, and sketch (optional), respectively. The generator  $G$  takes the  $\hat{x}$ ,  $m$ , and  $s$  and outputs the generated image  $G(\hat{x})$ . The proposed CAG includes four modules: 1) Stem module that reduces the resolution by 1/4 each, 2) dilation module, 3) U-net module, and 4) Scaler module that up-scales the image resolution back to the original.

The stem module has the same parameters as the layers from the first to the fifth of DeepFill v2. Features extracted by this module become the input to the U-Net module and dilation module.

The proposed model mixes a dilated module designed to prevent pixel information loss [10] and a U-Net module that can increase the performance of the model by stacking several layers relative to the former structure [17], [41]. When using the U-Net module, instead of a skip connection that concatenates channels, the Hadamard identity skip connection (HISC) is applied, which can increase inpainting performance and reduce network parameters by replacing valid pixels of the decoder with those of the encoder for each pixel [42]. To avoid breaking the direct correspondence between channels and their weights, both modules consist of the proposed CIA with the same number of channels.

Finally, the scaler module receives the outputs of the dilation and U-Net modules and outputs the image that matches the original resolution.

TABLE II  
THE TOTAL NUMBER OF LEARNABLE PARAMETERS FOR EACH INPAINTING MODEL.

Structure	Model	# of learnable parameter
U-net	SC-FEGAN	42.1M
	DFNet	32.9M
Dilated convolution	EdgeConnect	12.1M
	DeepFill v2	4.1M
	HiFill	2.7M
Both	<b>CAG (Ours)</b>	<b>1.8M</b>

Table II presents the parameter number of inpainting models according to the model structure, which proves that the proposed CAG acquires the fewest number of parameters. The proposed CAG architecture can achieve high performance with a reduced number of model parameters.

## C. CAPP: Channel Attention Projection patchGAN for discriminator

In Figure 2, CAPP represents the proposed discriminator. This discriminator considers the  $\hat{x}$  as fake data with  $m$  and  $s$  as conditions, and the  $x$  as real data with the same  $m$  and  $s$  as conditions. CAPP is motivated by three representative discriminators. Boundless [34]’ discriminator prevents performance degradation in restoring where the erased area is large. An attention-guided discriminator [32] focuses on missing

regions. An SN-PatchGAN [16] provides stable learning with global and local features.

To take advantage of these three discriminators, CAPP consists of SN-PatchGAN ( $\phi$  and  $f_\phi$ ) as the baseline, pre-trained Inception v3 ( $C$ ) for extracting perceptual feature of an image, and a convolutional layer  $f_C$  matching the dimension of the output by  $C$  and  $\phi$  to project its feature as the condition. Formally, our discriminator  $D$  is expressed by Equation (4).

$$D([\hat{x}, x, m, s]) = f_\phi([\hat{x}, m, s], [x, m, s]) + f_C(C(x)) \odot \phi([\hat{x}, m, s], [x, m, s]) \quad (4)$$

The SN-patchGAN is expressed by Equation (5).

$$D([\hat{x}, x, m, s]) = f_\phi([\hat{x}, m, s], [x, m, s]) \quad (5)$$

The Boundless’ discriminator is expressed by Equation (6).

$$D([\hat{x}, x, m, s]) = f_\phi([\hat{x}, m, s], [x, m, s]) + \langle f_C(C(x)), \phi([\hat{x}, m, s], [x, m, s]) \rangle \quad (6)$$

Where  $\langle \cdot, \cdot \rangle$  denotes an inner product. In Equation (4),  $f_\phi(\phi([\hat{x}, m, s], [x, m, s]))$  is from the SN-PatchGAN in Equation (5).  $C(x) \odot \phi([\hat{x}, m, s], [x, m, s])$  is from Boundless’s discriminator, differs in three respects. First, the inner product is changed to the Hadamard product to preserve the authenticity of each neuron. Second, the features extracted from  $\phi$  can contain sketch information. Although the inpainting task without sketch information was performed in this study, the proposed CAPP model can also receive sketch information as an optional input for user-intended inpaintings. Finally, the Boundless discriminator extracted the features of Inception v3’ from the output layer, but the proposed discriminator extracted features from the layer before pooling [37]. This change indicates that this model reflects the derived information of each pixel inferred from the model.

All convolutional layers in this architecture are changed to CIAs for dynamic feature selection. Compared to the earlier attention-guided discriminator with a constant threshold-based attention map [32], the CAPP effectively focuses on the erased parts with CIA layers, and it is possible to extract optimal features for the masked regions.

## D. Loss function

The adversarial loss function for this model is adjusted from the loss of SN-patchGAN to reflect the output of each neuron [16]. The model is trained with a mixture of three losses: reconstruction loss  $L_{rec}$ , adversarial loss  $L_{adv}$  [26], and gradient penalty loss  $L_{gp}$  [43] as shown in Equation (7), where  $\lambda_{rec} = 100$ ,  $\lambda_{adv} = 1$ , and  $\lambda_{gp} = 10$  [18], [34].

$$L_{total} = \lambda_{rec}L_{rec} + \lambda_{adv}L_{adv} + \lambda_{gp}L_{gp} \quad (7)$$

The pixel-wise L1 loss is selected as  $L_{rec}$ . The hinge loss [44] is applied to  $L_{adv}$  [30], [45]. Hinge loss consists



Fig. 3. Examples of a inpainting results using Places2 and CelebA-HQ by each model.

of  $L_G$  for training the generator and  $L_D$  for training the discriminator.  $L_G$  is derived using Equation (8).

$$L_G = -\mathbb{E}_{z \sim \mathbb{P}_Z} [D(G(z))] \quad (8)$$

$\mathbb{E}_{\bullet \sim \mathbb{P}_{\blacksquare}}$  represents the expectation of variables  $\bullet$  with probability distribution function  $\mathbb{P}_{\blacksquare}$ .  $z$  represents a set of  $\hat{x}$ ,  $m$ , and  $s$ .  $Z$  is the probability distribution of  $z$ .

The discriminator is trained using Equation (9).  $data$  represents the probability distribution of  $x$ ,  $Relu$  is a rectified linear unit [46].

$$L_D = \mathbb{E}_{x \sim \mathbb{P}_{data}} [Relu(1 - D(x))] + \mathbb{E}_{z \sim \mathbb{P}_Z} [Relu(1 + D(G(z)))] \quad (9)$$

The gradient penalty loss is represented by Equation (10) for high-quality inpainting performances.

$$L_{GP} = \mathbb{E}_{u \sim P_u} [(\|\nabla_u D(u)\|_2 - 1)] \quad (10)$$

In Equation (10),  $U$  represents the probability distribution function of  $u$ , which is a uniformly sampled data point along the straight line between the discriminator inputs from  $x$  and  $\hat{x}$ . In this study, the generator and the discriminator used Adam [47] optimizer for training and set the learning rate to  $1e-5$  and  $1e-4$  until convergence, respectively.

## IV. EXPERIMENTS

### A. Experimental setting

TensorFlow [48] 1.15, CUDA 10 [49], and cudnn 7.4 were used for this experiment. Two computers were used - Intel(R) Xeon(R) Silver 4114 as CPU and Intel(R) Xeon(R) W-2145 as CPU including NVIDIA TITAN RTX as GPU and 64GB of RAM.

Two datasets were used in this experiment: Places2 [50] and CelebA-HQ [51]. Places2 includes 18 million scene photographs with scene categories and is cropped to 256 pixels in both width and height for the experiments. CelebA-HQ is a dataset of face images in which 30,000 high-resolution images are resized to 512 pixels in both width and height. Free-form masks or an irregular mask dataset provided by [17] are used to create irregular holes. The Canny edge algorithm [52] was applied to the datasets to generate the sketch dataset (optional). However, for the sake of fairness, the sketch was not used in the comparative experiments. In all tables in this chapter, the bold fonts and underline indicate the best and the second performance in the same column, respectively. L1 error, Structural SIMilarity (SSIM) [53], and Fréchet Inception Distance (FID) [54] were used as the evaluation metrics to measure how much it is restored like the original, how structurally it is similar to the original, and how similar it is to the original data distribution, respectively.

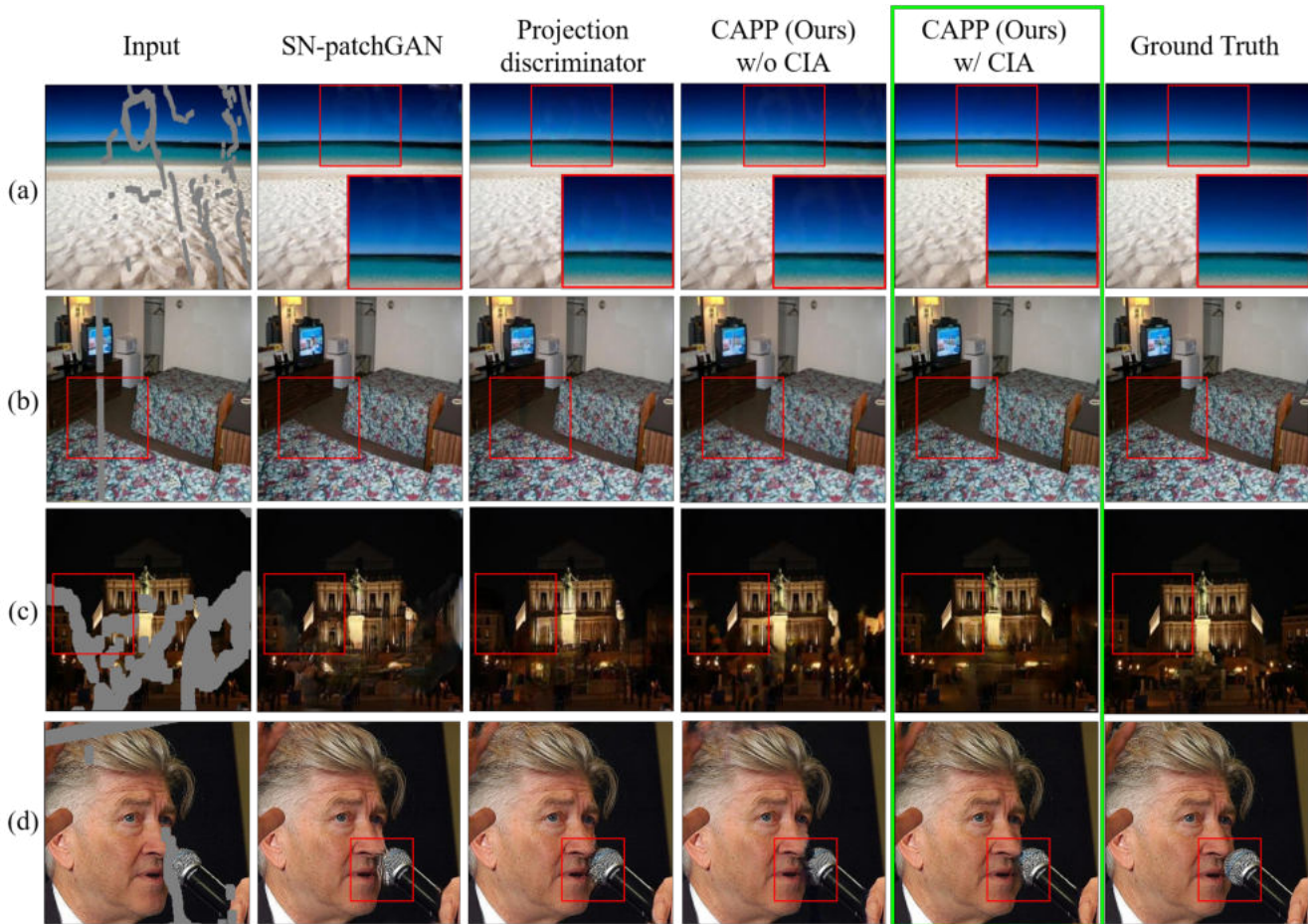


Fig. 4. Examples of inpainting results using Places2 and CelebA-HQ by each discriminator.

### B. Qualitative and quantitative results

Four models - DeepFill v2 [16], HiFill [20], EdgeConnect [45], and DFNet [30] - were set up to compare the performances with the proposed architecture. The quantitative and qualitative results of each comparative model were implemented using the pre-trained models published by the authors.

Figure 3 illustrates certain inpainting results of the five models. In the case of (a), the image could not be restored by DeepFill v2 without utilizing the NVIDIA irregular mask dataset proposed for general-purpose inpainting. As shown in (b), the proposed CIAFill successfully restored the shape of a building unlike DeepFill v2 or HiFill because CIAFill concentrates only on the pixel index without spatial information during attention. (c) indicated that only two models, DFNet and CIAFill utilizing U-net, consistently inpainted the black pillars because U-net could consider the full context of the image with stacking strided convolution. In the case of (d), edge connect, DFNet makes lips restoration unnatural. Overall, CIAFill architecture visually outperformed the other models.

Table III presents the average inference time per image, L1 error, SSIM, and FID of five models in the Places2 datasets and four models in the CelebA-HQ datasets. Our model performed

TABLE III  
PERFORMANCES OF INPAINTING RESULTS BY EACH MODELS.

Places2 Model	Time (ms)	L1 (%)	SSIM	FID
DeepFill v2	51	8.94	0.885	8.61
HiFill	43	8.07	0.884	8.54
EdgeConnect	78	<b>6.98</b>	0.902	7.97
DFNet	85	7.11	<u>0.905</u>	<u>7.45</u>
<b>CIAFill (Ours)</b>	<b>39</b>	<u>7.00</u>	<b>0.909</b>	<b>7.22</b>
CelebA-HQ Model	Time (ms)	L1 (%)	SSIM	FID
DeepFill v2	69	4.20	0.921	<u>5.87</u>
EdgeConnect	110	4.57	0.910	6.55
DFNet	126	4.11	<u>0.927</u>	6.16
<b>CIAFill (Ours)</b>	<b>41</b>	<b>4.02</b>	<b>0.936</b>	<b>5.41</b>

the best in all four metrics except for the L1 error in the Places2 testset. In CelebA-HQ, the proposed model achieved the best performance in all four metrics.

### C. Comparisons of discriminator

Figure 4 shows examples of generated images when trained by SN-PatchGAN, projection discriminator, CAPP without CIA, and CAPP with CIA. As shown in (a), it was restored

seamlessly because CAPP with CIA using the attention module concentrated only on the deleted part. In the case of (b), the complex texture of the quilt could be reproduced because the features of the original image were used for projection during training. However, the projection discriminator that did not consider local features could not erase the mask marks in the narrowly erased area. As confirmed by (c), considering both local features and attention was essential to prevent unintentional spots because the region of interest can be identified. In the case of (d), mike and the celeb lip were distorted in the outputs of SN-PatchGAN and projection discriminator. There, the proposed discriminator with CIA outperformed other discriminators in this experiment.

TABLE IV  
EVALUATION PERFORMANCES BY EACH DISCRIMINATOR.

Places2 Discriminator	L1 (%)	SSIM	FID
SN-patchGAN	8.81	0.866	8.82
projection discriminator	<u>7.18</u>	<u>0.893</u>	<u>7.27</u>
CAPP (Ours) w/o CIA	7.25	0.892	7.84
<b>CAPP (Ours) w/ CIA</b>	<b>7.00</b>	<b>0.909</b>	<b>7.22</b>
CelebA-HQ Discriminator	L1 (%)	SSIM	FID
SN-patchGAN	4.22	0.927	5.61
projection discriminator	4.19	0.916	<u>5.47</u>
CAPP (Ours) w/o CIA	4.15	<u>0.933</u>	5.70
<b>CAPP (Ours) w/ CIA</b>	<b>4.02</b>	<b>0.936</b>	<b>5.41</b>

Table IV reports the performances of the proposed discriminator. CAPP with CIA performed the best on all three metrics in both datasets. In this experiment, whereas the Places2 dataset included a variety of foregrounds, the CelebA-HQ dataset included front-facing human faces. Therefore, CelebA-HQ shared more similar features than Places2. In this case, the correct judgment about attention can be a more important contribution to performance improvement than in other cases. Therefore, CAPP without CIA recorded lower performance than projection discriminator in the Places2 dataset. In the CelebA-HQ dataset, CAPP without CIA performed better than the projection discriminator.

#### D. Comparisons of attention-based mechanisms

In these experiments, DeepFill v2 was the baseline model, and GC, LWGC, and the proposed CIA changed the generator’s convolution mechanism. Five perspectives were applied for the analysis measures of results: L1 error, SSIM, FID, the total number of gatings’ learnable parameters, and the time it takes to generate an image of  $256 \times 256$  in the Places2 dataset.

Figure 5 depicts certain samples for qualitative comparison. The results from the LWGC indicated a noticeable problem with regions that were blurred or erased. Because GC and LWGC utilize spatial information for attention, the channel features were reduced, leaving mask-shaped marks when restored.

As shown in Table V, the proposed CIA outperformed the compared mechanisms in all evaluation measures. In this study, it was proved that the proposed CIA is the most efficient

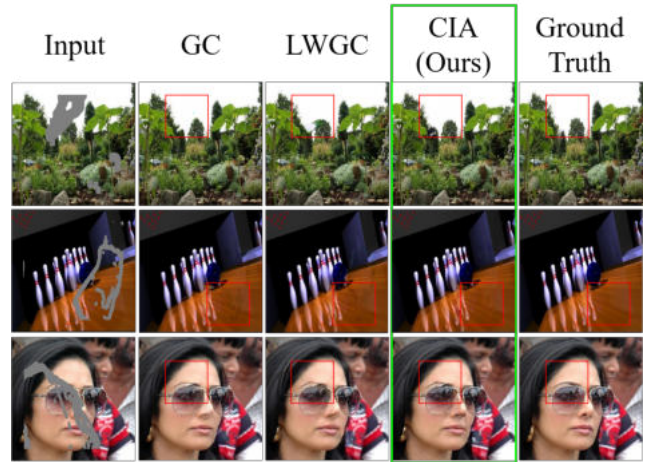


Fig. 5. The inpainting results using Places2 and CelebA-HQ by each attention mechanisms.

TABLE V  
PERFORMANCE COMPARISONS OF DEEPFILL v2 BASED MODELS USING PLACES2.

Mechanism	# of gatings’ parameter	Time (ms)	L1 (%)	SSIM	FID
GC	1,793,928	51	8.87	0.893	7.65
LWGC	<u>110,564</u>	<u>38</u>	<u>8.82</u>	<u>0.904</u>	<u>7.56</u>
<b>CIA (Ours)</b>	<b>1,119</b>	<b>33</b>	<b>8.69</b>	<b>0.911</b>	<b>7.49</b>

gating method in terms of computational amount and speed than earlier methods.

#### V. CONCLUSION

This paper introduces the CIAFill architecture for fast and lightweight image inpainting. The CIAFill included CAG and CAPP utilizing the proposed CIA as a core mechanism based on independent channel attention. The experiments in this study proved that the proposed mechanisms performed better than existing models in terms of performance, speed, and model size. The strengths of the proposed methods would contribute to real-time inpainting or inpainting in mobile environments. The CIAFill still requires improvements. It maintains its performance only by using both CAG, which uses both the dilation module and U-Net module, and CAPP, which combines attention-guided discriminator, SN-PatchGAN, and projection discriminator. Therefore, we intend to improve the channel attention mechanism applicable to the general model in our future work.

#### REFERENCES

- [1] A. Criminisi, P. Perez, and K. Toyama, “Object removal by exemplar-based inpainting,” in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, vol. 2. IEEE, 2003, pp. II–II.
- [2] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, “Filling-in by joint interpolation of vector fields and gray levels,” *IEEE transactions on image processing*, vol. 10, no. 8, pp. 1200–1211, 2001.
- [3] S. Darabi, E. Shechtman, C. Barnes, D. B. Goldman, and P. Sen, “Image melding: Combining inconsistent images using patch-based synthesis,” *ACM Transactions on graphics (TOG)*, vol. 31, no. 4, pp. 1–10, 2012.

- [4] J.-B. Huang, S. B. Kang, N. Ahuja, and J. Kopf, "Image completion using planar structure guidance," *ACM Transactions on graphics (TOG)*, vol. 33, no. 4, pp. 1–10, 2014.
- [5] D. Liu, X. Sun, F. Wu, S. Li, and Y.-Q. Zhang, "Image compression with edge-based inpainting," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 10, pp. 1273–1287, 2007.
- [6] D. Kim, S. Woo, J.-Y. Lee, and I. S. Kweon, "Deep video inpainting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5792–5801.
- [7] W. Wang, Q. Huang, S. You, C. Yang, and U. Neumann, "Shape inpainting using 3d generative adversarial network and recurrent convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2298–2306.
- [8] S. Esedoglu and J. Shen, "Digital inpainting based on the mumford-shah-euler image model," 2001.
- [9] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2536–2544.
- [10] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, pp. 1–14, 2017.
- [11] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5505–5514.
- [12] Y. Wang, X. Tao, X. Qi, X. Shen, and J. Jia, "Image inpainting via generative multi-column convolutional neural networks," *arXiv preprint arXiv:1810.08771*, 2018.
- [13] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *IEEE transactions on neural networks*, vol. 8, no. 1, pp. 98–113, 1997.
- [14] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 922–928.
- [15] C. Zhang, H. Li, X. Wang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 833–841.
- [16] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Free-form image inpainting with gated convolution," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4471–4480.
- [17] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, "Image inpainting for irregular holes using partial convolutions," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 85–100.
- [18] Y. Jo and J. Park, "Sc-fegan: face editing generative adversarial network with user's sketch and color," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1745–1753.
- [19] Y.-G. Shin, M.-C. Sagong, Y.-J. Yeo, S.-W. Kim, and S.-J. Ko, "Pepsi++: Fast and lightweight network for image inpainting," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 252–265, 2020.
- [20] Z. Yi, Q. Tang, S. Azizi, D. Jang, and Z. Xu, "Contextual residual aggregation for ultra high-resolution image inpainting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7508–7517.
- [21] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3156–3164.
- [22] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.
- [23] J. Hu, L. Shen, S. Albanie, G. Sun, and A. Vedaldi, "Gather-excite: Exploiting feature context in convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 31, pp. 9401–9411, 2018.
- [24] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.
- [25] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "Eca-net: efficient channel attention for deep convolutional neural networks, 2020 ieee," in *CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020.
- [26] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv preprint arXiv:1406.2661*, 2014.
- [27] J. Park, S. Woo, J.-Y. Lee, and I. S. Kweon, "Bam: Bottleneck attention module," *arXiv preprint arXiv:1807.06514*, 2018.
- [28] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.
- [29] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.
- [30] X. Hong, P. Xiong, R. Ji, and H. Fan, "Deep fusion network for image completion," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 2033–2042.
- [31] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.
- [32] Y. A. Mejjati, C. Richardt, J. Tompkin, D. Cosker, and K. I. Kim, "Un-supervised attention-guided image to image translation," *arXiv preprint arXiv:1806.02311*, 2018.
- [33] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *arXiv preprint arXiv:1802.05957*, 2018.
- [34] P. Teterwak, A. Sarna, D. Krishnan, A. Maschinot, D. Belanger, C. Liu, and W. T. Freeman, "Boundless: Generative adversarial networks for image extension," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 10521–10530.
- [35] T. Miyato and M. Koyama, "cgans with projection discriminator," *arXiv preprint arXiv:1802.05637*, 2018.
- [36] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.
- [37] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.
- [38] R. A. Horn, "The hadamard product," 1990.
- [39] F. Rosenblatt, "Principles of neurodynamics. perceptrons and the theory of brain mechanisms," Cornell Aeronautical Lab Inc Buffalo NY, Tech. Rep., 1961.
- [40] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint arXiv:1511.07289*, 2015.
- [41] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.
- [42] Y. C. E. H. Chung-II Kim, Jehyeok Rew, "Ufc-net with fully-connected layers and hadamard identity skip connection for image inpainting," *Computers, Materials & Continua*, vol. 68, no. 3, pp. 3447–3463, 2021. [Online]. Available: <http://www.techscience.com/cmc/v68n3/42518>
- [43] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein gans."
- [44] J. H. Lim and J. C. Ye, "Geometric gan," *arXiv preprint arXiv:1705.02894*, 2017.
- [45] K. Nazeri, E. Ng, T. Joseph, F. Qureshi, and M. Ebrahimi, "Edgeconnect: Structure guided image inpainting using edge prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.
- [46] K. Fukushima, "Visual feature extraction by a multilayered network of analog threshold elements," *IEEE Transactions on Systems Science and Cybernetics*, vol. 5, no. 4, pp. 322–333, 1969.
- [47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR (Poster)*, 2015.
- [48] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-

- scale machine learning,” in *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, 2016, pp. 265–283.
- [49] NVIDIA, P. Vingelmann, and F. H. Fitzek, “Cuda, release: 10.2.89,” 2020. [Online]. Available: <https://developer.nvidia.com/cuda-toolkit>
- [50] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, “Places: A 10 million image database for scene recognition,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 6, pp. 1452–1464, 2017.
- [51] Z. Liu, P. Luo, X. Wang, and X. Tang, “Deep learning face attributes in the wild,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3730–3738.
- [52] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.
- [53] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [54] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “Gans trained by a two time-scale update rule converge to a local nash equilibrium,” *arXiv preprint arXiv:1706.08500*, 2017.

# Anomaly Detection for Alzheimer's Disease in Brain MRIs via Unsupervised Generative Adversarial Learning

Jean Nathan Cabreza, Geoffrey A. Solano, Sun Arthur Ojeda, Vincent Munar

<sup>1</sup>Department of Physical Sciences and Mathematics  
College of Arts and Sciences, University of the Philippines Manila

**Abstract**—Alzheimer's disease (AD) is a neurodegenerative disease that results in cognitive decline, and even dementia, in patients. To diagnose AD, a combination of tools is typically used, with structural magnetic resonance imaging (sMRI) being one of them. sMRI images have mostly been used in supervised deep learning approaches, which requires large amounts of labeled data. To alleviate the need for labels, unsupervised deep learning could be used as an alternative. This study proposes an unsupervised model based on the deep convolutional generative adversarial network that performs anomaly detection on brain MRIs to diagnose AD. The model is able to yield an AUROC of 0.7951, a precision of 0.8228, a recall of 0.7386, and an accuracy of 74.44%.

**Index Terms**—Alzheimer's disease, generative adversarial networks, unsupervised learning, anomaly detection

## I. INTRODUCTION

Alzheimer's disease (AD) is a neurodegenerative disease that causes irreversible neuronal loss in affected patients [1]. AD typically manifests in the form of gradual memory loss [2], although it may also manifest itself as other forms of cognitive decline in patients. As AD progresses, it may also eventually cause dementia in those affected, with it being the leading cause of dementia around the world [3]. As the world now has a generally aging population, AD is now becoming an increasingly large problem since susceptibility to it also increases with a person's age.

To get a fully accurate AD diagnosis, a post-mortem brain examination is required, but it may still be diagnosed with high accuracy using a combination of tools [2]. Among these tools is structural magnetic resonance imaging (sMRI), a medical imaging technique that can capture the anatomical structure of a patient's brain. In an MRI, neuronal damage is reflected as a reduction in brain volume [4] and neurofibrillary tangle (NFT) density, a pathological hallmark of AD, is indirectly reflected through this since it is negatively correlated to neuronal count [5]. Since they reflect factors that are relevant to AD, MRIs can then be said to be viable biomarkers for AD diagnosis.

sMRI images have, in recent times, been used in studies together with supervised deep learning to diagnose AD in patients. While these studies have yielded considerable results, a caveat of supervised deep learning is that it needs large amounts of labeled data, which may not be feasible to gather in a medical context [6]. Preparing labeled medical datasets requires domain knowledge and a significant amount of time [7], making the process both expensive and hard to do. In addition to this, one has to also consider privacy

concerns and the availability of data when trying to create a labeled medical dataset.

To eliminate the problem of having to provide labels, unsupervised deep learning may be an alternative approach to explore. To this end, methods based on generative adversarial networks (GANs) and anomaly detection have recently been proposed. Anomaly detection involves finding datapoints that are dissimilar from all other data points in a data-driven fashion [8], that is, its main concern is to find outliers in some given data. To do so, anomaly detection models must be able to model non-outlier (or "normal") data, and a GAN is able to do so, even for high-dimensional data [9] like images, hence their use in anomaly detection approaches.

The authors then propose an unsupervised deep learning approach as decision support for AD diagnosis in the form of an anomaly detection model largely based on the deep convolutional GAN (DCGAN) [10]. The rest of the paper is structured as follows: related work in GAN-based anomaly detection using brain MRIs is first discussed. Then, the dataset and the model architecture is discussed, along with the specific techniques that the authors use in the study. Lastly, the results of the study are presented along with a discussion of the results.

## II. RELATED WORK

Several anomaly detection techniques have been proposed over the years for a wide variety of fields. In recent times, however, deep learning has become prevalent and many studies make use of methods such as convolutional neural networks and long short-term memory networks, among others. These particular methods have been used for detecting anomalies in areas such as time series data, medical images, and in more industrial applications as well. GANs have similarly been used for much of the same tasks, but this study focuses only on medical applications, and in particular, applications to brain MRI images. Chalapathy et al. give an overview of deep learning in anomaly detection in [8], which also includes other GAN-based anomaly detection approaches.

Anomaly detection in brain MRI images is often used to detect other brain conditions such as lesions, for example. Chen and Konukoglu [11] have used adversarial autoencoders for pixel-wise anomaly detection to detect lesions in brain MRIs, which resulted in a high AUC. van Hespen et al. [12] used a GAN architecture based on GANomaly [13] as an anomaly detection model to detect brain infarcts in

brain MRI images, which was then able to detect most of the infarcts, in terms of volume, in their dataset.

Han et al. [6] performed anomaly detection using a self-attention GAN that was trained to reconstruct multiple adjacent brain MRI slices and then diagnose an input MRI scan using the reconstructions. Their model yields considerable performance with an AUC of 0.783 when detecting AD in scans and an AUC of 0.921 when detecting brain metastases. The use of self-attention modules in GANs, however, may result in them needing significant amounts of memory and computing power [14]. In light of this, this study then aims to explore the use of simpler models in an effort to reduce the amount of computational resources needed in GAN-based anomaly detection for AD diagnosis.

### III. METHODOLOGY

#### A. Generative Adversarial Networks

Generative adversarial networks were proposed by Goodfellow et al. [15] as a framework for training generative models. The GAN framework usually involves two models: a generator  $G$  and a discriminator  $D$ . Both models are trained at the same time, and during training,  $D$  is made to classify data samples as either real or fake, whilst  $G$  tries to "fool"  $D$  by making fake data samples that are similar to the real data samples. The original GAN framework has both  $G$  and  $D$  solving the same value function at the same time:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} \log D(x) + \mathbb{E}_{z \sim p_z(z)} \log (1 - D(G(z))) \quad (1)$$

If both models are large enough and  $D$  is trained to optimality for every training step of  $G$ , then  $G$  eventually learns to model the distribution of the real data, and is then able to make samples from that same distribution. However, while that may be the case in theory, GANs have been shown to be difficult to train in practice, with problems such as instability and non-convergence manifesting themselves during the training process. Moreover, training  $D$  to optimality for every training step of  $G$  is largely impractical in real world situations.

Several techniques have since been proposed to improve GAN training, however, and this study explores a select few, which are further discussed in *Model Architecture*. The overall workflow of the experiment in this study is shown in figure 1.

#### B. Data Preprocessing

This study makes use of the OASIS-3 dataset [16], a longitudinal dataset containing brain scans from both cognitively healthy patients and patients with AD.

To distinguish between healthy patients and those with AD, the clinical dementia ratings (CDR) of each subject are used. Patients with a CDR of zero throughout the OASIS-3 dataset are considered as cognitively healthy (or "normal"), and patients otherwise are considered as those with AD. That is, non-cognitively healthy patients are considered as anomalies. Specific scans are considered either normal or anomalous, depending on the condition of the patient from which they are taken from.

80% of the normal scans were used as the training set for the anomaly detection model, whilst the rest, along with all the anomalous scans, were put into the test set. Afterwards, each scan, which were originally stored in NifTI files, was converted into a set of .PNG images using the NifTI Image Converter (nii2png) from Laurnce [17]. This study makes use of only axial images, and so non-axial images were discarded. In addition, since the hippocampus, amygdala, and ventricles are the structures most significant to AD [6], images that did not contain them were discarded as well.

#### C. Model Architecture

The models, and the GAN architectures, used in this study are based on deep convolutional GAN (DCGAN) [10], which, at the time, enabled more stable training for GANs on images by using techniques such as removing fully connected layers and using batch normalization in  $G$  and  $D$ . It is also relatively simple and small in terms of architecture, so a similar architecture may be less expensive in terms of computation cost.

Self-attention modules [18] are also used in the architecture as they have been shown to improve the quality of the images produced by GANs by enabling better modeling of long-range dependencies in images. Since they are computationally expensive [14], however, the modules are used only for the larger feature maps produced by the models.

Spectral normalization [19] used in both the generator and the discriminator has been shown to also improve training [18], and so the same technique is adopted in this study. This allows for better training and possibly better results for a relatively low computational cost, as opposed to other GAN formulations. The architecture of the models is shown in tables I and II.

In addition to the said techniques, this study also makes use of the two time-scale update rule (TTUR) [20], latent optimization [21], and differential augmentation [22] during GAN training in order to try and improve both training and results.

In addition to the proposed GAN, an encoder was also trained to learn the inverse of the generator, which takes in as input an image and then returns a vector which the generator can then use to make a reconstruction of the input, similar to f-AnoGAN [7]. After training, the discriminator is discarded and the encoder is combined with the generator to create an autoencoder-like architecture for reconstructing brain MRI images. The encoder has an architecture similar to that of the discriminator, as shown in table III.

Layers
$z \in \mathbb{R}^{128} \sim \mathcal{N}(0, I)$
$4 \times 4$ Deconvolution, Batch Norm 256 ReLU
Upsample (Scale=2), $3 \times 3$ Convolution, Batch Norm 128 ReLU
Upsample (Scale=2), $3 \times 3$ Convolution, Batch Norm 64 ReLU
Self-Attention Module
Upsample (Scale=2), $3 \times 3$ Convolution, 3 Tanh

TABLE I: Generator architecture



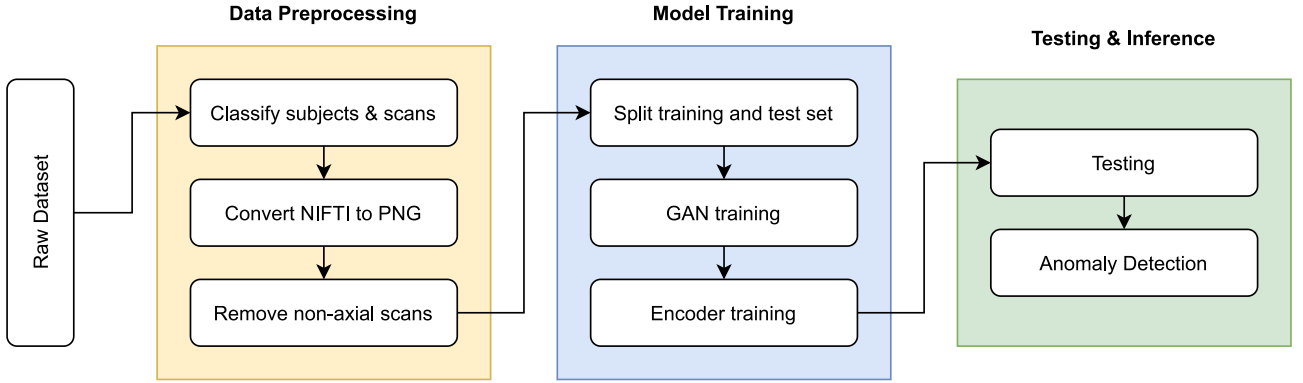


Fig. 1: Workflow of the experiment.

Layers
Greyscale Image $x \in \mathbb{R}^{32 \times 32 \times 1}$
$4 \times 4$ Convolution, 64 Leaky ReLU
Self-Attention Module
$4 \times 4$ Convolution, 128 Leaky ReLU
$4 \times 4$ Convolution, 256 Leaky ReLU
$4 \times 4$ Convolution, 1

TABLE II: Discriminator Architecture

Layers
Greyscale Image $x \in \mathbb{R}^{32 \times 32 \times 1}$
$4 \times 4$ Convolution, 64 Leaky ReLU
Self-Attention Module
$4 \times 4$ Convolution, 128 Leaky ReLU
$4 \times 4$ Convolution, 256 Leaky ReLU
$4 \times 4$ Convolution, 512 Leaky ReLU
Dense $\rightarrow$ 256 Leaky ReLU
Dense $\rightarrow$ 128 Leaky ReLU

TABLE III: Encoder architecture

#### D. Training

Before being used as input for the models, images are resized to  $33 \times 48$  and then zero-padded to  $49 \times 48$  from their original sizes of  $176 \times 256$  and  $176 \times 240$ . Afterwards,  $32 \times 32$  center crops are taken from each image so as to remove the black borders from each image. These center crops are then used as input for the models.

The GAN was then trained for 300,000 steps, with one discriminator update per generator update. The Adam optimizer [23] was used with a learning rate of 0.0001 for  $G$ , a learning rate of 0.0004 for  $D$ , and with the betas set to 0.0 and 0.9, respectively. Brock et al. [24] have shown that GANs can benefit from large batch sizes, and so a batch size of 256 was used. Moreover, the hinge loss [25] was used as well. The generator was set to receive a 128-dimensional vector as input.

After GAN training, the encoder was trained for 250,000 steps with the Adam optimizer as well. The learning rate was set to 0.0005 and the betas were set to 0.5 and 0.9,

respectively. The  $\ell_1$  loss between the input image and its reconstruction was used as the loss function for encoder training.

#### E. Anomaly Detection

Anomaly detection is done via a combination of comparing the original image and the image reconstruction, and comparing the latent vector given by the original image and the latent vector given by the reconstruction. More formally, each image is given an anomaly score based on the following:

$$A_{image} = \alpha|x - G(E(x))| + (1 - \alpha)|E(x) - E(G(E(x)))|, \quad (2)$$

where  $\alpha$  is used to weigh the contribution of the reconstruction error and the error between the latent vectors.

The latent vectors were considered for anomaly detection since it is assumed that the latent vectors contain the most relevant information about the image, and so this information should mostly be the same in the reconstruction if the image is normal. The anomaly score for a given scan is then the sum of all the anomaly scores in the images that make up the scan:

$$A_{scan} = \sum_{x \in S} A_{image}(x), \quad (3)$$

where  $S$  is the set of images in a given scan. The rationale behind this is that since the GAN is trained only on normal images, then it should be able to reconstruct normal images with relatively low error. On the other hand, the GAN should poorly reconstruct anomalous images since it was not trained on them, which should lead to a higher error on average in images from anomalous scans. This then, theoretically, enables discernment between normal and anomalous samples.

For testing, the anomaly scores for all the scans in the test set are normalized to the probabilistic range of  $[0, 1]$ , similar to [13]. Performance metrics are then derived from the normalized set of anomaly scores.

## IV. RESULTS AND DISCUSSION

Table IV shows the performance metrics that the proposed anomaly detection model yielded. In addition to this, a similar anomaly detection model was trained but without the use of self-attention modules. It, however, yielded a

Metric	Result
AUROC	0.7954
Precision	0.8228
Recall	0.7386
Accuracy	0.7444

TABLE IV: Performance metrics taken from model

lower AUROC of 0.7807. Figures 2 and 3 show some of the reconstructions that the GAN was able to produce.

The model is mostly able to reconstruct images from healthy brain MRI scans, but has relatively poor reconstruction performance on anomalous brain MRI scans. Since the GAN was trained to model only healthy brain MRI images, then the GAN should theoretically only be able to produce healthy brain MRI images. This particular result is then, more or less, to be expected.

The reduction in performance when the self-attention modules were taken out of the architecture suggests that the quality of images generated by the GAN affects its ability to reconstruct images, which is to be expected as low quality images are more likely to be erroneous if they were reconstructions. The authors did not perform further ablation studies to confirm which techniques provided the most significant boosts to the anomaly detection model's performance.

The use of a simple architecture and a limited number of self-attention modules allowed the anomaly detection model to both train and test on only a mid-range graphics card, and in particular, an Nvidia GTX1060 6GB, which was used by the authors in this study. That said, however, whilst the model has a relatively cheap computational cost, its performance is also relatively poor compared to supervised deep learning methods.

Due to the wide variety of brain MRI scans, a simple architecture, and possibly a small image size, the GAN may have been unable to capture subtle details in the scans, which in turn, may have been a source of error for the classifications of the anomaly detection model. Moreover, the CDR of a patient is not particularly based off of the results of their brain MRI scans, but rather it is based on a scoring system that cuts across different cognitive domains. This may then have been another source of error in the classification and testing of each brain MRI scan.

## V. CONCLUSION

This study proposes an anomaly detection model based on the DCGAN for diagnosing Alzheimer's disease using brain MRIs. The GAN is trained on only healthy brain MRI scans, and afterwards, an encoder was trained so that the model would be able to make reconstructions of brain MRI. To detect anomalies, it would reconstruct brain MRI scans and compare the reconstruction to the original scan, and it was able to yield a considerable level of performance.

The model, however, is limited in that it may be unable to capture subtle details in brain MRIs given its simple architecture, the small image sizes used to train it, and the wide variety of brain MRIs. Future work may then include exploring other GAN formulations and techniques to improve

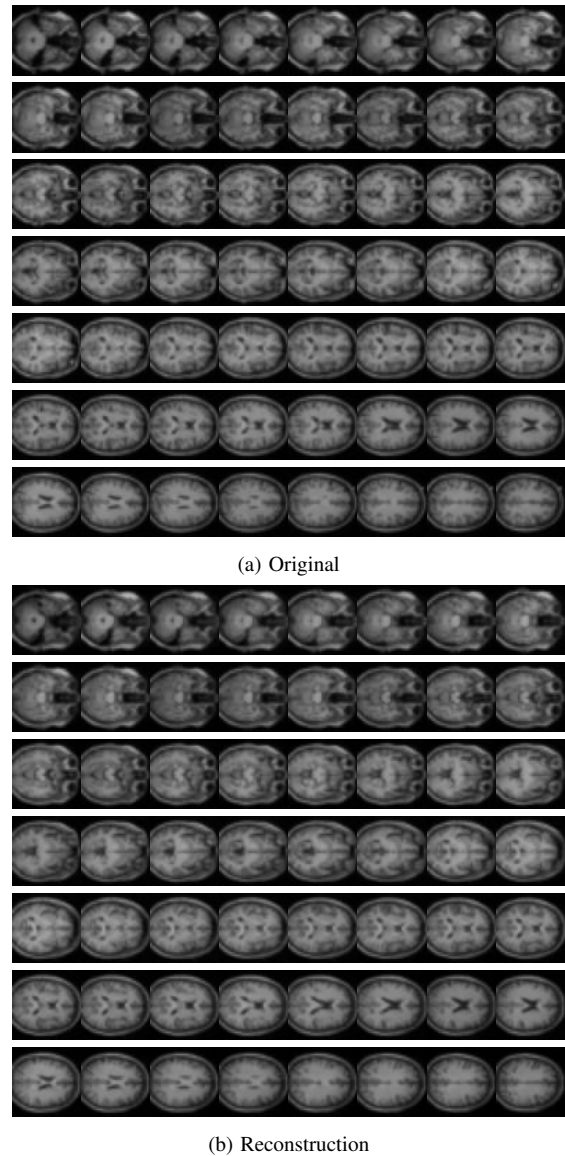


Fig. 2: Reconstruction of a normal, healthy brain MR scan.

image generation performance and using the other axes of the brain MRI. Exploring bigger GAN models would also be interesting, albeit at the cost of computational resources.

## ACKNOWLEDGMENT

Data were provided by OASIS-3: Principal Investigators: T. Benzinger, D. Marcus, J. Morris; NIH P50 AG00561, P30 NS09857781, P01 AG026276, P01 AG003991, R01 AG043434, UL1 TR000448, R01 EB009352. AV-45 doses were provided by Avid Radiopharmaceuticals, a wholly owned subsidiary of Eli Lilly.

## REFERENCES

- [1] R. L. Nussbaum and C. E. Ellis, "Alzheimer's disease and parkinson's disease," *New england journal of medicine*, vol. 348, no. 14, pp. 1356–1364, 2003.
- [2] L. Mucke, "Alzheimer's disease," *Nature*, vol. 461, no. 7266, pp. 895–897, 2009.
- [3] J. Weller and A. Budson, "Current understanding of alzheimer's disease diagnosis and treatment," *F1000Research*, vol. 7, 2018.

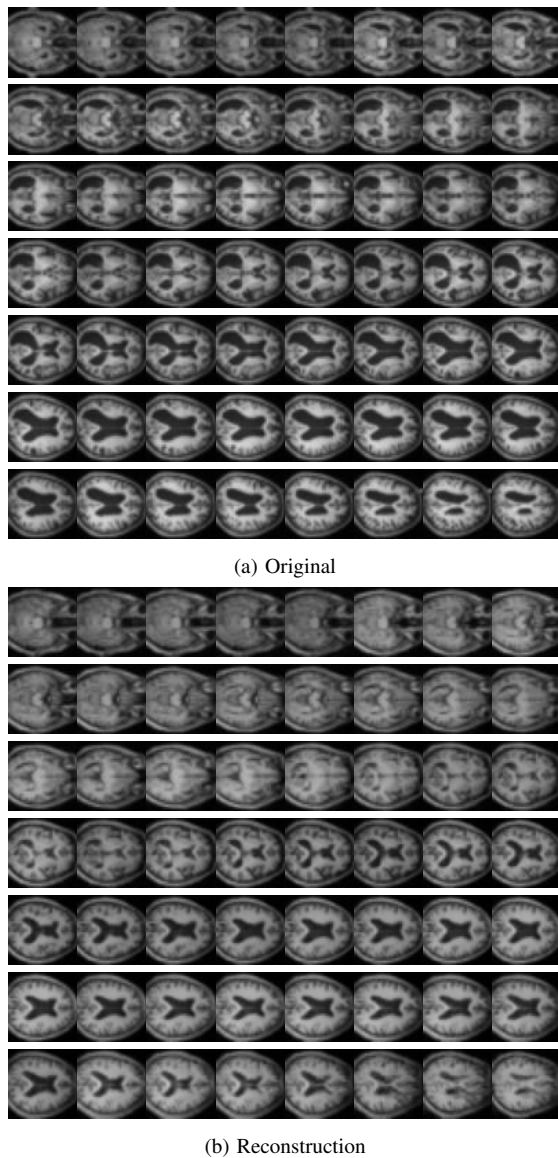


Fig. 3: Reconstruction of an anomalous brain MR scan.

[4] M. De Leon, S. DeSanti, R. Zinkowski, P. Mehta, D. Pratico, S. Segal, C. Clark, D. Kerkman, J. DeBernardis, J. Li *et al.*, "Mri and csf studies in the early diagnosis of alzheimer's disease," *Journal of internal medicine*, vol. 256, no. 3, pp. 205–223, 2004.

[5] P. Vemuri and C. R. Jack, "Role of structural mri in alzheimer's disease," *Alzheimer's research & therapy*, vol. 2, no. 4, pp. 1–10, 2010.

[6] C. Han, L. Rundo, K. Muraio, T. Noguchi, Y. Shimahara, Z. Á. Milacski, S. Koshino, E. Sala, H. Nakayama, and S. Satoh, "Madgan: unsupervised medical anomaly detection gan using multiple adjacent brain mri slice reconstruction," *BMC bioinformatics*, vol. 22, no. 2, pp. 1–20, 2021.

[7] T. Schlegl, P. Seeböck, S. M. Waldstein, G. Langs, and U. Schmidt-Erfurth, "f-anogan: Fast unsupervised anomaly detection with generative adversarial networks," *Medical image analysis*, vol. 54, pp. 30–44, 2019.

[8] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," *arXiv preprint arXiv:1901.03407*, 2019.

[9] H. Zenati, M. Romain, C.-S. Foo, B. Lecouat, and V. Chandrasekhar, "Adversarially learned anomaly detection," in *2018 IEEE International conference on data mining (ICDM)*. IEEE, 2018, pp. 727–736.

[10] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[11] X. Chen and E. Konukoglu, "Unsupervised detection of lesions in brain mri using constrained adversarial auto-encoders," *arXiv preprint arXiv:1806.04972*, 2018.

[12] K. M. van Hespén, J. J. Zwanenburg, J. W. Dankbaar, M. I. Geerlings, J. Hendrikse, and H. J. Kuijf, "An anomaly detection approach to identify chronic brain infarcts on mri," *Scientific Reports*, vol. 11, no. 1, pp. 1–10, 2021.

[13] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "Ganomaly: Semi-supervised anomaly detection via adversarial training," in *Asian conference on computer vision*. Springer, 2018, pp. 622–637.

[14] Z. Wang, J. Li, G. Song, and T. Li, "Less memory, faster speed: refining self-attention module for image reconstruction," *arXiv preprint arXiv:1905.08008*, 2019.

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[16] P. J. LaMontagne, T. L. Benzinger, J. C. Morris, S. Keefe, R. Hornbeck, C. Xiong, E. Grant, J. Hassenstab, K. Moulder, A. Vlassenko *et al.*, "Oasis-3: longitudinal neuroimaging, clinical, and cognitive dataset for normal aging and alzheimer disease," *MedRxiv*, 2019.

[17] A. A. Laurence, "Nifti image converter (nii2png) for python and matlab."

[18] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *International conference on machine learning*. PMLR, 2019, pp. 7354–7363.

[19] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *arXiv preprint arXiv:1802.05957*, 2018.

[20] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017.

[21] Y. Wu, M. Rosca, and T. Lillicrap, "Deep compressed sensing," in *International Conference on Machine Learning*. PMLR, 2019, pp. 6850–6860.

[22] S. Zhao, Z. Liu, J. Lin, J.-Y. Zhu, and S. Han, "Differentiable augmentation for data-efficient gan training," *arXiv preprint arXiv:2006.10738*, 2020.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[24] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," *arXiv preprint arXiv:1809.11096*, 2018.

[25] J. H. Lim and J. C. Ye, "Geometric gan," *arXiv preprint arXiv:1705.02894*, 2017.

# Heart Disease Prediction Using Adaptive Infinite Feature Selection and Deep Neural Networks

Sudipta Modak, *Member, IEEE* and Esam Abdel-Raheem, *Senior Member, IEEE*, Luis Rueda, *Senior Member, IEEE*  
Department of Electrical and Computer Engineering  
School of Computer Science  
University of Windsor  
401 Sunset Ave, Windsor, ON N9B 3P4, Canada  
E-mail: {modak, lrueda, eraheem}@uwindsor.ca

**Abstract**—Prediction of heart disease is one of the most important fields of study in modern science. By studying data such as cholesterol levels, blood sugar, and blood pressure, heart disease can be predicted. In recent years, several machine learning techniques have been used to aid in fast prediction by learning from the data. However, the prediction accuracy still remains low. This is due to lower number of records contained in the databases available. In this paper, we propose a new method of heart disease prediction using a modified variation of infinite feature selection and multilayer perceptron. The method shows a high accuracy of 87.70%, a high F1-score of 87.21%, a high sensitivity of 88.50%, a high specificity of 87.02%, and a high precision in prediction of 86.05% on the Cleveland, Hungarian, Switzerland, Long Beach, and Statlog datasets. For evaluation purposes, we have combined all the datasets together and then divided the combined dataset into training and test samples with a 20 % percent of the samples allocated for testing.

**Index Terms**—Heart disease prediction, Infinite feature selection, Multilayer perceptron, Neural Networks

## I. INTRODUCTION

Automatic disease detection, classification, and prediction have been important areas of research for several decades. For this purpose, several algorithms have been developed to aid doctors with accurate predictions of several types of diseases. One such field is that of heart disease, which is one of the biggest causes of death in the modern world [1]. By studying the patterns of the electrocardiogram (ECG) signals and correlating them to existing data, common anomalies in the heart can be identified. Several techniques for the detection of QRS complex exist with high accuracy such as the works in [2]–[5]. However, the problem with the prediction of heart diseases remains quite challenging due to the low number of records contained in the available databases.

By learning from the data available, machine learning models can predict these diseases at early stages. The attributes taken into account for such techniques can be obtained from an individual's body such as electrocardiogram, blood pressure, sugar levels, age, sex, cholesterol levels, etc [6]. However, there can be redundant features present in the datasets. These redundant features make predictions inaccurate and use up precious memory and time. A significantly large amount of data has been collected by the healthcare industry from previous cases of heart-related disease from patients all over

the world [1]. These datasets contain hidden information that is directly related to the condition of the heart and needs to be identified. Due to the presence of such a huge quantity of data, it is impossible to manually analyze them and create methods for prediction [6]. Therefore, machine learning techniques are required to deal with such data to predict diseases at early stages.

The work of [7] compares six different types of algorithms, including Linear, Quadratic, Cubic and Medium Gaussian support vector machines (SVM), as well as Decision Tree and Ensemble Subspace Discriminant for prediction accuracy. Deep learning has been used in the work of [8], where different combinations of a number of hidden layers and the number of epochs have been tested to learn which combination produces the best accuracy of prediction. Heart disease prediction using artificial neural networks can be found in [9]. This method uses six different classifiers to test the data and employs deep neural networks (DNN) for classification to achieve high accuracy in prediction. Similar algorithms can be found in [10], [11].

Feature selection techniques for heart disease prediction have been used in the works of [12]–[14]. In [12] an optimized version of genetic algorithms with SVM to achieve good accuracy in prediction, while in [13], a brute-force approach was used to select relevant features. That technique takes a small subset of features with at least three features and evaluates the combination of such features on several classifiers such as Logistic Regression (LR),  $k$ -nearest neighbour, decision tree, Naive Bayes, SVM, neural network, and vote. An integer-coded genetic algorithm has been used to select features as well [14]. That method aids the SVM-based prediction of heart disease and improves accuracy. A combination of the genetic algorithm and recursive feature elimination followed by a random forest classifier for better prediction has been presented in [15]. The work of [15] uses a genetic algorithm and recursive feature elimination to select relevant features from the data sets along with different classifiers for classification. Based on that work, the random forest classifier provides the best results when combined with the hybrid feature selection technique used in [15]. The work of [16] investigates the performance of several classifiers such as

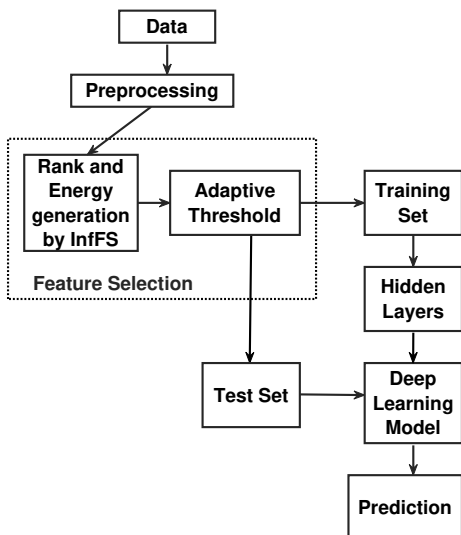


Fig. 1. Block diagram for the proposed method.

decision trees, Naive Bayes,  $k$ -nearest neighbor, and neural networks on heart disease prediction by varying the number of features provided as inputs. According to [16] the Naive Bayes classifier works best if the number of features is low.

This paper presents a new technique for prediction of heart disease using an advanced version of Infinite Feature Selection (InfFS) [17] and deep neural networks (DNN). The method is evaluated on five different datasets, namely, Cleveland, Hungary, Switzerland, Statlog, and Long Beach V [18]. A comparison with state-of-the-art methods in the field is included in this context as well.

## II. MATERIALS AND METHODS

The proposed method involves four stages, namely, preprocessing, which reads the raw data and converts it into usable quantities; a feature selection stage that selects a suitable subset of features and eliminates the redundant ones; a deep learning stage that is used to learn from the training datasets; finally, a prediction stage that predicts the outcomes from the test dataset. The block diagram for the entire process is summarized in Fig. 1.

### A. Dataset Description

The datasets used in the experiments contain 1,190 records from five distinct databases. It has 14 distinct features of which eight are categorical features and six are numeric features. The features are age, sex, chest pain type (cp), resting blood pressure (restbps), serum cholesterol (chol), fasting blood sugar greater than 120 mg/d (fbs), resting electrocardiographic results (restecg), maximum heart rate achieved (thalach), exercise-induced angina (exang), ST depression induced by exercise relative to rest (oldpeak), the slope of the peak exercise ST segment (slope), number of major vessels colored by fluoroscopy (ca), thallium scan (thal), and class attribute (num). Out of these 14 features 13 of them are taken as inputs to the proposed algorithm and class attribute (num)

is taken as the output. The aim is to design an algorithm that takes the 13 attributes and predicts the output. Since the data is labeled, the algorithm is regarded as a supervised learning algorithm.

### B. Pre-processing

The algorithm begins with combining all five datasets into one complete dataset. Any record with missing values was eliminated from across the five databases and therefore only 1,025 records out of 1,190 were used in this work. The reason for this is that, if more data is fed into a neural network then it can learn better. Similarly, any missing values will hamper the process of learning as it creates ambiguity in the learning mechanism. The output is also changed to binary output, that is anyone with no disease is regarded as '0', and an individual with heart disease is regarded as '1'. Originally, the data had classes from 0 to 3 which indicated the type of heart disease present in the individual with '0' being the absence of any heart disease and '1,' '2,' and '3' being the presence of three different types of diseases. For our work, we have considered only the binary case of not having any disease as '0' and the presence of disease as '1.'

### C. Feature Selection

A dataset of data might have hundreds of features of which many of them can be uncorrelated to the output of the data. The main objective of feature selection (FS) algorithms is to pick out a subset of variables from the input that directly influence the outcome of the data while reducing the noise and filtering out the unwanted variables from the data. For each dataset, a feature selection algorithm needs a particular selection criterion that can evaluate the applicability of each of the features on the output classes. Once this measure is calibrated, the irrelevant features are identified one by one and eliminated if they do not satisfy the conditions being imposed.

In our research, we have employed an improved version of InfFS to select a distinct subset of feature from the dataset. The method was initially developed in [17] and is used to map the features on an affinity graph as nodes and then connects them. It then considers moving from feature to feature and by doing so, it creates a path by selecting several features as a subset of the original list of features. Given a list of features,  $F$ , the algorithm considers two aspects of the features, the vertices,  $V$ , and the edges,  $E$ .  $V$  contains a set of values that represent a feature distribution for each of the values in  $F$ , while  $E$  represents the relations between two features in the feature distribution space [17]. An adjacency matrix  $A$ , containing all the pairwise energies, which is the maximal feature dispersion and correlation between two features [17], is formulated. Once all the features are mapped onto the graph based on their weights in  $A$ , several pathways are selected with more than two features at each iteration. These features are all connected nodes and the energies of each of these paths are calculated as follows:

$$a_{i,j} = \alpha \sigma_{i,j} + (1 - \alpha) c_{i,j}, \quad (1)$$

$$\xi_\gamma = \prod_{k=0}^{l-1} a_{v_k, v_{k+1}}, \quad (2)$$

where  $\alpha$  is a loading coefficient,  $\sigma$  is the standard deviation, and  $\xi_\gamma$  accounts for the pairwise energies of all the feature pairs that compose the path [17]. Here,  $l$  is the length of the path,  $i$  and  $j$  are the positions of the feature, while  $a$  is the feature. The cycles are recorded in  $R_l$ , that are computed as follows:

$$R_l(i, j) = \sum_{\gamma \in P^l(i, j)} \xi_\gamma = A^l(i, j), \quad (3)$$

where  $P^l(i, j)$  contains all the paths of length  $l$  between  $i$  and  $j$ . The single feature energy score  $s(i)$  is given by:

$$S(i) = \sum_{j \in V} R_l(i, j) = \sum_{j \in V} A^l(i, j), \quad (4)$$

and is equal to  $R_l(i, j)$ . The vector  $\mathbf{S}$  stores the feature energies individually. The feature energies are then arranged in decreasing order with the feature having the highest energy first and stored in vector  $\mathbf{M}$ .

Once all the feature rank energies are calculated, the algorithm automatically selects the number of features to keep, and hence some features are deemed redundant and are eliminated. A vector ( $\mathbf{B}$ ) is formulated, which stores the square of the individual rank energies. Equation 5 shows how each element in  $\mathbf{B}$  is calculated.

$$B(k) = M(k)^2. \quad (5)$$

Here,  $k$  is the element number in both  $\mathbf{B}$  and  $\mathbf{M}$ . Finally, a threshold of  $T$  is used to decide how many features to keep for the classification step. This threshold is calculated as follows:

$$T = C \sum_{k=1}^n B(k), \quad (6)$$

where  $n$  is the number of elements in  $\mathbf{B}$  and  $C$  is a constant taken as 0.325 for the entire dataset.

If the individual energy of a feature exceeds the threshold  $T$ , then that feature is kept, or else it is discarded. The remaining features are then arranged in the order of their energy values in vector  $\mathbf{M}$ . The number feature kept is denoted by  $N$ .

#### D. Classification

Artificial neural networks (ANN) are just like the neurons in a human brain [9]. Neural networks consist of several components such as neurons, synapses, weights, and biases. Deep neural networks (DNN) are an extension to ANN where multiple hidden layers are included to maximize the accuracy of decisions. These networks are feedforward networks, where the data always propagates in the forward direction, that is, in the direction of the output from the input and does not loop backward. Figure 2 shows an example of a DNN with one input and one output layer and two connected hidden layers.

For our research, we have used two hidden layers in the deep learning platform. The learning algorithm takes  $N$  inputs

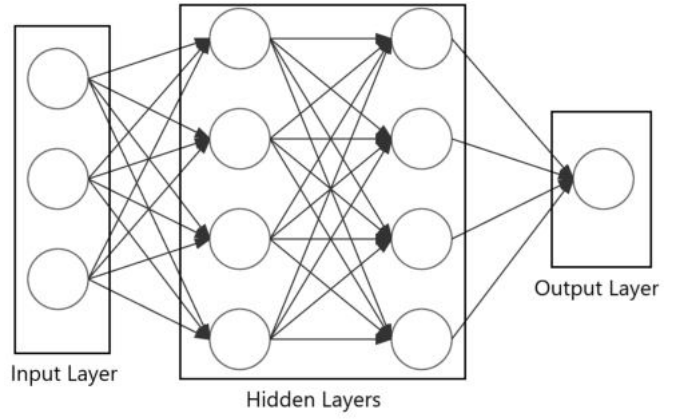


Fig. 2. Schematic view of a deep neural network.

from the FS stage and formulates the same number of neurons in the input stage. The training dataset is then fed to the neural network in batches of size 32. The activation function used for the input stage is the rectified linear activation function (Relu), which is a piecewise linear function that will transfer the input directly to the output if it is positive. However, if the input is negative then it will pass zero to the output. The hidden layers consist of eight neurons each and the activation function of the first hidden layer is also kept as 'relu'. The second hidden layer is initialized with the activation function of 'sigmoid', which is a logistic function that is represented by the following equation:

$$\phi(x) = \frac{1}{1 + E^{-x}}, \quad (7)$$

where  $E$  is Euler's number and  $x$  is the element number in the input range of the sigmoid function. The dropouts for all three layers, that is the input layer and the two hidden layers are 0.25 each.

The activation function used for the output layer is 'softmax' which is a normalized exponential function and provides the probability of obtaining a '0' and a '1' as the output. Equation 8 represents the 'softmax' function.

$$\gamma(\vec{z}) = \frac{e^z}{\sum_{q=1}^P e^{z_q}}. \quad (8)$$

Here,  $P$  is the number of classes in the multi-class classifier,  $\vec{z}$ , input vector  $z$  standard exponential function for the input vector, and  $z_l$  standard exponential function for the output vector.

### III. EXPERIMENT AND RESULTS

First, the datasets are combined into one database and then the combined database is divided into training and test datasets. The reason for this is that, the datasets all come from the same repository which is the UCI and according to other methods such as the one in [15] only the fourteen features mentioned in the dataset description of this paper are selected for training and testing. These fourteen features

are common for all five datasets and therefore they can be combined together into one comprehensive database. Now, the combined database is divided into separate segments for training the DNN and then testing the model. This is done by using five-fold cross-validation, that is the database is divided into five equal segments and out of the five one segment is kept as the test set and the rest four are used for training the neural networks. This process is repeated five times with a separate segment each time used for testing and the rest being used for training to cross-validate the results. Out of the 1,025 instances, each time, 205 samples are kept as the test samples which are later used to generate the metrics for the algorithm. Once, the design of the neural network is finalized using the above procedure, it is finally tested on the Cleveland database to ensure a fair comparison with other methods included those in the literature. The main metrics are *accuracy*, *sensitivity*, *specificity*, *precision*, and *F1* score which are represented by the following equations,

$$accuracy = \frac{tp + tn}{tp + tn + fp + fn}, \quad (9)$$

$$sensitivity = \frac{tp}{tp + fn}, \quad (10)$$

$$specificity = \frac{tn}{tn + fp}, \quad (11)$$

$$precision = \frac{tp}{tp + fp}, \quad (12)$$

$$F1 = \frac{tp}{tp + 0.5 * (fp + fn)}, \quad (13)$$

where  $tp$  is the number of true positives,  $tn$  is the number of true negatives,  $fn$  is the number of false negatives, and  $fp$  is the number of false positives.

Table I shows the results on the test samples of five different folds. The values of *accuracy*, *sensitivity*, *specificity*, *precision*, and *F1 – score* are included in the table. Figure 3 shows the test and training accuracy trends. The algorithm requires approximately 50 epochs to reach the desired weights for the DNN classifier, therefore there is not much need to increase the number of epochs as it might lead to overfitting. As shown in the graph, the training accuracy is close to 86% and stabilizes at this value at approximately epoch 45. The testing accuracy stabilizes at approximately 85 % which is roughly the same epoch number as the training accuracy. The figure shows the accuracy versus the number of epochs for Fold 5.

In addition, the proposed method is compared to five state-of-the-art methods in the field. The performances of these methods are collected from the papers mentioned in the literature. For a fair comparison, we have only used the Cleveland dataset, which has also been used by all methods for testing and comparison of metrics. The results are presented in Table II, while Figure 4 displays the comparison more vividly. It is observed that the proposed method outperforms the methods of Latha et al. [5], Gokulnath et al. [12], and

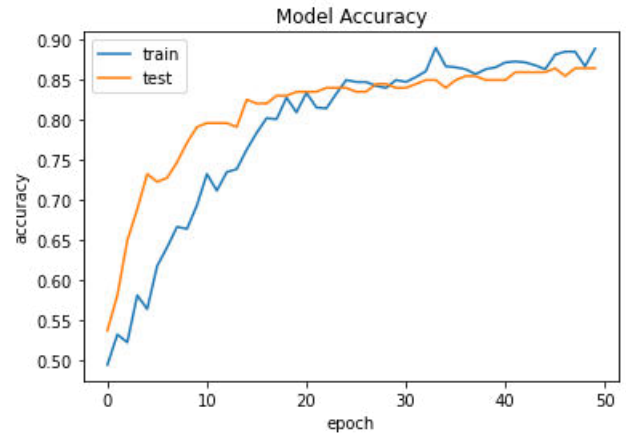


Fig. 3. Accuracy versus Epochs.

Rani et al. [15] in terms of accurate predictions. Similarly, the accuracy achieved by the proposed method is approximately equal to the method Amin et al. [13]. However, the method of Bharti et al. [9] shows higher accuracy than the proposed method. This method is evaluated further using merits such as sensitivity and specificity. Figures 5 and 6 illustrate the comparison of both quantities between the proposed method and the one in [9]. It is noticeable that the proposed method outperforms the method of Bharti et al. [9] in both cases with higher sensitivity and specificity. Furthermore, the method of Bharti et al. uses three dense layers with 128, 64, and 32 units in layers 1, 2, and 3, respectively. This increases the number of computations and time for processing significantly and makes the model quite complex. On the other hand, the proposed algorithm uses only 11, 8, and 8 units for the input, hidden layer 1 and hidden layer 2, respectively, and so has lesser time complexity. Considering this, it is safe to say that the proposed algorithm performs better than all methods included in the context.

#### IV. CONCLUSION

We have presented a new method of heart disease prediction using adaptive infinite feature selection and deep neural networks. The proposed method has shown high accuracy of prediction on five datasets which include Cleveland, Hungary, Switzerland, Statlog, and Long Beach V. The proposed method has been tested on different folds of data and shows high accuracy in terms of prediction of heart disease. In the future, the proposed method can be tested with eigenvector centrality for feature selection and more neural network-based classifiers can be implemented such as graph neural networks. The work can also be extended to the prediction of more acute and chronic diseases such as anemia, diabetes, and tumors.

#### ACKNOWLEDGEMENT

This work has been partially supported by a grant provided by the Natural Science and Engineering Council of Canada (NSERC). The authors would also like to thank the Office

TABLE I  
METRICS OF EVALUATION ON THE TEST SET PER FOLD.

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average
<b>Accuracy (%)</b>	90.24	88.29	90.24	83.41	84.91	87.70
<b>Sensitivity (%)</b>	92.63	84.69	92.55	84.38	88.24	88.50
<b>Specificity (%)</b>	88.18	91.59	88.29	82.57	84.47	87.02
<b>Precision (%)</b>	87.13	90.22	87.00	81.00	84.91	86.05
<b>F1-Score (%)</b>	89.80	87.37	89.69	82.65	86.54	87.21

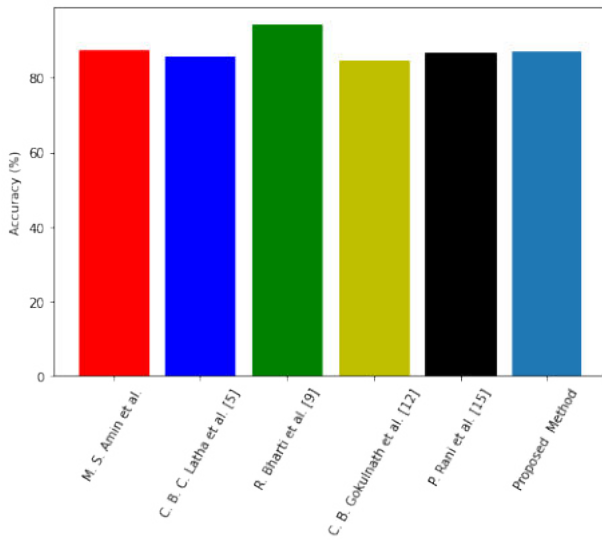


Fig. 4. Comparison of accuracy with state-of-the-art methods.

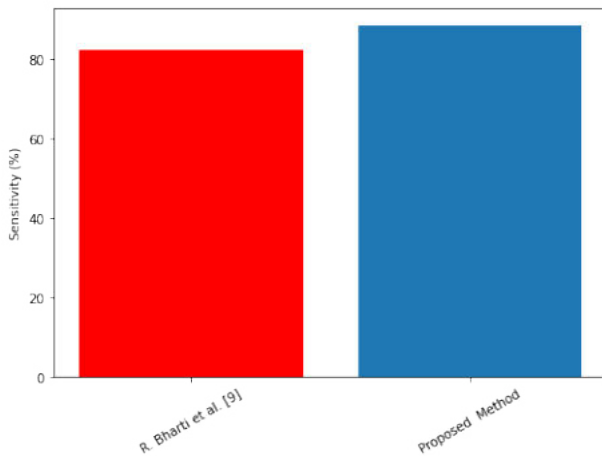


Fig. 5. Comparison of sensitivity with Bharti et al. [9].

TABLE II  
COMPARISON WITH THE STATE-OF-THE-ART METHODS ON CLEVELAND DATABASE.

Method	Year	Accuracy (%)
M. S. Amin et al. [13]	2019	87.41
C. B. C. Latha et al. [6]	2019	85.48
R. Bharti et al. [9]	2021	94.20
C. B. Gokulnath et al. [12]	2019	84.40
P. Rani et al. [15]	2021	86.60
<b>Proposed Method</b>	<b>2021</b>	<b>87.13</b>

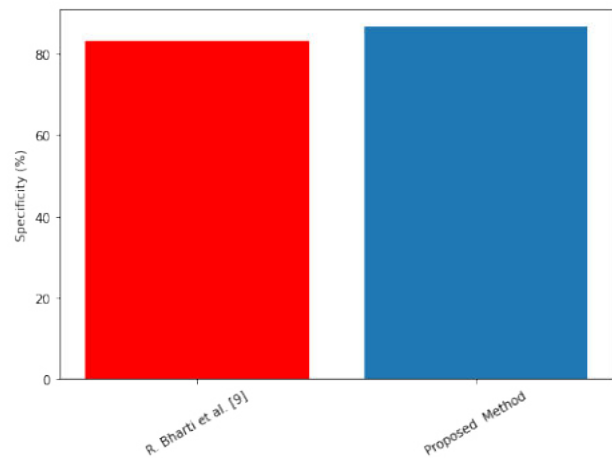


Fig. 6. Comparison of specificity with Bharti et al. [9].

of Research and Innovation Services of the University of Windsor.

## REFERENCES

- [1] R. Chitra and V. Seenivasagam, "Review of heart disease prediction system using data mining and hybrid intelligent techniques," *ICTACT journal on soft computing*, vol. 3, no. 04, pp. 605–09, 2013.
- [2] S. Modak, L. Y. Taha, and E. Abdel-Raheem, "A novel method of qrs detection using time and amplitude thresholds with statistical false peak elimination," *IEEE Access*, vol. 9, pp. 46 079–46 092, 2021.
- [3] S. Modak, E. Abdel-Raheem, and L. Y. Taha, "A novel adaptive multilevel thresholding based algorithm for qrs detection," *Biomedical Engineering Advances*, p. 100016, 2021.



- [4] S. Modak, L. Y. Taha, and E. Abdel-Raheem, "Single channel qrs detection using wavelet and median denoising with adaptive multilevel thresholding," in *2020 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*. IEEE, 2020, pp. 1–6.
- [5] Z. Zhang, Q. Yu, Q. Zhang, N. Ning, and J. Li, "A kalman filtering based adaptive threshold algorithm for qrs complex detection," *Biomedical Signal Processing and Control*, vol. 58, p. 101827, 2020.
- [6] C. B. C. Latha and S. C. Jeeva, "Improving the accuracy of prediction of heart disease risk based on ensemble classification techniques," *Informatics in Medicine Unlocked*, vol. 16, p. 100203, 2019.
- [7] S. Ekız and P. Erdođmuş, "Comparative study of heart disease classification," in *2017 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT)*. IEEE, 2017, pp. 1–4.
- [8] P. Ramprakash, R. Sarumathi, R. Mowriya, and S. Nithyavishnupriya, "Heart disease prediction using deep neural network," in *2020 International Conference on Inventive Computation Technologies (ICICT)*. IEEE, 2020, pp. 666–670.
- [9] R. Bharti, A. Khamparia, M. Shabaz, G. Dhiman, S. Pande, and P. Singh, "Prediction of heart disease using a combination of machine learning and deep learning," *Computational Intelligence and Neuroscience*, vol. 2021, 2021.
- [10] A. Khemphila and V. Boonjing, "Heart disease classification using neural network and feature selection," in *2011 21st International Conference on Systems Engineering*. IEEE, 2011, pp. 406–409.
- [11] Y. E. Shao, C.-D. Hou, and C.-C. Chiu, "Hybrid intelligent modeling schemes for heart disease classification," *Applied Soft Computing*, vol. 14, pp. 47–52, 2014.
- [12] C. B. Gokulnath and S. Shantharajah, "An optimized feature selection based on genetic approach and support vector machine for heart disease," *Cluster Computing*, vol. 22, no. 6, pp. 14777–14787, 2019.
- [13] M. S. Amin, Y. K. Chiam, and K. D. Varathan, "Identification of significant features and data mining techniques in predicting heart disease," *Telematics and Informatics*, vol. 36, pp. 82–93, 2019.
- [14] S. Bhatia, P. Prakash, and G. Pillai, "Svm based decision support system for heart disease classification with integer-coded genetic algorithm to select critical features," in *Proceedings of the world congress on engineering and computer science*, 2008, pp. 34–38.
- [15] P. Rani, R. Kumar, N. M. S. Ahmed, and A. Jain, "A decision support system for heart disease prediction based upon machine learning," *Journal of Reliable Intelligent Environments*, pp. 1–13, 2021.
- [16] T. J. Peter and K. Somasundaram, "An empirical study on prediction of heart disease using classification data mining techniques," in *IEEE-International conference on advances in engineering, science and management (ICAESM-2012)*. IEEE, 2012, pp. 514–518.
- [17] G. Roffo, S. Melzi, and M. Cristani, "Infinite feature selection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4202–4210.
- [18] D. Dua, C. Graff *et al.*, "Uci machine learning repository," 2017.

# A federated binarized neural network model for constrained devices in IoT healthcare services

Hyeontaek Oh

Institute for Information Technology  
Convergence, KAIST, Daejeon,  
Rep. of Korea, 34141  
Email: hyeontaek@kaist.ac.kr

Jongmin Yu

Institute for Information Technology  
Convergence, KAIST, Daejeon,  
Rep. of Korea, 34141  
Email: andrew.yu@kaist.ac.kr

Nakyoung Kim

Institute for Information Technology  
Convergence, KAIST, Daejeon,  
Rep. of Korea, 34141  
Email: nkim71@kaist.ac.kr

Dongyeong Kim

Institute for Information Technology  
Convergence, KAIST, Daejeon,  
Rep. of Korea, 34141  
Email: deathquin@kaist.ac.kr

Jangwon Lee

Institute for Information Technology  
Convergence, KAIST, Daejeon,  
Rep. of Korea, 34141  
Email: walker0723@kaist.ac.kr

Jinhong Yang

Department of Healthcare Information  
Technology, Inje University,  
Gimhae, Rep. of Korea, 50834  
Email: jinhong@inje.ac.kr

**Abstract**—In IoT healthcare environment, the devices are not sufficiently powerful for operating recent deep learning models, and data collected by the devices are usually decentralized. Moreover, data are unavailable to share between devices because of information security issues. Therefore, a concept of federated learning has emerged to overcome data sharing issues, and a concept of binarized neural network has emerged to generate lightweight deep learning models. This paper proposes a federated binarized neural network model to derive a reliable healthcare system in this circumstance. This paper shows an overview of considered system model with constrained IoT healthcare devices. In addition, this paper shows illustrations of implementing the proposed federated learning model with the proposed binarized MLP networks by utilizing an open-source library. The experiment results show that the binarized MLP network shows comparable performances compared to the full-precision MLP network while the binarized MLP requires about 10-times less model size for training.

**Index Terms**—Federated learning, Binarized neural network, Internet of Things, Healthcare

## I. INTRODUCTION

Today, with a huge development of machine learning (ML) and artificial intelligence (AI) techniques, the application of AI/ML is inevitable to all IT based applications and services. Particularly, the emergence of deep learning has accelerated the innovations in all IT related areas [1], [2].

Deep learning based AI techniques require the entire dataset as centralized manner. However, these kinds of centralized approaches have raised several issues, such as data security and privacy. Particularly, to preserve privacy of users, the necessity of distributed approaches for AI/ML techniques has increased. Therefore, a concept of federated learning has been emerged [3], [4]. Unlike to centralized AI models, in federated learning environment, each device contains AI models for train

and inference. The device sends AI models, including detailed information of parameter weights. Since data from the device are not shared to public, using federated learning becomes more popular in industrial sectors, particularly, that handles various personal data such as healthcare.

With the advantages of federated learning, various approaches have been proposed in IoT based healthcare services [5]–[7]. Particularly, Wu *et al.* [8] proposed a personalized federated learning model for in-home health monitoring by exploiting generative convolutional autoencoder in cloud-edge computing environment. Chen *et al.* [9] proposed a federated transfer learning framework for wearable healthcare devices. These kinds of approaches produced remarkable performances; however, deep neural network based federated learning models have a very high complexity model in general. Therefore, it may not suitable for constrained devices. In IoT healthcare environments, the complexity of AI/ML models becomes an issue because many applications have utilized constrained devices.

To overcome this issue, as one of possible approaches to reduce the complexity of AI model, a concept of binarized (or binary) neural network has emerged [10]. Since the concept of binarized neural network can significantly reduce size and complexity of deep neural networks, it has been mainly considered in image processing area [11] to reduce computational complexity of convolutional neural network. In IoT environments, few studies have leveraged BNN. Using the concept of binarized neural network, this paper proposes a binarized multi-layer perceptron model for constrained devices in IoT environment. Cerutti *et al.* [12] proposed a sound event detection model based on binary neural network for power-constrained IoT devices. Verca *et al.* [13] proposed a method for detecting network intrusion with binarized neural network that can be utilized in embedded IoT devices.

Based on these backgrounds, this paper proposes a framework with federated binarized neural network model in IoT

This research was financially supported by the Ministry of Trade, Industry and Energy (MOTIE) and Korea Institute for Advancement of Technology (KIAT) through the International Cooperative R&D program. (Project No. 0011879). Corresponding author: Jongmin Yu (andrew.yu@kaist.ac.kr).

healthcare environments with various constrained devices. The contributions of this paper are summarized as follows:

- This paper proposes a federated learning based binarized neural network model for considering the characteristics of constrained devices in IoT healthcare environments.
- This paper proposes a binarized multi-layer perceptron (MLP) network model for handling IoT data. It also illustrates a implementation procedures of the proposed federated binarized MLP using an open-source [14].
- Using the real-world dataset [15] captured by a radar-based contactless biometric monitoring testbed, this paper shows the performance of the proposed binarized neural network model in federated learning environment. The results show that the proposed binarized neural network model performs comparable performance, compared with the full-precision model, with 10-times less model size for training.

## II. SYSTEM MODEL

This paper considers a healthcare monitoring system consisting of a global server and  $N$  distributed gateways deployed at users' house. Each gateway manages various health/biometric monitoring devices in the house. Each monitoring device collects various healthcare data from users. The collected data are transferred to the local gateway device. Figure 1 shows an illustration of the proposed federated learning framework.

### A. Federated learning for IoT devices

For applying federated learning based AI applications, both local clients and global server share the same neural network structures. To update the model parameters, for each communication round, a federated learning is operated as follows.

- 1) Local clients utilize the data collected from monitoring devices to train local neural network model.
- 2) When a local model training is done, the trained model is sent to the global server. At this time, some information such as general statistics of dataset is also transferred for applying a federated learning algorithm in the global server; however, no raw data are directly transferred to the global server.
- 3) When the global server aggregates all local models from the gateways, the server combines the local models into one single global model and validates the new global model.
- 4) After the validation, the newly updated global model is sent back to all local gateways so that local clients can also update their local model.

A federated learning iteratively performs to improve the performance of both local and global model.

In the proposed model, both clients and server have deep neural network based model. The primary objective of federated learning with deep neural network is to minimize risk, which means how accurately achieve a global objective by combining the results from local objective functions. Let  $F^k$  and  $D_k$  denote the local objectives and the set of indexes of

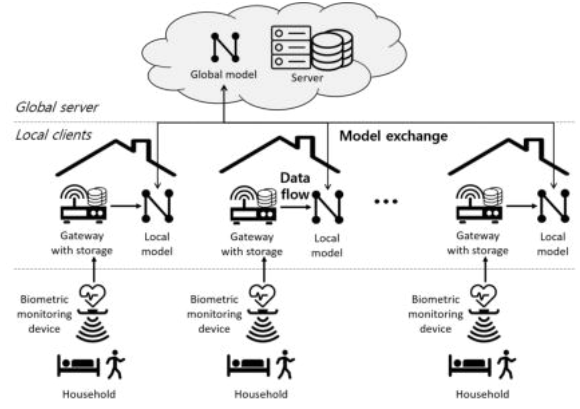


Fig. 1. An overview of considered IoT healthcare environment

local data on device  $k$ . Then, the risk minimization problem or federated learning can be shown as follows [16].

$$\min_{\omega \in \mathcal{R}_d} f(\omega) = \sum_{k=1}^K \frac{n_k}{n} F^k(\omega), \text{ where } F^k(\omega) = \frac{1}{n_k} \sum_{i=D_k} F_i^k(\omega). \quad (1)$$

In equation (1),  $n = \sum_{k=1}^K n_k$  means the total number of samples, where  $K$  is the number of active devices participating in federated learning. The global objective  $f(\omega)$  in federated learning is able to be represented as a linear sum of the local objectives  $F_k(\omega)$ , where  $\omega$  denotes the parameters of the model (e.g., the weights and bias in a deep neural network). Each local objective is defined by averaging the outputs of the local objectives with respect to each local dataset:  $F^k(\omega) = \frac{1}{n_k} \sum_{i=D_k} F_i^k(\omega)$ .

*FederatedAverage* (FedAvg) algorithm, which is one of the most well-known approaches in federated learning, is utilized as the fundamental framework in federated learning setting [16]. FedAvg utilizes an iterative model averaging scheme with collected local stochastic gradient descent (SGD) from local devices. In each iteration  $t$ , each device  $k$  calculates the average gradient  $g_k = \nabla F_k(\omega_t)$  of the local model at the current model parameters  $\omega_t$ . Then, the server aggregates all of gradients from devices and updates the global as follows,

$$\omega_{t+1} \leftarrow \omega_t - \eta \nabla f(\omega_t), \text{ where } \nabla f(\omega_t) = \sum_{k=1}^K \frac{n_k}{n} g_k. \quad (2)$$

This paper also adopts the concept of FedAvg algorithm for performing a federated learning with a binarized neural network. The global model in server can be derived by using FedAvg algorithm.

After the global model is learned, the model can be applied to the local devices. However, if the global model is directly overwritten to local models, each local model has lost its locality that contains the characteristics of individual. Therefore, in this paper, the local model is updated as follows

$$\omega_{k,t+1} = \omega_{G,t+1} + \lambda \omega_{k,t}. \quad (3)$$

In equation (3),  $\omega_{G,t+1}$  means the results of FedAvg in global server at time  $t + 1$ , and  $\omega_{k,t}$  and  $\omega_{k,t+1}$  indicate model

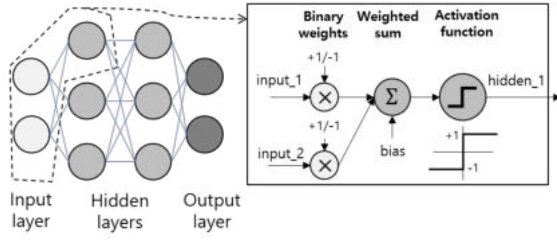


Fig. 2. A concept of binarized neural network

parameters of a local client  $k$  at time  $t$  and  $t + 1$ , respectively. Here,  $\lambda$  is a balance weight for the local model parameters at the previous time frame. If  $\lambda = 0$ , it ignores the local model parameters.

### B. Binarized neural network model

Binarized neural network (BNN) is a deep neural network that has weight parameters only consists of either  $-1$  or  $1$  [17]. Figure 2 shows a concept of BNN. When an input  $x$  is quantized into either  $-1$  or  $1$  as follows.

$$q(x) = \begin{cases} 1 & \text{if } x \geq 0, \\ -1 & \text{if } x < 0, \end{cases} \quad (4)$$

where  $x$  is a weight parameter of a original model and  $q(x)$  is an activation function to make binarized neural network.

In general, each weight parameter of a neural network model is represented as floating number. Therefore, it requires at least 32-bit for storing one weight parameter of the model. However, in a BNN, it requires only 1-bit for storing the weight parameter because the weight is either  $-1$  or  $1$ . Therefore, theoretically, BNN can consume 32-times lower storage and communication costs. Since the communication costs are expensive in federated learning, these kinds of cost reduction is very helpful to operating IoT systems, particularly, consists of many constrained devices. For example, the throughput of LoRaWAN varies from 300 bps to 37.5 Kbps [18].

Various neural network models can be applied to IoT environment. In this paper, a binarized multi-layer perceptron (MLP) network is utilized to train and predict healthcare datasets from constrained devices. The reason to choose MLP is that the dimensionality of input data from constrained devices is relatively lower than other environments.

By using the proposed federated learning framework and binarized MLP network, the next section shows how the proposed federated binarized neural network model is implemented for experiments with the real-world dataset from IoT healthcare environment.

## III. IMPLEMENTATION

For implementing a binarized MLP and a federated learning process, this paper utilizes Python, *Tensorflow*, and an open-source *Larq* [14] that provides BNN library.

```

1 class BNN:
2     @staticmethod
3     def build(attributes, classes):
4         kwargs = dict(input_quantizer="ste_sign",
5                       kernel_quantizer="ste_sign",
6                       kernel_constraints="weight_clip",
7                       use_bias=False)
8         model = tf.keras.models.Sequential()
9         model.add(larq.layers.QuantDense(64, kernel_quantizer="ste_sign",
10                                         kernel_constraint="weight_clip",
11                                         use_bias=False, input_shape=(attributes,),
12                                         ))
13         model.add(tf.keras.layers.BatchNormalization(momentum=0.9, scale=False))
14         model.add(larq.layers.QuantDense(32, **kwargs))
15         model.add(tf.keras.layers.BatchNormalization(momentum=0.9, scale=False))
16         model.add(larq.layers.QuantDense(16, **kwargs))
17         model.add(tf.keras.layers.BatchNormalization(momentum=0.9, scale=False))
18         model.add(larq.layers.QuantDense(classes, **kwargs))
19         model.add(tf.keras.layers.BatchNormalization(momentum=0.9, scale=False))
20         model.add(tf.keras.layers.Activation("softmax"))
21         return model

```

Fig. 3. An illustration of binarized MLP using *Larq*

```

1 with larq.context.quantized_scope(True):
2     global_model = BNN().build(attributes, classes)
3
4 for comm_round in range(EPOCHS):
5     global_weights = global_model.get_weights()
6     scaled_local_weight_list = []
7
8     for device_id in devices:
9         bnn_local = BNN()
10        local_model = bnn_local.build(attributes, classes)
11        local_model.compile(optimizer='adam',
12                          loss='categorical_crossentropy', metrics=['accuracy'])
13
14        weights = []
15        lw = scale_model_weights(local_model.get_weights(), 1)
16        weights.append(lw)
17        gw = scale_model_weights(global_model.get_weights(), 0.2) # lambda
18        weights.append(gw)
19        local_weights = sum_scaled_weights_bin(weights)
20        local_model.set_weights(local_weights)
21
22        dataset = devices_dataset[device_id]
23        local_model.fit(dataset['X_train'],
24                      tf.keras.utils.to_categorical(dataset['y_train'],
25                                                    num_classes=classes),
26                    batch_size=32, epochs=1, verbose=0)
27
28        local_models[device_id] = local_model
29
30        scaling_factor = weight_scaling_factor(device_id)
31        scaled_weights = scale_model_weights(local_model.get_weights(),
32                                          scaling_factor)
33        scaled_local_weight_list.append(scaled_weights)
34
35        y_pred = np.argmax(local_model.predict(dataset['X_test']), axis=1)
36        K.clear_session()
37
38    average_weights = sum_scaled_weights_bin(scaled_local_weight_list)
39
40    global_model.set_weights(average_weights)
41    global_model.compile(optimizer='adam',
42                      loss='categorical_crossentropy', metrics=['accuracy'])
43
44    y_pred = np.argmax(global_model.predict(global_dataset['X_test']), axis=1)

```

Fig. 4. An illustration of federated learning using *Tensorflow* with *Larq*

### A. Binarized Multi-Layer Perceptron

Figure 3 shows an implementation of the proposed binarized MLP network using *Larq* library. Since the dimension of data is relatively low, a MLP network is adopted on both the server and the devices at users' house for model training and prediction. The dimension of input is 14 (from a dataset [15]). The MLP network is composed of 4 fully-connected layers with 64, 32, 16, and 6 (i.e., the number of classes) units for extracting features from input data. A binary quantizer (which can be used as activation function in BNN), called SteSign quantizer with a standard weight clip constraint is used. Using this quantizer, the gradient is estimated using the straight-through estimator. *Softmax* is adopted in the final fully-connected layer of the MLP network to calculate the probability of the output results. According to the guideline from [14] inspired by [19], batch normalization layers with momentum of 0.9 are inserted between network layers.

### B. Federated learning with binarized MLP

Figure 4 shows an implementation of the proposed federated learning framework for the simulation. The implementation of

TABLE I  
ACCURACY AND F1-SCORE OF VARIOUS METHODS

Model	Accuracy	F1-score	Model Size
Fed-Bin-MLP (global)	0.783	0.783	1.36 KB
Fed-Full-MLP (global)	0.802	0.802	14.80 KB
Fed-Bin-MLP (local avg.)	0.843	0.843	1.36 KB
Fed-Full-MLP (local avg.)	0.833	0.833	14.80 KB

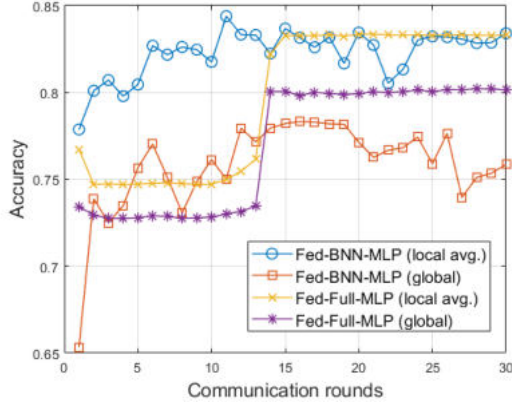


Fig. 5. A trend of accuracy for federated MLP models

federated learning is inspired by [20]. The detailed implementation is follows.

Line 1 means the model utilizes the binary weights (i.e., either  $-1$  or  $1$ ). First, global model is initialized in line 2. A variable in line 6 uses to calculate scales weights for applying equations (1) and (2).

For each local device, local model learning is performed from line 8. From line 9 to 11, it shows a local model initialization. The model is trained by Adam optimizer with a loss function of categorical cross-entropy.

From line 14 to 20, it represents the proposed local weight update procedure in equation (3). The balance weight  $\lambda$  is set to 0.2 tuned using cross-validation procedure.

From line 22 to 28, it loads a local datasets of each device and trains a local binarized MLP network model. Batch size for training local models is set to 32. In line 28, the trained local model is saved for performing the proposed local weight update procedure described above.

From line 30 to 33, it performs weight calculation for federated learning according to equation (1). After local model training is done, from line 39, the global binarized MLP is updated according to equation (2).

#### IV. EXPERIMENTS

This section shows various simulation results of the proposed federated binarized MLP model compared with various methods. For the simulation, this paper refers a dataset from [15] that consists of various biometric information (e.g., heart rate, breath rate, the intensity of movement etc.) collected from radar-based contactless biometric monitoring testbed.

The total number of participants considered in this experiment is 20. The dataset contains 14 attributes (e.g., heart

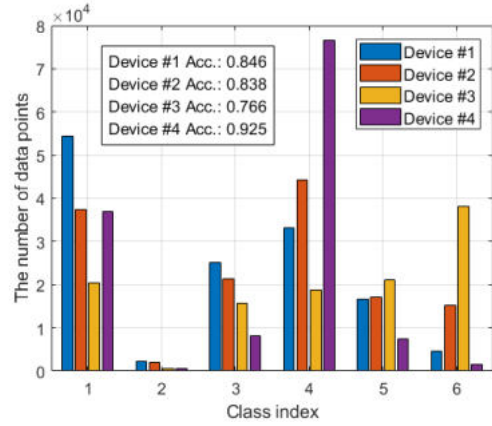


Fig. 6. Data distribution of selected devices and their prediction accuracies for federated binarized MLP

rate, breath rate, etc.) collected from radar-based contactless biometric monitoring system [15]. The total number of classes is 6 that represents the status of users. Data with *Null* element are removed for experiments. Consequently, each user has the average 100K data points. Each dataset from user has non-IID characteristics. A min-max scalar is applied for the entire dataset. For binarized network, the data additionally normalized for setting the value in  $[-1, 1]$ . Since it is a multi-class classification problem, accuracy and f1-score (which calculates metrics globally by counting the total true positives, false negatives and false positives) are chosen as performance metrics. RTX 2070 SUPER is used for these experiments.

Table I shows the performance of the proposed federated binarized MLP model (i.e., Fed-Bin-MLP) compared with federated MLP with full precision networks (i.e., Fed-Full-MLP). In the table, “global” means the performance of the global model and “local avg.” means the averaged performance of all local models. As shown in Table I, full precision MLP shows a better performance than that of binarized MLP. Fed-Full-MLP shows accuracy and f1-score of 0.802 for the global model and those of 0.833 as average for local models, respectively. On the other hand, Fed-Bin-MLP shows accuracy and f1-score of 0.783 for the global model and those of 0.843 as average for local models. The values represent that MLP networks with either binary precision or full precision show similar performance. However, as shown in the table, a model size of Fed-Bin-MLP is about 1.4 KB while that of Fed-Full-MLP is about 15 KB.

Since constrained IoT devices produce relatively lower complex data (i.e., data with a lower dimensionality) compared to other devices, the binarized MLP network also can capture enough discriminative information compared with the MLP network with full precision weights. In addition, since local models is updated using raw datasets, the overall performance of local models is higher than that of the global model for both MLPs with binary and full precision weights.

Meanwhile, Table I shows the best case for each model, so it is necessary to check some trends of accuracy changed during the communication rounds. Figure 5 shows a trend of accuracy

for federated MLP models with respect to communications rounds (i.e., as time goes). Basically, when time goes (i.e., the number of communication rounds is increased), overall performance of the federated MLP models increases. However, the trends of binarized MLP and full-precision MLP networks are a little different. The full-precision MLP networks show stable results. Both local and global models show significant performance improvement at a certain points. On the other hand, the binarized MLP networks show relatively unstable results. Since the binarized MLP contains less information in model parameters, it is more sensitive to variations of input data. Therefore, the accuracy values of both local and global models are fluctuated and sometimes overfitted.

Lastly, since this paper handles non-iid datasets, 4 devices, which show the different range of accuracy for local models, are chosen for microscopic analysis. Figure 6 shows a data distribution of the selected devices (i.e., the number of data points of each class) and the accuracy performance of the devices with federated binarized MLP. As shown in the figure, each device collects data with different characteristics. For example, the largest amount of data from device 1 is class 1, but that from device 4 is class 4. The device 3, which has the largest amount data of class 6, shows the lowest performance among 4 devices. These kinds of non-iid characteristics significantly impacts on the performance of local and global models.

Since the global model can contain more generalized information from all local clients, therefore, it is possible to provide better performances for general circumstances. Meanwhile, from a local client points of view, generalization of the model means that it loses some local specific features for customization. Therefore, it is necessary that some kinds of adaptation techniques for both local and global models to improve the overall performance of the entire systems. Not only the characteristics of non-iid data environments but also the input sensitive characteristics of binarized neural networks should be considered. In future, these kinds of issues should be considered in future to improve overall performance of a federated binarized neural networks.

## V. CONCLUSION

This paper has proposed a federated binarized neural network model that can be utilized in IoT healthcare environments with various constrained devices. For the proposed federated binarized neural network model, the detailed methods for model exchange process and a binarized multi-layer perceptron (MLP) network model are proposed. With implementation of the proposed federated learning model with the proposed binarized MLP networks, the experiment results show that the binarized MLP network shows comparable performances compared to the full-precision MLP network while the binarized MLP requires much less model size for training. To improve the overall performance of a federated binarized neural network for IoT healthcare environments, various issues should be more considered in future such as local-global model adaptation, handling non-iid datasets, more input stable binarized neural network models.

## REFERENCES

- [1] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Commun. Surv. Tutor.*, vol. 22, no. 2, pp. 869–904, 2020.
- [2] W. Chen, X. Qiu, T. Cai, H.-N. Dai, Z. Zheng, and Y. Zhang, "Deep reinforcement learning for internet of things: A comprehensive survey," *IEEE Commun. Surv. Tutor.*, vol. 23, no. 3, pp. 1659–1692, 2021.
- [3] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. Vincent Poor, "Federated learning for internet of things: A comprehensive survey," *IEEE Commun. Surv. Tutor.*, vol. 23, no. 3, pp. 1622–1658, 2021.
- [4] A. Imteaj, U. Thakker, S. Wang, J. Li, and M. H. Amini, "A survey on federated learning for resource-constrained iot devices," *IEEE Internet of Things Journal*, pp. 1–1, 2021.
- [5] N. Rieke, J. Hancox, W. Li, F. Milletari, H. R. Roth, S. Albarqouni, S. Bakas, M. N. Galtier, B. A. Landman, K. Maier-Hein, S. Ourselin, M. Sheller, R. M. Summers, A. Trask, D. Xu, M. Baust, and M. J. Cardoso, "The future of digital health with federated learning," *npj Digital Medicine*, vol. 3, no. 1, p. 119, Sep 2020.
- [6] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, and F. Wang, "Federated learning for healthcare informatics," *Journal of Healthcare Informatics Research*, vol. 5, no. 1, pp. 1–19, Mar 2021.
- [7] L. Yang, K. Yu, S. X. Yang, C. Chakraborty, Y. Lu, and T. Guo, "An intelligent trust cloud management method for secure clustering in 5g enabled internet of medical things," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2021.
- [8] Q. Wu, X. Chen, Z. Zhou, and J. Zhang, "Fedhome: Cloud-edge based personalized federated learning for in-home health monitoring," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2020.
- [9] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, "Fedhealth: A federated transfer learning framework for wearable healthcare," *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 83–93, 2020.
- [10] H. Qin, R. Gong, X. Liu, X. Bai, J. Song, and N. Sebe, "Binary neural networks: A survey," *Pattern Recognition*, vol. 105, p. 107281, 2020.
- [11] T. Simons and D.-J. Lee, "A review of binarized neural networks," *Electronics*, vol. 8, no. 6, 2019.
- [12] G. Cerutti, R. Andri, L. Cavigelli, E. Farella, M. Magno, and L. Benini, "Sound event detection with binary neural networks on tightly power-constrained iot devices," in *Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design*, ser. ISLPED '20. New York, NY, USA: ACM, 2020, p. 19–24.
- [13] J. Vreća, I. Ivanov, G. Papa, and A. Biasizzo, "Detecting network intrusion using binarized neural networks," in *2021 IEEE 7th World Forum on Internet of Things (WF-IoT)*, 2021, pp. 622–627.
- [14] L. Geiger and P. Team, "Larq: An Open-Source Library for Training Binarized Neural Networks," *Journal of Open Source Software*, vol. 5, no. 45, p. 1746, Jan. 2020. [Online]. Available: <https://doi.org/10.21105/joss.01746>
- [15] H. Oh, J. Yu, C. J. Jung, and J. K. Choi, "Data Analysis and Consideration of Radar-Based Contactless Biometrics Monitoring Testbed for Single Elderly Households," *The Journal of Korean Institute of Communications and Information Sciences*, vol. 46, pp. 1056–1064, 2021.
- [16] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*, 2017, pp. 1273–1282.
- [17] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016.
- [18] A. Lavric and V. Popa, "Performance Evaluation of LoRaWAN Communication Scalability in Large-Scale Wireless Sensor Networks," *Wireless Communications and Mobile Computing*, vol. 2018, p. 6730719, Jun 2018.
- [19] M. Courbariaux and Y. Bengio, "Binarynet: Training deep neural networks with weights and activations constrained to +1 or -1," *CoRR*, vol. abs/1602.02830, 2016. [Online]. Available: <http://arxiv.org/abs/1602.02830>
- [20] "Federated Learning: A Step by Step Implementation in Tensorflow," accessed Nov. 25, 2021. [Online]. Available: <https://towardsdatascience.com/federated-learning-a-step-by-step-implementation-in-tensorflow-aac568283399>

# Hierarchical User Status Classification for Imbalanced Biometric Data Class

Nakyoung Kim

*Institute for Information Technology Convergence  
Korea Advanced Institute of Science and Technology  
Daejeon, Rep. of Korea, 34141  
nkim71@kaist.ac.kr*

Gyeong Ho Lee

*School of Electrical Engineering  
Korea Advanced Institute of Science and Technology  
Daejeon, Rep. of Korea, 34141  
gyeongho@kaist.ac.kr*

Hyeontaek Oh

*Institute for Information Technology Convergence  
Korea Advanced Institute of Science and Technology  
Daejeon, Rep. of Korea, 34141  
hyeontaek@kaist.ac.kr*

Hyunseo Park

*School of Electrical Engineering  
Korea Advanced Institute of Science and Technology  
Daejeon, Rep. of Korea, 34141  
tkf92001@kaist.ac.kr*

Jaeseob Han

*School of Electrical Engineering  
Korea Advanced Institute of Science and Technology  
Daejeon, Rep. of Korea, 34141  
j89449@kaist.ac.kr*

Jun Kyun Choi

*School of Electrical Engineering  
Korea Advanced Institute of Science and Technology  
Daejeon, Rep. of Korea, 34141  
jkchoi59@kaist.ac.kr*

**Abstract**—With the proliferation of Internet of Things technologies, health care services that target a household equipped with IoT devices are widely emerging. In the meantime, the number of global single households is expected to rapidly grow. Contactless radar-based sensors are recently investigated as a convenient and practical means to collect biometric data of subjects in single households. In this paper, biometric data collected by contactless radar-based sensors installed in single households of the elderly under uncontrolled environments are analyzed, and a deep learning-based classification model is proposed that estimates a user's status in one of the predefined classes. In particular, the issue of the imbalance class sizes in the generated dataset is managed by reorganizing the classes into a hierarchical structure and designing the architecture for a deep learning-based status classification model. The experimental results verify that the proposed classification model has a noticeable impact in mitigating the issue of imbalanced class sizes as it enhances the classification accuracy of the individual class by up to 65% while improving the overall status classification accuracy by 6%.

**Index Terms**—Radar-based status monitoring, status classification, imbalanced data

## I. INTRODUCTION

With the proliferation of Internet of Things (IoT) technologies, diverse applications and services targeting a household equipped with IoT devices are widely developed. In the meantime, single households are expected to globally grow over the next decades, and the single households of the elderly are

This research was financially supported by the Ministry of Trade, Industry and Energy (MOTIE) and Korea Institute for Advancement of Technology (KIAT) through the International Cooperative R&D program. (Project No. 0011879)

anticipated to take a significant portion [1]. Accordingly, the need for health care services for various health problems (e.g., monitoring of underlying diseases, detecting emergencies, etc.) that can occur in single households of the elderly is increasing.

Many health care services utilize on-body or contactless sensors to collect and monitor the data about their service users in their households [2]. The on-body method obtains the data through a wearable device such as a smartwatch, which requires to be attached to the body of the user for service provision. Since the on-body sensors are often tightly placed on the user's body, the biometric data collected from the on-body sensors are relatively accurate. However, the user has to continuously wear the monitoring device for reliable service provision, and the devices need to be periodically charged. This increases the inconvenience and leads to a low usage rate, in particular, for the high age group who often has difficulties in using smart devices [3]. Accordingly, the methods to monitor the users' status in their households with externally installed contactless sensors are widely investigated for reliable service provision without affecting the users' daily life [4].

With the advances of the sensor technology, radar-based data acquisition methods are recently investigated, which obtains an individual's biometric information (e.g., heart rate, respiration, etc.) from the phase change of the radar reflected by the movement of the human body, and the development of relevant data processing methods are in progress [5]–[8]. In this paper, a classification method is studied to estimate the status of subjects from their real-world biometric information

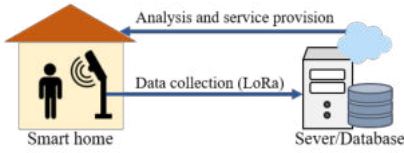


Fig. 1. High level radar-based biometric monitoring system model



Fig. 2. Actual illustration of the radar sensor installed in a single household

TABLE I  
SPECIFICATIONS OF THE INSTALLED RADAR SENSOR

Category	Specification
Chip	Sharp DC6M4JN3000
Method	Microwaves (24.05 24.25GHz)
Resolution	60cm
Range (Max.)	1.5m (Heartbeats, Breathing) 7m (Body motion)
Directionality	Azimuth: 25°, Elevation: 20°
Error rate	±10% ( 3m)

that is collected through contactless radar-based sensors installed in single households. As the frequency and duration for the occurrence of the subjects' status differ, the issue of status class imbalance inevitably rises in uncontrolled environments. In this paper, the status classes in the collected data are reorganized into a hierarchical structure to handle the issue of the imbalanced classes. In particular, this paper analyzes the biometric information of 22 elders collected in uncontrolled environments and proposes a subject's status classification model with the hierarchically structured data that alleviates the imbalance issue.

The rest of this paper is organized as follows. Section II provides previous studies. Section III introduces the status classification method. The performance evaluation is presented in Section IV, followed by the conclusion in Section V.

## II. RELATED WORKS

Biometric information monitoring technologies with contactless radar-based sensors are largely divided into two fields of research: research on increasing the accuracy of the measurement of monitoring devices and research on accurately and effectively analyzing the obtained information. As the sensor technologies have become more mature than before, the studies to accurately analyze the obtained data are actively in progress nowadays. In relation to contactless radar-based biometric information monitoring, research on cardio-respiration

TABLE II  
SAMPLE DISTRIBUTION OF THE HR DATASET

Class	Status	Number of data samples	Ratio
1	Not detected	3,673	34.85%
2	In sleep	4,831	45.84%
3	In active activity	602	5.71%
4	In stationary Activity	554	5.26%
5	In unidentified activity	879	8.34%
Total		10,539	100%

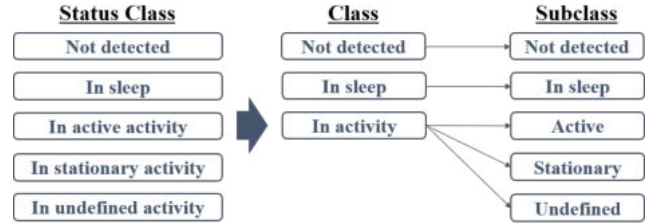


Fig. 3. Reorganization of the status dataset structure with biometric data

monitoring is widely investigated, which is mainly conducted in an impulse-radio ultra wide band (IR-UWB) radar-based environment [5]. These studies include the research on accurate measuring of heart rate, continuous monitoring of cardio-respiration rate [9], and monitoring and estimating the sleep stages [8]–[10]. In particular, with the development of machine learning and artificial intelligence technologies, various studies have now adapted deep learning technologies to accurately analyze biometric data. In [10], the cardio-respiratory signals collected through an IR-UWB radar-based device are analyzed with recurrent neural networks and an attention mechanism. In [11], the performance of various deep learning technologies with the publicly available biometric datasets is evaluated. In [12] and [13], electrocardiogram signals are analyzed based on a convolutional neural network.

However, to the best knowledge of the authors, the previous studies have not managed the issue of the imbalance classes in the dataset that inevitably occurs in practice under uncontrolled environments. Accordingly, this study specifically targets a method to alleviate the adverse effect caused by the imbalanced dataset that is used to develop a user's status classification model with the biometric data collected by a contactless radar-based sensor.

## III. STATUS CLASSIFICATION FROM IMBALANCED DATA OF BIOMETRIC FEATURE

In this section, we propose a deep learning-based status classification model based on the biometric data of a single targeted subject. The proposed model estimates the status of the subject in one of the predefined status classes. In prior to providing the details of the proposed model, we introduce the dataset first to ease the understanding of the reason behind the proposed classification model.



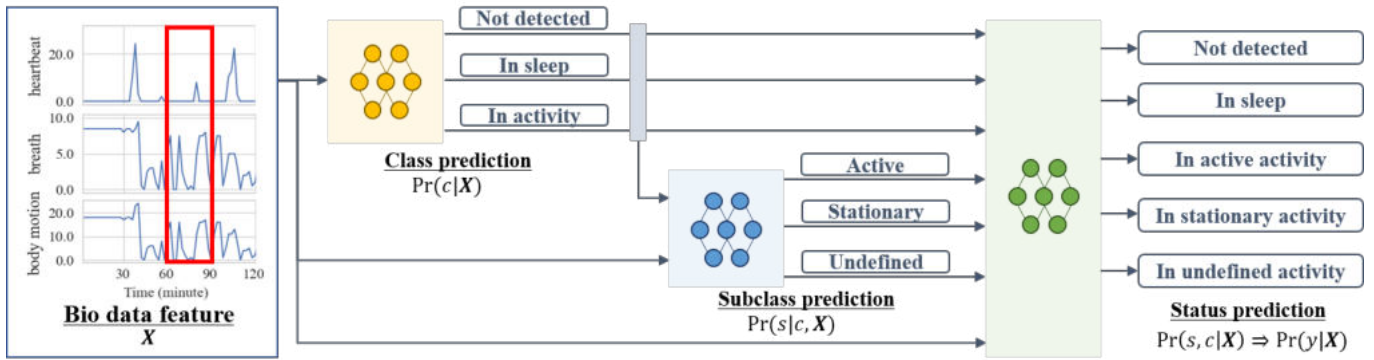


Fig. 4. Proposed deep learning-based user status estimation model with biometric data with imbalanced classes

### A. Dataset Generation and Data Preprocessing

The real-world biometric data that were collected from uncontrolled environments are used in this study. The radar sensors with the specifications given in Table I were installed in a bed room of over 100 individual single households, as described in Fig. 1 and Fig. 2. The dataset was generated by collecting the heart rate, respiratory rate, and body motion, and other types of biometric features for every 2 minutes at different times for each individual from 2020 to 2021. Whereas the dataset includes various types of biometric features, the heart rate, respiratory rate, and body motion data are used in this paper for simplicity. However, the method proposed in this paper is not restricted to the specific combination of the biometric features but can be extended to many feature compositions.

The subject's status for the collected biometric features are labeled by experts as one of the five predefined classes: the subject is 1) not detected, 2) in sleep, 3) in an active activity, 4) in a stationary activity, and 5) in an undefined activity. From the generated dataset, the parts that are stably collected without a network disconnection are extracted, resulting in a subset of the data from 22 out of 112 subjects. Min-max normalization is applied to each dimension of the biometric features. The data are segmented into fully non-overlapping data samples for 30 minutes, and the dimension of a data sample accordingly becomes 15. The status of a segmented data sample is set as the most frequently detected status in that period. Table. II shows the sample distributions of the dataset with respect to the status classes.

As the dataset is collected with externally installed sensors in the subjects' places, the dataset includes the data when the subjects are in sleep as well as when they are out of the range of the sensors. Subsequently, the frequency and duration of the subjects' statuses are different under uncontrolled environments, and the imbalance in the status classes inevitably occurs. In the remainder of this section, the issue of the imbalanced classes is handled by reorganizing the status classes into a hierarchical structure with classes and subclasses as described in Fig. 3 and by developing a neural network model that is particularly designed to learn the hierarchical

structure.

### B. Status Classification

The objective of a status classification problem is to estimate the probability of occurrence of a status in a given period when a sequence of biometric features is observed, such that

$$\Pr(y | \mathbf{X}), \quad (1)$$

where  $y$  denotes the status class before the reorganization, and  $\mathbf{X}$  is the input sequence of biometric features. As the dataset is reorganized in a hierarchical structure, the status classification problem in (1) can be now considered as estimating the joint probability of classes and subclasses, which is given by

$$\Pr(s, c | \mathbf{X}), \quad (2)$$

where  $c$  denotes the class, and  $s$  denotes the subclass. The joint probability in (2) then can be further divided into

$$\Pr(c | \mathbf{X}) \Pr(s | c, \mathbf{X}). \quad (3)$$

where  $\Pr(c | \mathbf{X})$  indicates the probability of the occurrence of a class when a biometric data feature is observed, and  $\Pr(s | c, \mathbf{X})$  indicates the probability of the occurrence of a subclass when the information on the class and biometric data are given.

That is, the status classification problem in (1) can be investigated as a combination of two separate classification problems, which are respectively predicting the class and subclass. Through separately developing these classification models, the problem to find the decision boundaries among the classes and subclasses can be simplified. In addition, the issue of imbalance status class can be mitigated to some degree by transferring the classification models that are separately pre-trained into a comprehensive model for integration and parameter tuning as described in Fig. 4.

As the main focus of this paper is to investigate the impact of the proposed classification model structure particularly designed for the hierarchically structured dataset, multi-layered perceptrons (MLP) are applied for the classification models for simplicity. However, the proposed model is not restricted to MLP but other advanced deep learning models can be alternatively used to enhance the performance.

TABLE III  
CONFUSION MATRIX FOR ERROR MEASUREMENTS

Predicted		Actual	
		Positive	Negative
Positive	True positive (TP)	False negative (FN)	
Negative	False positive (FP)	True negative (FN)	

#### IV. PERFORMANCE EVALUATION

In this section, the performance of the proposed status classification model is evaluated in terms of its classification accuracy.

##### A. Evaluation Metric

The performance of the proposed model is evaluated with a well-known error measurement,  $F_1$  score, based on the confusion matrix illustrated in Table III. The  $F_1$  score is the harmonic mean of the precision and recall, where the precision indicates the ratio between the true positive instances and all positive instances while the recall indicates the ratio between the true positive instances and all of the instances that are identified as positive. For a multi-label classification problem, the confusion matrix and  $F_1$  score of each class are computed and combined afterward.

The  $F_1$  score for the  $i$ -th class is given by

$$F_1 \text{ score} = \frac{TP_i}{TP_i + \frac{1}{2}(FP_i + FN_i)}, \quad (4)$$

where  $TP_i$ ,  $TN_i$ ,  $FP_i$ , and  $FN_i$  are the numbers of the true positive, true negative, false positive, and false negative instances for the  $i$ -th class. The  $F_1$  score has a range of  $[0, 1]$ , and a value closer to 1 implies better classification performance. The unweighted mean of each class'  $F_1$  scores is widely considered an adequate metric to evaluate the performance of a classifier on a dataset with imbalanced class samples. Accordingly, the equally weighted  $F_1$  scores of the classes regardless of their numbers of instances are used for performance evaluation.

##### B. Experimental Settings and Results

To evaluate the performance of the proposed classification model, experiments on the classification methods with conventional machine learning and deep learning techniques are conducted. As the benchmarks, the performance of Gaussian naïve Bayes (NB), k-nearest neighbor (kNN), support vector machine (SVM), and random forest (RF) are experimentally investigated. In addition, MLP with different numbers of layers and nodes are used as the baselines. The conventional machine learning methods are implemented with the scikit-learn library in python. Each part of the proposed model is designed with two fully connected layers where each layer is respectively composed of 16 and 8 nodes. The class and subclass classification models are pre-trained and combined with the status classification part in the end for further training. The baseline MLP models are designed with a combination of layers with 32, 16, and 8 nodes. The deep learning models are implemented with TensorFlow and Keras. For all deep

TABLE IV  
 $F_1$  SCORE OF THE STATUS CLASSES (%)  
(STATUS CLASS: 1. NOT DETECTED, 2. IN SLEEP, 3. IN ACTIVE ACTIVITY, 4. IN STATIONARY ACTIVITY, AND 5. IN UNDEFINED ACTIVITY)

Model	Status class					Avg.
	1	2	3	4	5	
NB	<b>83.3</b>	<b>74.6</b>	<b>39.0</b>	<b>18.9</b>	<b>63.2</b>	<b>55.8</b>
kNN	88.1	84.1	11.9	12.9	61.1	51.6
SVM	90.2	88.6	7.6	3.7	72.8	52.6
RF	92.3	87.4	5.5	15.9	73.3	54.9
MLP (16-8)	90.7	91.3	53.2	13.1	80.9	65.8
MLP (16-16-8)	<b>91.0</b>	<b>90.9</b>	<b>44.2</b>	<b>24.2</b>	<b>78.9</b>	<b>65.8</b>
MLP (16-16-16-8)	90.6	90.6	46.4	5.3	79.0	62.4
MLP (32-16-8)	91.0	91.6	49.3	11.9	80.5	64.9
MLP (32-16-16-8)	91.0	90.8	50.0	9.8	80.7	64.5
Proposed	<b>91.3</b>	<b>91.0</b>	<b>46.1</b>	<b>40.2</b>	<b>80.9</b>	<b>69.9</b>

learning models, Adam optimizer is applied with a learning rate 0.001, and they are trained until the losses are converged with the learning rate decay and early stopping. For training, validation, and test, 60%, 20%, and 20% of instances are randomly selected from the generated dataset, respectively.

The experimental results are provided in Table IV. The results show that the deep learning approaches outperform the conventional machine learning approaches. In addition, as the number of instances in Class 1 (not detected) and Class 2 (in sleep) are much larger than those of the other classes, the classification models are trained biased to Class 1 and Class 2. As result, the  $F_1$  scores of the classes with small numbers of instances, Class 3 (in an active activity) and Class 4 (in a stationary activity), are fairly lower than the others in general. In particular, Class 4 is relatively difficult to be separated from Class 2 as the biometric features of these two classes are not much different from each other but Class 4 has much fewer instances compared to Class 2. Even though the MLP models achieve higher  $F_1$  scores for Class 3 than the conventional machine learning models do, their  $F_1$  scores for Class 4 are still comparatively low. In the meantime, the proposed model achieves noticeable improvements on the  $F_1$  scores of both Class 3 and Class 4, resulting in the highest  $F_1$  score on average over the classes.

In summary, whereas there have not been significant improvements in the classification accuracy for the classes with a large number of instances, the results show that the proposed model improves the classification accuracy for the classes in small sizes. Hence, these results verify the effectiveness of the proposed classification model in the structure that is particularly designed for a dataset with imbalanced class sizes.

#### V. CONCLUSION

In this paper, the biometric data (i.e., heart rate, respiratory rate, and volume of motion) collected in single households of the elderly under uncontrolled environments are investigated, and a classification model is proposed that estimates a user's status in one of the five predefined classes (i.e., not detected, in sleep, in an active activity, in a stationary activity, and in an undefined activity). In particular, the issue of the imbalanced

class is managed by reorganizing the classes into a hierarchical structure and separating the status classification problem into two disjoint problems based on probabilistic analysis. The separated classification models are deliberately designed to manage the individual objective of the disjoint classification problems. The scope of this study is to verify the impact of the structure of the proposed classification model in alleviating the adverse effects of the imbalanced class. Accordingly, simple feed-forward networks are applied for classification to rule out the influences caused by the choice of deep learning technologies other than the structure of the classification model itself. The experimental results show that the proposed model enhances the status classification accuracy of the individual class by over 65% while improving the overall accuracy by 6%. Whereas the improvements in the overall accuracy are not significant, the results verify that the proposed model has a noticeable impact in managing the issue of imbalanced class. To improve the overall classification accuracy, more advanced deep learning technologies (e.g. convolutional neural network, long-short-term memory, etc.) can be extensively applied to the proposed classification model structure as a future work of this study.

#### ACKNOWLEDGMENT

The authors of this paper thank Seung Chul Kim from KULS, João Garcia from Ubiwhere, and Ricardo Gonçalves from PROEF for co-working to develop the dataset and prototype for the proposed system as partners of a research project (KIAT, Project No. 0011879).

#### REFERENCES

- [1] Organisation for Economic Co-operation and Development (OECD). The Future of Families to 2030. Accessed: Feb., 2020. [Online]. Available: <https://doi.org/10.1787/9789264168367-en>
- [2] H. Mshali, T. Lemlouma, M. Moloney, and D. Magoni, "A survey on health monitoring systems for health smart homes," *International Journal of Industrial Ergonomics*, vol. 66, pp. 26–56, 2018.
- [3] J. Kim, "How much do we know about the use of smartphones in the silver generation?: Determinants of the digital divide within the silver generation," *Information Society & Media*, vol. 21, no. 3, pp. 33–64, 2020.
- [4] M. Bahache, J. P. Lemayian, W. Wang, and J. Hamareh, "An inclusive survey of contactless wireless sensing: A technology used for remotely monitoring vital signs has the potential to combating covid-19," 2020.
- [5] J. Kranjec, S. Beguš, G. Geršak, and J. Drnovšek, "Non-contact heart rate and heart rate variability measurements: A review," *Biomedical Signal Processing and Control*, vol. 13, pp. 102–112, 2014.
- [6] A. Ni, A. Azarang, and N. Kehtarnavaz, "A Review of Deep Learning-Based Contactless Heart Rate Measurement Methods," *Sensors*, vol. 21, no. 11, 2021.
- [7] F. Zhang, C. Wu, B. Wang, M. Wu, D. Bugos, H. Zhang, and K. J. R. Liu, "Smars: Sleep monitoring via ambient radio signals," *IEEE Transactions on Mobile Computing*, vol. 20, no. 1, pp. 217–231, 2021.
- [8] Y. Lee, J. Y. Park, Y. W. Choi, H. K. Park, S. H. Cho, S. H. Cho, and Y. H. Lim, "A Novel Non-contact Heart Rate Monitor Using Impulse-Radio Ultra-Wideband (IR-UWB) Radar Technology," *Scientific Reports*, vol. 8, no. 13053, 2018.
- [9] C. H. Chang and W. W. Hu, "Design and implementation of an embedded cardiorespiratory monitoring system for wheelchair users," *IEEE Embedded Systems Letters*, vol. 13, no. 4, pp. 150–153, 2021.
- [10] H. B. Kwon, S. H. Choi, D. Lee, D. Son, H. Yoon, M. H. Lee, Y. J. Lee, and K. S. Park, "Attention-based lstm for non-contact sleep stage classification using ir-uwband radar," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 10, pp. 3844–3853, 2021.
- [11] S. Purushotham, C. Meng, Z. Che, and Y. Liu, "Benchmarking deep learning models on large healthcare datasets," *Journal of Biomedical Informatics*, vol. 83, pp. 112–134, 2018.
- [12] M. Kachuee, S. Fazeli, and M. Sarrafzadeh, "Ecg heartbeat classification: A deep transferable representation," in *2018 IEEE International Conference on Healthcare Informatics (ICHI)*, 2018, pp. 443–444.
- [13] N. Ahmed, A. Singh, S. KS, G. Kumar, G. Parchani, and V. Saran, "Classification Of Sleep-Wake State In A Ballistocardiogram System Based On Deep Learning," 2020.

# Increasing Accuracy of Hand Gesture Recognition using Convolutional Neural Network

1<sup>st</sup> Gyutae Park  
Electronic engineering  
Gyeongsang National University  
Jinju, Republic of Korea  
erbxo4321@gnu.ac.kr

2<sup>nd</sup> V.K. Chandrasegar  
Electronic engineering  
Gyeongsang National University  
Jinju, Republic of Korea  
san5432155@gmail.com

3<sup>rd</sup> JoongGun Park  
JD Co., Ltd  
Jinju, Republic of Korea  
jddesign0@gmail.com

4<sup>th</sup> Jinhwan Koh  
dept. of Electronics  
Engineering/Engineering  
Research Institute (ERI)  
Gyeongsang National University  
Jinju, Republic of Korea  
jikoh@gnu.ac.kr

**Abstract**—Human gestures play important roles in the interaction between humans and machines. These human gestures are becoming more important, yet complex gesture input and noise induced by external elements are important problems to solve in order to improve the accuracy of hand gesture recognition methods. Convolutional Neural Networks (CNN) are offered as a technology that can solve this problem in this research. CNN has the advantage of being able to learn image data, and this technology will greatly improve human-machine interaction accuracy. Data was extracted using Vivaldi antennas with a frequency bandwidth of 7.4-9.0 GHz and gain characteristics of 8 dB in five sign language operations, and data that went through the preprocessing process was learned through CNN. The classification results of the proposed CNN showed about 90% accuracy.

**Keywords**—IR-UWB Radar, 2D-FFT, Hand Gesture, CNN, Machine Learning

## I. INTRODUCTION

Human gestures play a very important role in the interaction between humans and machines. A representative example is a technology that replaces a switch or remote control that requires existing physical contact with only a gesture [1]. However, while the importance of hand gesture recognition technology increases, the accuracy of hand gesture recognition technology is still insufficient. The impulse radio ultra-wideband radar (IR-UWB RADAR) technology is effective in addressing these issues.

The radar we used to increase recognition accuracy is the Impulse Radar-Ultra Wide Bandwidth (IR-UWB), which uses a wide frequency band with low power, and has characteristics such as an occupancy bandwidth of 25% or more of the FCC's central frequency and an occupancy bandwidth of 500 MHz or more.

And because it instantaneously transmits a very narrow pulse, there is a very low spectral power density over a very wide frequency band. These characteristics can improve the accuracy of hand gesture recognition as they provide high security and high data transmission characteristics, high resolution as accurate distance and location measurements are possible [2-4]. Recognizing human gestures using radar requires the extraction of meaningful information from the received signal, which is difficult to do in big datasets containing a variety of human gestures [5]. To overcome this issue, 2D-FFT was utilized to convert the data into 2D data with important properties, and a convolutional neural network (CNN) was used to classify the results. Recently, several neural network technologies have been studied, and results have been derived that CNN is easy to learn image data. Therefore, CNN was judged to be useful for classifying image data output by radar, so it was used for hand gesture identification [6]. The composition of this paper is as follows. This paper is structured as follows. It is divided into three parts, each of which has the following contents. Section 2 describes the theories related to radar and the Fourier transform and Section 3 describes the experimental process and results, and finally Section 4 describes the conclusions.

## II. THEORY

### A. IR-UWB Radar

Radars are largely divided into continuous wave radars and pulsed wave radars depending on the radio waves used. Continuous-wave radar refers to a radar that continuously radiates radio waves with a constant frequency, and pulsed wave radar refers to a radar that radiates radio waves that instantaneously increase and returns along a specific cycle. Here, there is a problem in that it is impossible to measure when the reflected wave returns because the same radio wave is continuously received in the receiving unit of the continuous wave radar. Therefore, it is difficult for the continuous wave radar to measure the distance between objects. Therefore, it is

the frequency modulation continuous wave radar FMCW radar that has improved the distance measurement capability by modulating the frequency modulation continuous wave radar. However, since it is impossible to distinguish reflected waves (a kind of noise) generated by the speed, angle, and surrounding terrain of the target, the FMCW Doppler radar method solved the problem using the Doppler effect. In other words, when continuous waves are used, additional functions such as frequency modulation must be added to measure the most basic distance, and a high-spec signal processing system is needed to process a large amount of information that continues to flood, making the system larger and more complex. On the other hand, radars using pulse waves radiate waves with different amplitudes for a moment, so they can easily measure the distance to the counterpart by knowing when the reflected wave returned. Here, by using the Doppler effect, 3D information such as target speed and altitude and noise caused by topographic features can be removed. And the pulse-Doppler radar may constitute a small system. Therefore, since a large amount of information does not need to be processed compared to the continuous wave radar, the size of the device becomes smaller, the configuration becomes simpler, and the energy efficiency is increased.

### III. EXPERIMENT AND RESULTS

An experimental environment was created as shown in Figure 1 to measure hand gestures using radar. The experiment used NVA-R661 radar from Novelda, which includes two Vivaldi antennas and has a bandwidth of 6.0-8.5 GHz and a gain characteristic of 8dB. The sampling rate of the radar is 39GS/s, and objects up to 1.5m away can be detected. The radar's Tx antenna transmits a signal toward an object, reaches the object, and the reflected signal is transmitted to the Rx antenna to receive the signal.



Fig. 1. Experiment environment and NVA-R661

The experiment was conducted in a long corridor where no surrounding objects existed, as shown in Figure 1, to minimize the noise component measured on the radar. In the experiment, three experimenters performed sign language operations, and measurements were performed at a distance of 50 cm from the radar. The hand movements used are five American sign language movements, the first being all done sign, the second being Eat sign, the third being More sign, and the fourth being Thank you sign. Lastly, Sorry sign.

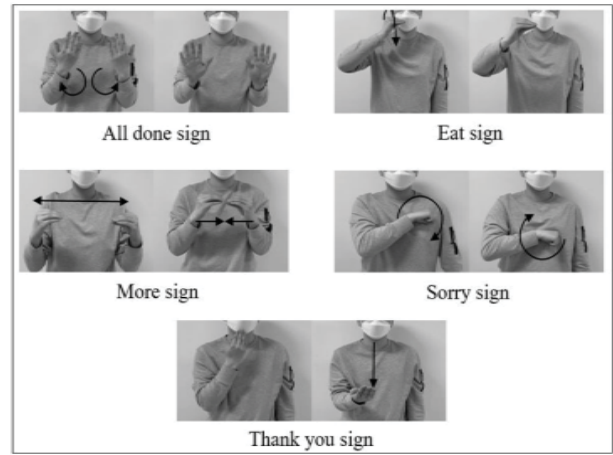


Fig. 2. Selected five sign language

In this paper, five sign language operations were measured 600 times for each operation. Of the 600 data, 500 were measured in general and 100 were measured in a form of behaviour that may be somewhat difficult to recognize (recognition distance, speed of sign language motion, height change of hand).

Figure 3 shows the measurement results of general motion, and Figure 4 shows the measurement results of motion that have changed various elements. Each result is an image obtained by adding all data and then averaging it.

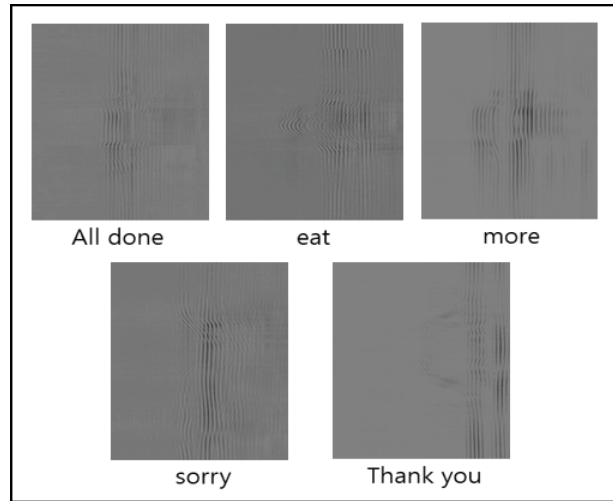


Fig. 3. The measurement results of general motion

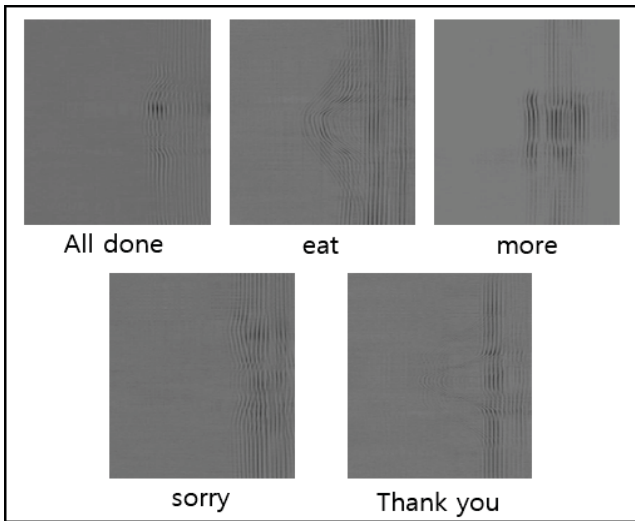


Fig. 4. The measurement results of motion that has changed various elements

In addition, a two-dimensional Fourier transform was performed to extract features from the normal hand gesture data. Image data of the five sign language operations thus obtained were converted into frequency domains through 2D-FFT using MATLAB, and frequency components at each corner, that is, zero frequency values, were collected in the middle to facilitate analysis. Figure 5 is the result of the conversion of All done sign among the five sign language operations.

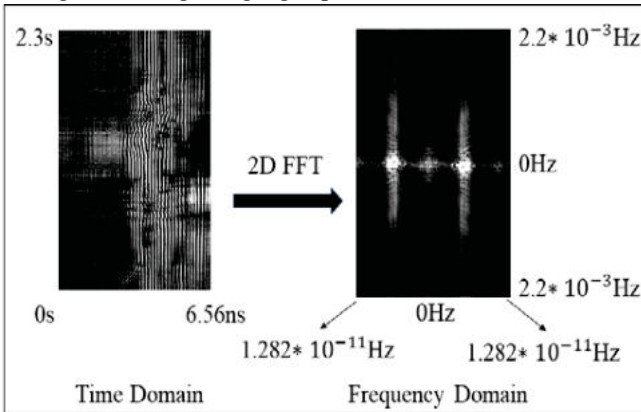


Fig. 5. Converted result (All done sign)

2D-FFT is maintained as double-type data, the same data type as the initial measurement data, but when learning, if data that is not standardized in the gradient descent algorithm of the artificial neural network is input, there is a difference in the learning process and results. To solve this problem, an 8-bit standardization process was additionally performed using Matlab.

Finally, in the normalized 2D-FFT image file, the main feature is the zero-frequency component present in the centre of the image, so only the centre portion of the image was extracted to increase the learning speed of CNN and improve recognition accuracy, and finally, data of 191 by 191 by 1 was obtained.

In this paper, learning was conducted using Matlab's Deep Learning Tool, and the CNN model proposes two CNN models.

The first proposes a two-stage serial CNN model that learns by connecting two CNN terminals, and the second proposes a double parallel CNN model that connects two CNN terminals in parallel. Figure 6 shows the structure of the two models.

The filter size of the Convolution Layer used was 3 by 3, the number of filters was 32 strides 1 by 1, the padding was the same, and the weights initializer was the glorot. The pool size of the Maxpool layer was selected as 5 by 5, the stride was 1 by 1, and the padding was the same. Glorot was used as the Weights Initializer of the Fully Connected Layer.

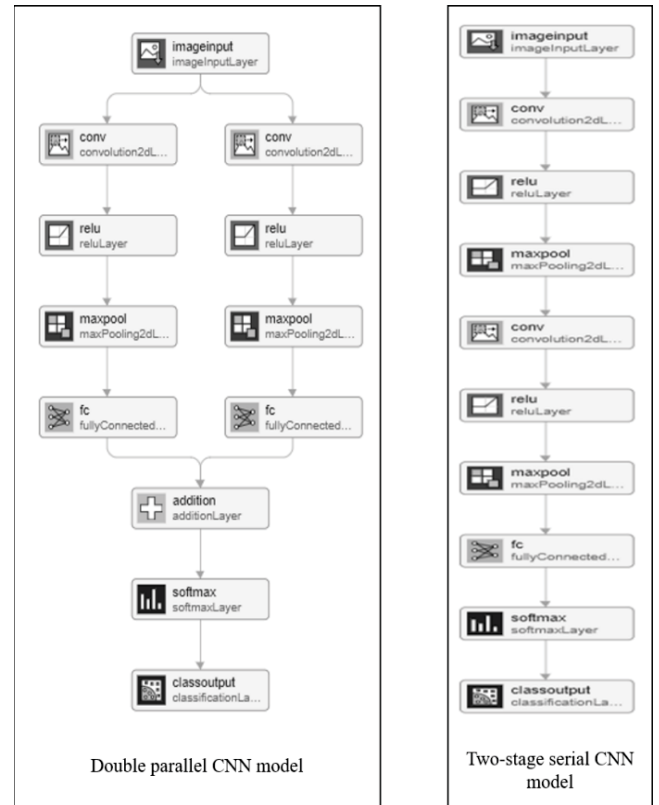


Fig. 6. Block diagram of the proposed model

The main specifications of the system in which learning was conducted are as follows.

CPU: 11th Gen Intel(R) Core(TM) i7-11700 @ 2.50GHz

RAM: 32.0GB

GPU: Geforce RTX3090

A total of four data were used for learning. Raw data(Range-Time map), data performed only 2D-FFT, data performed with 2D-FFT and normalization, and final data performed with 2D-FFT, normalization, and feature partial extraction were divided.

The evaluation method was divided into two methods.

First, the process of learning with 350 general gestures and evaluating with 150 general gestures.

The second is the process of learning with 500 general gestures and evaluating with 100 change gestures.

The results are shown in Tables I and II below.

TABLE I. TWO-STAGE SERIAL CNN MODEL

Data Type	Result of Recognition	
	<i>The results of the general hand gesture evaluation.</i>	<i>The result of the change hand gesture evaluation.</i>
Raw data	81.60%	51.00%
Only 2D-FFT	92.53%	64.40%
2D-FFT and normalization	91.47%	62.00%
Final data	92.53%	83.40%

TABLE II. DOUBLE PARALLEL CNN MODEL RESULT

Data Type	Result of Recognition	
	<i>The results of the general hand gesture evaluation.</i>	<i>The result of the change hand gesture evaluation.</i>
Raw data	90.53%	35.60%
Only 2D-FFT	95.73	75.80%
2D-FFT and normalization	96.13%	80.60%
Final data	96.50%	87.20%

And for comparison with the proposed model, learning was additionally conducted using Google Net, Resnet-50, VGG-19, and AlexNet. Learning and evaluation were conducted using four types of data as mentioned above, and learning and evaluation were conducted three times to prevent fragmentary results of each model, and in the case of three, the maximum result was selected and displayed in a table. Each result is shown in Tables III, IV, V, and VI.

TABLE III. GOOGLNET MODEL RESULT

Objects Type	Result of Recognition	
	<i>The results of the general hand gesture evaluation.</i>	<i>The result of the change hand gesture evaluation.</i>
Raw data	99.20%	71.20%
Only 2D-FFT	98.27%	86.80%
2D-FFT and normalization	95.20%	87.20%
Final data	92.13%	84.00%

TABLE IV. RESNET-50 MODEL RESULT

Data Type	Result of Recognition	
	<i>The results of the general hand gesture evaluation.</i>	<i>The result of the change hand gesture evaluation.</i>
Raw data	99.20%	88.40%
Only 2D-FFT	98.93%	82.20%
2D-FFT and normalization	96.40%	82.40%
Final data	96.67%	84.40%

TABLE V. VGG-19 MODEL RESULT

Data Type	Result of Recognition	
	<i>The results of the general hand gesture evaluation.</i>	<i>The result of the change hand gesture evaluation.</i>
Raw data	91.73%	47.20%
Only 2D-FFT	93.73%	86.60%
2D-FFT and normalization	92.27%	78.60%
Final data	87.33%	78.60%

TABLE VI. ALEXNET MODEL RESULT

Data Type	Result of Recognition	
	<i>The results of the general hand gesture evaluation.</i>	<i>The result of the change hand gesture evaluation.</i>
Raw data	86.67%	59.40%
Only 2D-FFT	93.60%	83.60%
2D-FFT and normalization	92.53%	79.00%
Final data	88.93%	83.20%

Even though using Alexnet, Googlenet, Resnet-50, and VGG-19 provides better hand gesture recognition accuracy than shown in the above tables, Googlenet and Resnet-50 confirm that the accuracy is higher than the other two models. However, VGG-19 and ALEXNET have higher accuracy than serial models, but lower accuracy than parallel CNN models. The classification accuracy is very high in the case of the prominent CNN model, however, leading Googlenet and Resnet in the longer learning rate. Googlenet took 7 minutes and Resnet 10 minutes, and Googlenet and Resnet-50 took about three to five times longer than parallel models. Whereas the proposed model took only 2 minutes with comparatively higher accuracy. Moreover, a model with a shorter learning time is judged to be a more competitive model, and the model proposes a parallel model with less time execution and higher accuracy.

#### IV. CONCLUSION

In this paper, a radar and deep learning model are proposed as a way to improve the accuracy of hand gesture recognition technology effective in interaction with machines. Five different sign language movements were directly measured by 600 for each operation using IR-UWB radar, and then made into learning data through preprocessing such as 2D-FFT, normalization, and feature extraction with MATLAB. The deep learning model used CNN Layer and Pooling Layer as the main layers, and a two-stage CNN model and a double parallel CNN model were proposed, and learning and evaluation were also conducted through several prominent models to compare the accuracy of classification results.

Compared with the classification results with existing prominent CNN models, the accuracy of the model proposed in this paper was judged to be a model with sufficient competitiveness.

In the future, it plans to conduct research to further increase the accuracy of recognition of gestures that are difficult to recognize with research plans.

#### REFERENCES

- [1] X. Guo, W. Xu, W. Q. Tang and C. Wen, "Research on Optimization of Static Gesture Recognition Based on Convolution Neural Network," 2019 4th International Conference on Mechanical, Control and Computer Engineering (ICMCCE), Hohhot, China, pp. 398-3982, 2019.
- [2] F. Wang, M. Tang, Y. Chiu and T. Horng, "Gesture Sensing Using Retransmitted Wireless Communication Signals Based on Doppler Radar Technology," in *IEEE Transactions on Microwave Theory and Techniques*, vol. 63, pp. 4592-4602, Dec. 2015
- [3] U.S FCC, "Amendment of Part 97 of the Commission's Amateur Service Rules", 2003
- [4] X. Wang, A. Dinh and D. Teng, "Reliability modeling for wireless Ultra Wideband biomedical radar sensing network," 2010 International Conference on Bioinformatics and Biomedical Technology, Chengdu, China, pp. 69-73, 2010.
- [5] J. Park and S. H. Cho, "IR-UWB Radar Sensor for Human Gesture Recognition by Using Machine Learning," 2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Sydney, NSW, Australia, pp. 1246-1249, 2016.
- [6] K. Nakada, A. Ito, H. Hatano and H. Aratame, "New Switchless and Free Positioning Gesture Recognition System Using RNN and CTC Loss Function," 2018 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, pp. 450-453, 2018.



# Impacts of Behavioral Biases on Active Learning Strategies

Deepesh Agarwal  
Department of Electrical and Computer Engineering  
Kansas State University  
Manhattan, Kansas, USA  
deepesh@ksu.edu

Obdulia Covarrubias-Zambrano  
Department of Cancer Biology  
The University of Kansas Medical Center  
Kansas City, Kansas, USA  
ocovarrubias@kumc.edu

Stefan Bossmann  
Department of Cancer Biology  
The University of Kansas Medical Center  
Kansas City, Kansas, USA  
sbossmann@kumc.edu

Balasubramaniam Natarajan  
Department of Electrical and Computer Engineering  
Kansas State University  
Manhattan, Kansas, USA  
bala@ksu.edu

**Abstract**—Cyber-Physical-Human Systems (CPHS) interconnect humans, physical plants and cyber infrastructure across space and time. Industrial processes, electromechanical systems operations and medical diagnosis are some examples where one can see the intersection of humans, physical and cyber components. Emergence of Artificial Intelligence (AI) based computational models, controllers and decision support engines have improved the efficiency and cost effectiveness of such systems and processes. These CPHS typically involve a collaborative decision environment, comprising of AI-based models and human experts. Active Learning (AL) is a category of AI algorithms which aims to learn an efficient decision model by combining domain expertise of the human expert and computational capabilities of the AI model. Given the indispensable role of humans and lack of understanding about human behavior in collaborative decision environments, modeling and prediction of behavioral biases is a critical need. This paper, for the first time, introduces different behavioral biases within an AL context and investigates their impacts on the performance of AL strategies. The modelling of behavioral biases is demonstrated using experiments conducted on a real-world pancreatic cancer dataset. It is observed that classification accuracy of the decision model reduces by at least 20% in case of all the behavioral biases.

**Index Terms**—Active Learning, Behavioral Biases, Cyber-Physical-Human Systems, collaborative decision environment, human behavior modelling

## I. INTRODUCTION

Active Learning (AL) is a form of semi-supervised machine learning (ML) approach where the learning algorithm leverages information from external sources in order to predict labels for the unlabeled instances in the dataset. The primary motive is to accomplish a higher prediction accuracy with fewer labelled instances as compared to traditional supervised ML methodologies. It has proved to be advantageous in modern ML frameworks involving expensive or wearisome labelling procedures [1]. The learning algorithm in AL settings is referred to as *Active Learner* and the external information source is termed as the *Oracle*. The AL framework can be

represented as a collaborative decision environment comprising of Artificial Intelligence (AI) engine, in the form of ML-based classification/regression models; and human experts, in the form of Oracle (i.e., a domain expert). Typically, in such environments, the aim is to learn an efficient decision model by combining domain expertise of the human expert and computational capabilities of the AI model.

Although there is a plethora of literature published on handling practical AL challenges, like cold-start problem, oracle uncertainty, variable labelling costs and performance evaluation in the absence of ground truth, the collaboration of human and AI engine in a decision environment is neither straightforward nor well understood. There are anomalies and biases associated with both human and AI components of the decision environment. Algorithmic biases (like, selection bias, sampling bias, correlation fallacy, etc.) arises due to inability of algorithms to appropriately adjust to differences in data from different population subgroups [2]. On the other hand, behavioral biases (like, overconfidence, cognitive bias, hot hand fallacy, regret aversion bias, etc.) creep in due to uncertainties associated with human decisions [3]. This paper, first simulates different behavioral biases in an AL context. Then, the impact of these behavioral biases on the performance of AL strategies is quantified by comparing against an ideal case, where behavioral biases are absent.

### A. Related Work

Human experts are crucial components of AI-enabled services in cyber-physical-human systems (CPHS). They form a collaborative decision environment with the support of AI-based computational models. This is pertinent in a wide variety of domains, including fault diagnosis, predictive maintenance, optimal control, process and manufacturing industry operations and medical diagnosis. Given the compelling role of humans in such decision environments, it is an important research challenge to model, predict and use the limits of

human behavior (e.g., behavioral bias and cognitive fatigue) in CPHS design [4]. Modeling human behavior in a decision environment is not straightforward. Humans use cognitive mechanisms and decision heuristics to process information and make decisions under uncertainty [5], [6].

Behavioral biases have been studied in numerous fields, including investment and finance [7], radiology [8], medical diagnosis [9], and human-in-the-loop systems [10]. The existence of behavioral biases for investment decisions is studied, supported by evidence from the Indian stock market [11]. Protte et al. [3] have presented the impacts of overconfidence bias and hot-hand fallacy with the help of an experimental framework involving surveillance drone piloting. Cognitive bias and carelessness are parameterized, and their impact on users' reliability is evaluated for personal context recognition [12]. Furthermore, several recent studies have proposed methodologies to address algorithmic biases using effective sampling approaches [13], [14], [15], [16] and adversarial learning [17], [18], [19]. Among all these studies presented in the literature, a generic framework for modelling and prediction of behavioral biases in the context of a collaborative decision environment has not been proposed so far. An interactive framework with the flexibility to simulate and predict different behavioral biases would be highly beneficial to study, analyze and use the limits of human decision under uncertainty in human-AI collaborative decision-making tasks.

### B. Contributions

The article demonstrates the simulation of different behavioral biases in an AL context. AL is represented as a collaborative decision environment consisting of AI engines (ML-based models) and human experts (Oracle). The user inputs are designed to be taken in two steps: agreement or disagreement with the AI model, followed by class labels based on human judgement (in case of disagreement). The behavioral biases are simulated by providing pre-engineered human decisions during the input steps. All the bias models are validated by performing experiments on a real-world pancreatic cancer dataset. Further, the impacts of all the simulated biases on the performance of AL strategies are examined by comparing classification accuracy against an ideal case, which does not subsume any type of behavioral bias. It is observed that the accuracy score of the decision model is reduced by at least 20% (in cases of hot-hand fallacy and representative bias) to around 85% (in case of gambler's fallacy). Such a collaborative decision framework, with the flexibility to study multiple behavioral biases, has not been proposed in the literature and is a novel contribution of this work.

The remainder of this article is organized as follows: background on AL is presented in Section II. Section III elaborates upon the modelling of behavioral biases within AL frameworks, followed by experimental setup in Section IV and results in Section V. The article ends with concluding remarks in Section VI.

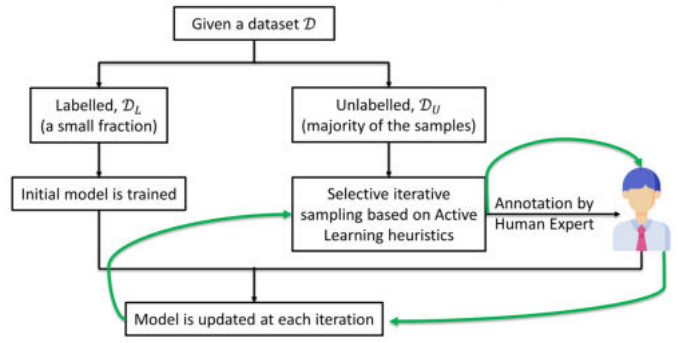


Fig. 1. General Approach of AL frameworks.

## II. BACKGROUND ON ACTIVE LEARNING

The general approach of AL frameworks is presented in Figure 1. Given a dataset  $\mathcal{D}$ , a small fraction of the samples ( $\mathcal{D}_L$ ) are labelled and majority of them ( $\mathcal{D}_U$ ) are unlabelled. The primary aim is to accurately predict labels for all instances in  $\mathcal{D}_U$  with much fewer labelled instances for training, as compared to the conventional supervised ML frameworks. This is executed by allowing the Active Learner to choose the data it wants to learn from – by posing *queries* to Oracle in the form of unlabeled instances and requesting for the corresponding labels. An initial ML model is trained using  $\mathcal{D}_L$ . This is followed by the iterative selection of queries by optimizing appropriate AL heuristics, like entropy of class probabilities, margin uncertainty or classifier uncertainty. The model is updated at each query step by including the query and associated label within  $\mathcal{D}_L$ . This interactive modelling procedure between AI engine (ML model) and human (Oracle, i.e., domain expert) can be well represented as a collaborative decision environment.

The inherent assumptions in AL frameworks give rise to several challenges during implementation in practical scenarios. There is a wealth of literature on methodologies for handling each of these practical challenges: cold-start problem [20], [21], [22], oracle uncertainty [23], [24], [25], hybrid query strategies [22] and performance evaluation in the absence of ground truth [25]. However, the representation of AL in the form of a collaborative decision environment is not well examined in the literature. The human and AI components in this collaboration engender behavioral and algorithmic biases respectively. In this work, different types of behavioral biases are simulated within an AL context and their impacts is studied by comparing the accuracy score of associated AL strategy with an ideal case which does not incorporate any sort of behavioral bias.

## III. BEHAVIORAL BIAS MODELS

The irrational behaviors of humans which abstractly hinder the logical decision process are known as behavioral biases. The human decision or judgement methodically deviates from rationale, under the influence of these biases. This can lead to serious consequences, especially in domains like human health

and medicine, where the stakes associated with decision-making are high. In this work, we consider the following behavioral biases: herding bias, cognitive bias, hot-hand fallacy, representative bias, anchoring bias, gambler’s fallacy and regret-aversion bias.

In order to simulate the behavioral biases, the AL framework has been designed to query human experts in two steps:

- (I) Firstly, the instance selected by the Active Learner is labelled as per the AI model trained at the current step. This instance is then presented to the human expert along with the predicted label, who is asked to specify whether he/she agrees or disagrees with the decision of the AI model.
- (II) If the human expert agrees with the decision of AI model, the predicted label is considered to update the model, otherwise the human expert is prompted to provide a label as per his/her judgement.

In this work, we simulate each of the behavioral biases by supplying pre-engineered human decisions during both the input steps, based on the foundational understanding of respective biases. On the other hand, the human inputs corresponding to “Ideal Case” are formulated based on the ground-truth labels in the dataset, which justifies the absence of behavioral biases.

Herding bias is the tendency of humans to take a specific decision just because it is being supported by many other people, rather than relying on their own judgement. This is simulated in our AL environment by making the human expert to indiscriminately agree with the AI model during step (I) of each query. Cognitive bias arises from the generation of a strong, falsified preconceived notion in human minds. Henceforth, there is a tendency to form mental shortcuts to process the information quickly, rather than making rational decisions. We simulate this by making the human expert to blindly disagree with the AI model during step (I) of the input process. Further, their decisions are simulated by supplying uniformly distributed random numbers as shown in (1) during step (II) of each query. Here,  $C$  is the number of classes and  $d_j$  is decision of the human expert at step (II) for  $j$ th query.

$$d_j \sim U(1, C) \quad (1)$$

Hot-hand fallacy causes humans to overconfidently believe that their decision will be correct based on sequences of immediate correct decisions in the past. This is a “fallacy” because a future outcome is independent of the past performance. This is simulated by considering ground-truth labels during an initial set of queries, similar to that in Ideal Case. After an initial set of queries, the inputs are formulated so as to make the human expert to always disagree with AI model in step (I) and generating uniformly distributed random numbers in step (II) to mimic the overconfidence effect in hot-hand fallacy. Representative bias leads to decisions being taken based on an erroneous prototype already existing in the human minds. This “prototype” is typically the most relevant example of a particular object or event. It results in overestimation of similarity between two things that are being compared

by the humans. In the AL environment, ground truth labels are considered during initial fraction of queries. Once the representative bias sets in, the inputs in step (I) are designed to have the human expert randomly agree/disagree with the AI model, followed by uniformly distributed random numbers in step (II), as indicated in (1).

Anchoring bias induces the human decisions to over-rely on first piece of information about a particular event or object. This skews the human judgement and prevent them from making rational decisions. This is simulated in our AL environment by having inputs so as to make the human expert to always agree with the AI model after an initial set of queries. This emulates the decision of human experts to be anchored based on the information learn during initial queries. Gambler’s fallacy causes humans to erroneously predict the probability of a random event based on the outcomes corresponding to sequences of immediate events in the past. Although the human expert would have made a series of incorrect decisions, he/she would still go ahead for another wrong decision overconfidently, in the hope of making a correct one. We simulate this by having the human experts to forcibly make wrong decisions, i.e., shuffling the ground-truth class labels for a fraction of instances in the query set.

Regret-aversion bias occurs when human experts make decisions, so as to avoid regretting alternate decisions in the future. Under the influence of this bias, the expert prefers to select the option that would carry the least regret, even if it is not the most appropriate choice. We simulate this in our AL environment by modifying the ground-truth labels to replace them with the ones corresponding to a pessimistic choice (for example, replacing the label corresponding to lower grade of a disease with the one corresponding to higher grade of the same) for a fraction of instances in the set of queries.

#### IV. EXPERIMENTAL SETUP

In this work, we demonstrate simulation of all the behavioral biases discussed in Section III in an AL context, on a pancreatic cancer dataset adapted from [26]. It comprises of data from 159 participants, classified into 4 classes (healthy, pancreatitis, localized and metastatic) on the basis of an enzymatic signature consisting of arginase, matrix metalloproteinase-1, -3, and -9, cathepsin-B and -E, urokinase plasminogen activator, and neutrophil elastase [26]. 10% of the total instances in the dataset are selected randomly to create an initial labeled dataset, which is used to train an initial ML-based classification model. Further, 50% of the total instances are used for querying iteratively (one query per iteration), and the classification model is updated after each query step. k-Nearest Neighbors (kNN) is chosen as the base classification method because it is versatile, simple and easy to implement and a non-parametric classification algorithm. Moreover, it does not make any inherent assumptions about the distribution of input data. Uncertainty Sampling (US) query strategy is implemented in Python 3.8 to select instances for annotation by the human experts. US selects instances for querying from the pool of unlabeled samples which minimizes the classifier

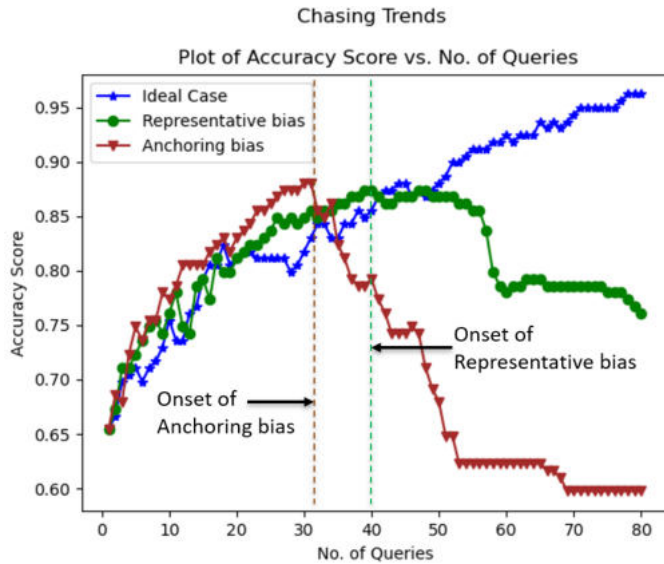


Fig. 2. Plot of Accuracy Score vs. No. of Queries: Chasing Trends.

uncertainty, as described mathematically in (2). Here,  $\hat{y}$  is the predicted label for the instance  $x$  under the model  $\theta$ . In the US query strategy,  $\hat{y}$  is the prediction with the highest posterior probability under the model  $\theta$  (indicated in eq. (3)) and  $x^*$  is the instance chosen for annotation by the human expert.

$$x^* = \arg \min_x P_\theta(\hat{y}|x) = \arg \max_x 1 - P_\theta(\hat{y}|x) \quad (2)$$

$$\hat{y} = \arg \max_y P_\theta(y|x) \quad (3)$$

## V. RESULTS

For the sake of convenience, the behavioral biases discussed in Section III are categorized into 4 categories: Representative bias and Anchoring bias are classified as *Chasing Trends*; Hot-hand and Gambler's fallacies are *Overconfidence* biases; Herding bias and Cognitive bias fall under *Limited Attention Span*; and *Regret-aversion* bias is treated as a separate category. In order to study the impacts of all these behavioral biases in AL setting, the performance (i.e., classification accuracy score) of the model is recorded after each query step for all the cases. The plots of accuracy scores for all categories of biases are presented in Figures 2 - 5.

It can be seen that the accuracy score increases with increase in no. of queries for the Ideal Case. The model trained with initial labelled dataset classifies around 65% of the instances correctly. This score gradually increases to around 96% after 80 queries are made to the human annotator and model being updated after each query step. The corresponding confusion matrix is shown in Table I. However, this trend is not observed in case of any of the behavioral biases. For Representative bias (Figure 2), the accuracy score increases upto 40% of the queries. The inputs are provided so as to set in the Representative bias at this point. Once its sets in, the accuracy score reduces with increasing no. of queries. This is because

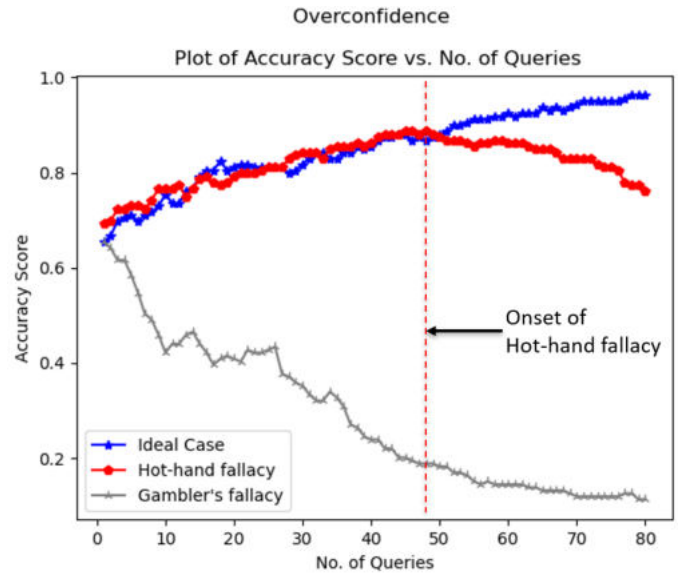


Fig. 3. Plot of Accuracy Score vs. No. of Queries: Overconfidence.

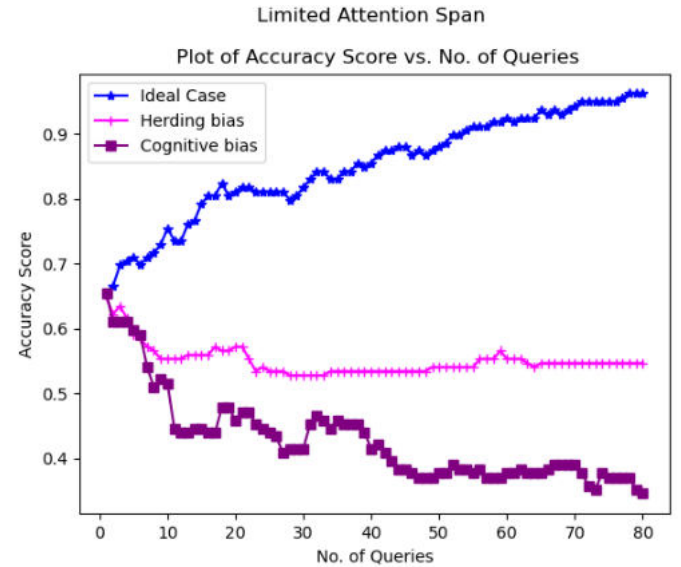


Fig. 4. Plot of Accuracy Score vs. No. of Queries: Limited Attention Span.

the human decisions are biased due to an erroneous prototype already existing in their minds. Similarly, for the cases of Anchoring bias (Figure 2) and Hot-hand fallacy (Figure 3), the accuracy score start decreasing after the corresponding biases set in at 50% and 60% query steps respectively. Further, it can be seen that in the case of Cognitive bias (Figure 4), the accuracy score consistently reduces with increase in no. of queries. This is because the human experts make biased decisions due to a strong, falsified preconceived notions. They tend to form mental shortcuts for quick information processing, rather than making rational decisions. Similar trends can be observed for Gambler's fallacy (Figure 3), Herding bias (Figure 4) and Regret-aversion bias (Figure 5). In each of these cases, the human experts make decisions biased on several

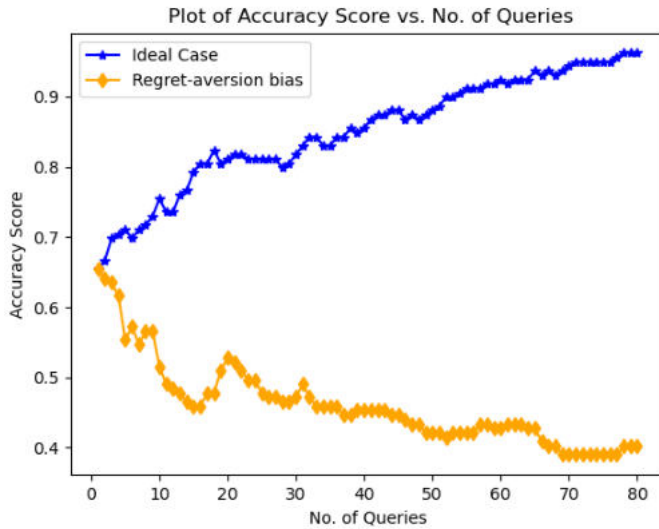


Fig. 5. Plot of Accuracy Score vs. No. of Queries: Regret-aversion.

TABLE I  
CONFUSION MATRIX FOR IDEAL CASE AFTER 50% QUERIES

True Class	Predicted Class			
	Healthy	Pancreatitis	Localized	Metastatic
Healthy	50	0	0	0
Pancreatitis	2	23	0	1
Localized	0	0	32	1
Metastatic	0	2	0	48

factors as discussed in Section III, rather than relying on their own logical judgement.

## VI. CONCLUSION

In this paper, the impact of seven behavioral biases, namely, herding bias, cognitive bias, hot-hand fallacy, representative bias, anchoring bias, gambler’s fallacy and regret-aversion bias is illustrated using experiments conducted on a real-world pancreatic cancer dataset. Firstly, AL is represented in the form of a collaborative decision environment of AI engines and human experts, and the annotation by human experts is formulated as a two-step process. Secondly, the behavioral biases are simulated by dispensing pre-manipulated user inputs based on the foundational understanding of respective biases during the iterative query steps. Finally, the impacts of these biases on the performance of AL strategies are assessed by comparing classification accuracy score of the decision model against a reference case, which does not assimilate any sort of behavioral bias. It is observed that the performance deteriorates significantly when the human decisions are influenced by each of the behavioral biases. Future extensions of this work include ways to detect behavioral biases within a collaborative decision setting, incorporate algorithmic biases and implement the corresponding framework across datasets from different domains.

## ACKNOWLEDGEMENT

This material is based upon work supported by National Science Foundation under award number 2129617.

## REFERENCES

- [1] S. Hao, P. Hu, P. Zhao, S. C. Hoi, and C. Miao, “Online active learning with expert advice,” *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 12, no. 5, pp. 1–22, 2018.
- [2] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, “A survey on bias and fairness in machine learning,” *ACM Computing Surveys (CSUR)*, vol. 54, no. 6, pp. 1–35, 2021.
- [3] M. Protte, R. Fahr, and D. E. Quevedo, “Behavioral economics for human-in-the-loop control systems design: Overconfidence and the hot hand fallacy,” *IEEE Control Systems Magazine*, vol. 40, no. 6, pp. 57–76, 2020.
- [4] Y. Yildiz, “Cyberphysical human systems: An introduction to the special issue,” *IEEE Control Systems Magazine*, vol. 40, no. 6, pp. 26–28, 2020.
- [5] D. Kahneman, “Maps of bounded rationality: Psychology for behavioral economics,” *American economic review*, vol. 93, no. 5, pp. 1449–1475, 2003.
- [6] D. Ariely and S. Jones, *Predictably irrational*. Harper Audio New York, NY, 2008.
- [7] S. A. Zahera and R. Bansal, “Do investors exhibit behavioral biases in investment decision making? a systematic review,” *Qualitative Research in Financial Markets*, 2018.
- [8] L. P. Busby, J. L. Courtier, and C. M. Glastonbury, “Bias in radiology: The how and why of misses and misinterpretations,” *Radiographics*, vol. 38, no. 1, pp. 236–247, 2018.
- [9] E. D O’Sullivan and S. Schofield, “Cognitive bias in clinical medicine,” *Journal of the Royal College of Physicians of Edinburgh*, vol. 48, no. 3, pp. 225–231, 2018.
- [10] E. Wall, L. M. Blaha, L. Franklin, and A. Endert, “Warning, bias may occur: A proposed approach to detecting cognitive bias in interactive visual analytics,” in *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*, IEEE, 2017, pp. 104–115.
- [11] S. Mehta and J. Chaudhari, “The existence of behavioural factors among individual investors for investment decision in stock market: Evidence from indian stock market,” *Global Journal of Research in Management*, vol. 6, no. 1, p. 57, 2016.
- [12] F. Giunchiglia, M. Zeni, and E. Big, “Personal context recognition via reliable human-machine collaboration,” in *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, IEEE, 2018, pp. 379–384.

- [13] A. Amini, A. P. Soleimany, W. Schwarting, S. N. Bhatia, and D. Rus, "Uncovering and mitigating algorithmic bias through learned latent structure," in *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, 2019, pp. 289–295.
- [14] V. Iosifidis and E. Ntoutsi, "Dealing with bias via data augmentation in supervised learning scenarios," *Jo Bates Paul D. Clough Robert Jäschke*, vol. 24, 2018.
- [15] M. Ngxande, J.-R. Tapamo, and M. Burke, "Bias remediation in driver drowsiness detection systems using generative adversarial networks," *IEEE Access*, vol. 8, pp. 55 592–55 601, 2020.
- [16] F. P. Calmon, D. Wei, B. Vinzamuri, K. N. Ramamurthy, and K. R. Varshney, "Optimized pre-processing for discrimination prevention," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 3995–4004.
- [17] M. Alvi, A. Zisserman, and C. Nellåker, "Turning a blind eye: Explicit removal of biases and variation from deep neural network embeddings," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 0–0.
- [18] Z. Wang, K. Qinami, I. C. Karakozis, *et al.*, "Towards fairness in visual recognition: Effective strategies for bias mitigation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8919–8928.
- [19] B. H. Zhang, B. Lemoine, and M. Mitchell, "Mitigating unwanted biases with adversarial learning," in *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 2018, pp. 335–340.
- [20] A. Primpeli, C. Bizer, and M. Keuper, "Unsupervised bootstrapping of active learning for entity resolution," in *European Semantic Web Conference*, Springer, 2020, pp. 215–231.
- [21] J. Shao, Q. Wang, and F. Liu, "Learning to sample: An active learning framework," in *2019 IEEE International Conference on Data Mining (ICDM)*, IEEE, 2019, pp. 538–547.
- [22] D. Agarwal, P. Srivastava, S. Martin-del-Campo, B. Natarajan, and B. Srinivasan, "Addressing practical challenges in active learning via a hybrid query strategy," *arXiv preprint arXiv:2110.03785*, 2021.
- [23] M.-R. Bouguelia, S. Nowaczyk, K. Santosh, and A. Verikas, "Agreeing to disagree: Active learning with noisy labels without crowdsourcing," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 8, pp. 1307–1319, 2018.
- [24] R. Saeedi, K. Sasani, and A. H. Gebremedhin, "Collaborative multi-expert active learning for mobile health monitoring: Architecture, algorithms, and evaluation," *Sensors*, vol. 20, no. 7, p. 1932, 2020.
- [25] D. Agarwal, P. Srivastava, S. Martin-del-Campo, B. Natarajan, and B. Srinivasan, "Addressing uncertainties within active learning for industrial iot," in *2021 IEEE World Forum on Internet of Things (WF-IoT)*, IEEE, in press.
- [26] M. Kalubowilage, O. Covarrubias-Zambrano, A. P. Malalasekera, *et al.*, "Early detection of pancreatic cancers in liquid biopsies by ultrasensitive fluorescence nanobiosensors," *Nanomedicine: Nanotechnology, Biology and Medicine*, vol. 14, no. 6, pp. 1823–1832, 2018.

# Effect of the Period of the Fourier Series Approximation for Binarized Neural Network

SeonYong Lee  
Department of Electrical and  
Computer Engineering  
INMC, Seoul National University  
Seoul, Korea  
sonyongyi@snu.ac.kr

Hee-Youl Kwak  
Department of Electrical and  
Computer Engineering  
INMC, Seoul National University  
Seoul, Korea  
ghy1228@gmail.com

Jong-Seon No  
Department of Electrical and  
Computer Engineering  
INMC, Seoul National University  
Seoul, Korea  
jsno@snu.ac.kr

**Abstract**—The construction of low complexity models for the neural networks is an important issue in practical, real-world scenarios. One of the most famous construction methods for a simple neural network model is to represent weights and activations by the 1-bit quantization, called binarized neural networks (BNNs). However, it is still under research on how to represent the gradient in the backpropagation of BNNs because the activation function is the sign function whose gradients are zero almost everywhere. One way to address this problem is to approximate the gradient of the sign function by the Fourier series representation. In this paper, we analyze the effect of the period and the number of terms of the Fourier series representation on the network accuracy. Since the period has a direct relationship with the degree of the approximation for the sign function and the oscillation behavior of the gradient function, the choice of the period significantly affects the accuracy of the BNN model. The experiments on the CIFAR-10 dataset demonstrate that a proper choice of the period can outperform the conventional BNNs with straight through estimator.

**Index Terms**—Binary neural network (BNN), Fourier series representation (FSR), gradient approximation, period, Straight through estimator (STE)

## I. INTRODUCTION

Recently, machine learning techniques have been applied to a wide range of engineering fields and make a remarkable achievement in such fields. One of the most famous examples is superhuman performance in vision recognition with convolutional neural networks (CNNs) [1]. However, CNNs are not suitable for low-complexity applications such as mobile devices because CNNs need lots of memory and computation (energy) requirements. Since mobile devices (e.g., smartphone, laptop) have several limitations of a small battery, insufficient memory, and low GPU performance, many researchers have proposed several methods such as AlexNet [1], VGGNet [2], and MobileNetV2 [3] to implement cost-efficient architectures, and  $1 \times 1$  convolution [4] to reduce the computational complexity of the arithmetic operation.

Another promising way to reduce the hardware cost dramatically is binarized neural networks (BNNs) [5], where weights and activations are represented by the 1-bit quantization. BNNs have a competitive advantage over other methods in

that they can be constructed from a given well-designed neural networks without changing the key idea of underlying architectures. However, the simply converted BNN using the binary sign function is not feasible to train because the gradient of the sign function is zero almost everywhere, which hinders backpropagation in training. Therefore, BNNs need to employ special techniques to facilitate backpropagation such as straight through estimator (STE) [6] and the approximation by the Fourier series representation (FSR) [7].

Using the FSR, we can transform a periodic function with period  $T$  into the summation of  $n$  triangular functions. For BNNs, the sign function is approximated by the summation of  $n$  differentiable sinusoidal functions, and then their gradients are used for the backpropagation [7]. However, in [7], they did not consider the problem of selecting a proper period  $T$  and number of terms  $n$ .

In this paper, we investigate the effect of the period and number of terms on the approximation of the sign function by the FSR method and the resulting accuracy of the BNN model. As the period  $T$  decreases, the approximation becomes more accurate around the zero-point, but a small period  $T$  induces a problem of the fast oscillating in the gradient domain. In addition, as the number of terms  $n$  increases, the FSR represents the original function more accurately, but it grows the computational complexity linearly. In other words, there is a compromise on the period and number of terms to maximize the accuracy of the model with a reasonable complexity. We evaluate the model accuracy by simply replacing the STE method with the FSR method under the given BNN architecture [5]. Evaluation using the CIFAR-10 dataset shows that the BNN with the FSR can outperform the BNN with the STE if we choose proper values of the period and the number of terms in the FSR.

The remainder of the paper is organized as follows. Section II introduces preliminaries for the BNNs and the STE method. Section III describes the FSR method and its training algorithm. Section IV shows the performance evaluation results and discussions on the period and number of terms in the FSR. Finally, conclusion is given in Section V.

## II. PRELIMINARIES

### A. Binarized Neural Networks

The CNNs are included in a class of full precision neural networks, where the weight matrix  $W$  and activation matrix  $A$  are represented with 32-bit or 64-bit. Instead, the quantized neural networks (QNNs) are those that represent the weight and activation with lower precision. As the extreme case of QNNs, BNNs use 1-bit to represent the binarized weight matrix  $W^b$  and binarized activation matrix  $A^b$  using the sign function,  $\text{Sign}(x)$ , as

$$\text{Sign}(x) = \begin{cases} 1, & \text{if } x \geq 0, \\ -1, & \text{otherwise.} \end{cases}$$

### B. Straight Through Estimator (STE)

Since the gradient of the sign function is zero almost everywhere, training BNNs with the traditional backpropagation method is nearly impossible. Thus, the STE method [6] is proposed to train the conventional BNNs. The key idea of the STE method is to alter the actual gradient to the coarse gradient, which enables to train BNNs. The STE method represents the coarse gradient of the sign function as

$$g_a \approx \text{Clip}\left(\frac{\partial C}{\partial a}, -1, 1\right),$$

where  $g_a$  is real gradient,  $C$  is the cost function,  $a \in A$  is the layer activation, and the clip function,  $\text{Clip}(x, -1, 1)$ , is defined as

$$\text{Clip}(x, -1, 1) = \begin{cases} -1, & \text{if } x < -1, \\ x, & \text{if } -1 \leq x < 1, \\ +1, & \text{otherwise.} \end{cases} \quad (1)$$

Then, the gradient of (1),  $\text{Clip}'(x, -1, 1)$  is given as

$$\text{Clip}'(x, -1, 1) = \begin{cases} 1, & \text{if } -1 \leq x < 1, \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

Although the idea is simple, this method works pretty well. But still, the model accuracy of such BNNs is not accurate enough to satisfy practical demands. For those reasons, researchers have tried to find a way to increase the model accuracy by replacing the STE into BNN+ [8], DSQ [9], and FDA-BNN [7].

## III. FOURIER SERIES BNN

The FSR is a very useful mathematical tool to approximate an original function. A function with period  $T$  can be represented by the summation of  $n$  triangular functions. Let  $\text{FSR}(x; n, T)$  denote the FSR of the sign function,  $\text{Sign}(x)$ , with a period  $T$  and a finite number of terms  $n$ , which is given as

$$\text{FSR}(x; n, T) = \frac{c_0}{2} + \sum_{i=1}^n \left[ c_i \cos \frac{2\pi i x}{T} + s_i \sin \frac{2\pi i x}{T} \right],$$

---

### Algorithm 1 Training process of Fourier series BNN

---

**Require:** A layer weight matrix  $W$ , binarized weight matrix  $W^b$ , layer input  $a_k$ , binarized layer input  $a_k^b$  batchnormalization parameter  $\theta_k$

**Forward propagation:**

**for**  $k = 1$  to  $L$  **do**

$W_k^b \leftarrow \text{Binarize}(W_k)$

$s_k \leftarrow a_{k-1}^b W_k^b$

$a_k \leftarrow \text{BatchNorm}(s_k, \theta_k)$

**if**  $k < L$  **then**

$a_k^b \leftarrow \text{Binarize}(a_k)$

**end if**

**end for**

**Backward propagation:**

Compute  $g_{a_L} = \frac{\partial C}{\partial a_L}$  knowing  $a_L$

**for**  $k = L$  to  $1$  **do**

**if**  $k < L$  **then**

$g_{a_k} \leftarrow g_{a_k^b} \text{FSR}'(x; n, T)(a_k)$

**end if**

$(g_{s_k}, g_{\theta_k}) \leftarrow \text{BackBatchNorm}(g_{a_k}, s_k, \theta_k)$

$g_{a_{k-1}^b} \leftarrow g_{s_k} W_k^b$

$g_{W_k^b} \leftarrow g_{s_k}^\top a_{k-1}^b$

**end for**

{Accumulating the parameters gradients:}

**for**  $k = 1$  to  $L$  **do**

$\theta_k^{t+1} \leftarrow \text{Update}(\theta_k, \eta, g_{\theta_k})$

$W_k^{t+1} \leftarrow \text{Clip}(\text{Update}(W_k, \gamma k \eta, g_{W_k^b}), -1, 1)$

$\eta^{t+1} \leftarrow \lambda \eta$

**end for**

---

where  $c_i$  and  $s_i$  are the  $i$ th Fourier series coefficients. For the sign function, the Fourier series coefficients are given as

$$c_i = 0 \text{ for all } i \text{ and } s_i = \begin{cases} \frac{4}{i\pi}, & \text{if } i \text{ is odd,} \\ 0, & \text{otherwise.} \end{cases}$$

Now we can express the FSR of the sign function as the following reduced form

$$\text{FSR}(x; n, T) = \frac{4}{\pi} \sum_{i=0}^n \frac{\sin(2i+1)\frac{2\pi}{T}x}{2i+1}, \quad |x| < T.$$

Thus, the gradient of the  $\text{FSR}(x; n, T)$  is given as

$$\text{FSR}'(x; n, T) = \frac{8}{T} \sum_{i=0}^n \cos(2i+1)\frac{2\pi}{T}x, \quad |x| < T. \quad (4)$$

The above approximation of the gradient can replace the STE of BNNs. Algorithm 1 represents the Fourier series BNN using the derived gradient of  $\text{FSR}'(x; n, T)$  in (4). Note that the main part of the algorithm follows the conventional BNN in [5], and we just replace the STE method with  $\text{FSR}'(x; n, t)$  as shown in (3). During the backpropagation, the backpropagated gradient from the  $(i+1)$ th layer is multiplied with the gradient of the  $i$ th layer. The conventional BNN using the STE method calculates the backpropagated gradient by (2)



TABLE I  
MODEL ACCURACIES WITH  $n = 20$  AT 100 EPOCHS FOR THE CIFAR-10 DATASET

Model	Method	Period $T$	Accuracy(%)
VGG-Small	STE		90.9
	FSR	10	80.1
	FSR	50	90.4
	FSR	100	90.8
	FSR	150	<b>91.1</b>
	FSR	200	91.0
Resnet-18	FSR	1000	87.4
	STE		86.2
	FSR	10	53.0
	FSR	50	86.0
	FSR	100	<b>87.0</b>
	FSR	150	85.2
	FSR	200	84.8
	FSR	1000	74.3

while our method calculates the backpropagated gradient from the  $(i + 1)$ th layer by (3).

#### IV. RESULTS AND DISCUSSION

##### A. Evaluation Result

We evaluate the proposed FSR method in the BNN on the CIFAR-10 dataset. The CIFAR-10 dataset is one of the famous image classification benchmark datasets. It consists of 50,000 training data and 10,000 test data. Each image consists of  $32 \times 32$  color pixels. Tested BNNs architectures are based on the VGG-small network and the ResNet-18 network. We train the BNN model with the FSR by an optimized learning rate setting in [5]. The stochastic gradient descent optimizer is used with a momentum of 0.9 and a weight decay of 0.999. Test environment builds on cuda toolkit==11.5 and pytorch. And the GPU specification is RTX 2080 Ti.

The results are summarized in Table I and Table II, which show that the period  $T$  and the number of terms  $n$  affect the model accuracy. From the result in Table I, we can see the model accuracy is improved by increasing  $T$  until a certain point but is degraded after that. In addition, Table II shows that the model accuracy is also improved as  $n$  grows but it does not after  $n > 20$ .

##### B. Effect of the Period and Number of Terms in the FSR

Fig. 1 and Fig. 2 show the variation of  $\text{FSR}(x; n, T)$  and  $\text{FSR}'(x; n, T)$  as a function of the period  $T$  and the number of terms  $n$ , respectively. The results apparently show that the better approximation for the sign function,  $\text{Sign}(x)$ , can be achieved by the smaller period  $T$  and the larger number of terms  $n$ . Therefore, one can assume that a smaller period and a larger number of terms lead to better performance. However, the evaluation results in Section IV-A show that the smaller period does not guarantee better performance, and neither larger number of terms do too. We investigate the results and draw the following discussions.

First, the selection of a proper period  $T$  is very important to improve the accuracy of the model using the FSR method.

TABLE II  
MODEL ACCURACIES WITH  $T = 150$  FOR VGG-SMALL AND  $T = 100$  FOR RESNET-18 AT 100 EPOCHS FOR THE CIFAR-10 DATASET

Model	Method	Number of terms $n$	Accuracy(%)
VGG-Small	STE		90.9
	FSR	5	88.3
	FSR	10	90.0
	FSR	20	<b>91.0</b>
	FSR	50	90.8
Resnet-18	STE		86.2
	FSR	5	80.6
	FSR	10	84.6
	FSR	20	<b>87.0</b>
	FSR	50	85.3

Fig. 1(b) shows that small periods such as  $T = 10$  induce a large variation of the gradient near the zero-point  $x = 0$ , which in turn causes the noisy gradient problem [11]. The noisy gradient problem is known to interfere with the training of the network and occur more often as the variation of the gradient increases [11]. Thus, small periods are not preferable in terms of the gradient. On the contrary, large periods such as  $T = 1,000$  in Fig. 1(a) cannot achieve the accurate approximation for the sign function. In other words, there is a trade-off between the accuracy of the approximation and the noisy gradient problem, which can be controlled by period  $T$ . Thus, there is a compromise on the period and Table II shows that the proper period  $T$  is 150 that maximizes the model accuracy.

Second, it is unnecessary to increase the number of terms  $n$  more than a threshold value because there is no additional accuracy gain as  $n$  grows over the threshold while the computation complexity grows linearly by  $n$ . Table 2 shows that the model accuracy has not improved over  $n > 20$ , which means a finite value of  $n$  is sufficient.

#### V. CONCLUSION

In this paper, we investigated the effect of the period and the number of terms in the FSR method for BNN. We replaced the STE method with the FSR method and conducted the evaluations with the various period and the various number of terms in the FSR method. We show that the proper period of the FSR method improves the model accuracy and the certain number of the terms improves the model accuracy. Since the proper period deals with a tradeoff between oscillation of the approximated gradient and precision of the approximated gradient of the sign function around the zero point, the proper period guarantees the better performance. The number of terms has a direct relationship with computational complexity. However, the number of terms does not always improve the model accuracy. Therefore, the proper number of terms is enough to generate the best performance of the BNN model. The experimental results proved that the proper period and the number of terms in the FSR of BNNs outperform the STE method in the model VGG-small network and the ResNet-18 network for the CIFAR-10 dataset.

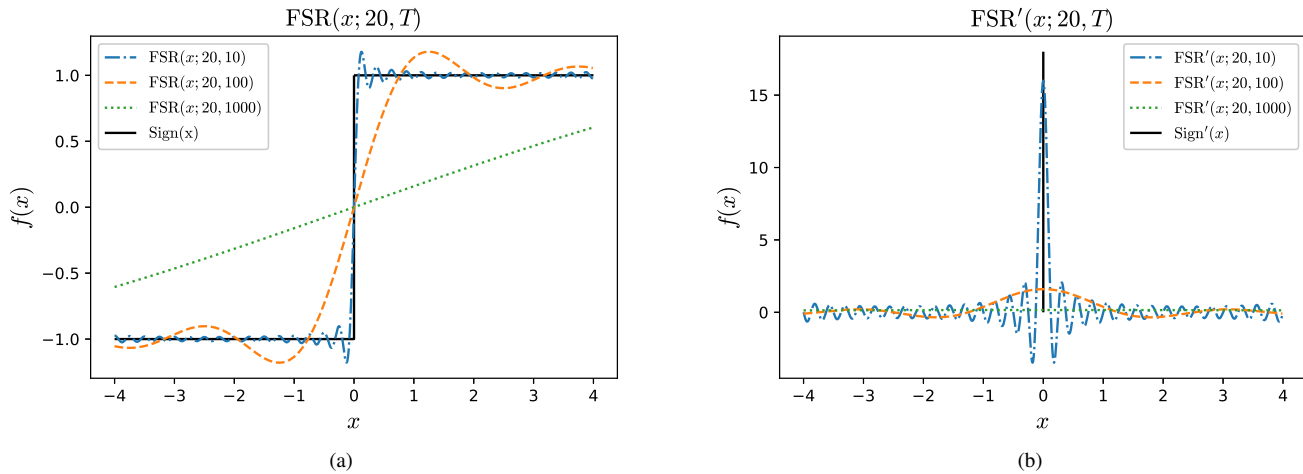


Fig. 1. Comparison of  $\text{Sign}(x)$  and  $\text{FSR}(x; 20, T)$  with various values of  $T$  in terms of (a) original functions and (b) their gradients.

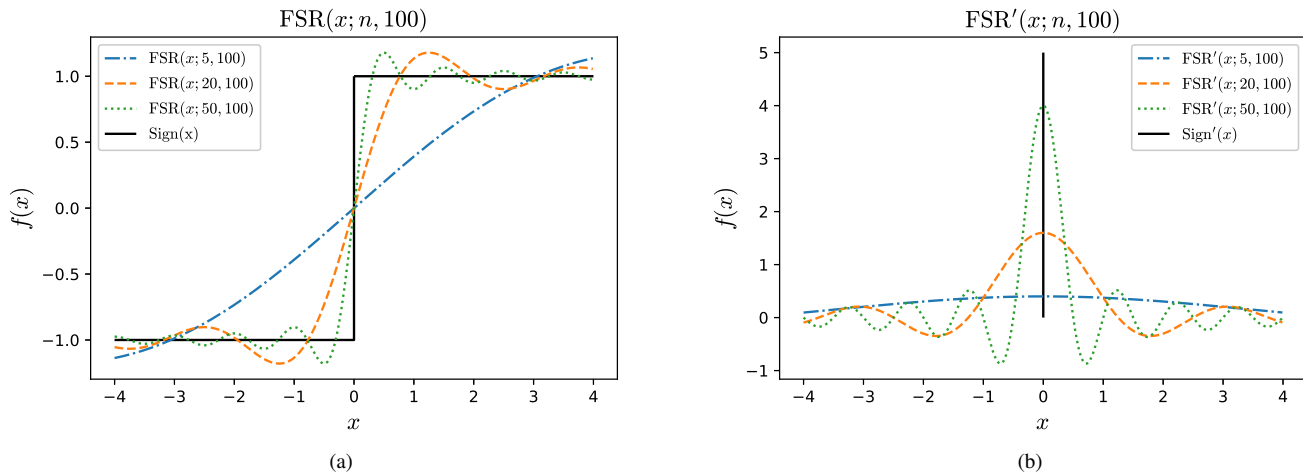


Fig. 2. Comparison of  $\text{Sign}(x)$  and  $\text{FSR}(x; n, 100)$  with various values of  $n$  in terms of (a) original functions and (b) their gradients.

#### ACKNOWLEDGMENT

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2021-0-00400, Development of Highly Efficient PQC Security and Performance Verification for Constrained Devices)

#### REFERENCES

- [1] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097-1105.
- [2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2015.
- [3] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510-4520.
- [4] M. Lin, Q. Chen, and S. Yan, "Network in network," 2014. [Online]. Available: <http://arxiv.org/abs/1312.4400>
- [5] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 4107-4115.
- [6] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," 2013. [Online]. Available: <http://arxiv.org/abs/1308.3432>
- [7] Y. Xu, K. Han, C. Xu, Y. Tang, C. Xu, and Y. Wang, "Learning frequency domain approximation for binary neural networks," 2021. [Online]. Available: <http://arxiv.org/abs/2103.00841>
- [8] S. Darabi, M. Belbahri, M. Courbariaux, and V. P. Nia, "Regularized binary network training," 2020. [Online]. Available: <https://arxiv.org/abs/1812.11800>
- [9] R. Gong, X. Liu, S. Jiang, T. Li, P. Hu, J. Lin, F. Yu, and J. Yan, "Differentiable soft quantization: Bridging full-precision and low-bit neural networks," in *Proc. Int. Conf. Comput. Vision*, 2019, pp. 4852-4861.
- [10] M. Rastegar, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *Proc. Eur. Conf. Comput. vision*, 2016, pp. 525-542.
- [11] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the variance of the adaptive learning rate and beyond," 2019. [Online]. Available: <http://arxiv.org/abs/1908.03265>

# CMCL: Clustering-based Memory Management for Continual Learning

Jiae Yoon and Hyuk Lim  
Gwangju Institute of Science and Technology (GIST)  
Gwangju 61005, Republic of Korea  
jiaeyoon@gm.gist.ac.kr, hlim@gist.ac.kr

**Abstract**—Continual learning (CL) is an incremental learning method to accumulate and refine knowledge over time by continually processing datasets belonging to new tasks. While learning a new task, CL may not retain essential information on previous tasks, which is known as catastrophic forgetting (CF). The CF of information about previous tasks is a challenging problem to overcome for CL. We consider a memory-based approach to combine the previous and new data for CL and propose a memory management method using unsupervised clustering to mitigate the CF. The proposed method generates a set of clusters for the combined datasets by an unsupervised clustering and stores the most representative data belonging to each cluster in memory. The number of clusters is determined by the unsupervised clustering depending on the features of the combined dataset rather than the number of tasks. Further, an experiment was performed to compare the proposed method with existing methods that store a certain amount of data for each task in the memory. The experiment results indicate that the proposed method outperformed the existing methods.

**Index Terms**—Machine learning, continual learning, memory management

## I. INTRODUCTION

Continual learning (CL) refers to the technique of learning multiple tasks sequentially rather than learning the entire data at once. As such the previously learned model is relearned using the newly available dataset. If the model continues to learn new tasks, the learning performance for previous tasks will be gradually reduced, and eventually, only information about the new task remains in the model. This phenomenon is called catastrophic forgetting (CF) in [1]. Numerous studies have been conducted to overcome CF. Kirkpatrick *et al.* proposed elastic weight consolidation (EWC) in [2], which restricts the update of weights that significantly contribute to old tasks. Rusu *et al.* proposed a method of progressively extending the network structure in [3], which adds nodes to the network to learn a new task each time that a new task. Rebuffi *et al.* proposed incremental classifier and representation learning, called iCaRL, in [4], which avoids the CF by storing  $\frac{K}{N}$  samples for every  $N$  class. Shin *et al.* proposed a deep generative replay model in [5], which remembers the previous tasks using a generator that replays the previously learned tasks. Yoon *et al.* proposed a dynamically expandable network, made by combining EWC and progressive networks in [6], which uses regularization to distinguish important weights and dynamically expands the network to prevent CF.

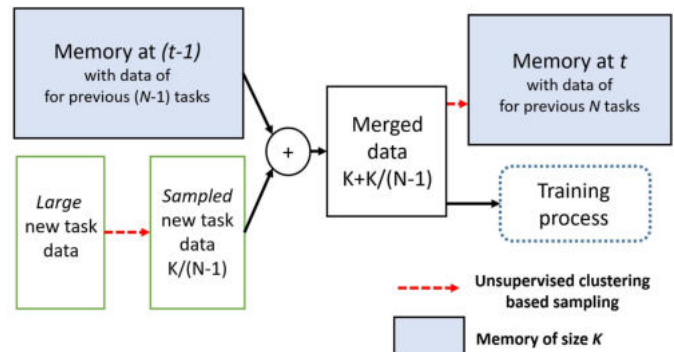


Fig. 1. Proposed memory management method. Data from the previous task stored in the memory and data from a new task are merged to be used for continual learning.

## II. CLUSTERING-BASED MEMORY MANAGEMENT METHOD

We propose a memory management method that decides which sample data belonging to the previous tasks are kept in the memory to be blended with the new sample data for the next task for mitigating CF in CL. The memory management method is based on unsupervised clustering, which generates a certain number of the most representative sample groups to describe the features of data samples used in previous tasks. Notably, the number of clustering groups depends on the characteristics of data sample features and is determined by the clustering algorithm. If the number of groups is  $G$ , and the size of memory is  $K$ ,  $\frac{K}{G}$  samples belonging to each clustering group are stored in the memory. Here, the number of sample groups to be stored in the memory is not the same as that of tasks or classes. It may vary depending on the features of the tasks. Among all samples in each group, the ones closest to the cluster center are chosen to be stored in the memory. The CL is performed using samples belonging to previous tasks and those for the new task to mitigate the CF problem.

Figure 1 shows the procedures of storing training data from the previous task in memory and combining the new task data to create the current training data. The first task trains data without updating the memory. Once the training is completed, the clustering for the data is performed, and the selected samples are stored in the memory. The subsequent tasks train the data comprising the new data and stored data in memory.

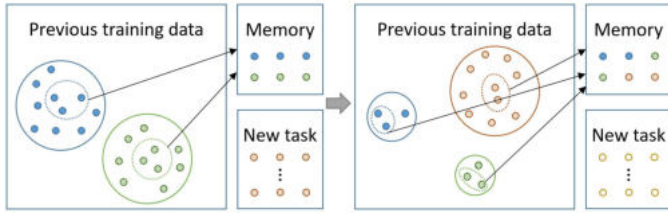


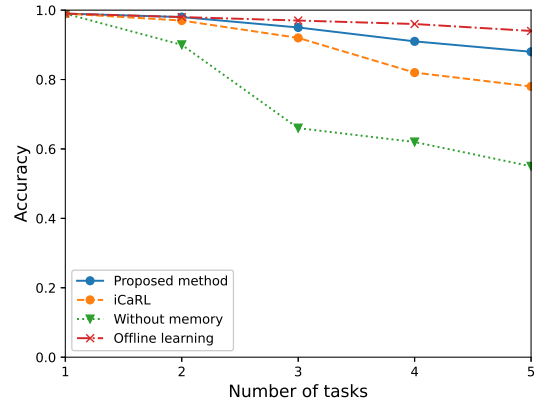
Fig. 2. The process of updating the data to be stored in the memory.

Suppose that the task to be learned at time  $t$  is the  $N$ th task and there were  $(N - 1)$  tasks up to time  $(t - 1)$ . The amount of data samples stored in the memory is  $K$ , and the average number of samples per task is  $\frac{K}{N-1}$ . The amount of data for the new task could be much larger than  $\frac{K}{N-1}$ . This data imbalance may cause a bad effect on training outcomes. To resolve the data imbalance problem among tasks, the amount of training data for the new task is set to  $\frac{K}{N-1}$ . As a result, the amount of the combined dataset is given by  $K \cdot \frac{N}{N-1}$ . The combined data are fed to an unsupervised clustering algorithm. After that,  $K \cdot \frac{N}{N-1}$  data is clustered again and sampled. Eventually,  $K$  data samples remain in the memory. Figure 2 shows how to update the data to store in the memory. For each cluster, the most representative  $\frac{K}{G}$  samples are selected. The most representative samples are the ones closest to the center of the cluster. On average,  $\frac{N-1}{N}$  fraction of the combined data is stored in the memory. The training is performed with the combined data in the memory.

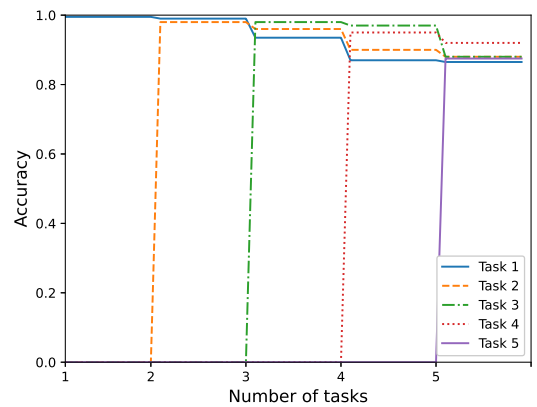
For the unsupervised clustering, we use the density-based spatial clustering of applications with noise (DBSCAN) [7], groups points closely packed with high density to form clusters. According to two DBSCAN parameters for a distance  $\epsilon$  and the minimum number of neighboring points for core points, an arbitrary number of clusters are formed. In general, clustering algorithms can be divided into center-based algorithms and density-based algorithms. Since the center-based clustering algorithm selects data belonging to a cluster according to the distance from the center of a cluster, clusters are formed in the shape of a circle. On the other hand, since a density-based clustering algorithm allows neighboring data to be merged into the same cluster, an unspecified shape of clusters is formed. The advantage of using DBSCAN is that it is not required to specify the number of clusters. In addition, since it can perform clustering and classify noise data at the same time, it can alleviate the decline in clustering performance due to outliers. The data samples of a task may be mapped to multiple clusters, and those of multiple tasks may be mapped to the same cluster. This clustering algorithm can also be exploited to select  $\frac{K}{N-1}$  samples from the dataset belonging to the new task.

### III. EXPERIMENT

We have applied the proposed method to a classification problem. In the experiments, new classes were sequentially



(a) Overall accuracy



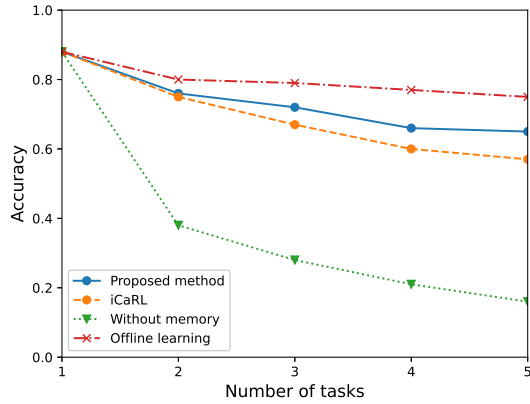
(b) Accuracy of each task for the proposed method

Fig. 3. Results of MNIST dataset experiments in the proposed method, iCaRL, CL without a memory, and offline learning.

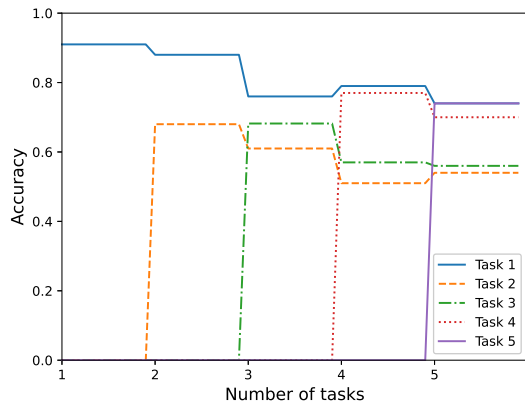
added to a learning model to investigate whether the proposed method was suitable for CL. We have conducted the experiments on the MNIST dataset and CIFAR10 dataset. Both datasets are image datasets with ten classes. The MNIST dataset consists of black and white images, and each image is of  $28 \times 28$  size in one dimension. Unlike MNIST, the CIFAR10 dataset includes the image data of a three-dimensional  $32 \times 32$  size in three colors, RGB. In the experiments, the datasets are divided into five tasks (Tasks 1–5), and two classes are learned in one task. The hyperparameter of DBSCAN was adjusted so that the number of samples stored in the memory for each task was balanced. We compare the proposed method with offline learning and two other methods. The offline learning algorithm learns all data of the current task and previous tasks at the same time. Note that the offline learning’s accuracy is the most ideal result in CL.

#### A. MNIST dataset experiment

In the first experiment, MNIST dataset was used. PCA and t-SNE were used for data feature extraction in this experiment. The learning model consisted of two dense layers and one



(a) Overall accuracy



(b) Accuracy of each task for the proposed method

Fig. 4. Results of CIFAR10 dataset experiments in the proposed method, iCaRL, CL without a memory, and offline learning.

dropout layer. For DBSCAN, the distance was set to 2.7, and the minimum number of neighboring points was set to 100. The size of the memory  $K$  was 20,000. We compared the result of the proposed method with that of iCaRL, LC without a memory, and offline learning. The accuracy of all methods is shown in Figure 3(a). The overall accuracy was measured as the average value of each task’s accuracy. The classification accuracy for the first task was the same. As the number of tasks increased, the classification accuracy gradually decreased. After the training of Task 5, the accuracy of offline learning is 94%. The proposed method, iCaRL, and LC without memory had 88%, 78%, and 55% accuracies, respectively. Figure 3(b) shows the accuracy changes of each task when a new task was added. The accuracy of all tasks is the highest when they first were learned, and each time a new task was added, the accuracy dropped little by little. The width of the reduction in accuracy of each task is somewhat constant.

## B. CIFAR10 dataset experiment

The second experiment was conducted on the CIFAR10 dataset. ResNet50 was used for learning and data feature extraction of clustering in the experiment. For DBSCAN, the distance was set to 8, and the minimum number of neighboring points was set to 100. The size of the memory  $K$  was 20,000. We compared the result of the proposed method with that of iCaRL, LC without a memory buffer, and offline learning. Figure 4(a) shows the overall accuracy with respect to the number of tasks for the three methods. The classification accuracy for the first task was the same. As the number of tasks increased, the classification accuracy gradually decreased. After the training of Task 5, the proposed method, iCaRL, LC without memory, and the offline learning had an accuracy of 65%, 57%, 16%, and 75%, respectively. Figure 4(b) shows the accuracy changes of each task when a new task was added. The accuracy of Task 1 decreased as new tasks were added, but the degradation rate was not severe compared with the other tasks. The accuracies of Tasks 4 and 5 were almost the same as that of Task 1, whereas Tasks 2 and 3 achieved a lower performance than others.

## IV. CONCLUSION

We proposed an unsupervised clustering-based memory management method for CL. The proposed method keeps a certain amount of previous data samples in the memory to mitigate CF. Instead of dividing the memory resource into the same number bins as the number of classes or tasks, it clusters the combined dataset of the previous and new tasks and equally allocates the memory resource to the samples belonging to each cluster. The results of the experiments using two different datasets indicated that the proposed method performed better than the other LC methods.

## ACKNOWLEDGMENTS

This work was supported by IITP grant funded by the Korea government (MSIT) (No. 2021-0-00379, Privacy risk analysis and response technology development for AI systems)

## REFERENCES

- [1] R. M. French, “Catastrophic forgetting in connectionist networks,” *Trends in Cognitive Sciences*, vol. 3, no. 4, pp. 128–135, 1999.
- [2] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Venessa, and G. Desjardins, “Overcoming catastrophic forgetting in neural networks,” in *Proceedings of the National Academy of Sciences of the United States of America*. PMLR, 2017, pp. 3521–3526.
- [3] A. Rusu, N. Rabinowitz, and G. Desjardins, “Progressive neural networks,” in *arXiv*. PMLR, 2016, pp. 2021–2031.
- [4] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. Lampert, “icarl: Incremental classifier and representation learning,” in *Computer Vision Foundation Open Access*. PMLR, 2017, pp. 2001–2010.
- [5] H. Shin, J. K. Lee, J. Kim, and J. Kim, “Continual learning with deep generative replay,” in *arXiv*. PMLR, 2017, pp. 2021–2031.
- [6] J. Yoon, E. Yang, J. Lee, and S. J. Hwang, “Lifelong learning with dynamically expandable networks,” 2018.
- [7] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD’96. AAAI Press, 1996, p. 226–231.

# TinyML: A Systematic Review and Synthesis of Existing Research

Hui Han

Data Science Department

Fraunhofer Institute for Experimental Software Engineering IESE

Kaiserslautern, Germany

hui.han@alumnos.upm.es

Julien Siebert

Data Science Department

Fraunhofer Institute for Experimental Software Engineering IESE

Kaiserslautern, Germany

julien.siebert@iese.fraunhofer.de

**Abstract**— Tiny Machine Learning (TinyML), a rapidly evolving edge computing concept that links embedded systems (hardware and software) and machine learning, with the purpose of realizing ultra-low-power and low-cost and efficiency and privacy, brings machine learning inference to battery-powered intelligent devices. In this study, we conduct a systematic review of TinyML research by synthesizing 47 papers from academic and grey publication since 2019 (the early TinyML publication starts from 2019). Relevant TinyML literature is analyzed from five aspects: hardware, framework, datasets, use cases, and algorithms/model. This systematic review will serve as a roadmap for understanding the literature within the new emerging field of TinyML.

**Keywords**—TinyML, Systematic review, Data synthesis, MCUs, TensorFlow Lite, Neural networks

## I. INTRODUCTION

TinyML (tiny machine learning) is a relatively new term that encompasses research and development at the intersection of embedded systems and machine learning [1]. The goal of TinyML is to apply machine learning inference on extremely low-power (under a milliwatt), low-cost (55 cents in 2023) microcontrollers (MCUs) [2]. Through the implementation of various battery-powered MCUs and streaming applications, TinyML facilitates real-time, on-site data collection, processing, analysis and interpretation. [3]. As a result, TinyML provides low latency, low power, and high privacy while avoiding the energy cost and data loss associated with wireless communication between edge devices and the cloud [4].

Although TinyML is an important and growing field, the academic research associated with the term remains at a very early stage at this time. As a result, efforts to synthesize TinyML research into an integration of a broad body of knowledge have been relatively limited [5]. To fill this gap, we propose a systematic method and synthesis of current research on TinyML.

A synthesis from various aspects can clarify issues and identify relations in a structured manner [6]. Therefore, to assist scholars to improve the understanding of TinyML, we adopt synthesis with a comprehensive and structured list of elements. Specifically, this article contributes to the TinyML literature by synthesizing current research with five aspects: hardware, framework, datasets, use cases, and algorithms/model.

This review serves as guidance for research exploration on TinyML. It aims to gain a better understanding of the state of the art of TinyML and to offer guidelines to TinyML practice. This study also provides researchers and practitioners with directions for future research in this emerging field.

The remainder of the paper is organized as follows. In Section II, we present a note on review methodology. Section III discuss the search results and Section IV describes data synthesis results. Finally, in Section V, we provide future research needs, and conclude.

## II. METHODOLOGY

A systematic review approach is used to select studies and identify relevant articles [7]. Additionally, a synthesis methodology is adopted to synthesize the broad scope of selected papers in an integrative way [8]. PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) instruction is applied here to display a flow diagram of the literature search process (shown in Fig. 1).

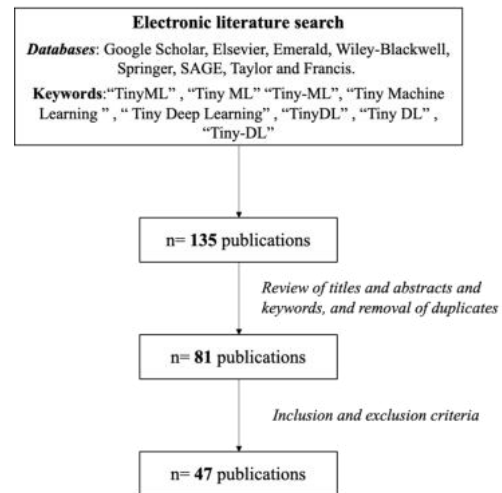


Fig. 1. PRISMA flow diagram.

### A. Searching

We utilize web-based resources (Google Scholar) as the database. In addition, we manually search for articles from key authors in the TinyML field. The search strategy (each keyword for each time separately) is detailed below:

- The keyword “TinyML” is used for the first-round search.
- The keyword “Tiny ML” is used for the second-round search.
- The keyword “Tiny-ML” is used for the third-round search.
- The keyword “Tiny Machine Learning” is used for the fourth round search.
- The keyword “ Tiny Deep Learning” is used for the fifth-round search.
- The keyword “Tiny-DL” is used for the sixth-round search.
- The keyword “TinyDL” is used for the seventh-round search.
- The keyword “Tiny DL” is used for the eighth-round search.

In order to full search, we also use the six main databases as complementary sources:

- Elsevier (<https://www.sciencedirect.com/>);
- Emerald (<https://www.emeraldinsight.com/>);
- Wiley-Blackwell (<https://onlinelibrary.wiley.com/>);
- Springer (<https://www.springer.com/fr>);
- SAGE (<https://us.sagepub.com/en-us/nam/home>);
- Taylor and Francis (<https://www.tandfonline.com/>).

### B. Selection

In order to give broader insights, selected studies not only contain academic literature but also grey publishing such as reports and working paper and the online newspaper. The publication is chosen in two steps: first by review of title and abstract and keywords, then by full-text review. The selection is not restricted by year of publication but papers have to be written in English because of limited translation resources (such as lack of language experts for translation work). All web-based resources are accessible in printed or downloadable form.

The initial literature search yielded 135 results. After reviewing the titles, abstracts and keywords, as well as removing duplicates, we obtained 81 unique records for further evaluation. After applying the inclusion and exclusion criteria when reading the full text, we excluded 34 articles for a final total of 47 articles (36 primary studies and 11 reviews) used for data synthesis.

### C. Data Extraction and Data Synthesis

For data extraction and synthesis, we employ the content analysis method, which identifies the appearance of specific words, topics, or concepts within a text [9].

The purpose of the data synthesis approach is to integrate diverse range of studies into a conceptual map describing five TinyML elements: hardware, framework, datasets, use cases, and algorithms/model. Full details of the data extraction and synthesis are available from the first author upon request.

## III. SEARCH RESULTS

Of the 47 publications in the review, 26 publications are conference proceedings, 15 journal articles including 13 research articles and 2 review articles, 2 book sections and 1 book, 1 report, 1 working paper and 1 newspaper article. We list the reference in Table 1.

TABLE I. PUBLICATION TYPES

Publication Types	References
Conference Proceedings	[10][11][2][12][13][14][15][16][17][18][19][20][21][22][23][24][25][26][27][28][29][30][31][32][33][34]
Research Article	[3][35][36][37][38][39][40][41][42][43][44][45][46]
Review Articles	[1][4]
Book Section	[47]
Book	[48][49]
Report	[5]
Working Paper	[50]
Newspaper Article	[51]

Figure 2 lists the main sources (by year) that have issued two or more publishing items. In total, there are 24 different types of sources involved in this review. Among the publication, most papers are published in different conferences, and 6 articles are published in 4 types of journals respectively: Sensors (3 papers), IEEE Circuits and Systems Magazine (1 paper), IEEE Transactions on Circuits and Systems II: Express Briefs (1 paper) and Journal of Sensor and Actuator Networks (1 paper). TinyML Research Symposium (10 papers) is very popular among these conferences referring to TinyML. In addition, 11 papers were pre-print from arXiv and one book was published in 2019 by O’Reilly Media and one newspaper article is from IEEE IoT Newsletter.

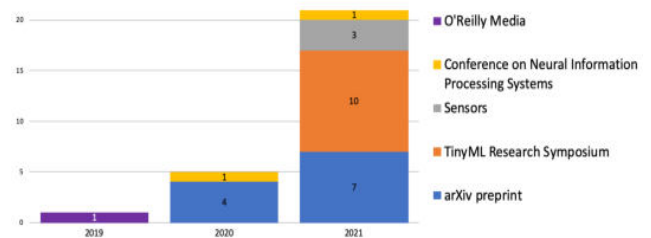


Fig. 2. Publication source since 2019.

## IV. DATA SYNTHESIS RESULTS

Synthesis can spot the location of every issue on an integrative map of TinyML. This step aims to paint one abstraction frame to precisely record the information gained from selected studies. We use Mendeley and Microsoft Excel spreadsheets to synthesis the 47 papers into five elements:

hardware, framework, datasets, use cases, and algorithms/model [1], [2], [4], [41].

### A. Hardware

TinyML is operated on low-power microcontrollers boards with extensive hardware extraction [50]. In this study, we list the main hardware (Table 2) applied by two or more publishing items. 8 papers generally mentioned Microcontroller units (MCUs) without hardware details. On the other hand, 5 papers clearly specified STM32 MCU and 2 papers on Apollo3 MCU. ARM Cortex-M processors (mentioned by 4 papers) are commonly integrated by these MCUs when applied TinyML.

TABLE II. HARDWARE

Hardware	Count
MCUs	8
STM32	5
ARM Cortex-M (M0 and M7) series	4
Amiq Apollo 3	2
Arduino Nano 33 BLE Sense	2
FPGA	2

In contrast to cell phone and cloud platforms, MCUs are generally small (around 1cm<sup>3</sup>), low-cost (around \$1) and energy-saving (around 1mW). Therefore, they are the ideal hardware platforms for TinyML. An MCU consists of a CPU, embedded flash (eFlash) memory for code and Static Random Access Memory (SRAM) for data bit, as well as input/output peripherals [18]. Specifically, microcontrollers from the STM32 (32-bit) family, based on ARM Cortex-M processors, support both small projects and end-to-end platforms. In detail, the ARM Cortex-M processors perform a single operation at a time, which are optimized for low-cost, low latency and low power [52]. Additionally, the Apollo3 MCU is designed for ultra-low power and portable, smart devices with an integrated ARM Cortex-M processor [53]. Arduino Nano 33 BLE Sense contains a series of embedded sensors, a Cortex-M4 microcontroller and BLE [50]. Field Programmable Gate Arrays (FPGAs) are semiconductor-integrated circuits that execute all operations in a parallel way [33].

### B. Framework

TinyML frameworks are typically used to enable ML models into various MCU-based edge devices [1].

The variety of embedded systems including hardware and software needs to be addressed for TinyML to obtain board understanding [5]. Therefore, we analyzed the framework (software) concerning hardware shown in Fig. 3. From Figure 3, we can find the relation between hardware and the framework (software). For instance, TensorFlow Lite (10 papers) is the most well-known framework (software) adopted in hardware. It is frequently an alternative for the term “TinyML”. TensorFlow Lite is Google’s open-source machine learning framework that deploys a special format model on mobile and embedded devices [5], such as STM32, Apollo3 and ARM Cortex-M7. The

Framework TinyOL is adopted by Arduino Nano 33 BLE Sense. Tiny RespNet TensorFlow and PYNQ are used by FPGA.

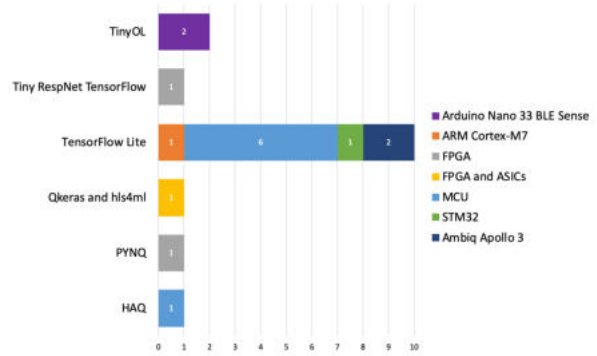


Fig. 3. TinyML frameworks by hardware.

### C. Use Cases

Although TinyML is in its infancy, there are a large amount of well-established use cases that apply TinyML in solving real-life issues [4]. In this study, the use cases are keyword spotting, image classification, visual wake words, object detection, anomaly detection, semantic segmentation, motor control, gesture recognition, forecasting, face recognition and activity detection. In addition, we noticed 16 papers that proposed a new tool or improve the current technology without belonging to any use cases type. Therefore, we added the technology improvement/new tools as another new type of use case. Fig. 4 shows an overview of the use cases by categorizing 47 papers (we named the new use case type as “technology improvement/new tools”, therefore it is not shown in the Fig. 4).

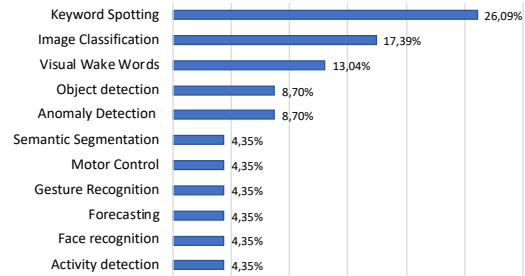


Fig. 4. TinyML use cases.

The dominant use case is keyword spotting with 6 articles which cover 26.09% of TinyML studies. The second use case is image classification, where 17.39% of the studies classified, followed by visual wake words with 13.04%. Object detection and anomaly detection are the same ordered with 8.07% (2 papers). The remaining six use cases are semantic segmentation, motor control, gesture recognition, forecasting, face recognition and activity detection (4.35%).

The most popular and largely deployed case of TinyML is keyword spotting [37]. Keyword spotting is the sensible detection of certain words and short sentences. For example, the initiation of virtual assistants like Siri (Apple), Cortana (Microsoft), and Alexa (Amazon). TinyML can be widely used



for keyword spotting because a specific word or phrase identification needs accordingly low power consumption [2].

Anomaly detection is generally deployed on MCUs that divide normal samples from abnormal samples [2]. It has numerous applications such as checking for anomalous audio, temperature or IMU (inertial measurement unit) data to issue early warnings of potential breakdowns [50].

Machine learning inference of various sensor data from low-power image sensors, photoplethysmogram (PPG) optical sensors, gyroscope sensors, microphones, accelerometers, and other embedded devices enable market and industrial applications such as image classification, face recognition, object detection, gesture recognition, semantic segmentation, and forecasting [37]. Some use cases have been proven feasible, but have yet to contact consumers as they are too new, like visual wake words [4].

#### D. Datasets

Many free and public datasets are relevant to TinyML use cases [4]. Fig. 5 shows the relationship between the datasets and TinyML use cases. Speech commands are audio datasets released by Google for training and evaluating keyword spotting systems, which sorts short audio clips into a distinct set of classes [23]. COCO dataset is applied to train, validate and test for visual wake words models [2]. When referring to the application of two datasets or more, we find the use case of image classification commonly employs datasets ImageNet [16] and MNIST [19], [25]. ImageNet is also used in the use case of visual wake words [18]. From Figure 5, we could find the suitable datasets for each use case model evaluation.

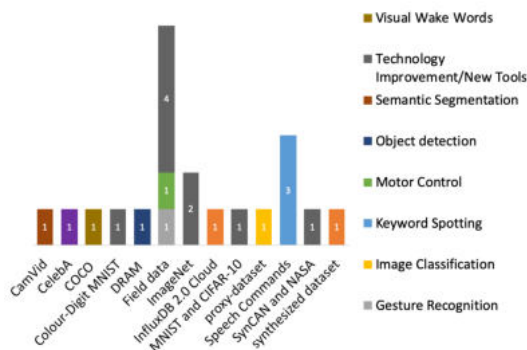


Fig. 5. Existing datasets by TinyML use cases.

Notwithstanding the accessibility of these open-source datasets, most of deployed TinyML models are trained and evaluated on massive datasets. These proprietary datasets that are huge in size are not specific for developing TinyML applications. The lack of suitable TinyML datasets poses a substantial obstacle to the progress of academic research [4].

#### E. Algorithms/Model

Fig. 6 lists the typical algorithm/model for solving TinyML use cases problems. As we know, neural networks (NN) are the main force for both traditional machine learning and TinyML [15]. In particular, due to low CPU and memory usage, some

TinyML use cases can also use non-NN algorithms like random forest [24].

From Fig. 6, we find convolutional neural networks (CNN) are employed in image recognition and computer vision such as TinyML use cases image classification, face recognition and activity detection. Deep neural networks (DNNs) have been used by many TinyML use cases such as visual wake words, keyword spotting and image classification. However, one of the main challenges to use DNNs for solving TinyML use cases is the steadily growing number of parameters (from millions to billions to 1 trillion parameters within the next decade) [13]. In addition, some papers only mentioned NN in general without pointing out exact type of NN (CNN, DNN, RNN) used. Therefore, Fig. 6 displays CNN and DNN and NN at the same time.

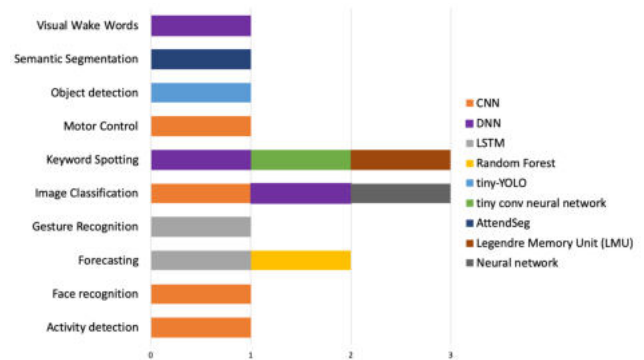


Fig. 6. TinyML algorithm/model by use cases.

Long short-term memory (LSTM) is used for natural language processing such as speech recognition or music generation [54]. Using LSTM particularly with CNN for human gesture recognition and motor control is well documented to obtain a high level of accuracy [14].

#### V. CONCLUSION

TinyML is an essential and fast-developing field that requires trade-offs among diverse integral components (hardware, software, machine learning algorithms) [5]. In this paper, we contribute a systematic literature review in reference to the data synthesis results of 47 publications on TinyML since 2019.

We focus on five elements: hardware, framework, datasets, use cases, and algorithms/models. Future studies could add more elements like the TinyML application area (Industry 4.0, vehicular services, smart spaces, smart agriculture and farming, eHealth, etc.).

TinyML has the potential to exploit an entirely new domain of smart applications throughout manufacturing and business and personal life areas [27]. TinyML proposes innovative solutions in different fields and provides novel study directions, which would be a promising area for scholars to explore [5].

#### ACKNOWLEDGMENT

The research of Dr. Hui Han is funded by the European Research Consortium for Informatics and Mathematics (ERCIM) Alain Bensoussan Fellowship Programme and the Fraunhofer Institute for Experimental Software Engineering IESE.

## REFERENCES

- [1] T. S. Ajani, A. L. Imoize, and A. A. Atayero, "An overview of machine learning within embedded and mobile devices – optimizations and applications," *Sensors*, vol. 21, no. 13, pp. 1–44, 2021.
- [2] C. Banbury *et al.*, "MLPerf tiny benchmark," *Conference on Neural Information Processing Systems*, 2021. [Online]. Available: <http://arxiv.org/abs/2106.07597>.
- [3] G. Signoretti, M. Silva, P. Andrade, I. Silva, E. Sisinni, and P. Ferrari, "An evolving tinyml compression algorithm for IOT environments based on data eccentricity," *Sensors*, vol. 21, no. 12, pp. 1–25, 2021.
- [4] C. R. Banbury *et al.*, "Benchmarking tinyML systems: challenges and direction," *arXiv preprint arXiv:2003.04821*, 2020. [Online]. Available: <http://arxiv.org/abs/2003.04821>.
- [5] S. Soro, "TinyML for ubiquitous edge AI," 2021.
- [6] H. Han, H. Xu, and H. Chen, "Social commerce: A systematic review and data synthesis," *Electron. Commer. Res. Appl.*, vol. 30, pp. 38–50, 2018.
- [7] B. Kitchenham, "Procedures for performing systematic reviews," *Keele Univ.*, vol. 33, pp. 1–26, 2004.
- [8] C. L. Downey, W. Tahir, R. Randell, J. M. Brown, and D. G. Jayne, "Strengths and limitations of early warning scores: A systematic review and narrative synthesis," *Int. J. Nurs. Stud.*, vol. 76, pp. 106–119, 2017.
- [9] K. Krippendorff, *Content analysis: an introduction to its methodology*. SAGE Publications Inc, 2004.
- [10] F. Alongi, N. Ghielmetti, D. Pau, F. Terraneo, and W. Fornaciari, "Tiny neural networks for environmental predictions: an integrated approach with Miosix," in *IEEE International Conference on Smart Computing*, 2020, pp. 350–355.
- [11] M. Lootus, K. Thakore, S. Leroux, G. Trooskens, A. Sharma, and H. Ly, "A VM / containerized approach for scaling tinyML applications," in *TinyML Research Symposium*, 2021, pp. 1–6.
- [12] P. Blouw, G. Malik, B. Morcos, A. R. Voelker, and C. Eliasmith, "Hardware aware training for efficient keyword spotting on general purpose and specialized hardware," in *TinyML Research Symposium*, 2021, pp. 1–5.
- [13] S. Ghamari *et al.*, "Quantization-guided training for compact tinyML models," in *TinyML Research Symposium*, 2021.
- [14] B. Coffen and M. S. Mahmud, "TinyDL: edge computing and deep learning based real-time hand gesture recognition using wearable sensor," in *IEEE International Conference on E-Health Networking, Application and Services*, 2020, pp. 1–6.
- [15] G. Crocioni, G. Gruosso, D. Pau, D. Denaro, L. Zambrano, and G. di Giore, *Characterization of neural networks automatically mapped on automotive-grade microcontrollers*. Association for Computing Machinery, 2021.
- [16] S. Disabato and M. Roveri, "Incremental on-Device tiny machine learning," in *International Workshop on Challenges in Artificial Intelligence and Machine Learning for Internet of Things*, 2020, pp. 7–13.
- [17] H. Doyu, R. Morabito, and M. Brachmann, "A tinyMLaaS ecosystem for machine learning in IoT: overview and research challenges," in *International Symposium on VLSI Design, Automation and Test*, 2021, pp. 1–6.
- [18] C. Banbury *et al.*, "Micronets: neural network architectures for deploying tinyml applications on commodity microcontrollers," in *Proceedings of Machine Learning and Systems*, 2021, pp. 1–16.
- [19] F. Fahim *et al.*, "hls4ml: an open-source codesign workflow to empower scientific low-power machine learning devices," in *TinyML Research Symposium*, 2021, pp. 1–10.
- [20] M. Giordano, P. Mayer, and M. Magno, "A battery-free long-range wireless smart camera for face detection," in *International Workshop on Energy Harvesting and Energy-Neutral Sensing Systems*, 2020, pp. 29–35.
- [21] J. Kwon and D. Park, "Toward data-adaptable tinyML using model partial replacement for resource frugal edge device," in *International Conference on High Performance Computing in Asia-Pacific Region*, 2021, pp. 133–135.
- [22] H. F. Langroudi, V. Karia, T. Pandit, and D. Kudithipudi, "TENT: efficient quantization of neural networks on the tiny edge with Tapered FixEd PoiNT," in *TinyML Research Symposium*, 2021, pp. 1–8.
- [23] J. Lin, W.-M. Chen, Y. Lin, J. Cohn, C. Gan, and S. Han, "MCUNet: tiny deep learning on IoT devices," in *Conference on Neural Information Processing Systems*, 2020, pp. 1–15.
- [24] A. Navaas Roshan, B. Gokulapriyan, C. Siddarth, and P. Kokil, "Adaptive traffic control with tinyML," in *International Conference on Wireless Communications, Signal Processing and Networking*, 2021, pp. 451–455.
- [25] A. J. Paul, P. Mohan, and S. Sehgal, "Rethinking generalization in american sign language prediction for edge devices with extremely low memory footprint," in *IEEE Recent Advances in Intelligent Computational Systems*, 2020, pp. 147–152.
- [26] H. Ren, D. Anicic, and T. A. Runkler, "The synergy of complex event processing and tiny machine learning in industrial IoT," in *ACM International Conference on Distributed and Event-based Systems*, 2021, pp. 126–135.
- [27] B. Sudharsan *et al.*, "TinyML benchmark: executing fully connected neural networks on commodity microcontrollers," in *IEEE World Forum on Internet of Things*, pp. 19–21.
- [28] F. Svoboda *et al.*, "Resource efficient deep reinforcement learning for acutely constrained tinyML devices," in *TinyML Research Symposium*, 2021, pp. 1–8.
- [29] C. Toma, M. Popa, and M. Doinea, "A.I. neural networks inference into the IoT embedded devices using tinyml for pattern detection within a security system," in *International Conference on Informatics in Economy Education, Research and Business Technologies*, 2020, pp. 14–22.
- [30] C. Vuppapapati, A. Ilapakurti, K. Chillara, S. Kedari, and V. Mamidi,

- “Automating tiny ML intelligent sensors DevOPS using microsoft Azure,” in *IEEE International Conference on Big Data*, 2020, pp. 2375–2384.
- [31] C. Vuppapapati, A. Ilapakurti, S. Kedari, J. Vuppapapati, S. Kedari, and R. Vuppapapati, “Democratization of AI, albeit constrained IoT devices & Tiny ML, for creating a sustainable food future,” in *International Conference on Information and Computer Technologies*, 2020, pp. 525–530.
- [32] S. Siddiqui, C. Kyrkou, and T. Theocharides, “Mini-NAS: a neural architecture search framework for small scale image classification applications,” in *TinyML Research Symposium*, 2021, pp. 1–8.
- [33] B. Jiao *et al.*, “A 0.57-GOPS / DSP object detection PIM accelerator on FPGA,” in *Asia and South Pacific Design Automation Conference*, 2021, pp. 13–14.
- [34] H.-A. Rashid, H. Ren, A. N. Mazumder, and T. Mohsenin, “Tiny RespNet: a scalable multimodal tinyCNN processor for automatic detection of respiratory symptoms,” in *TinyML Research Symposium*, 2021, pp. 1–8.
- [35] X. Wen, M. Famouri, A. Hryniowski, and A. Wong, “AttendSeg: a tiny attention condenser neural network for semantic segmentation on the edge,” *arXiv preprint arXiv:2104.14623*, 2021. [Online]. Available: <http://arxiv.org/abs/2104.14623>.
- [36] A. Capotondi, M. Rusci, M. Fariselli, and L. Benini, “CMix-NN: mixed low-precision CNN Library for Memory-Constrained Edge Devices,” *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 67, no. 5, pp. 871–875, 2020.
- [37] R. David *et al.*, “TensorFlow Lite Micro: embedded machine learning on tinyml systems,” *arXiv preprint arXiv:2010.08678*, 2020. [Online]. Available: <http://arxiv.org/abs/2010.08678>.
- [38] M. de Prado *et al.*, “Robustifying the deployment of tinyML models for autonomous mini-vehicles,” *Sensors*, vol. 21, no. 1339, pp. 1–16, 2021.
- [39] L. Heim, A. Biri, Z. Qu, and L. Thiele, “Measuring what really matters: optimizing neural networks for tinyML,” *arXiv preprint arXiv:2104.10645*, 2021. [Online]. Available: <http://arxiv.org/abs/2104.10645>.
- [40] H. Ren, D. Anicic, and T. Runkler, “TinyOL: tinyML with online-learning on microcontrollers,” *arXiv preprint arXiv:2103.08295*, 2021. [Online]. Available: <http://arxiv.org/abs/2103.08295>.
- [41] R. Sanchez-Iborra and A. F. Skarmeta, “TinyML-enabled frugal smart objects: challenges and opportunities,” *IEEE Circuits Syst. Mag.*, vol. 20, no. 3, pp. 4–18, 2020.
- [42] A. Wong, M. Famouri, and M. J. Shafiee, “AttendNets: tiny deep image recognition neural networks for the edge via visual attention condensers,” *arXiv preprint arXiv:2009.14385*, 2020. [Online]. Available: <http://arxiv.org/abs/2009.14385>.
- [43] M. Z. H. Zim and B. Dhaka, “TinyML: analysis of Xtensa LX6 microprocessor for neural network applications by ESP32 SoC,” *arXiv preprint arXiv:2106.10652*, 2021. [Online]. Available: <http://arxiv.org/abs/2106.10652><http://dx.doi.org/10.13140/RG.2.2.28602.11204>.
- [44] C. Campolo, G. Genovese, A. Iera, and A. Molinaro, “Virtualizing AI at the distributed edge towards intelligent IOT applications,” *J. Sens. Actuator Networks*, vol. 10, no. 1, pp. 1–14, 2021.
- [45] H. Miao and F. X. Lin, “Enabling large NNs on tiny MCUs with swapping,” *preprint arXiv:2101.08744*, 2021. [Online]. Available: <http://arxiv.org/abs/2101.08744>.
- [46] A. Wong, M. Famouri, M. Pavlova, and S. Surana, “TinySpeech: attention condensers for deep speech recognition neural networks on edge devices,” *arXiv preprint arXiv:2008.04245*, 2020. [Online]. Available: <http://arxiv.org/abs/2008.04245>.
- [47] P. Mohan, A. J. Paul, and A. Chirania, “A tiny cnn architecture for medical face mask detection for resource-constrained endpoints,” in *Innovations in Electrical and Electronic Engineering*, vol. 756, 2020, pp. 657–670.
- [48] P. Warden and D. Situnayake, *TinyML: machine learning with TensorFlow Lite on Arduino and ultra-low-power microcontrollers*. 2019.
- [49] M. Rusci, M. Fariselli, A. Capotondi, and L. Benini, “Leveraging automated mixed-low-precision quantization for tiny edge microcontrollers,” in *IoT Streams for Data-Driven Predictive Maintenance and IoT, Edge, and Mobile for Embedded Machine Learning*, 2020, pp. 296–308.
- [50] V. J. Reddi *et al.*, “Widening access to applied machine learning with tinyML,” 2021.
- [51] H. Doyu, E. Research, R. Morabito, and J. Höller, “Bringing machine learning to the deepest IoT edge with tinyML as-a-service,” *IEEE IoT Newsletter*, pp. 1–4, 2020.
- [52] Cortex-M, “Arm Cortex-M series processors,” *ARM Developer*, 2021. [Online]. Available: <https://developer.arm.com/ip-products/processors/cortex-m>.
- [53] Ambiq, “Apollo3 Blue MCU,” *Ambiq*, 2021. [Online]. Available: <https://www.ambiq.top/en/apollo3-blue-mcu-eval-board>.
- [54] R. Jain, V. B. Semwal, and P. Kaushik, “Deep ensemble learning approach for lower extremity activities recognition using wearable sensors,” *Expert Syst.*, pp. 1–17, 2021.

# A Survey of Procedural Content Generation of Natural Objects in Games

Tianhan Gao  
Software College  
Northeastern University  
Shenyang, China  
gaoth@mail.neu.edu.cn

Jiahui Zhu  
Software College  
Northeastern University  
Shenyang, China  
2071361@stu.neu.edu.cn

**Abstract**—The natural environment is one of the important research fields in game design and development. A good game environment is a key factor in the process of game development and player experience. Procedural Content Generation (PCG) is currently a wide range of fully automatic game environment generation technology. This paper introduces some algorithms and experimental results of PCG for natural objects (such as vegetation, river, and terrain), with special attention to the applicability and aesthetic visualization. It is found that the appearance of PCG greatly reduces the time needed to design large-scale natural landscape for game levels. Furthermore, PCG is able to adjust the natural landscape in real time to improve the overall game development efficiency.

**Keywords**—game development, procedural content generation, natural objects, procedural modeling methods

## I. INTRODUCTION

With the rise of mobile games, video games have become the most profitable parts in the entertainment industry. According to the "2020 China Game Industry Report", the actual sales revenue in China's game market has reached 278.687 billion yuan in 2020. The game industry has developed for more than ten years, and the technology of games and modeling is also constantly changed. From the early arcade video games to the present 3D virtual world, the automatic creation of content has a huge attraction for game production. For most designers, creating a game world still requires a lot of repetitive labor. In order to cope with such heavy work, designers have introduced a novel procedural content generation method that enables designers to create a complete 3D world within a short time. Procedural Content Generation (PCG) is a game design technology which uses automation generation technology instead of traditional manual creation. This automatic production mode greatly improves the production speed and effectively reduces the cost, and the error rates are low. *Elite* (1984) and its sequel *Frontier* (1993) are typical examples, putting an endless universe into a small floppy disk, dynamically generated galaxy system, dynamically generated checkpoints, and a whole set of features is all generated by programs. *Krieger* (2004) is a first-person shooting game in which everything is created dynamically while the program is running, including level, map, model, animation, and sound effects. But it is incredible that the entire game is only 96kb. According to the traditional game storage method, such a beautiful game may take a few hundred megabytes of space. Procedural content generation rules are not only applicable to video games. Desktop games such as *Catan: World Explorers* (1995) [1] requires players to create colonies using randomly assigned natural resources, and the player who gets ten points first through action is WINNER. In each game, randomly distributed resources will bring a new experience for players,

and procedural content generation has brought great success for this game.

The natural environment is becoming more and more important to the game world. Hand-built virtual worlds are very strict, and once built, they cannot be easily modified. In particular, the modification of some large scenes may make designers have to start over. Procedural content generation (PCG) can perform high quality treatment for specific plant models, animal models, rivers, terrain, and buildings.

Obviously, this technology of PCG can mitigate the burden of content creation. In addition to entertainment [2], PCG can also be used for simulation, training, education and decision-making in other sectors of society, Such as military [3] training peacekeeping missions and simulating tactical decision-making scenarios, rescue troops [4] need to train to rescue trapped people in buildings, the road network generated by PCG can help driving school students learn more conveniently, and PCG can provide the required scenes from all walks of life in society to education and training in schools.

The paper first introduces the definition and classification of PCG, then some existing procedural content generation methods of natural environment such as vegetation, river, and terrain has been summarized. And in the last section, the combination of PCG and other fields in the future is discussed.

## II. DEFINITION AND CLASSIFICATION OF PCG

After understanding the source of procedural content generation, this chapter will detail the concept of procedural content generation. Julian Togelius [5] and others believe that PCG refers to "All aspects of the game that affect gameplay other than nonplayer character (NPC) behavior and the game engine itself", including "terrain, maps, levels, stories, dialogue, quests, characters, rulesets, dynamics, and weapons". Hendrikx [6] and others believe that the idea behind procedural content generation technology is that the game content is not produced manually by human designers, but is generated by a well-defined process executed by computers. It is another choice to make complex game world in a limited time, and will not bring heavy burden to game designers.

Freiknecht and Effelsberg [7] give a more detailed explanation of the concept of PCG: Procedural content generation is to automatically create digital assets of games, simulations or movies based on predefined algorithms and modes, requiring minimal user input.

With the growing game world, PCG technology is now more mature. Pixar Studio has been named the company that uses process content generation in RenderMan, which shows

that the automated content generation has long been accepted by the movie industry.

The following figure shows the objects with the most frequent procedural content generation in the game world. Taking the game world as the background, the procedural content generation technology is used to create the required objects. In the following chapters, this paper will introduce the generation methods of natural objects such as plants, rivers and terrain.

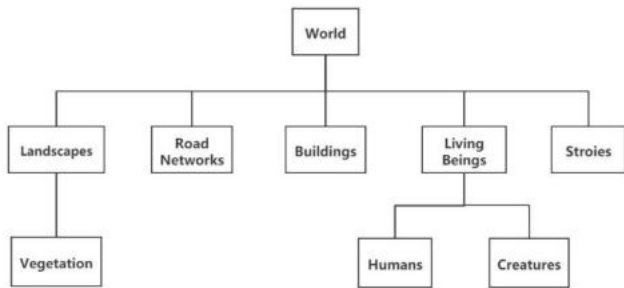


Fig. 1. Type classification of generated content

### III. PCG OF NATURAL OBJECTS

#### A. Plant

Thousands of years of biological evolution have led to a complex ecological environment, which play a crucial role in various species and ecosystems. Object generation in nature, such as vegetation and terrain, has received much attention in the field of procedural content generation, and its development has reached quite mature so far.

##### 1) Fractal of plants

Some of the plants in nature have inevitable self-similarities, and Mandelbort [8] is called fractal. Fractal [9] is a roughly divided geometry, which is divided into several parts of the original geometry, each of which is only different to size of the original whole. Fractal is defined as a geometric branch of describing the geometric patterns contained in nature. Fractal has global determinism and local randomness. Fractal structure [10] has higher stability and fault tolerance than Euclidean geometry with stronger certainty, this is why fractals are so common to nature, from trunks and branches to complex leaf vein structures.

Fractal objects have infinite details, have similar self-structure at different magnification levels, and the details of fractal objects are not directly visible, so they can gradually display after being magnified. This means that the higher the magnification, the more detail you get. Tree [11] has the greatest degree of self-similarity among fractal plants. Although tree is a very complex structure, it is well defined. Fractal geometry can be generated using a variety of methods, the most important and the most mature are L-System. In L-system, fractal can be used to model a two-dimensional tree according to its self-similarity and recursion.

##### 2) L-System

As early as 1969, biologist Aristid Lindenmayer [12] used Lindenmayer (L-system) system to simulate the growth process of complex organisms such as algae and fungi, and later extended it to simulate higher plant species and complex branching systems.

System [13] is a context-free syntax. Each rule generated is only applicable to one symbol in geometry, and other symbols are not affected by the rules, starting from the initial structure. By replacing some parts of the syntax notation to form an object, and iteratively applying some rules for the string symbols to create a branch structure, each recursive iteration will increase the growth level of the string, and finally the string can represent the branch structure of the growing tree.

Define L-system  $G$  as a tuple  $G = (V, \omega, P)$ .

$V$ :  $V$  (alphabet) or letter is a set up to  $V$  and formal symbols.

$\omega$ : The initial state of the system is defined, which is called axiom. Axiom is a string composed of  $V$  symbols, The string set of  $V$  is denoted as  $V^*$ .

$P$ : A set of production rules a symbol, map  $a \in V$  to string,  $\omega \in V^*$  is written as  $P: a \rightarrow \omega$ , Variables can be replaced by a combination of constants and other variables, Predecessor or successor.

Rewriting—The basic idea of L-system. The rewriting rule defines that the left side of the generation can be replaced by the right side, and it can be replaced repeatedly as needed. For example, given two symbols A and B, the results obtained according to the rewriting rules are as follows:

$$a \rightarrow ab,$$

$$b \rightarrow a.$$

This principle was originally used by Chomsky [7] in describing programming languages. However, unlike Chomsky's language, L-Systems requires every rewriting rule to be applied once in each round, on the grounds that plant growth is based on cell division and occurs in parallel for all cells. If 'a' in the above example is used as the initial string, the process can be expressed as shown in Figure 2.

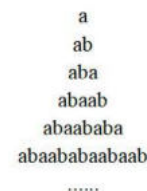


Fig. 2. String tree generated by rewrite rule

##### 3) Visualization of L-system

The recursion of L-system leads to self-similarity, so it is easier to obtain fractal and branching forms. Increasing recursion will lead to the "growth" of models and generate more complex self-similar structures, which can be represented by symbols and graphics. Assuming that the length is  $h$  and the rotation angle is  $\delta$ , then the symbolic commands of L-system used to describe tree visualization in the paper are as follows.

- F: Move one unit forward and draw a line.
- +: Turn right  $\delta$  degrees (clockwise).
- -: Turn left  $\delta$  degrees (counterclockwise).

- [: Save the current position and move according to the next command.
- ]: back to the original position stored by the symbol "[".

The problems with the previous syntax explanation can be described by the steps arranged by the system. Then, with the help of the general algorithm of fractal object interpretation on L-system, the syntax can be interpreted as a graph through the following steps:

- Enter the number of rewriting rules( $n$ ), inclination angle ( $\delta$ ), and segment length ( $h$ ).
- Determine the starting angle  $a_0$ , enter the value of  $a_0$  to get the starting point  $F$ . Then, enter  $F_0$  in the production rule formula  $p$  to obtain  $P_0$ .
- After each iteration, the next angle  $a_n$  and the next point  $F_n$  will be obtained. Then, enter  $F_n$  on  $P_{n-1}$  in the production rule to obtain  $P_n$ .
- Some line segments are obtained from axioms and production rules.

For graphical representation, Use the *turtle* models to reconstruct the tree graph, and the following are the experimental results from several generation rules used in figures 3 to 5.

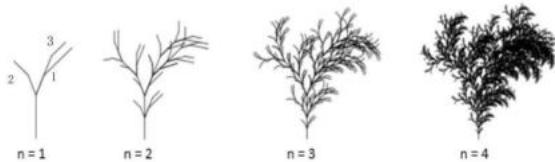


Fig. 3. Fractal tree generated by L-system,  $\delta=25^\circ$ , Axiom  $P: F:\omega: FF[+F+FF][-F-F][+FF+F]$

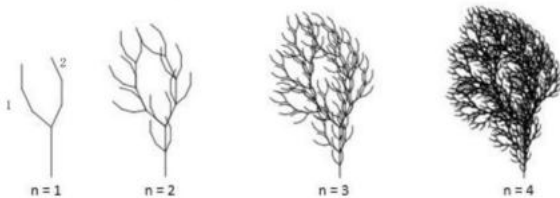


Fig. 4. Fractal tree generated by L-system,  $\delta=25^\circ$ , Axiom  $P: F:\omega: FF[-F-F+FF][+F-F-F]$

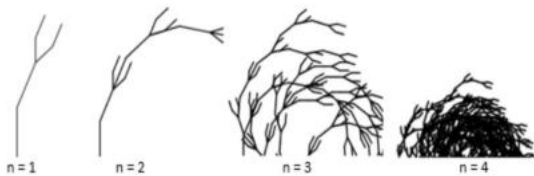


Fig. 5. Fractal tree generated by L-system,  $\delta=25^\circ$ , Axiom  $P: F:\omega: FF+FF[+F-F][-F+F]$

Figures 3 through 5 show the results of the spanning tree model. It can be seen that the tree model is different to different production rules. Figures 3 and 4 are close to the visualization of the tree and are more visually realistic. Figure 5 [11] successfully constructs the syntax, but fails to visualize the tree model.

Applying L-System to 3D plants is also very easy. Just add "tilt left", "tilt right", "tilt forward", "tilt backward" [7] at each decision point, replace the initial "turn left" and "turn right" operations with these operations, these short rules can be used to generate different 3D plants. Similarly, L-system is also suitable for generating shrubs and other types of plants. Up to now, the application of L-system is the most extensive method in plant procedural content generation. The latest research also confirms that the application of L-system has a high maturity.

### B. Rivers

Water always plays a vital role in games. To make the natural environment in the game as detailed as real life, and excellent water resources are indispensable if the natural environment in the game are to be meticulous as in real life. On the contrary, it will make the players who were there instantly to break away.

Although water resources are always the same as a single element, the formation of rivers, lakes, oceans, and waterfalls is very different in many ways. Under the calm sea, there are sometimes more turbulent undercurrents; Sometimes the lake is calm like a mirror. Rivers generally keep flowing, and the creation of rivers [7] is usually carried out in two ways: When the terrain is created; Or place the landscape in a separate step later. Several authors have proposed algorithms specifically for producing rivers:

Kelley et al. [7] was the first person to propose a river network by program. They started from a single river path, formed a river network by recursive subdivision, and then filled the river network with scatter data interpolation function. Next Prusinkiewicz and Hammel [17] put forward a fully automatic method, which combines L-system with topographic erosion, and combines the generation of curved rivers with height map subdivision scheme. In the starting triangle of the river, one edge is marked as the entry and the other edge is marked as the exit. In the subdivision process, triangles are decomposed into smaller triangles. The elevation of the triangle containing the river is set as the sum of all negative displacements on all recursive levels of rivers. While other triangles are treated with mid-point displacements. After 8 or more iterations, the river will be quite natural [15].

In the design and implementation of checkpoints, creating natural phenomena requires changing the height of terrain. Huijser [16] introduced the concept of "cross-section" to express the formation of rivers, and used shape features to overcome the problem of asymmetric cross-section when rivers bend. Shape characteristics describe the local width, curvature, and slope of the shape, and create a richer river landscape according to the shape features of the river.

In order to increase the authenticity of river landscape generated, A. Peytavie and T. Dupont [14] and others put forward a novel program framework to create River Landscape: taking bare-earth as input, deducing river network trajectories affected by water flow, carving riverbeds in terrain, and then automatically generating corresponding blend-flow tree for the water surface. The width, depth, and shape of the riverbed is derived from topography and river type. The water surface is defined by a time-varying continuous function encoded as a blend-flow tree, in which leaves are parameterized procedural flow

primitives. The resulting framework can produce various of river forms, ranging from delaying winding rivers to the torrent of surging currents. These models also include surface effects, such as foam and leaves flowing down the river.

### C. Terrain

Automatic terrain generation is one of the main topics of procedural content generation. It starts from natural phenomena such as plant growth and terrain elevation, and has been extended to the automatic generation of the urban environment. In previous experiments, the terrain is generally represented by height map [18], also known as Digital Terrain Model (DTM), which is a set of approximate elevation levels of a group of discrete points in the grid. The height of these points is the vertical distances between the terrain points and the reference surface. Usually, height map is composed of a gray bitmap, where elevation is represented by the gray shadow of the bitmap pixels. Then, use polygon mesh to visualize in 3D space. The whiter the pixel, the higher the elevation point. Figure 6 shows Mount Everest and other surrounding mountains represented by greyscale [7]. The higher areas are represented by lighter pixels and the lower regions are represented by darker pixels.

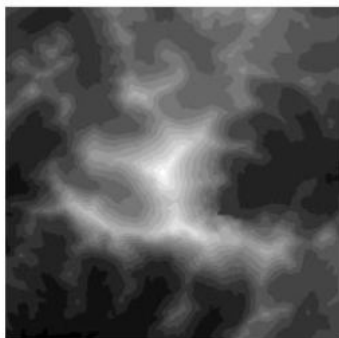


Fig. 6. Height map of mountains obtained in greyscale

The early height maps generation algorithms are based on subdivision methods, which refines the rough elevation map by iteration, and some random parameters are added in each iteration to change the elevation details. The first subdivision algorithm is called the mid-point displacement method: a rough height map is subdivided iteratively, and controllable randomness is used to add details in each iteration. In this method, Miller [18] set the elevation of a new point as the average of a triangle or diamond plus a random offset. According to the roughness of the generated height map, the range of offset at each iteration is reduced. The terrain generated by this method is fractal Brownian motioned (FBM) surface.

The generation of height map is usually based on fractal noise generator. The 2D Perlin noise map created by fractal Brownian Motion [20] is a series of fluctuation data onto smoothness and predictability (see Fig. 7). It generates noise by sampling and interpolating points in a random vector grid, and scales and accumulates several noises with increasing frequency into an elevation map, which is suitable for creating a landscape with mountains and valleys.

Perlin noise algorithm has excellent performance, because each grid point can be calculated independently of the values of neighbouring points, it is very suitable for parallel processing. However, the terrain generated by noise

algorithm is generally uniform without the change of surface details. Therefore, if you want to add detail features of the smooth terrain, you can further to modify the terrain using algorithms based on physical phenomena, such as erosion.



Fig. 7. Output of simple Perlin noise on terrain grid

Musgrave [20] et al. realize the physical erosion process of local or overall erosion characteristics in height field by simply simulating the natural erosion process. The terrain generated by this method has the characteristics of fractal tree and arbitrary local control of cross dimension, which is not available in previous methods. They also proposed a global simulation to simulate what is called thermal weathering, Hydraulic erosion forms valleys and drainage networks, thermal weathering wears steep slopes and forms pluvial rocks at their feet. Compared with hydraulic erosion simulation, thermal weathering simulation can get more real results from a shorter time.

Although these erosion algorithms greatly improve the authenticity of mountain terrain, they need to run hundreds to thousands of iterations. Another method based on natural phenomena is Voronoi Tessellation [19] method, which usually occurs to the valley where the mountains meet, that is, the collection of all points in the Voronoi Tessellation region closest to the center of the region. By occupying the terrain mesh, the author raises up the mountains and the surrounding terrain at the same time, so each of the adjacent vertices is lifted. From the top view, the Voronoi Tessellation can be finally obtained. This method greatly saves the time of generating mountain terrain by programming content.

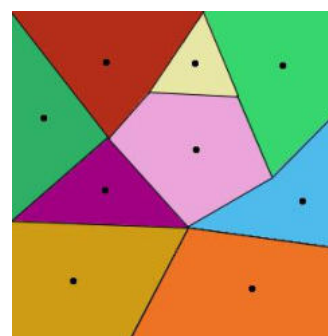


Fig. 8. Ordinary plane Voronoi diagram

## IV. CONCLUSION

Procedural content generation is becoming more and more comfortable in creating the natural objects and some complex aspects of the game. However, these technologies need intuitive parameter control, powerful editing, and result visualization, for real-time operation. This paper introduces the procedural content generation methods and examples of plants, rivers, and terrain in games.

According to the statistics in [21], the algorithm families generated by application in the papers on PCG in the past ten years are shown in Figure 9. The overall proportion of each algorithm is relatively low, but the selection of algorithms is usually related to the programmer's personal preferences. Grammar has faced a downward trend since it became popular in early 2010-2013. Declarative and constructive methods also seem to be used smoothly in this decade. In addition, artificial intelligence (AI) is rising with a steady trend, which has become a popular way in game development.

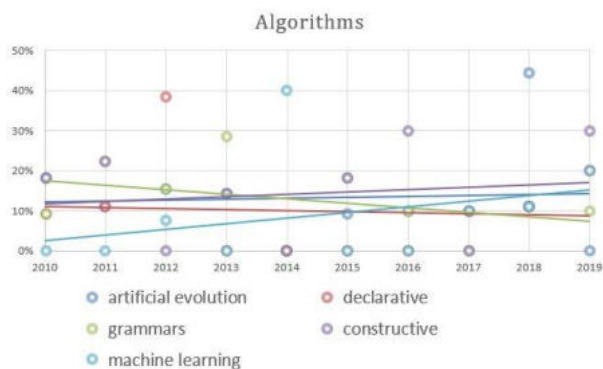


Fig. 9. Algorithm families applied in PCG papers in the past decade

Nowadays, AI has become an interdisciplinary field involving computer science, medicine, agronomy, mathematics, philosophy, and other disciplines. In terms of algorithm, AI can also be combined with PCG. Through machine learning, the concept of PCG is more extensive, which promotes the research of PCG seminars. Therefore, it is predicted that the combination of PCG and machine learning will continue to show an upward trend. In the era of abstract visualization of network traffic, PCG is a very important link in game design. PCG is able to stimulate designers' creativity. Designers can make full use of PCG to show us a new world they have never seen before.

#### ACKNOWLEDGMENTS

This paper is supported by the Fundamental Research Funds for the Central Universities under Grant Number: N2017003.

#### REFERENCES

[1] M. Wang, "Java Settlers Intelligente agentenbasierte Spielsysteme für intuitive Multi-Touch-Umgebungen," Ph.D. dissertation, Dept. Comput. Eng., Free Univ., Berlin, Germany, 2008.

[2] G. N. Yannakakis, and J. Togelius, "Experience-driven procedural content generation," *IEEE Tans On Affect Comput*, vol. 2, no. 3, pp. 147-161, 2011.

[3] K. Stanley, B. Bryant, and R. Miikkulainen. (2005, April). Real-Time Evolution in the NERO Video Game. [Online]. Available: <https://ac.scmor.com/>

[4] D. D. Djordjevich et al., "Preparing for the aftermath: Using emotional agents in game-based training for disaster response," *IEEE Sym On Comput Intell and Games*. 2008, pp. 266-275.

[5] J. Togelius, G. N. Yannakakis, et al., "Search-based procedural content generation: A taxonomy and survey," *IEEE Trans On Comput Intell and AI in Games*, vol. 3, no. 3, pp.172-186, 2011.

[6] M. Hendriks, S. Meijer, et al., "Procedural content generation for games: A survey," *TOMM*, vol. 9, no.1, pp. 1-22, 2013.

[7] J. Freiknecht, and W. Effelsberg, "A survey on the procedural generation of virtual worlds," *Multimodal Technol and Interact*, vol. 4, no. 4, 2017.

[8] B. B. Mandelbrot, "The fractal geometry of nature," in *WH freeman*, vol. 1, New York, 1982.

[9] M. F. Barnsle, "Fractals everywhere," Academic press, 2014.

[10] P. H. Carr, "Does God play dice? Insights from the fractal geometry of nature," *Zygon®*, vol. 39, no. 4, pp. 933-940, 2004.

[11] E. W. Hidayat, I. Putra, A. D. Giriantari, et al., "Visualization of a two-dimensional tree modeling using fractal based on L-system," in *IOP Conf Series: Mater Sci and Eng*, 2019.

[12] G. Ochoa, "An introduction to lindenmayer systems," 1998. [Online]. Available: <http://www.cogs.susx.ac.uk/users/gabro/lsys/lsys.html>.

[13] M. H. Tanveer, A. Thomas, X. Wu, et al., "Simulate forest trees by integrating l-system and 3d cad files," *IEEE 3rd ICICT*, pp. 91-95, 2020.

[14] A. Peytavie, T. Dupont, E. Guérin, et al., "Procedural Riverscapes," *Comput Graph Forum*, vol. 38, no. 7, pp. 35-46, 2019.

[15] R. M. Smelik, T. Tutenel, R. Bidarra, et al., "A Survey on Procedural Modelling for Virtual Worlds," *Comput Graph Forum*, vol. 33, no. 6, pp. 31-50, 2014.

[16] R. Huijser, J. Dobbe, W. F. Bronsvort, et al., "Procedural Natural Systems for Game Level Design," *IEEE Brazilian Symposium on Games and Digital Entertainment*, pp. 189-198, 2010.

[17] P. Prusinkiewicz, M. Hammel, "A fractal model of mountains with rivers," *InProceedings of Graphics Interface* , vol. 93, no. 4, pp. 174-180, 1993.

[18] R. M. Smelik et al., "A Survey of Procedural Methods for Terrain Modelling," *3AMIGAS*, vol. 2009, pp. 25-34, June, 2009.

[19] K. S. Emmanuel, C. Mathuram, A.R. Priyadarshi, et al., "A Beginners Guide to Procedural Terrain Modelling Techniques," *IEEE 2nd ICSPC*, pp. 212-217, March, 2019.

[20] F. K. Musgrave, C. E. Kolb, R.S. Mace, "The synthesis and rendering of eroded fractal terrains," *ACM Siggraph Comput Graph*, vol. 23, no. 3, pp. 41-50, 1989.

[21] A. Liapis, "10 Years of the PCG workshop: Past and Future Trends," in *International Conference on the Foundations of Digital Games*, pp.1-10, 2020.



# Reinforcement Learning for Neural Collaborative Filtering

Alexandros I. Metsai  
My Company Projects O.E.  
Thessaloniki, Greece  
alexandros.metsai@mycompany.com.gr

Konstantinos Karamitsios  
My Company Projects O.E.  
Thessaloniki, Greece  
kk@mycompany.com.gr

Konstantinos Kotrotsios  
My Company Projects O.E.  
Thessaloniki, Greece  
kotrotsios@mycompany.com.gr

Periklis Chatzimisios  
International Hellenic University  
Thessaloniki, Greece  
pchatzimisios@ihu.gr

George Stalidis  
International Hellenic University  
Thessaloniki, Greece  
stalidgi@ihu.gr

Kostas Goulianas  
International Hellenic University  
Thessaloniki, Greece  
gouliana@ihu.gr

**Abstract**—Artificial Intelligence (AI) has become an integral part of many modern technologies, with significant advances in real-world applications. With the rise of deep learning methods in the past decade, systems that utilize artificial neural networks have produced remarkable results in a variety of fields, such as computer vision, natural language processing and voice recognition, with performance exceeding that of humans in many cases. Examples of practical applications include self-driving cars, state-of-the-art text translators and generators, and robust object detection algorithms. The field of Recommender Systems has also taken advantage of this progress, with a plethora of novel neural networks being proposed that achieve significant improvements in providing automatic recommendations regarding the preferences of users. Aiming to further explore this area and the capabilities of different deep neural networks, we train top-performing neural collaborative filtering recommender systems under a reinforcement learning setting, which has been largely unexplored in favor of supervised learning for these models. Experimental evaluation on the MovieLens-1m dataset showcases the behavior of different neural architectures under this setting, and how the introduction of sophisticated components contributes to improved performance.

**Keywords**—artificial intelligence, deep learning, reinforcement learning, recommender systems

## I. INTRODUCTION

Over the last decade, Artificial Intelligence (AI) has witnessed a great rise in popularity and adoption, with various modern architectures being characterized as state-of-the-art and top-performing solutions in many different domains. Specifically, a plethora of systems based on deep neural networks (or “deep learning” in the relevant literature) have been applied outside of basic research, in real-world industrial settings, with outstanding results [20]. Examples include self-driving cars, significant improvements in computer vision related fields and major advances in natural language processing. A notable field that has made extensive use of these advances in AI is the field of Recommender Systems [16]. These systems concern the automatic generation of recommendations that satisfy a user’s preferences by utilizing various factors, such as their item selection history, ratings, demographic information, seasonal conditions, etc. [2].

A major category of algorithms for implementing Recommender Systems are Collaborative Filtering

---

This work has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship, and Innovation, under the call RESEARCH – CREATE – INNOVATE (project code: T2EDK-03843).

techniques. In essence, these are methods for performing predictions regarding the preferences of a single user by taking into account the information regarding the preferences of many users [20]. These methods usually require large amounts

of data; however, this restriction is usually satisfied for this particular domain, since e-commerce and related online services have access to an abundance of user-item interaction data. As an example, the volume of interactions that platforms such as Spotify and Netflix record is enormous, with the same being true for online retail platforms such as Amazon and eBay. Influenced by the rise of deep learning described above, many implementations of collaborative filtering methods that utilize neural networks were introduced in recent years, with promising results [4, 6, 7, 8].

Regarding the training or learning process of a neural network (and of a machine learning model in general), the three basic learning paradigms are supervised learning, unsupervised learning, and reinforcement learning [5]. Most of the algorithms in the collaborative filtering for recommender systems domain utilize supervised learning, with the unsupervised learning setting being less practical in this scenario, while the reinforcement learning setting has not been extensively explored. Motivated by this, we aim to evaluate the performance of modern neural collaborative filtering algorithms under a reinforcement learning setting.

## II. RELATED WORK

In both research and industry settings, many Recommender Systems have been proposed, with the first attempts dating to the late 1970s [19]. The first truly mature methods began to emerge in the mid-1990s, with the systems GroupLens [18], Video Recommender [9] and Ringo [21] providing remarkable solutions in automatic recommendations. The GroupLens system started as a recommender for relevant articles in the Usenet [11] platform, by taking into consideration a user’s previous ratings. Similarly, the Video Recommender system was tasked with selecting the most relevant videos from a larger set. Finally, the Ringo system’s goal was the personalized recommendations of relevant music artists and records.

The above contributions led to the first commercial recommender systems by the end of the decade. One of the first and most significant examples is the recommender system integrated in Amazon’s platform [14]. This platform provided recommendations in the form of lists “also seen by other users”. In the following years, even more commercial platforms followed this example, and by the middle of the next decade the field of recommender systems had become an area

highly active in both research and adoption by the industry. The Netflix Prize [1], organized in 2006 and offering 1 million dollars to the best collaborative filtering algorithm, is another example of this high activity, which continued in the following years.

The field of recommender systems has utilized a broad spectrum of machine learning techniques for the implementation of relevant algorithms. These include classification techniques and clustering, as well as methods for dimensionality reduction. Algorithms for matrix factorization [12] and factorization machines [17], which improve upon the former, were some of the most prominent solutions that first gained widespread attention after their exploitation during the Netflix prize.

During the past decade, AI algorithms based on deep neural networks have witnessed a great surge in popularity and adoption, with many applications and improvements in different domains. Influenced by these advances, many works in recommender systems adopted deep learning techniques for implementing collaborative filtering methods and factorization machines. In 2016, the authors of [4] introduced an influential neural network called “Wide and Deep”. As the title suggests, this model utilizes a linear model (wide component) combined with a deep network for modeling both high and lower-level relations. The deep component is constituted by three fully connected layers with ReLU activations, while the wide component is described by a cross-product transformation. The authors reported that this architecture was successfully put in production and evaluated on Google Play, a commercial application distribution service with more than a billion active users.

Motivated by the idea of combining a shallow and a deep component with the aim of modeling high and lower-level user and item relationships, similar works that improved upon the “Wide and Deep” network architecture emerged. In 2017, the authors of [6] proposed a variation of the model, that allows for a larger degree of flexibility and efficiency regarding the overall architecture. These improvements include the replacement of the wide model with a Factorization Machine and the introduction of an embedding vector for representing the input, that constitutes the shared input to both of the wide and deep components. Shortly after this work, in 2018, a model that further improves upon the latter was introduced [13]. This architecture, called xDeepFM, utilized a linear model combined with a Compressed Interaction Network (CIN) and a deep neural network. A different approach was taken by the authors of [7] that attempted to construct an improved factorization machine model with a neural network architecture, called Neural Factorization Machine.

### III. PROPOSED APPROACH

Aiming to investigate the capabilities of different deep learning networks for utilizing recommender systems, we attempt to train the most prominent networks under a reinforcement learning setting. Though supervised learning has been the norm for training these architectures, reinforcement learning can be more suitable for modeling real-world applications, by utilizing the network as an agent that is tasked with performing actions in an environment defined by the user and the possible items that can be recommended.

#### A. Factorization Machines

Matrix Factorization techniques have been an important contribution to the field of recommender systems and have gained a significant adoption [10]. However, due to the way they function, which is by deconstructing a user-item interaction matrix to two matrices of lower dimensionality, the product of which approximates the original, they are limited to modeling lower-order relationships. Since these relationships may be better described by considering higher-order relationships too, there arises the need for a more descriptive modelling method.

Factorization Machines, first described in 2010 [17], have been widely adopted by the industry and influenced the relevant research. These models map any input features to vectors of lower dimensions and are able to estimate parameters using very sparse data, thus being able to scale into larger datasets. Equation (1) describes the output of a Factorization Machine:

$$\hat{y}_{FM}(x) = w_0 + \sum_{i=1}^n x_i w_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^n \langle v_i, v_j \rangle x_i x_j \quad (1)$$

Where the input vector is  $x \in R^n$ , with the value  $x_i = 0$  signifying that the  $i$ th feature is not present in the current input, and  $w_0, w_i, \langle v_i, v_j \rangle$  are values that must be configured during the training/fitting processes. The first term represents the global bias, the second term represents the weights of the  $i$ th variable, and the third term represents the dot product of the  $i$ th and  $j$ th elements of the user-item interaction matrix.

#### B. Neural Network Architectures

As mentioned previously, DeepFM [6] is a neural network architecture that extends the core idea presented by [4], that is combining a wide component with a deep component, by replacing the wide part of the architecture, which was previously a linear model, with a Factorization Machine. This allows for modelling both first and second order relationships, while at the same time being more robust to sparse data, as is the case with most user-item interaction matrices. Moreover, the authors introduced an embedding layer for representing the input and used this new representation as the shared input to the deep and wide components. The model thus allows for end-to-end training of the network instead of the manual feature engineering by human experts that the original method required.

Authors of [13] further extended the above work by introducing xDeepFM (eXtreme Deep Factorization Machine). This architecture utilized a linear model coupled with a deep model, as proposed in [4], and adopted the embedding layer introduced in DeepFM. However, they introduced an additional component, the Compressed Interaction Network (CIN). This component’s properties include user-item interactions being applied at a vector-wise level, measuring high-order feature interactions explicitly, and its complexity increasing in a non-exponential manner as the dimensionality of interactions increases.

In a similar manner to the above works, [7] proposed a novel architecture, called Neural Factorization Machine (NFM), that combines the functionality of a Factorization Machine with the abilities for modeling nonlinear higher order relationships that characterize a deep neural network. By definition, a NFM is more flexible than a Factorization

Machine since the latter can be considered as a special case of the former. Given a sparse vector  $x \in R^n$  as input, with the value  $x_i = 0$  signifying that the  $i$ th feature is not present in the current input, a NFM calculates it input as:

$$\hat{y}_{NFM}(x) = w_0 + \sum_{i=1}^n x_i w_i + f(x) \quad (2)$$

Where the first two terms are known from equation (1), that describes a Factorization Machine, while the third term,  $f(x)$ , is a deep neural network, the core component that this architecture introduces. Finally, as was the case with both neural network architectures described previously, the input is passed through an embedding layer in order to reduce dimensions and sparsity.

### C. Reinforcement Learning

Reinforcement learning is a machine learning paradigm that describes the process where an intelligent agent performs actions in an environment, with the aim of maximizing some reward. Fields of application include robotics, autonomous driving, natural language processing, etc. [20]. For the needs of this work, we define the environment as a user and a list of items on which the agent (neural network model) will perform actions, which correspond to recommending some of the available items. The agent will be rewarded for each item that would be selected by the user, and therefore will aim to maximize its total reward during the training process.

In more detail, for a given user and a list of items, for which the user’s preference is known, the agent iterates through the list, recommending some items to the user and discarding others. After the whole list of items has been processed, the agent will have produced a subset corresponding to the model’s recommendations, with each selection or rejection being treated as a separate action. Actions that led to a correct recommendation, that is the selection of a positively labeled item and the rejection of a negatively labeled item, yield a positive reward, while incorrect actions yield zero reward.

## IV. EXPERIMENTS

### A. Dataset

The MovieLens-1m [15] is an established benchmarking dataset in the Recommender Systems literature. It is provided by the GroupLens research lab of the university of Minnesota, and its goal is the facilitation and testing of relevant algorithms. This dataset contains 1 million anonymized ratings for 4,000 movies from 6,000 unique users, as well as additional information concerning the users and the movies (user gender, age, movie title, description, etc.). In correspondence with our research goals, we only utilize the information regarding the ratings (ranging from 1 to 5), user IDs and item IDs. Ratings with a score lower or equal to 3 are considered as negative samples, while ratings with a larger score are considered as positive samples.

### B. Evaluation Approach

For evaluating the performance of each architecture, we utilize the Precision, Recall and AUC, which are common metrics in the field of machine learning [3]. Precision concerns the percentage of relevant samples among the retrieved samples, i.e., the percentage of samples that belong in the positive class from the total samples classified as positive. Recall describes the percentage of positive samples

retrieved, i.e., the percentage of positive samples retrieved from the total positive samples in the dataset. The Receiver Operator Characteristic (ROC) is a curve that plots the true positive rate against the false positive rate in various threshold values. Resulting from the ROC curve, the Area Under the Curve (AUC) metric describes an algorithm’s ability to separate in between two classes, with values higher than 50% signifying an increasingly better performance in distinguishing between positive and negative samples.

### C. Implementation Details

We proceed to describe our implementation details. We follow the typical evaluation setting of splitting the dataset into two non-overlapping sets, keeping 80% of the data for training and 20% for validation and testing of the models. For fair comparison, the learning rate is set to  $10^{-3}$  for all models using the Adam optimizer, the maximum number of epochs is set to 100, while we also apply early stopping when performance ceases to improve in the validation set.

The neural network part of DeepFM consists of 2 hidden layers, each with 16 neurons and ReLU activations, and an output layer with a sigmoid activation function, with the same applying for the xDeepFM model. The Neural Factorization Machine model consists of an embedding layer followed by a Bi-Interaction pooling layer and two hidden layers, with the output being extracted by a single neuron with linear activation. All of the presented deep neural networks are implemented using the PyTorch framework and are trained in an end-to-end manner.

### D. Performance Evaluation

In table 1, we present the results of the comparative evaluation of the three deep neural models, along with the results yielded by random selection. We observe that the DeepFM and xDeepFM models exhibit significantly better results than those of the Neural Factorization Machine model. This can be attributed to the more sophisticated structure of the first two architectures since they consist of both wide and deep components. The xDeepFM model yields generally better results than DeepFM, especially for the Recall metric, which further strengthens the significance of the addition of the CIN network. This is also a reasonable outcome, since xDeepFM builds and improves upon the DeepFM architecture, as described previously. We conclude that xDeepFM is the best performing architecture for this reinforcement learning setting, with DeepFM being a close second. It is therefore recommended to still test the performance of both methods if another dataset is evaluated in future work. Finally, we note that all methods are significantly better than the performance of random selection.

TABLE I. PERFORMANCE COMPARISON ON THE MOVIELENS-1M DATASET [15].

<i>Model</i>	<i>Precision</i>	<i>Recall</i>	<i>AUC</i>
DeepFM [6]	<b>75.01 %</b>	65.27 %	71.10 %
xDeepFM [13]	73.50 %	<b>73.30 %</b>	<b>72.10 %</b>
NFM [7]	69.98 %	57.71 %	63.27 %
Random	57.19 %	49.96 %	49.72 %

## V. CONCLUSIONS

In this work, we compared three state-of-the art collaborative filtering recommender system architectures, based on deep neural networks, under a reinforcement

learning setting. The popular MovieLens-1m dataset was utilized to benchmark the selected models using common metrics for evaluating the performance machine learning algorithms. Our experiments showcased that the training of all the networks converges successfully under this setting, yielding significantly better performance than random selection, with the xDeepFM architecture exhibiting the best results. Future work could utilize additional datasets, suitable for further evaluating the performance and overall properties of the recommender system algorithms. Moreover, the whole learning process could benefit from the introduction of a trainable critic for facilitating a more robust evaluation of the actor's actions, which for our case are the system's recommendations, in an actor-critic reinforcement learning setting.

#### ACKNOWLEDGMENT

This work has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship, and Innovation, under the call RESEARCH – CREATE – INNOVATE (project code: T2EDK-03843).

#### REFERENCES

- [1] J. Bennett and S. Lanning, "The Netflix Prize," in Proceedings of KDD Cup and Workshop, 2007.
- [2] S. Blanda, "Online Recommender Systems – How Does a Website Know What I Want?" American Mathematical Society, 2015.
- [3] C. D. Brown and H. T. Davis, "Receiver operating characteristics curves and related decision measures: A tutorial," *Chemometrics and Intelligent Laboratory Systems*, Volume 80, Issue 1, 2006, Pages 24-38. ISSN 0169-7439. <https://doi.org/10.1016/j.chemolab.2005.05.004>
- [4] [3] H. T. Cheng, et al., "Wide & Deep Learning for Recommender Systems," in Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, 2016, DOI:<https://doi.org/10.1145/2988450.2988454>
- [5] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*, MIT Press, 2016.
- [6] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "DeepFM: a factorization-machine based neural network for CTR prediction," in Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17), AAAI Press, 2017, 1725–1731.
- [7] X. He and T. S. Chua, "Neural Factorization Machines for Sparse Predictive Analytics," in Proceedings of SIGIR '17, Shinjuku, Tokyo, Japan, 2017.
- [8] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. S. Chua, "Neural Collaborative Filtering," in Proceedings of the 26th International Conference on World Wide Web (WWW '17), International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2017, 173–182. DOI:<https://doi.org/10.1145/3038912.3052569>
- [9] W. Hill, L. Stead, M. Rosenstein, and G. Furnas, "Recommending and evaluating choices in a virtual community of use," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '95). ACM Press/Addison-Wesley Publishing Co., USA, 1995, 194–201. DOI:<https://doi.org/10.1145/223904.223929>
- [10] KismetK. Netflix: Recommendations Worth a Million. Retrieved March 28, 2021 from [https://studio-pubs-static.s3.amazonaws.com/190289\\_5089121940cc4f74b7e87c963f6e3b65.html](https://studio-pubs-static.s3.amazonaws.com/190289_5089121940cc4f74b7e87c963f6e3b65.html)
- [11] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl, GroupLens: applying collaborative filtering to Usenet news. *Commun. ACM*, 40, 3 (March 1997), 77–87. DOI:<https://doi.org/10.1145/245108.245126>
- [12] Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," *Computer*, vol. 42, no. 8, Aug. 2009. doi: 10.1109/MC.2009.263.
- [13] J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie, and G. Sun, "XDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems," in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18), Association for Computing Machinery, New York, NY, USA, 2018, 1754–1763. DOI:<https://doi.org/10.1145/3219819.3220023>
- [14] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," in *IEEE Internet Computing*, vol. 7, no. 1, 2004, pp. 76-80.
- [15] F. M. Harper and J. A. Konstan, "The MovieLens Datasets: History and Context," in *ACM Trans. Interact. Intell. Syst.* 5, 4, Article 19 (January 2016), 2015, 19 pages. DOI:<https://doi.org/10.1145/2827872>
- [16] A. I. Metsai, I. Tabakis, K. Karamitsios, K. Kotrotsios, P. Chatzimisios, G. Stalidis, and K. Goulianas, "Customer Journey: Applications of AI & Machine Learning in E-Commerce," in Proceedings of the International Conference on Interactive Mobile and Communication Technologies and Learning (IMCL), 2021.
- [17] S. Rendle, "Factorization Machines," in Proceedings of the IEEE International Conference on Data Mining, 2010. DOI:<https://doi.org/10.1109/ICDM.2010.127>
- [18] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: an open architecture for collaborative filtering of netnews," in Proceedings of the 1994 ACM conference on Computer supported cooperative work (CSCW '94). Association for Computing Machinery, New York, NY, USA, 1994, 175–186. DOI:<https://doi.org/10.1145/192844.192905>
- [19] E. Rich, "User modeling via stereotypes," in *Cognitive Science*, vol. 3, no. 4, 1979, p. 329–354.
- [20] T. J. Sejnowski, *The Deep Learning Revolution*. MIT Press, 2018.
- [21] U. Shardanand and P. Maes, "Social information filtering: algorithms for automating word of mouth," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '95). ACM Press/Addison-Wesley Publishing Co., USA, 1995, 210–217. DOI:<https://doi.org/10.1145/223904.223931>

# A Survey of Markov Model in Reinforcement Learning

Tianhan Gao  
Software College  
Northeastern University  
Shenyang, China  
gaoth@mail.neu.edu.cn

Baicheng Chen  
Software College  
Northeastern University  
Shenyang, China  
2071264@stu.neu.edu.cn

Qingwei Mi  
Software College  
Northeastern University  
Shenyang, China  
2110491@stu.neu.edu.cn

**Abstract**—There was a famous mathematician from Russian in the early 20<sup>th</sup> century whose name is Andrey Andreyevich Markov. After he proposed the Markov process, the theories have been effectively developed in the past years and have been widely adopted in different fields. This paper both summarizes and does incomplete statistics on Markov models which are related to reinforcement learning. It will help the readers to quickly clarify the relationship between these theories and have an overall understanding of them.

**Keywords**—Markov model, Reinforcement Learning, Markov chain, Markov process

## I. INTRODUCTION

The theories about the Markov process will be called uniformly Markovian theories in this paper for convenience. Although there are still a lot of theories except Markovian theories in the field of reinforcement learning, Markovian theories can be always considered as the basis for the implementation of most of the algorithms in reinforcement learning. It will be better to know what the Markov model is before move on to other theories, and after that, it will be much easier to understand for example the Markov chain, process, decision process, etc. Among them, the Markov decision-making process (MDP) can be considered as the basis theory of reinforcement learning. Besides the theories about the Markov process, many variants have been proposed based on them, and people have come up with a lot of new ideas for the use of them. This paper presents incomplete statistics on the Markovian theories especially in the field of reinforcement learning, which mainly summarizes the variation and usage ideas of kinds of Markovian theories, so that the reader can both quickly understand the structure of the theories and reinforcement learning in other fields, and have a view of reinforcement learning from the perspective of Markovian theories.

In the early 20th century, Andrey Andreyevich Markov, who is a mathematician from Russian, worked on the Markov process[1] and published a paper on this subject in 1906[2]. Before his work, the Poisson process had been discovered, and it can be considered as a kind of Markov process which is continuous in time.

## II. MARKOV PROPERTY

Because Markovian theories contain a lot of knowledge, learning them in a certain order will lead to better learning results. While the Markov model, process, and chain make up the bulk of Markovian theories, Markov property should be considered the most valuable to learn at the beginning. Because almost any theory that can be called a Markov theory should have or fit the Markov property.

### A. Introduction

In probability theory and statistics, stochastic processes can be considered to have memorability. This kind of property was later summarized as the Markov Property named after Andrey Markov. It is a kind of property that can be used to define a special environment and the environment's state signal. In the field of reinforcement learning, ideally, researchers who want to predict the future will need the past state signal that retains all the information which can summarize the past. If a state signal successfully retains all the information which can summarize the past, then this signal is Markovian or has Markov Property.

### B. Definition

The number of states and reward values assumed should not be infinite so that the problem can be calculated based on 'sum' of different kinds of data and "probabilities" rather than "integral" and the "probability density", otherwise researchers will have to worry about the assumption where the number of states and reward values are unlimited, as the argument can be difficult to be extended to include continuous states and rewards.

Assuming the action is taken at time  $t$ , then what the environment will react at time  $t+1$  will be the response. In the most general causal case, this response depends on what happened before. In this case, the dynamic [4] of the environment can be defined by specifying the full joint probability distribution:

$$Pr\{S_{t+1}=s', R_{t+1}=r|S_0, A_0, R_1, \dots, S_{t-1}, A_{t-1}, R_t, S_t, A_t\} \quad (1)$$

for all  $r, s'$ , and all possible values of the past events:  $S_0, A_0, R_1, \dots, S_{t-1}, A_{t-1}, R_t, S_t, A_t$ . On the other hand, if the state signal has Markov Property, then the response of the environment at  $t+1$  depends only on the state and action representations at  $t$ , in which case the dynamics of the environment should be defined as:

$$p(s', r|s, a) \triangleq Pr\{S_{t+1}=s', R_{t+1}=r|S_t=s, A_t=a\} \quad (2)$$

for all  $r, s', s$ , and  $a$ .

### C. Strong Markov property

Strong Markov property is similar to Markov property. The most important difference between them is that the Strong Markov property contains a stopping time, which is a specific type of "random time". This kind of "random time" is always defined by a stopping rule which is a mechanism for deciding whether the process should be stopped or not. The rule is usually based on current or past state. And stopping time cannot be infinite in general.

The Markov property means that it is sufficient to predict the future from the current state because the current state is the result of all previous states. In another word, it is not necessary to collect all the past states' information, just use the current state's information.

The Markov property understands time in the dimension of time, but the Strong Markov attribute understands time in the perspective of regular logic.

### III. MARKOV MODELS IN REINFORCEMENT LEARNING

In probability theory, the Markov model is a stochastic model used to simulate pseudo-randomly changing systems. It assumes that the future state depends only on the current state and not on the events that occurred before it (The Markov Property). In general, this assumption makes the model accessible for inference and computation that it would otherwise be difficult to handle. Thus, in the field of prediction models and probabilistic prediction, it is desirable to assume that a given model exhibits the Markov Property [5].

#### A. The relationship between Markov model and Markov process

When the system state is both automatic and completely visible, the Markov model can be called a Markov chain. And the Markov process is a continuous-time version of the Markov chain.

#### B. Classification of the Markov models

There are four common Markov models in different scenarios, depending on whether each sequence state is observable, and whether the system is to be adjusted for the observations:

- Markov chain.
- Hidden Markov models.
- Markov decision process.
- Partially observable Markov decision process.

TABLE I. CLASSIFICATION OF THE MARKOV MODELS

	<i>System state is completely visible</i>	<i>System state is partially observable</i>
System is autonomous	Markov chain	Hidden Markov model
System is controlled	Markov decision process	Partially observable Markov decision process

#### 1) Markov chain

a) *Definition:* The Markov property can be called "memorylessness" sometimes because it means that in a stochastic process that satisfies it, the predictions of the process can be made based solely on the process's present state and the predictions are as good as the one that could be made knowing all the process's history. And this kind of process can be called the Markov process. There are different definitions of a Markov chain. The most common is that a Markov chain is a Markov process having discrete-time in either countable or continuous state space. But some definitions are regardless of the nature of time and say that a Markov chain is a Markov process with a countable state space. For example, if you ignore time, you can define a Markov chain as a Markov process in a countable state space, and a Markov chain as a Markov process in discrete time if you ignore a state space.

b) *Types of the Markov chain:* Depending on different kinds of state spaces and discrete-time v. continuous time, there will be four kinds of Markov chains:

- (Discrete-time) Markov chains on a countable or finite-state space.
- Markov chains on a measurable state space (e. g., Harris chains).
- Continuous time Markov process or a Markov jump process.
- Any continuous stochastic process with the Markov property (for example, the Wiener process).

TABLE II. TYPES OF MARKOV CHAINS

	<i>Countable state space</i>	<i>Continuous or general state space</i>
Discrete-time	(discrete-time) Markov chain on a countable or finite state space	Markov chain on a measurable state space (for example, Harris chain)
Continuous-time	Continuous-time Markov process or Markov jump process	Any continuous stochastic process with the Markov property (for example, the Wiener process)

The Markov process is a stochastic process where, given the present case, the future is independent of the past. Sometimes, the Markov process is also known as a version of the Markov chain with continuous time [6]. A simple representation of the Markov process is shown below:

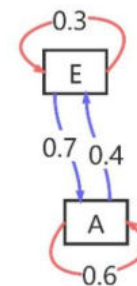


Fig. 1. Example of a Markov process with two states marked E and A. And each number represents the probability that the Markov process changes from one state to another, with the direction indicated by the arrow.

#### 2) Hidden Markov model

The Hidden Markov model (HMM) is a statistical model that was first proposed by Baum L.E. He uses a Markov process that contains hidden and unknown parameters. In this model, the observed parameters are used to identify the hidden parameters. Its state cannot be directly observed but can be identified by observing the vector series [7]. The hidden Markov models are probabilistic frameworks where the observed data are modeled as a series of outputs generated by one of several internal states [8].

In general, when considering a Markov model, all its processes should be observable. However, an HMM is a doubly stochastic process with an underlying stochastic process that is not observable but can only be observed through another set of stochastic processes that produce the sequence of observed symbols [9].

If the state of a Markov model is only partially observable or noisily observable, this Markov model can be called a hidden Markov Model.

#### 3) Markov decision process

a) *Definition*: Markov decision process (MDP) is also a kind of Markov chain. In general, the one that tries to control the system will be called the agent. And there may be many agents in one system. The MDP state transitions are depending on the current state and the actions the agents take. Typically, an MDP is always being used to compute a policy of actions that will maximize some utility concerning expected rewards. And that is why nearly all the reinforcement learning algorithms are based on MDP. Sometimes, a MDP will be defined by the tuple whose number of parameters is larger than four for the computation of reinforcement learning (RL) because RL needs parameters to representing the learning rate, discount factor and so on. In general, A MDP will be represented by a four-parameter tuple  $\{S, A, P_a, R_a\}$ . S is a finite set of states, called the state space. A is a finite set of actions, called behavioral space, while as are a series of actions that can be performed under state S.

$$P_a(s, s') = Pr(s_{t+1} = s' | s_t = s, a_t = a) \quad (3)$$

is the probability that taking behavior a at t in state s leads to state s' at time t+1.  $R_a(s, s')$  is the instantaneous reward (or the expected instantaneous reward) when the state s turns to s' due to behavior a. State and action space can be finite or infinite. Some processes with countable infinite states and action space can be reduced to processes with finite state and action space [10].

#### b) Reinforcement Learning

Deep reinforcement learning based on Markovian theories first gained widespread attention in 2013, when Google's Deepmind team first implemented image-based reinforcement learning AI [11]. The algorithm they used was Deep Q-learning (DQN).

Deep Q-learning is a method that tries to get the best strategy by using Q-learning and deep neural networks. Q-learning is a kind of model-free algorithm that tries to get the best policy by calculating the value of each action in each particular state.

After the DQN, algorithms related such as DDPG, AC, A3C, NAF, TRPO, PPO, TD3, SAC were invented in the single-agent field based on the Markovian theories. In the field of multi-agents, algorithms for example IQL, VDN, COMA, QMIX, QTRAN, QTRAN++, Qatten have also been invented.

The DQN cannot be straightforwardly applied to continuous domains since it relies on finding the action that maximizes the action-value function, which in the continuous-valued case requires an iterative optimization process at every step. Based on the deterministic policy gradient (DPG) [12] algorithm, countzero et al. present a model-free, off-policy actor-critic algorithm using deep function approximators that can learn policies in high-dimensional, continuous action spaces which is called the DDPG [13].

And there are some algorithms called Actor-Critic algorithms that can combine the strong points of actor-only and critic-only methods [14].

The Advantage Actor-Critic (A2C) algorithm replaces the original return in the critic network by advantage function. The Asynchronous advantage actor-critic (A3C) algorithm makes it possible to calculate asynchronously while each worker gets the data directly from the global network and interacts with the environment [15].

As a continuous variant of Q learning, the NAF can reduce the sample complexity for continuous control tasks and it can

be regarded as an alternative to the more commonly used policy gradient and actor-critic methods [16].

The policy gradient algorithm has four challenges: (1) The large policy change will destroy the training. (2) It cannot map changes between policy and parameter space easily. (3) Improper learning rate causes vanishing or exploding gradient. (4) Low sample efficiency. The trust region policy optimization (TRPO) combines the MM algorithm, Trust region, and Importance sampling and will improve the performance in most cases [17].

PPO algorithm is a new kind of Policy Gradient algorithm. The performance of the Policy Gradient algorithm is very sensitive to the step size, but it is difficult to select the appropriate step size. If the difference between the old strategy and the new strategy is too large in the training process, it will be always difficult to calculate. PPO proposed a new objective function that can be updated in small batches by multiple training steps, which solved the problem that the step size in the Policy Gradient algorithm was difficult to determine. TRPO is also actually trying to solve this problem but it is much easier to do PPO than TRPO [18].

Although DDPG can sometimes achieve excellent performance, it is often not very easy to adjust the hyper-parameters and other things that can be adjusted. A common problem with DDPG is that Q functions learned will sometimes overestimate the Q values. It then causes the policy to break because it exploits an error in the Q function. Twin delayed DDPG (TD3) [19] is an algorithm that solves this problem by introducing three key tricks: clipped double-Q learning, "delayed" policy updates, and target policy smoothing.

There are several algorithms based on MDP and are famous in the field of multi-agent reinforcement learning.

The independent q learning (IQL) [20] algorithm regards the other agents directly as a part of the environment, which means that each agent in the environment is in its single-agent task. It is impossible to guarantee convergence and the agents will easily get lost in the endless exploration because the environment is non-stationary for any one of the agents. But this algorithm's performance is still relatively acceptable in practice.

In cooperative multi-agent reinforcement learning, each agent chooses actions based on its local observations to maximize team rewards. The Value-Decomposition Networks for Cooperative Multi-Agent Learning (VDN) [21] proposes a way to decompose the team's reward signal to each agent through back-propagation.

There is a credit assignment problem in MARL because the immediate reward of each agent are the same which means the agents who have made a huge contribution and those who have not much contribution will get the same rewards. To solve the problem, the Counterfactual Multi-Agent Policy Gradients (COMA) [22] algorithm uses a centralized critic to estimate the Q-function and decentralized actors to optimize the agents' policies. In addition, to address the challenges of multi-agent credit assignment, it uses a counterfactual baseline that marginalizes out a single agent's action, while keeping the other agents' actions fixed. COMA also uses a "critic" representation that allows the counterfactual baseline to be computed efficiently in a single forward pass.

The full factorization of VDN is not necessary to extract decentralized policies. The Monotonic Value Function Factorization for Deep Multi-Agent Reinforcement Learning (QMIX) [23] is a novel value-based method that can train decentralized policies in a centralized end-to-end fashion.

QMIX employs a network that estimates joint action values as a complex non-linear combination of per-agent values that condition only on local observations. It enforces a monotonicity constraint on the value's relationship between all the agents and one single agent.

VDN and QMIX address only a fraction of factorizable MARL tasks due to their structural constraint in factorization such as additivity and monotonicity. QTRAN [24] guarantees more general factorization than VDN or QMIX, thus covering a much wider class of MARL tasks than previous methods.

#### 4) Partially observable Markov decision process

While doing reinforcement learning research in most computer games, the agents will know with full certainty the state of the environment. In another word, the agent's sensors will allow it to perfectly monitor the state at all times, where the state captures all aspects of the environment relevant for optimal decision making. However, this kind of situation will rarely happen in the real world. For example, in many robotic applications, the robot's onboard sensors may not be able to enable the robot to unambiguously identify its location or pose. Furthermore, a robot's sensors are often limited to observing its direct surroundings, and there will always be features of the environment's state beyond the robot's visibility which can be called the hidden state. Another source of uncertainty regarding the true state of the system is imperfections in the robot's sensors. For instance, let us suppose a robot uses a camera to identify the person it is interacting with. The face-recognition algorithm processing the camera images is likely to make mistakes sometimes and report the wrong identity. Although in some domains the issues resulting from imperfect sensing might be ignored, the severe performance degradation caused by it is inevitable. The POMDP captures the partial observability in a probabilistic observation model, which relates possible observations to states [25].

#### 5) Semi-Markov process

The difference between the semi-Markov process and the Markov process is the type of time for which the state is defined. In the Markov process, the state is defined at the jump times. But in the semi-Markov process, the state is defined for every given time. The semi-Markov process is an actual stochastic process that evolves over time [26].

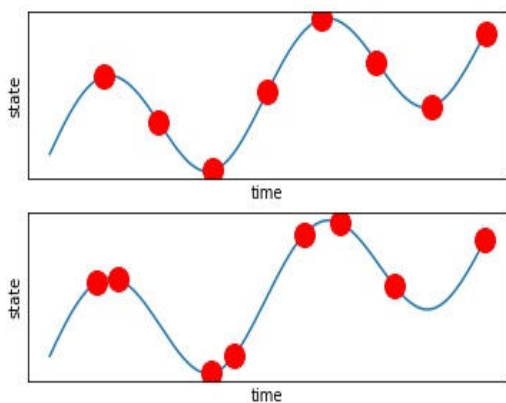


Fig. 2. The difference between Markov process and semi-Markov process

## IV. CONCLUSION

Since the birth of Markovian theories, there has been more than one hundred years of research history. Markovian theories have strongly promoted the development of science and technology. Especially in the field of reinforcement

learning, the application of Markov decision process theory has greatly promoted the development of reinforcement learning in recent years. On this basis, a large number of reinforcement learning algorithms have been proposed, although these algorithms are far from perfect. There is still much to be explored in Markovian theories. And the popularity of reinforcement learning will in turn promote the further development of them.

## ACKNOWLEDGMENT

This paper is supported by the Fundamental Research Funds for the Central Universities under Grant Number: N2017003.

## REFERENCES

- [1] P. A. Gagniu, *Markov chains: From theory to implementation and experimentation*, 1st ed. Nashville, TN: John Wiley & Sons, 2017.
- [2] A. A. Markov and N. M. Nagorny, *The theory of algorithms*. Dordrecht, Netherlands: Kluwer Academic, 2010.
- [3] S. M. Ross, *Stochastic Processes*, 2nd ed. Nashville, TN: John Wiley & Sons, 1996.
- [4] R. S. Sutton and A. G. Barto, *An Reinforcement Learning: Introduction*. Mit Press, 2012.
- [5] P. A. Gagniu, "Markov chains: From theory to implementation and experimentation."
- [6] R. Jarrow and P. Protter, "A short history of stochastic integration and mathematical finance: The Early Years, 1880–1970."
- [7] Y. Lan, D. Zhou, H. Zhang, and S. Lai, "Development of early warning models."
- [8] M. H. Swat, G. L. Thomas, J. M. Belmonte, A. Shirinifard, D. Hmeljak, and J. A. Glazier, "Multi-scale modeling of tissues using compucell3d."
- [9] L. Rabiner and B. Juang, "An introduction to hidden Markov models," *IEEE ASSP mag.*, vol. 3, no. 1, pp. 4–16, 1986.
- [10] A. Wrobel, "On Markovian decision models with a finite skeleton," *Zeitschrift für Operations Research*, vol. 28, no. 1, pp. 17–27, 1984.
- [11] V. Mnih *et al.*, "Playing Atari with deep reinforcement learning," *arXiv [cs.LG]*, 2013.
- [12] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proceedings of the 31st International Conference on Machine Learning*, 2014, vol. 32, pp. 387–395.
- [13] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," *arXiv [cs.LG]*, 2015.
- [14] K. V. R and T. J. N, "Actor-critic algorithms[C]//Advances in neural information processing systems." pp. 1008–1014, 2000.
- [15] V. Mnih *et al.*, "Asynchronous methods for deep reinforcement learning," *arXiv [cs.LG]*, 2016.
- [16] L. S., S. T., I., and S. Levine, "Continuous deep q-learning with model-based acceleration." pp. 2829–2838, 2016.
- [17] L. J., A. S., J. P., M., and P. Moritz, "Trust region policy optimization." pp. 1889–1897, 2015.
- [18] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," *arXiv [cs.LG]*, 2017.
- [19] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," *arXiv [cs.AI]*, 2018.
- [20] M. A., "Multiagent cooperation and competition with deep reinforcement learning," *PloS one*, vol. 12, no. 4, p. 0172395, 2017.
- [21] L. P., "Value-decomposition networks for cooperative multi-agent learning." 2017.
- [22] F. J., A. G., N. T., N., and S. Whiteson, "Counterfactual multi-agent policy gradients," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1), 2018.
- [23] T. Rashid, M. Samvelyan, C. S. de Witt, G. Farquhar, J. Foerster, and S. Whiteson, "QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning," *arXiv [cs.LG]*, 2018.
- [24] K. K., K. D., W. J., D. E. Hostallero, and Y. Yi, "Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning." pp. 5887–5896, 2019.
- [25] M. T, *Partially observable Markov decision processes*. Berlin, Heidelberg: Springer, 2012.
- [26] S. Z, *Hidden Semi-Markov models: theory, algorithms and applications*. Morgan Kaufmann, 2015.



# Fairness Enhancement of TCP Congestion Control Using Reinforcement Learning

Sang-Jin Seo\*  
School of Electronic and Electrical Engineering  
Kyungpook National University  
Daegu, Korea  
chil258@knu.ac.kr

You-Ze Cho  
School of Electronic and Electrical Engineering  
Kyungpook National University  
Daegu, Korea  
yzcho@ee.knu.ac.kr

**Abstract**—In TCP congestion control research, the use of machine learning to solve the issue of unused link bandwidth and to improve performance, such as maximizing link utilization or minimizing latency, is steadily increasing. Among such approaches, the Deep Q Network (DQN)-based TCP congestion control algorithm improves the link utilization but suffers from performance degradation when a specific link bandwidth is exceeded. In addition, inter-protocol fairness with other TCP congestion control algorithms has not been verified. In this paper, on a NS3 simulator, we conducted the experiments to enhance the improvement of the DQN-based TCP congestion control algorithm v2 in single flow and an inter-protocol fairness when several flows share the same bottleneck link. Our results confirmed that the average throughput was improved, and our approach is fairer than existing congestion control algorithms.

**Keywords**—Deep Q Network, TCP congestion control

## I. INTRODUCTION

TCP congestion control is a network congestion avoidance algorithm that involves slow start and adjustment of the congestion window (Cwnd), i.e., the maximum number of unacknowledged packets that can be transmitted. As an internet host function implemented in the operating system's protocol stack, TCP congestion control is used to reduce packet loss and avoid congestion collapse by limiting the Cwnd. Since TCP congestion control was first proposed in 1988 [1], various versions of the algorithm have been studied and proposed, e.g., NewReno [2], CUBIC [3], and BBR [4].

Research to improve the performance of TCP congestion control is still ongoing. In particular, since 2017, interest in congestion control research by applying machine learning has increased [5]. Machine learning is an approach that automatically improves performance through experience and learning. Machine learning has been applied to many congestion control algorithms such as an algorithm that has been improved to use the available link bandwidth as much as possible in various link environments [6], an algorithm to minimize latency in real-time traffic environments [7], and so on.

Among machine learning-based TCP congestion control algorithms, the Deep Q Network (DQN)-based TCP congestion control algorithm v1 we proposed earlier [6] has a higher

average throughput than CUBIC or NewReno when the link bandwidth is less than 50 Mbps. However, when the link bandwidth exceeds 50 Mbps, the average throughput is relatively low. Furthermore, inter-protocol fairness with existing congestion control was not verified. In this paper, on a NS3 simulator, we conducted an experiment to validate the improvement of the DQN-based TCP congestion control algorithm in single flow and an experiment to check the fairness when existing or DQN-based TCP congestion control algorithms share the same bottleneck link.

## II. RELATED WORK AND MOTIVATION

Existing congestion control algorithm such as NewReno and CUBIC use hand-tuned heuristics, so they only operate with a fixed Cwnd adjustment algorithm in any network environment. As Cwnd increases according to a fixed algorithm, the existing congestion control unconditionally causes congestion during the Cwnd increase.

Nowadays, due to the development of communication technology the internet speed increases, and the link bandwidth become larger. As the link bandwidth becomes larger such as 5G or over 100 Mbps of network environment, the existing congestion control algorithm has some issues that takes a longer time to increase Cwnd to use all link bandwidth, and the unused link bandwidth also increases due to unconditionally occurring congestion.

To overcome these issues by making it more adaptable to network environment, several algorithms, such as Orca [8], a hybrid congestion control protocol that depends on TCP fine-grained control action with DDPG, Aurora [9], a rate-based congestion control algorithm with PPO, and DQN-based TCP congestion control algorithm [6][10], that apply machine learning from various perspectives, have been proposed.

Among the machine learning methods, DQN is learned in real time by selecting an action that obtains the best reward from the state, it is suitable for adapting to the real-time changing network environment that has many state parameters, so there have been many attempts to apply it to the congestion control.

The DQN-based TCP congestion control algorithm v1 we proposed [6], has higher average throughput when the link bandwidth is under 50 Mbps but has lower average throughput than CUBIC when the link bandwidth is over 50 Mbps, and the inter-protocol fairness or round-trip time (RTT) has not yet been validated. So, to improve the issues of DQN-based TCP congestion control algorithm v1 and verify the inter-protocol fairness, we propose the DQN-based TCP congestion control algorithm v2.

### III. DQN-BASED CONGESTION CONTROL ALGORITHM

#### A. DQN Model

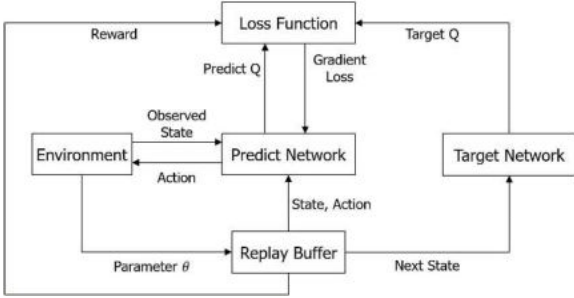


Figure 1. DQN algorithm.

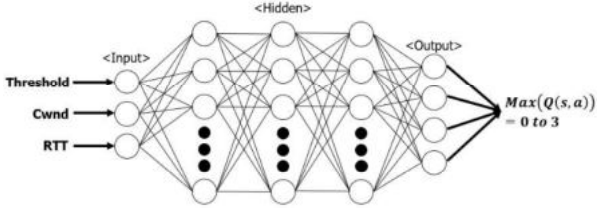


Figure 2. Deep Neural Network model for action.

Figure 1 is a diagram of DQN algorithm. The environment refers to the network environment in which communication is performing using congestion control, and the DQN agent is congestion control in environment. The DQN attempts to find a policy that maximizes the  $Q(s, a)$ , which is the value obtained when the agent takes an action ‘ $a$ ’ in a certain state ‘ $s$ ’. The  $Q$  value can be obtained as an output value of the deep neural network using the state as an input, and the one with the largest  $Q$  value is selected as the action like Figure 2. The parameter  $\theta$  is a tuple that [state ( $s$ ), reward ( $r$ ), action ( $a$ ), next state ( $s'$ )].

At the target network, the agent calculates the target  $Q$  using Bellman equation, as (1). The  $R(s, a)$  is the final reward value calculated using the equation described in section C, Reward function. The  $\gamma$  is a discount factor for weight, the  $s'$  is next state after action, and  $a'$  is the best action that the agent chose. So,  $\max_{a'} Q(s', a')$  means the maximum possible  $Q$  value at the next state, estimated by target network.

When the parameter is  $\theta_i$ , the loss for gradient descent is calculated with the loss function (2) using Mean Squared Error (MSE), and  $Q_{predict}$  is  $Q$  value of the current state after the DQN, from predict network.

$$Q_{target}(s, a) = R(s, a) + \gamma \max_{a'} Q(s', a') \quad (1)$$

$$L_i(\theta_i) = E_{s,a,r,s'} \left[ \left( Q_{target}(s, a; \theta_i) - Q_{predict}(s, a; \theta_i) \right)^2 \right] \quad (2)$$

The DQN-based TCP congestion control algorithm v1 [6], used [Threshold, Cwnd, RTT, Throughput] for state space. However, because the throughput overlaps the value calculated by Cwnd and RTT, we excluded the throughput from state space of the DQN-based TCP congestion control algorithm v2 to improve the learning performance. In this model, Cwnd is a byte unit.

In the hidden layer, the previous model used only ReLU as an activation function, so the model training was not good in some cases. We improved the performance by adding Dropout so that the model can be trained well in various network environments by reducing overfitting and using all nodes.

#### B. DQN output & Action Space [6]

$$a = \begin{cases} 0 : \text{Very high possibility of congestion} \\ 1 : \text{High possibility of congestion} \\ 2 : \text{Low possibility of congestion} \\ 3 : \text{Very low possibility of congestion} \end{cases} \quad (3)$$

#### Action – Cwnd Adjustment

$$= \begin{cases} Cwnd = Cwnd + SegmentSize \times (a - 1), & a > 0 \\ Cwnd = Cwnd - SegmentSize, & a = 0 \end{cases} \quad (4)$$

#### Epsilon – Greedy Probability

$$= \begin{cases} \epsilon, & \text{Random Action} \\ 1 - \epsilon, & \text{Best Rewards Action} \end{cases} \quad \epsilon = 0.015 \quad (5)$$

At first, after DQN learning, the output is the same value as (3), and the model learns by recognizing the output as network state. Action is selected using (4) according to the DQN output. At this time, the output value ‘ $a$ ’ is used as a variable in the Cwnd adjustment equation. By increasing/decreasing Cwnd according to the network state, the algorithm adjusts to maintain the link utilization rate learned through learning as much as possible.

Action selection probabilistically chooses between the best performance action and the random action using the Epsilon-greedy policy of (5), with the epsilon value as 0.015. Therefore, even if the network environment changes, the DQN model can explore to learn the new best performance action.

#### C. Reward function [6]

$$Rewards = \frac{Throughput_i}{Throughput_{DQN}} \quad (6)$$

$$Throughput_i = \frac{Cwnd_i}{RTT_i} \quad (7)$$

Equation (6) is a reward function indicating the link utilization rate, which is the improvement direction that this algorithm focuses on.  $Throughput_i$  is the throughput calculated by (7), and uses measured  $Cwnd_i$  and  $RTT_i$ , parameters when the DQN agent receives  $ACK_i$  for the  $i^{th}$  segment and confirms that the transmission completed without congestion.  $Throughput_{DQN}$  is the maximum throughput that will not cause congestion, obtained through DQN learning. By adjusting Cwnd using a Rewards function, it can minimize the congestion occurring in a micro-changing network environment.

#### D. Overall algorithm

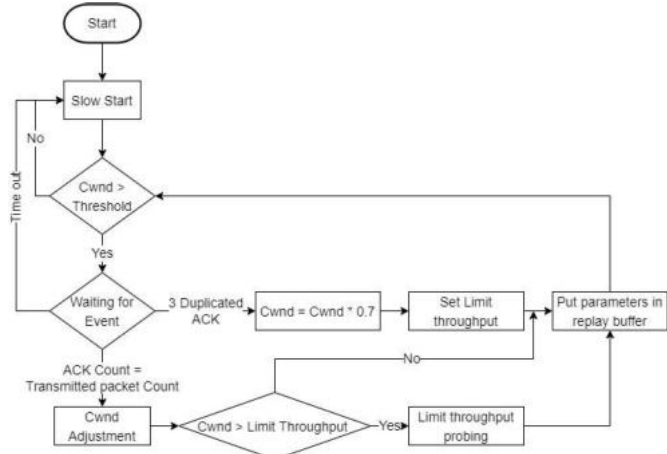


Figure 3. Flowchart of DQN-based TCP congestion control algorithm.

Figure 3 is a flowchart of the DQN-based TCP congestion control algorithm v2. When communication starts, it enters and operates in the standard TCP slow start until the Cwnd is bigger than threshold. After the slow start, the algorithm waits for the event. First, when three duplicated ACKs occur, it is judged as congestion, and Cwnd is reduced by using the standard CUBIC Cwnd reduction index of 0.7 which aims for fast Cwnd increase and scalability. Afterwards, set the limit throughput for  $Throughput_{DQN}$ . Second, when a timeout occurs, it re-enters a slow start according to the standard TCP operation. Finally, when ACKs equal to the number of all transmitted packets are received, it operates in an algorithm that takes an action based on (4). If Cwnd is bigger than the limit throughput without 3 duplicated ACKs, the agent enters limit throughput probing, otherwise it input parameters into the replay buffer.

#### IV. EXPERIMENT ENVIRONMENT

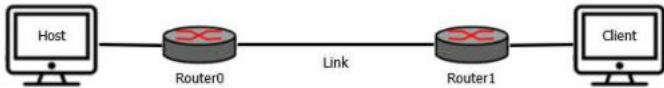


Figure 4. NS3 simulator experiment A setup.

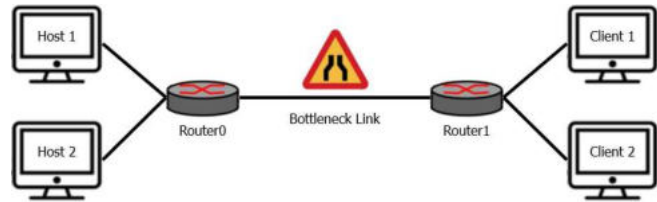


Figure 5. NS3 simulator experiment B setup.

Figure 4 illustrates the setup of experiment A on NS3. The experiment A involves checking the performance improvement of the DQN-based TCP congestion control algorithm v2 compared to existing congestion control algorithms. The performance is compared by transmitting NewReno, which uses the same AIMD algorithm, CUBIC, which is widely used as a standard, the DQN-based TCP congestion control algorithm v1, and the v2 with only one host and client in a single flow. The RTT of the experiment environment is set to 100ms, and the experiment time is 300 seconds. Link bandwidths of 5, 25, 50, 75, and 100 Mbps are used.

Figure 5 illustrates the setup of experiment B on NS3. The experiment B compares fairness when two flows using different congestion control algorithms share a bottleneck link. The experiment is conducted with a bottleneck link bandwidth of 50 Mbps, an RTT of 100 ms, and an experiment time of 300 seconds with two hosts and client. The competition uses the DQN-based TCP congestion control algorithm v2 vs CUBIC or NewReno.

#### V. EVALUATION

##### A. Congestion control comparison when single flow

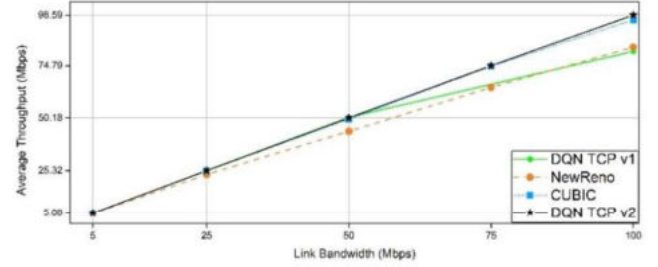


Figure 6. Average throughput comparison of each congestion control.

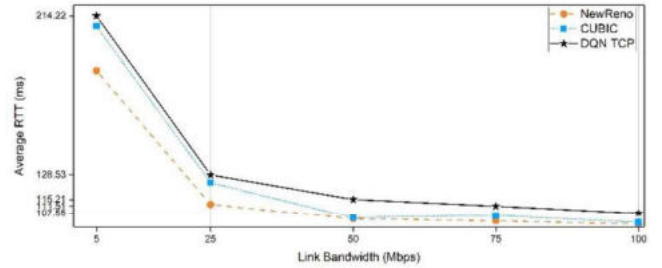


Figure 7. Average RTT comparison of each congestion control.

Figure 6 is a graph comparing the average throughput of each congestion control during a single flow. The DQN-based TCP congestion control algorithm v1 suffers from lower average throughput than CUBIC and NewReno when the link bandwidth exceeds 50 Mbps. However, we confirmed that the proposed algorithm has a higher average throughput than the existing congestion control because there is almost no Cwnd reduce operation due to congestion.

Figure 7 is a graph comparing the average RTT of each congestion control. Although the average RTT of the proposed algorithm was higher than existing congestion control in all link bandwidths, the difference was insignificant, i.e., approximately 5 ms.

##### B. Comparison of fairness when sharing bottleneck link

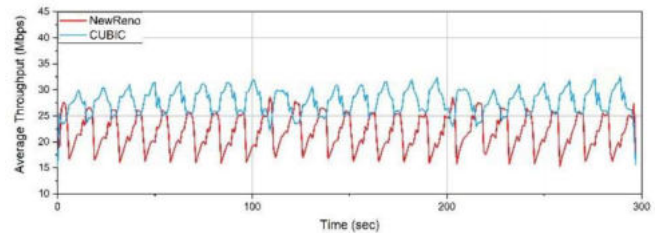


Figure 8. NewReno vs CUBIC average throughput.

## VI. CONCLUSION

In this paper, we propose an improved DQN-based TCP congestion control algorithm v2 and simulate the performance difference of each algorithm in a single flow as well as the fairness comparison when sharing a bottleneck link on an NS3 simulator. As a result of improving the DQN model, the average throughput improved to higher than existing TCP congestion control algorithms for all bandwidths. Through a fairness experiment when sharing a bottleneck link, the proposed algorithm and CUBIC confirmed that fairness was good by using the link bandwidth at an approximately 49:51 ratio. However, the proposed algorithm and NewReno do not significantly improve compared to CUBIC vs NewReno by using the link bandwidth at about 55:45. In future research, we will attempt to adjust Cwnd adaptively and make it more TCP-friendly in a more dynamic network environment by applying a learning method such as DDPG with continuous output rather than DQN with discrete output.

## ACKNOWLEDGMENT

This research was supported in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by Ministry of Education (No. NRF-2018R1A6A1A03025109) and by National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2019R1A2C1006249).

## REFERENCES

- [1] V. Jacobson, "Congestion avoidance and control," ACM SIGCOMM Computer Communication Review, vol. 25, pp. 157-187, January 1995.
- [2] S. Floyd, A. Gurtov, and T. Henderson, "The NewReno Modification to TCP's Fast Recovery Algorithm," RFC 2582, April 1999.
- [3] S. Ha, I. Rhee, and L. Xu, "CUBIC: A New TCP-Friendly High-Speed TCP Variant," ACM SIGOPS Operating Systems Review, vol. 42, pp. 64-74, July 2008.
- [4] N. Cardwee et al., "BBR: Congestion-Based Congestion Control," Commun. ACM, vol. 60, pp. 58-66, February 2017.
- [5] W. Wei, H. Gu, and B. Li, "Congestion control: A Renaissance with Machine Learning," IEEE Network, vol. 35, pp. 262-269, July/August 2021.
- [6] S. J. Seo and Y. Z. Cho, "DQN-Based TCP Congestion Control Algorithm to Improve Link Utilization," 2<sup>nd</sup> Korea Artificial Intelligence Conference, September 2021.
- [7] J. Fang et al., "Reinforcement Learning for Bandwidth Estimation and Congestion Control in Real-Time Communications," Proc. NeurIPS Workshop on Machine Learning for Systems, 2019.
- [8] S. Abbasloo et al., "Classic Meets Modern: A Pragmatic Learning-Based Congestion Control for the Internet," Proc. ACM SIGCOMM, pp. 632-647, 2020.
- [9] N. Jay, N. Rotman, B. Godfrey, M. Schapira, and A. Tamar, "A Deep Reinforcement Learning Perspective on Internet Congestion Control," 36th International Conference on Machine Learning, 2019.
- [10] Y. Wang, L. Wang, and X. Dong, "An Intelligent TCP Congestion Control Method Based on Deep Q Network," Future Internet, vol. 13, no. 10, pp. 261, October 2021.

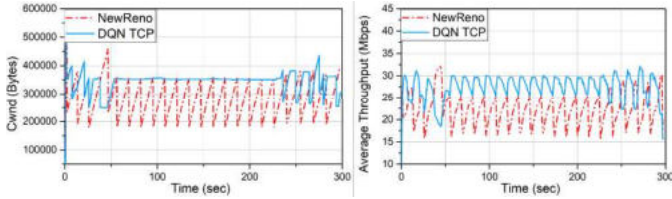


Figure 9. NewReno vs the proposed algorithm average throughput & Cwnd.

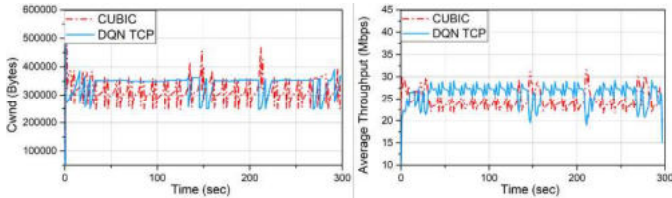


Figure 10. CUBIC vs the proposed algorithm average throughput & Cwnd.

Figure 8 is a graph of average throughput when NewReno and CUBIC compete by sharing the same bottleneck link. Ideally, each should have 25 Mbps of average throughput, but CUBIC has 27.7 Mbps and NewReno has 21.85 Mbps for average throughput because CUBIC has a faster Cwnd increase speed and small reduction index than NewReno.

Figure 9 is a graph of Cwnd and average throughput when NewReno and the proposed algorithm compete by sharing the bottleneck link. NewReno reduces Cwnd using 0.5 of reduction index when congestion occurs, and the Cwnd increase speed is slower than proposed algorithm or CUBIC. The proposed algorithm maintains the learned Cwnd, and NewReno repeats the increase and decrease in Cwnd due to congestion. Even if congestion occurs because of exploration of DQN or Cwnd increase of NewReno, the proposed algorithm increases the Cwnd back to the previous Cwnd again after decreasing it, and maintains. As a result, the average throughput of NewReno is 22.28 Mbps and that of the proposed algorithm is 27.39 Mbps, which is not significantly improved compared to CUBIC vs NewReno.

Figure 10 is a graph of Cwnd and average throughput when CUBIC and the proposed algorithm compete by sharing the same bottleneck link. CUBIC selects a concave/convex function as a Cwnd increment algorithm according to  $w_{max}$ . When the proposed algorithm occupies the bottleneck link by learning the result, CUBIC adjusts Cwnd using the remaining link bandwidth. The proposed algorithm maintains Cwnd as much as possible and even if congestion occurs, and the two algorithms have a faster Cwnd increase rate and a smaller reduced index than NewReno. For these reasons, the average throughput of CUBIC is 24.62 Mbps, and that of the proposed algorithm is 26.08 Mbps, confirming that is fairer than when NewReno competing with CUBIC.

# Merging Reinforcement Learning and Inverse Reinforcement Learning via Auxiliary Reward System

1<sup>st</sup> Wadhah Zeyad Tareq  
Computer Engineering  
Yıldız Technical University  
Istanbul, Turkey  
wadhah.zeyad.t.tareq@std.yildiz.edu.tr

2<sup>nd</sup> Mehmet Fatih Amasyali  
Computer Engineering  
Yıldız Technical University  
Istanbul, Turkey  
amasyali@yildiz.edu.tr

**Abstract**—In recent years, learning from demonstration has become one of the promising methods in robotics and interactive systems. Learning from demonstration is a model by which an agent learns by observing an expert. The expert could be a pre-trained agent or human. The main problem with learning from demonstrations is the difference between the reward representation in the demonstrations and the actual environment. During the construction of the demonstrations, it is easy to add new rewards to enhancement the agent’s performance. In contrast, it is not easy to do that in an actual environment. This work is built upon our previous work to solve this problem. In previous work, the agent uses Reinforcement Learning algorithms to learn how to play video games from demonstrations. The agent was supplied with an external reward to solve the problem of missing rewards in the hard exploration environments. In this work, Inverse Reinforcement Learning uses to extract the external rewards from the demonstration and make them available during the interaction period. The results showed that inverse learning enables the agent to interact with the environment after the pre-training. Furthermore, the performance of the agent becomes more stable.

**Index Terms**—deep reinforcement learning, inverse reinforcement learning, prioritized double deep q-networks, atari games

## I. INTRODUCTION

Deep reinforcement learning has succeeded in many sequential decision-making problems such as Atari games and robotic control. One of the challenges to applying reinforcement learning is the missing of environment reward which leads to makes the exploration difficult [1]. The reward is the only criterion that evaluates the efficiency of the reinforcement learning agent. Designing a reward function to assess the agent behavior depending on the environment is complex and impossible. One successful solution to learning the reward function is inverse reinforcement learning (IRL) [2].

IRL solves the reward engineering problem by obtaining the reward function from the expert’s demonstrations [3]. An expert can be a professional human player or an agent who has been previously trained using one of the learning algorithms. IRL agents aim is to find out the reward function that explains the behavior of the expert. Once the reward function is found,

the agent starts using the standard reinforcement learning methods to find out the optimal policy expectedly to behave as well as the expert. Reward function enables the agent to interact with the environment and improve its behavior without extra demonstrations [4].

In traditional Imitation Learning (IL), the agent is used to directly learn the expert’s behavior. IRL enables the agent to learn the strategies which lead to that behavior. Understanding the strategy increases the robustness of the new behavior and allows the agent to handle new challenges such as new initial states or any change in the environment. Another advance of IRL is the ability of the agent to explore the environment via online learning using the standard RL algorithms. The exploration enables the agent to visit new states without the need for further demonstrations. Being robust and adapting to the change is essential to an agent running in a dynamic world [5]. Unlike deep RL, few IRL algorithms have been developed to play video games. In this paper, we build on our previous work [6]. The earlier work represented the first phase, and the current work represented the second phase. In phase one, the agent is trained to learn from the demonstrations without interacting with the environment. In phase two, the agent uses the IRL principles to interact with the environment and enhance its performance.

## II. RELATED WORK

There are two parts of related work. The first part is about the deep RL in video games, especially Atari games. The second is about the IRL works in different areas.

The first most popular algorithm implemented Deep RL in Atari games is the deep Q-network algorithm (DQN) [7]. DQN merged the Q-learning [8] with the convolutional neural network (CNN) and tested it on many Atari games. The result showed that the DQN agent was able to reach the expert performance on many games. After that, the RL community has made many changes and extensions to the DQN algorithm to improve its performance and stability. Using the Double Q-learning algorithm [9] rather than the Q-learning algorithm was the first change to the DQN. The Double DQN (DDQN)

[10] showed that the DQN suffers from significant overestimation, and the Double Q-learning reduces that overestimation which leads to better performance. Prioritized experience replay (PER) [11] replaced random sampling with sampling experience by probability. This change allowed the DQN to sample experiences that have more information to learn than the other.

The dueling network [12] suggests decoupling the Q-learning into two estimators: state and action. The state estimator tells us it is (or is not) a valuable state regardless of the effect of the actions at that state. A3C [13] presented asynchronous four standard reinforcement learning algorithms and showed that parallel actor-learners stabilize training, allowing all four methods to train neural network controllers successfully. Distributional Q-learning [14] learns a categorical distribution of discounted returns instead of estimating the mean. Noisy DQN [15] uses stochastic network layers for exploration. Finally, Rainbow [16] studied the efficiency of the six previous extensions in one algorithm. Their algorithm provided an agent with performance better than each extension separately.

Deep Q-learning from demonstrations (DQfD) [17] was the first algorithm used demonstrations in Atari games. The DQfD included two learning phases. In phase one, the agent learning by imaging the demonstrations. In phase two, the agent is learning by interacting with the environment. The self-generated data and the demonstrations are used in the learning during phase two. The DQfD was the first work to enable RL agents to score in environments where the reward is rare or missed. Later, many works [18] – [20] attempt to use demonstrations to enhancement the performance of RL in such environments.

The second part of the related works is about the IRL. The IRL first appeared to solve the problem of the new state in the Imitation Learning (IL) approaches. In IL, when the agent visited a new state, it became impossible to return to the demonstrated states. IRL was first introduced by Ng, and Russell [2]. They proposed to solve the reward engineering problem by inferring the reward function from the expert demonstrations. The main challenge was the ambiguity. In the same demonstration, many reward functions could produce the expert policy. In [21], they developed a method that produces a policy that gets the same reward as the expert policy rewards. The later works are Bayesian IRL and Maximum Entropy IRL. Bayesian IRL [22] adopted the ambiguity by calculating the distributed rewards rather than focusing on the given function. This method produces the same guarantee as in [21]. On the other hand, Maximum Entropy IRL [23] builds a reward function that meets the expert features. This function guarantees the higher-entropy stochastic policy and allows the generalization in the environments using many different planning dynamics.

As mentioned before, few IRL algorithms were developed to play video games. The work by Uchibe [24] used logistic regression to classify the transitions into two groups: expert and non-expert. The classifier is used instead of the reward

function to train standard deep RL algorithms. The results showed that their performance rarely outperforms the IL. The other work is CNN-AIRL [25]. They modified the Adversarial IRL algorithm [3]. Their modification includes adding CNN to the AIRL baseline, normalizing the environment reward, and increasing the size of the discriminator dataset. Additionally, they represented the state with low-dimensional by auto-encoder architecture built especially for video games. Their algorithm achieves good performance on the simple Catcher video game. They applied their algorithm to the Enduro game, and the algorithm performance was lower than the expert performance.

### III. BACKGROUND

#### A. Markov Decision Processes

RL researchers adopt the Markov Decision Process (MDP) formalism for their works. MDP framework is a tuple  $(S, A, T, \gamma, R)$  where  $S$  is a set of states. Each state represents the status of the environment at that time.  $A$  is a set of actions. Actions are something that an agent can do in the state.  $T$  is a transition probability from one state to another.  $\gamma$  is a discount factor with a value between 0 and 1. The discount factor controls the dependability of the far future rewards. If  $\gamma = 0$ , the agent will only learn from the immediate reward.  $R$  is a signal that represents the reward of the agent. The reward evaluates the agent's action. For each state, there is a policy  $\pi$  determining which action the agent must select. The agent's primary goal is to find the policy that enables it to choose the optimal action, which increases the cumulative reward. The  $Q$  value for an  $(s, a)$  pair estimates the expected future reward when the agent follows that policy. The  $Q^\pi$  represents the  $Q$  value with that policy. The optimal  $Q$  value  $Q^*(s, a)$ , which achieve maximal reward in one episode, is determined by solving the Bellman equation [17] [22]:

$$Q^*(s, a) = E \left[ R + \gamma \sum_{s_{t+1}} P(s_{t+1}|s, a) \max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1}) \right] \quad (1)$$

From the above equation, the optimal policy can be found by:

$$\pi(a) = \operatorname{argmax}_{a \in A} Q^*(s, a) \quad (2)$$

DQN [7] approximates the  $Q$  value for all available actions in a state by using a deep neural network. The network's input is the stack of several states (to determine the direction and speed of the moving objects in the game), and the output is the  $Q$  value for each action. DQN used a separate network to calculate the target values. This network is updated after a specific number of steps by copying the weights of the regular network. The aim here is to stability the  $Q$  target values. DDQN [9] used the regular network to select the best action and the target network to calculate the target  $Q$  value for that action. The DDQN loss is [17]:

$$J_{DDQN} = (R(s, a) + \gamma Q(s_{t+1}, a_{t+1}^{max}; \theta^-) - Q(s, a; \theta))^2 \quad (3)$$

Where  $\theta$  is the weights of the regular network, the  $\theta^-$  is the weights of the target network, and  $a_{t+1}^{max} = \text{argmax}_a Q(s_{t+1}, a; \theta)$ .

PER [11] set a priority for each sample. These priorities define which sample must be selected first. In the classical DQN and DDQN, the samples were selected randomly. Due to the randomness, some important samples may be deleted before being selected. Also, the PER is sampling the necessary samples more frequently. The probability of sampling a particular transition  $i$  is proportional to its priority:

$$P_i = \frac{p_i^\alpha}{\sum_k p_k^\alpha} \quad (4)$$

Where  $p_i$  is the last temporal difference error (the difference between the new and old prediction) calculated for the  $i$  sample plus a small positive constant. This small value ensures that all samples are selected with some probability. The  $\alpha$  is a hyper-parameter used to reintroduce some randomness in the experience selection for the replay buffer. When the  $\alpha$  is equal to 0, all samples are selected randomly. When the  $\alpha$  is equal to 1, only necessary samples with the highest probability are selected.

### B. Deep Q-learning from demonstrations

The general structure of our work is like the DQfD structure [17]. DQfD contains two phases: the pre-training phase and the interacting phase. The pre-training phase aims to learn as much as possible before start interacting with the environment. In this phase, the agent imitates the demonstrator to satisfy the Bellman equation. During this phase, the agent updates the network by applying four losses: the 1-step double Q-learning loss, an n-step double Q-learning loss, a supervised large margin classification loss, and an L2 regularization loss on the network weights and biases. In the interacting phase, the agent acts on the environment by selecting action with its learned policy. In each time step, the agent saves current states, action, reward, and next state as self-generated data. Once the memory is full, the agent pulls out the oldest self-generated data and keeps the demonstration data without any change. The agent updates its network with a mixed mini-batch of demonstration and self-generated data. DQfD used PER [11] to control the ratio between demonstration and self-generated data while learning to improve the algorithm's performance.

### C. Inverse Reinforcement Learning

The main difference in IRL is the expert's demonstrations. In IRL, the demonstration sampled from the MDP without the reward. These samples represent the expert policy  $\pi_E$  or the behavior that the expert follows to solve a particular problem. IRL aims to find a reward function that explains the expert behavior. IRL algorithm receives as inputs a set of the expert sampled transitions  $D_E = (s, a, s')$  and if available, a set of non-experts sampled transitions  $D_{NE} = (s, a, s')$ . The goal of an IRL algorithm is to compute the reward  $R$  [4]:

$$\forall s \in S, \quad \text{argmax}_{a \in A} Q_R^*(s, a) = \text{Supp}(\pi_E(\cdot|s)) \quad (5)$$

Where  $Q_R^*$  represents the optimal score function, and  $\pi_E$  represents the expert policy. It is important to separate the expert sampled transitions from the non-experts sampled transitions to prevent the last one from being optimal, which is essential to imitate the expert behavior. To achieve that, one must search for a significant reward, which is a reward for which the expert policy is optimal and for which only expert actions are optimal or at least a subset of expert actions [4]:

$$\forall s \in S, \quad \text{argmax}_{a \in A} Q_R^*(s, a) \subset \text{Supp}(\pi_E(\cdot|s)) \quad (6)$$

The IRL researches, thus, included methods to find significant rewards [21], [23]. Once the reward is determined, the  $MDP = (S, A, T, \gamma, R)$  must be solved to compute the agent's policy, which is a problem as such.

## IV. LEARNING FROM AUXILIARY REWARDS METHOD

Our agent relies on the extra rewards for learning through two stages. The first stage is known as phase one or the pre-training phase. The second stage is known as phase two or the interacting phase. In the following two subsections, the details of these two phases will be explained.

### A. The Pre-Training Phase

Our work for this phase was published previously [6]. In this section, phase one is briefly described. For further details about the implementation, evaluation, and comparisons, the readers can check out our previous article. In this phase, the agent learns to play video games using human demonstrations. Two classical RL algorithms were used DDQN [9] and PER [11]. Our contribution is to use auxiliary rewards rather than environment rewards. The auxiliary rewards were involved within the demonstrations, and the agent learned to play video games depending on these demonstrations only. There is no interacting or self-generated data during this phase.

The auxiliary reward is represented by 0 or 1. A human player selects the correct action in each step and sets that action's reward to 1. The reward of the rest actions is 0. Assigning actions with 0 is essential to give all actions realistic values. These absolute values prevent the network from updating toward ungrounded variables. Five Atari games were used to benchmark the performance of the agent. Three of these games are considered hard exploration games (Environment rewards are miss or rare), and two are simple games. Our agent results in the hard exploration games exceeded the results of many baseline algorithms including DQfD. Fig. 1 shows the agent results in all five games [6].

### B. The Interacting Phase

Once the pre-training phase is complete, the agent starts acting on the environment by selecting action using its learned policy, collecting self-generated data, and learning from that data. The objectives of phase two are to explore new states and make the agent performance stable. The agent performance in some games is unstable. However, once the agent started interacting with the environment, the learned policy performance decreased. The reason here is the reward distribution. In phase

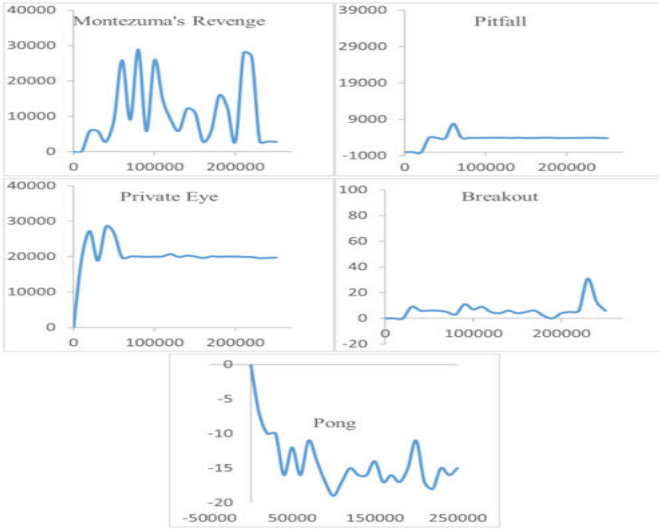


Fig. 1. Phase one scores averaged over ten episodes for each weight in 25 different weights starting from 10000 to 250000 steps of training for the five games: Montezuma’s Revenge, Pitfall, Private Eye, Breakout, and Pong.

one, the agent received a 0 or 1 reward for each action. On the other hand, phase two depends on environment rewards. These rewards have different time intervals variations from game to game. Fig. 2 shows the performance of the agent in Montezuma’s Revenge game after starting interacting.

Fig. 2 proves that the agent performance gets effect by the newly generated data. The agent runs on the environment for 50000 steps which is enough to show the problem. To solve this problem, the concept of IRL is used to extract new auxiliary rewards that help the agent during phase two. All details will explain in the next section.

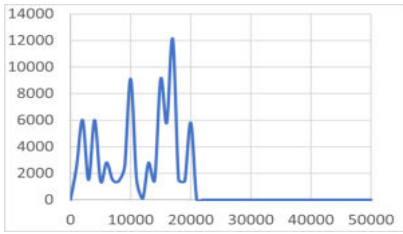


Fig. 2. The performance of our agent in phase two for Montezuma’s Revenge game. The agent starts learning from self-generated data after performing 20,000 steps.

### V. LEARNING USING IRL

IRL’s main concept is learning the reward function from the demonstration data. A new CNN is built to serve as a Reward Function (RF) to take advantage of that. The RF dataset is a particular transition from the phase one demonstration. These transitions are the transitions with rewards equal to 1. The current state in that transition is considered as an input to the RF. A zeros array with a size equal to action space size is a target output for the RF. Only the correct action in that transition is assigned with 1 in the target output. The

processor of building the RF dataset is shown in Fig. 3. The  $S_t$  refer to the current state, the  $S_{t+1}$  refer to the next state, (a) and (r) represent the selected action and the environment reward respectively. The RF has the same architecture as the agent regular CNN architecture from phase one. The main differences are the output activation function and the loss function. In the agent’s CNN, a linear activation function is used. In the reward function, the SoftMax activation function is used. For the loss function, the Categorical cross-entropy loss is used rather than Huber loss.

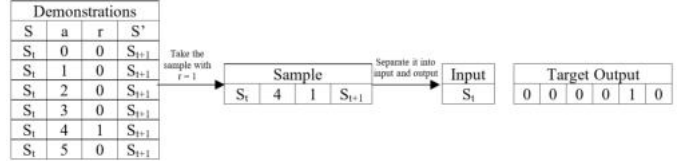


Fig. 3. Reward function dataset building procedure.

Fig. 4 shows the differences between the standard MDP used in classical RL algorithms and our new framework. In the new framework, the agent will select the action, and the reward function will provide a reward for the chosen action. This reward will be used alongside the environment reward. Fig. 5 compares our architecture with the DQfD architecture for phases one and two.

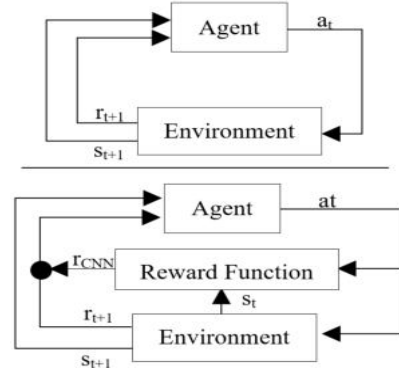


Fig. 4. The above plots show the standard MDP for solving RL problems. The down plots show our new framework by adding the RF.

## VI. RESULTS AND DISCUSSIONS

Our integrated approach has only been tested on two games: Montezuma’s Revenge and Breakout. Montezuma’s Revenge is a challenging exploration game, while Breakout is a simple game with routinely environmental feedback. The RF part has been tested on all five games from phase one. The reason for testing the integrated approach on two games only is the training time that is required.

### A. Reward Function Results

This part discusses the results of RF where this part contains the training and testing results for the RF CNN. The reward function trained with the dataset extracted from the



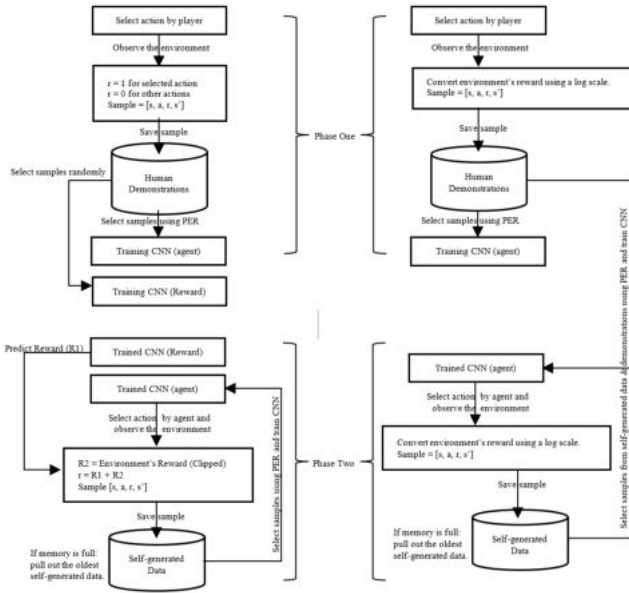


Fig. 5. The left plots show our architecture. The right plots show the DQfD architecture.

demonstration for 20,000 update steps. After that, the RF was tested on a new test set. This test set differs from the training set. The test set builds by playing each game for one episode. The episode starts from the initial state and finishes when all available lives become equal to zero. In general, one episode is enough to visit most states in the training test. These test sets find out if the reward function suffers from overfitting or underfitting. The details of the training dataset, test dataset, and test accuracy are shown in Tab. I. Furthermore, the RF learning F1 scores for each action are shown in Tab. II.

TABLE I  
DATASET DETAILS AND TEST ACCURACY

Game	Training Dataset		Test Dataset		Test Accuracy
	Size	Episode	Size	Episode	
MR	60877	5	9705	1	0.8646
Pitfall	132844	5	25135	1	0.9148
Private Eye	39098	5	8282	1	0.8443
Breakout	31297	9	3282	1	0.6453
Pong	26783	3	6245	1	0.7969

TABLE II  
F1 SCORE RESULTS

Accuracy	0	1	2	3	4	5	6	7	Weighted
MR	0.86	0.63	0.91	0.88	0.85	0.85	0.78	0.74	0.86
Pitfall	0.93	0.64	0.78	0.93	0.	0.59	0.76	0.	0.91
Private Eye	0.41	0.	0.54	0.87	0.92	0.	0.55	0.57	0.83
Breakout	0.68	0.	0.29	0.29	-	-	-	-	0.56
Pong	0.87	0.64	0.51	0.6	0.	0.	-	-	0.81

RF has low performance on both Breakout and Pong games. The reason is these games have an infinite state space. Furthermore, the size of the training set for both games is small. In such games, the RF output is not very important due to

the availability of environment reward, which can replace the reward function output.

## B. Interacting Results

The result of this phase is reported while the agent is running in the environment. The agent started with an empty buffer. This buffer will store the agent's self-generated data. The environment provides the current state for both the agent and the RF. As in DQfD, our agent selects action using the trained policy from the pre-trained phase. The RF role is to generate rewards array output depending on the current state. The current action reward and the environment clipped reward will add together and save with the current state, action, and next state. The agent uses these transitions for training.

In each game, the convolutional neural network is trained on a single GPU for 500k steps. We used the Nvidia GeForce GTX 1060 graphics card (6 GB memory version). For comparison, our agent, the DQfD agent, and the standard Prioritized Double DQN (DDQN) agent were trained on the same device. The standard PDDQN is the algorithm used to build our approach. The PDDQN agent differs from our agent because it does not have demonstration data, pre-training phase, and reward function. The PDDQN agent took about one week to finish the training for each game. The DQfD agent took about four days for each game, while our agent required 20 days to complete the training for each game. The reason for the difference in training time is due to the number of predications in the algorithms. In each step, our agent makes two predications, the first prediction is selecting the correct action, and the second is predicting the reward for that action. On the other hand, the DQfD and PDDQN agents predict the correct action only. The difference in training time is the only disadvantage in our work. Fig. 6 Showing the online results for the three algorithms in Montezuma's Revenge and Breakout games.

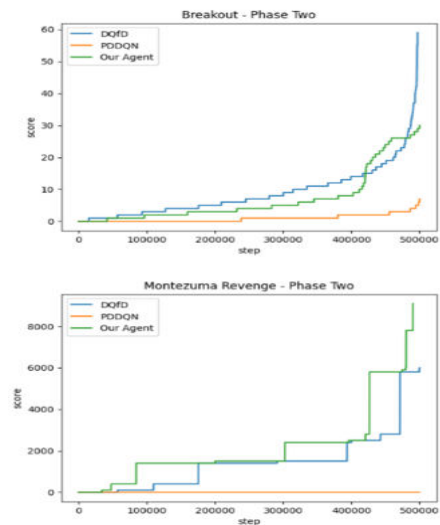


Fig. 6. Online rewards for the Montezuma's Revenge and Breakout games. The zero rewards between episodes are removed. Scores are from the Atari game, regardless of the internal representation of reward used by the agent.

In Montezuma’s Revenge game, our agent performance is good. The high score of our agent is about 9000 scores, while the DQfD agent scored 6000 only. As in all standard reinforcement learning algorithms, the PDDQN agent score is zero. In the Breakout game, our agent performance is poor compared with the DQfD agent. Our agent was able to score 30 only, while the DQfD agent scored 60. The reason here is the infinite state space in the simple games. For example, in the Breakout game, the state changes depending on the ball. So, it is impossible to train the RF with all possible states. The breakout game is a simple challenge for all classical reinforcement learning algorithms [7], [10] – [17]. All these works obtained results ranging from 300 to 600 after 200 million steps of training. For 500 steps, our agent performance is better than the performance of the PDDQN agent. Our approach can score higher in both games if trained with millions of steps. The last plot is the ratio between the demonstration and the self-generated data in the mini-batch. As mentioned before, the DQfD agent trained its current network using both demonstrations data and self-generated data. The ratio for both Montezuma’s Revenge and Breakout is shown in Fig. 7. Unlike our agent, the DQfD agent relies on demonstrations all-time of training. Furthermore, it uses the demonstrations only at the beginning of the training without returning to the self-generated data. Our agent uses the self-generated data only, which is the main concept in learning from interaction and makes it different from imitation learning.

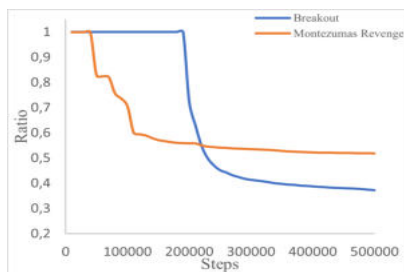


Fig. 7. The ratio of how often the demonstration data was sampled versus the self-generated data in the DQfD algorithm.

## VII. CONCLUSION

We have introduced an integrated approach that uses Reinforcement Learning, Learning from Demonstration, and Inverse Reinforcement Learning for solving the hard-exploration problem. Our approach solves the problem of sparse and/or deceptive rewards by using external rewards resulting in higher scores compared with prior works. The external reward system opens up a large number of new research directions including experimenting with different environments and different methods with one limitation: the availability of human demonstration.

## REFERENCES

[1] A. Ecoffet, J. Huizinga, J. Lehman, K. O. Stanley, and J. Clune, “Go-Explore: a New Approach for Hard-Exploration Problems,” arXiv preprint arXiv:1901.10995, 2019.

[2] A. Ng, and S. Russell, “Algorithms for inverse reinforcement learning,” In ICML, Vol. 1, 2000.

[3] J. Fu, K. Luo, and S. Levine, “Learning robust rewards with adversarial inverse reinforcement learning,” ICLR, 2018.

[4] B. Piot, M. Geist, and O. Pietquin, “Bridging the gap between imitation learning and inverse reinforcement learning,” IEEE Transactions on Neural Networks and Learning Systems, vol. 28, no. 8, 2017.

[5] T. Munzer, B. Piot, M. Geist, O. Pietquin, and M. Lopes, “Inverse reinforcement learning in relational domains,” in Proc. Of the 24th International Joint Conference on Artificial Intelligence. 2015.

[6] W. Z. Tareq, and M. F. Amasyali, “A New Reward System Based on Human Demonstrations for Hard Exploration Games,” CMC-Computers, Materials and Continua, vol. 70, no. 2, pp.2401–2414, 2022.

[7] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, et al., “Human-level control through deep reinforcement learning,” Nature, vol. 518, no. 7540, pp. 529–533, 2015.

[8] C. J. C. H. Watkins, “Learning from delayed rewards,” Ph.D. thesis, University of Cambridge England, 1989.

[9] H. V. Hasselt, “Double Q-learning,” Advances in Neural Information Processing Systems, vol. 23, pp.2613–2621, 2010.

[10] H. V. Hasselt, A. Guez, and D. Silver, “Deep reinforcement learning with double q-learning,” in Proc. of the 30th AAAI Conf. on Artificial Intelligence, Arizona, USA, vol. 30, pp. 2094–2100, 2016.

[11] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, “Prioritized experience replay,” in Proc. of the Int. Conf. on Learning Representations, San Juan, Puerto Rico, 2016.

[12] Z. Wang, T. Schaul, M. Hessel, H. V. Hasselt, M. Lanctot, et al., “Dueling network architectures for deep reinforcement learning,” in Proc. of the 33rd Int. Conf. on Machine Learning, New York, NY, USA, vol. 48, 2016.

[13] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Harley, et al., “Asynchronous methods for deep reinforcement learning,” in Proc. of the 33rd Int. Conf. on Machine Learning, New York, NY, USA, vol. 48, 2016.

[14] M. G. Bellemare, W. Dabney, and R. Munos, “A distributional perspective on reinforcement learning,” in Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, vol. 70, 2017.

[15] M. Fortunato, M. G. Azar, B. Piot, J. Menick, M. Hessel, et al., “Noisy networks for exploration,” in Proceedings of the International Conference on Learning Representations (ICLR), 2018.

[16] M. Hessel, J. Modayil, H. V. Hasselt, T. Schaul, G. Ostrovski, et al., “Rainbow: Combining improvements in deep reinforcement learning,” in Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI-18), vol. 32, no. 1, pp. 3215–3222, 2018.

[17] T. Hester, M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, et al., “Deep q-learning from demonstrations,” in Proceedings of the 32nd AAAI Conference on Artificial Intelligence, vol. 32, no. 1, pp. 3223–3230, 2018.

[18] T. Salimans, and R. Chen, “Learning montezuma’s revenge from a single demonstration,” in Proceedings of the 32nd Conference on Neural Information Processing Systems NIPS, Montréal, Canada, 2018.

[19] Y. Aytar, T. Pfaff, D. Budden, T. L. Paine, Z. Wang, et al., “Playing hard exploration games by watching YouTube,” in Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS), Montréal, Canada, 2018.

[20] T. Pohlen, B. Piot, T. Hester, M. G. Azar, D. Horgan, et al., “Observe and look further: Achieving consistent performance on atari,” arXiv preprint arXiv:1805.11593, 2018.

[21] P. Abbeel, and A. Ng, “Apprenticeship learning via inverse reinforcement learning,” In ICML, 2004.

[22] D. Ramachandran, and E. Amir, “Bayesian inverse reinforcement learning,” In IJCAI, 2007.

[23] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey, “Maximum entropy inverse reinforcement learning,” In Proc. of the 23rd AAAI Conf. on Artificial Intelligence, California, USA, vol. 8, pp. 1433–1438, 2008.

[24] E. Uchibe, “Model-free deep inverse reinforcement learning by logistic regression,” Neural Processing Letters, vol. 47, no. 3, pp. 891–905, 2018.

[25] A. Tucker, A. Gleave, and S. Russell, “Inverse reinforcement learning for video games,” In Proceedings of the Workshop on Deep Reinforcement Learning at NeurIPS, 2018.

# Pothole Detection Using Optical Camera Communication

Md. Osman Ali<sup>1,2</sup>, Israt Jahan<sup>1,3</sup>, Raihan Bin Mofidul<sup>1</sup>, ByungDeok Chung<sup>4</sup>, and Yeong Min Jang<sup>1</sup>

<sup>1</sup>Department of Electronics Engineering, Kookmin University, Seoul 02707, Korea

<sup>2</sup>Dept. of EEE, Noakhali Science and Technology University, Noakhali 3814, Bangladesh

<sup>3</sup>Dept. of EEE, Daffodil International University, Dhaka 1341, Bangladesh

<sup>4</sup>ENS. Co. Ltd., Ansan 15655, South Korea

Email: osman@kookmin.ac.kr; israt@kookmin.ac.kr; raihanbinmofidul@gmail.com; bdchung@ens-km.co.kr; yjang@kookmin.ac.kr

**Abstract**—Optical camera communication (OCC) is a potential candidate for the commercial deployment of vehicular communication. Recently, researchers have focused on the development of an OCC-based advanced driver assistance system. In this paper, we have proposed a pothole detection and road banking angle estimation technique from the rear LED shapes of the forwarding vehicle using OCC to ensure safe and comfortable driving. Mathematical approaches, as well as neural network-based methods, have been developed to provide highly accurate results. The proposed system is also applicable to detect stuck vehicles in icy conditions.

**Index Terms**—optical camera communication (OCC), advanced driver assistance system (ADAS), V2X communication, internet of vehicles (IoV)

## I. INTRODUCTION

Every year a significant number of people die and get injured in road accidents worldwide. The distraction of drivers is considered one of the key factors that cause accidents. Drivers can be assisted by providing necessary driving instructions based on the surrounding environments to reduce road accidents. At first, vehicle navigation systems drew attention in the late 1960s in the United States. The related primary goals included reducing highway congestion, increasing fuel efficiency, guiding routes, avoiding vehicle collisions, and collecting tolls electronically [1]. Nowadays, advanced driver-assistance systems are being extensively researched to reduce the number of casualties due to road accidents. Vehicle positioning and vehicle to vehicle (V2V) communication have immense potential to reduce the number of road accidents, making it possible to save the lives of a significant number of people by providing nearby vehicular position information to drivers [2]. In particular, an intelligent transportation system requires precise vehicle positioning to ensure a safe braking distance from surrounding vehicles. The primary focus of the researchers was on radio-frequency (RF) technology over the past decade. The selection of appropriate technology is a challenging task as the vehicular density is high in metropolitan areas. Using RF-based technologies, the system performance is expected to degrade owing to the huge amount of electromagnetic interference. Moreover, regular long-term driving can lead to adverse effects on drivers' health [3].

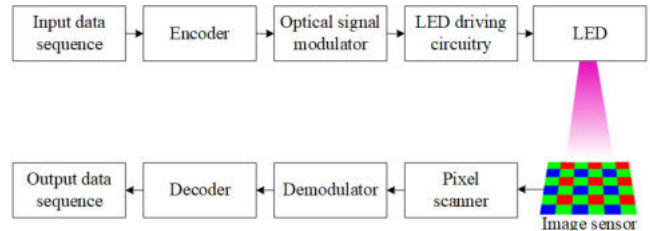


Fig. 1: Block diagram of a basic OCC system.

In addition, the number of connected devices in 5G is growing exponentially, which is supposed to become a more complex scenario in 6G. Therefore, an alternative of RF-based vehicle to everything system is under research as the RF spectrum is overcrowded and highly regulated. Currently, the unregulated massive optical spectrum (10 nm-1 mm) is considered a promising complement of RF technology to support the increasing demand for mobile data. Moreover, high security, immunity from interference, and high energy efficiency have brought unprecedented research attraction in this field [4], [5].

Recently, the research in optical camera communication (OCC) has attracted significant attention as most modern vehicles are equipped with one or multiple cameras for parking assistance and blind-spot monitoring, these cameras can also be used as receivers whereas the day time running light of vehicles can be modulated flicker freely to transmit data through the optical channel [6]. Therefore, OCC can be commercially deployed in the vehicles without adding too much cost to the existing system. Other key advantages of OCC technology include nearly interference-free communication as each pixel can be processed individually and an unlicensed spectrum that can be used as a complement for the nearly saturated RF spectrum. The basic operation of an OCC system is shown in Fig. 1 [7]. In a V2V communication scenario, the signal-to-noise ratio is high, as the light-emitting-diodes (LEDs) used for lighting have very high luminance. It offers a very strong line-of-sight link set up at a long distance with a low bit error rate. Additionally, the effect of sunlight, a major

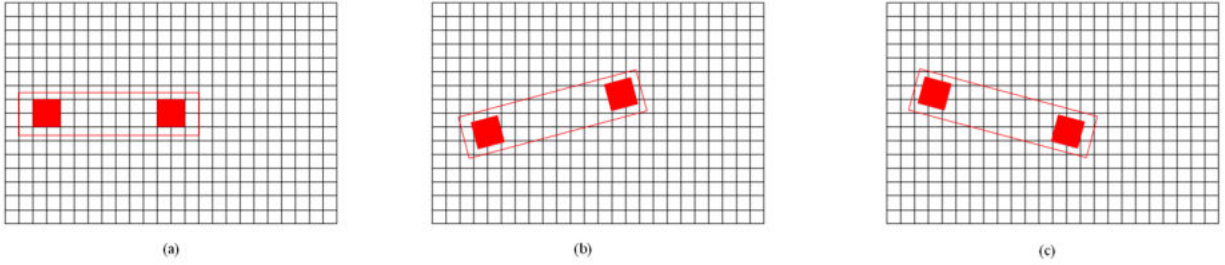


Fig. 2: Alignment of the rear LEDs when the FWV goes over a a) plain surface, b) pothole under the left wheel, and c) pothole under the right wheel.

challenge for other optical wireless communication systems in outdoor environments, can be effectively eliminated. Despite the high-speed switching capability of LEDs, the data rate of OCC systems is predominantly limited by the camera frame rate. Therefore, the significant advantages of OCC are often overshadowed by its low data rate [4]. Researchers have already proposed a high-speed camera and multiple-input multiple-output technique using an LED array to increase the data rate to one suitable for sensor data monitoring [8], patient monitoring [9], vehicular communication [10], and other low-rate indoor and outdoor applications [11], [12]. OCC can support numerous applications including platooning, emergency brake light detection, collision avoidance, traffic light recognition, and intersection assistance. Inter-distance between vehicles can also be estimated using OCC. Thus, it can be used to maintain a safe distance between vehicles [2], [10], [13]. Information on traffic accidents, road repair works, and traffic flow can be relayed from V2V to ensure safety and comfortable driving. Moreover, the front view of the forwarding vehicle (FWV) can be perceived by the following vehicle (FLV) using video streaming to enable the see-through feature [4]. In our previous work, we had developed a road curvature estimation technique from the rear LED shapes of the FWV using OCC to reduce the accidents at the road bendings [2]. In this work, we have designed a system that can measure the angle between the two rear LEDs of a vehicle so that potholes and vehicles stuck in the ice can be detected. Additionally, the road banking angle can also be estimated from the rear LED positions of the FWV.

The remainder of this paper is organized into the following sections. Section II presents the methodology where the overall architecture, dataset preparation, and mathematical approach are described. In the next section, the results are discussed. Then, the conclusion and future work are mentioned in Section IV.

## II. METHODOLOGY

### A. Overall Architecture

The 'x' coordinates of the LEDs vary with the lateral movement of the FWVs, FLVs, or both in the captured image in the FLV's camera receiver keeping the 'y' coordinate of the LEDs the same as shown in Fig. 2(a). If one of the

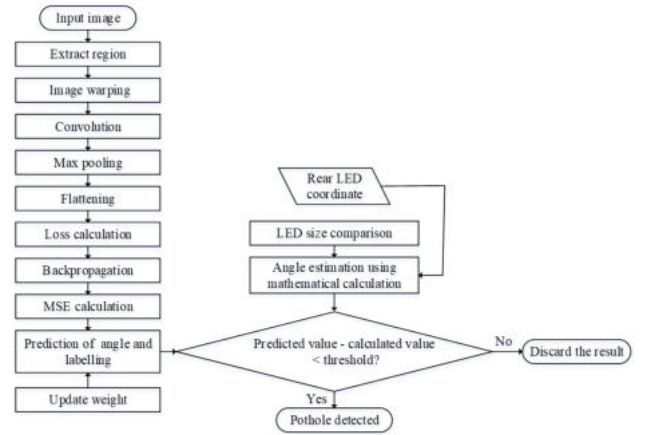


Fig. 3: Flowchart of the pothole detection and road banking angle calculation using OCC.

rear wheels runs over a pothole, the vertical coordinates of the rear LEDs of the FWV will no longer remain the same which is shown in Fig. 2(b) and Fig. 2(c). The flowchart of pothole detection and measurement using OCC is shown in Fig. 3. The deep learning-based method is more accurate than the mathematical approach. Both methods are compared to provide a more accurate result. If there is any discrepancy between the mathematical method and the neural network (NN)-based method, it is tolerated up to a certain level and the result of the deep learning-based approach is shown as output. However, if the difference is more than the threshold (for instance  $5^\circ$ ), that is removed from the calculation. It can be noted that this process doesn't affect the overall system performance. For instance, a 30 frame rate per second (fps) camera captures 30 images in one second and each image will provide one result. Therefore, if a high frame rate camera is used, the number of results obtained in one second will be equal to the frame rate of the camera. As a result, if some of the results are discarded, it will not create any significant performance degradation. The NN-based approach and mathematical approach are described in the following subsections.

## B. Dataset Preparation

5,000 images were taken at various angular positions with their associated angles. The dataset had a dimension of  $5,000 \times 6$  where the input and output vectors had the dimension of  $5,000 \times 5$  and  $5,000 \times 1$ , respectively. In a real-time scenario, numerous lights can be present in an image where most of the light sources are unwanted and can be considered as interfering light sources. Again, processing all the pixels of an image is unnecessary and time-consuming. Therefore, a convolutional NN (CNN) including two hidden layers was designed for image classification. The individual rear LED pair of vehicles were warped as individual images and a convolutional filter of  $4 \times 4$  was used for feature extraction. Then, max-pooling was performed with a  $2 \times 2$  filter to reduce the image size more. Then, it is flattened so that it can be used as the input of the NN.

## C. Mathematical Approach

When the vehicle goes over a pothole or got stuck in ice or there is road banking, the 'y' coordinates of the rear LEDs will be dissimilar. A hypothetical triangle can be formed which is shown in Fig. 4(a) and Fig. 4(b). The angle between the rear LEDs can be calculated from these triangles which require prior knowledge of the actual distance between the rear LEDs. This information is transmitted from the rear LEDs of the FWV to other FLVs using OCC. And, to calculate the apparent distance between the rear LEDs, the distance for each pixel is calculated as the distance between the rear LEDs is known. Then, the angle can be calculated using the cosine function as follows

$$\text{Angle between LEDs, } \theta = \cos^{-1} \left( \frac{\text{base}}{\text{hypotenuse}} \right), \quad (1)$$

$$\theta = \cos^{-1} \left( \frac{\text{apparent dist. between rear LEDs}}{\text{actual distance between rear LEDs}} \right). \quad (2)$$

TABLE I: Implementation parameters.

	Parameter	Value
Camera	Image resolution	$1080 \times 1920$
	Camera frame rate	30 fps
	Exposure time	2.5 ms
Backlight prototype	Distance between two LEDs	10 cm
	LED size	3 mm red LED
	Maximum distance	50 m

## III. RESULT AND DISCUSSION

The performance of our proposed scheme was verified experimentally. On the transmitter side, there were two parts such as the vehicle taillight prototype and an Arduino UNO to operate the LEDs. On the receiver side, a rolling shutter (RS) camera was used to capture the images and Python 3.7 was

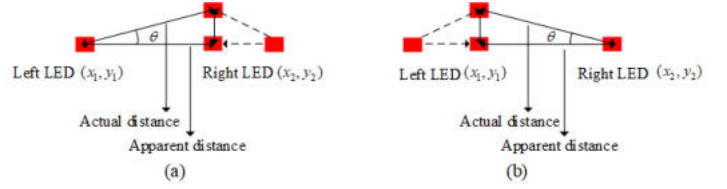


Fig. 4: Hypothetical triangle for calculation of pothole angle when the FWV goes over a pothole under the (a) left wheel and (b) right wheel.

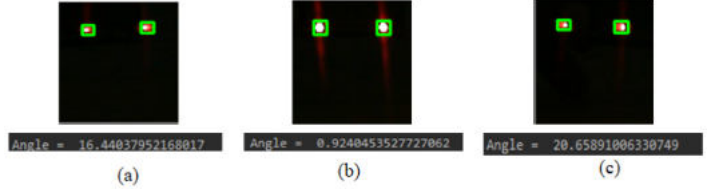


Fig. 5: A demonstration of the estimation of the angle between rear LEDs when the FWV goes over (a) a pothole under the left wheel, (b) pothole free road, and (c) a pothole under the left wheel.

used for LED detection and the rest of the processing. After LED detection, data was retrieved from the brightness of the stripes created on the image sensor due to the RS effect. And, the angle between LEDs was estimated from the rear LED coordinates of the FWV.

To evaluate the performance of our suggested system economically, we built a vehicle taillight prototype. The LEDs were connected with an Arduino UNO and DC power was supplied for the operation. It was implemented in an indoor environment where the sunlight and light from artificial sources were the interfering lights. Table I contains the implementation parameters. The pair was tilted to the left and right, and the results were recorded. The results were compared with the actual values to assess the performance of the proposed method. The threshold was set to  $5^\circ$  so that any discrepancy higher than this limit could be discarded. It was observed that the deviations remained below the predefined limit. A sample of the implementation results is shown in Fig. 5(a), Fig. 5(b), and Fig. 5(c).

## IV. CONCLUSION AND FUTURE WORK

This work adds a new feature in the OCC-based advanced driver assistance system (ADAS). The stuck vehicles in the ice, potholes, and road banking angles can be detected or estimated from the rear LED shapes of the FWVs using our proposed technique. Our proposed method is expected to assist the drivers even in adverse weather conditions, such as rainy, snowy, and foggy. Therefore, pothole detection can play a key role in safe and comfortable driving making OCC a promising candidate for vehicular communication because it not only provides communication but also ADAS features. The more the features will be developed, the higher the chance of commercial deployment of OCC-based ADAS. Therefore,

we will conduct our future research to connect the vehicles at roundabouts using OCC technology.

#### ACKNOWLEDGMENT

This work was supported by the Technology Development Program (S3098815) funded by the Ministry of SMEs and Startups (MSS, Korea).

#### REFERENCES

- [1] E. Abbott and D. Powell, "Land-vehicle navigation using GPS," *Proc. of the IEEE*, vol. 87, no. 1, pp. 145–162, 1999.
- [2] M. O. Ali et al., "Mono camera-based optical vehicular communication for an advanced driver assistance system," *Electronics (Basel)*, vol. 10, no. 13, p. 1564, 2021.
- [3] L. Rsa and M. Ga, "Effects of wireless devices on human body," *J. Comput. Sci. Syst. Biol.*, vol. 9, no. 4, 2016.
- [4] M. K. Hasan, M. O. Ali, M. H. Rahman, M. Z. Chowdhury, and Y. M. Jang, "Optical camera communication in vehicular applications: A review," *IEEE Trans. Intell. Transp. Syst.*, pp. 1–22, 2021.
- [5] M. H. Rahman, M. Shahjalal, M. K. Hasan, M. O. Ali, and Y. M. Jang, "Design of an SVM classifier assisted intelligent receiver for reliable optical camera communication," *Sensors (Basel)*, vol. 21, no. 13, p. 4283, 2021.
- [6] M. O. Ali, M. F. Ahmed, M. Shahjalal, M. H. Rahman, and Y. M. Jang, "Current challenges in optical vehicular modulation techniques," in *Proc. 2021 International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju Island, Korea (South), Oct. 2021, pp. 801–804.
- [7] M. O. Ali, M. M. Alam, M. F. Ahmed, and Y. M. Jang, "A new smart-meter data monitoring system based on optical camera communication," in *2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC)*, Jeju Island, Korea (South), Apr. 2021, pp. 477–479.
- [8] M. F. Ahmed, M. K. Hasan, M. Shahjalal, M. M. Alam, and Y. M. Jang, "Experimental demonstration of continuous sensor data monitoring using neural network-based optical camera communications," *IEEE Photonics J.*, vol. 12, no. 5, pp. 1–11, 2020.
- [9] M. F. Ahmed, M. K. Hasan, M. Shahjalal, M. M. Alam, and Y. M. Jang, "Design and implementation of an OCC-based real-time heart rate and pulse-oxygen saturation monitoring system," *IEEE Access*, vol. 8, pp. 198740–198747, 2020.
- [10] M. T. Hossan et al., "A new vehicle localization scheme based on combined optical camera communication and photogrammetry," *Mob. Inf. Syst.*, vol. 2018, pp. 1–14, 2018.
- [11] M. Shahjalal, M. T. Hossan, M. K. Hasan, M. Z. Chowdhury, N. T. Le, and Y. M. Jang, "An implementation approach and performance analysis of image sensor based multilateral indoor localization and navigation system," *Wirel. Commun. Mob. Comput.*, vol. 2018, pp. 1–13, 2018.
- [12] M. T. Hossan, M. Z. Chowdhury, A. Islam, and Y. M. Jang, "A novel indoor mobile localization system based on optical camera communication," *Wirel. Commun. Mob. Comput.*, vol. 2018, pp. 1–17, 2018.
- [13] M. S. Iftikhar, N. Saha, and Y. M. Jang, "Stereo-vision-based cooperative-vehicle positioning using OCC and neural networks," *Opt. Commun.*, vol. 352, pp. 166–180, 2015.

# Sensor Network System for Condition Detection of Harmful Animals by Step-by-step Interlocking of Various Sensors

Keigo UCHIYAMA

Ritsumeikan University  
Graduate School of Information  
Science and Engineering  
Shiga, Japan  
is0398vf@ed.ritsumei.ac.jp

Hiroshi YAMAMOTO

Ritsumeikan University  
Department of Information  
Science and Engineering  
Shiga, Japan

Eiji UTSUNOMIYA

KDDI Research, Inc.  
Saitama, Japan

Kiyohito YOSHIHARA

KDDI Research, Inc.  
Saitama, Japan

**Abstract**—In recent years, damage to crops and injuries to humans by harmful birds and animals have increased in areas of Japan. In order to support the coexistence of humans and wildlife, technologies for observing the ecology of wildlife and detecting harmful birds and animals are attracting attention. In our previous study, a method of detecting animals by utilizing a radio beacon has been proposed. In the system, the reception signal strength of radio waves transmitted between radio beacon devices is continuously measured and analyzed to identify the existence of the wildlife near the devices by using machine learning technology. However, the existing method of analyzing the radio wave strength cannot estimate the posture of the animal which can be utilized to identify the detailed behavior of the animals (e.g., the point where feeding behavior was taken) and to understand the situation of damage caused by them in more detail. Therefore, in this study, we propose a new sensor network system that detects the presence of wild animals by analyzing images taken by the thermal camera that can measure the body temperature of the organism day and night. In addition, in order to reduce the power consumption of the entire system that is assumed to be installed in mountain areas where the power supply is difficult, the thermal camera is activated only when the doppler sensor with low power consumption detects the moving object. After that, by analyzing the captured images using a machine learning technology, the system attempts to estimate not only the type and the number but also the posture of the animals.

**Index Terms**—IoT, wild animals damage prevention, machine learning, doppler sensor, thermal camera

## I. INTRODUCTION

In recent years, damage to crops caused by harmful birds and animals (e.g., wild boars, deer) becomes a serious problem. Although the total amount of damage is decreasing year by year, it still exceeds 15 billion yen per year [1]. As an example, Figure 1 shows the total amount of annual damage to crops caused by harmful animals nationwide, as published by the Ministry of Agriculture, Forestry and Fisheries, Japan. In order to reduce the damage to crops caused by such harmful birds and animals, it is necessary to monitor their habitats and the number in real-time and to detect their approaching to the human living areas in advance. However, since it is

difficult to prepare electricity in mountainous areas where harmful animals live, a system that consumes little power and can monitor harmful birds and animals for a long time is required. On the other hand, countermeasures to the harmful birds and animals such as the capture and extermination of harmful birds and animals by hunters and the installation of preventive fences have been taken, but require hunters to patrol the condition of the equipment deployed in a wide area which results in high cost.

Therefore, existing studies have developed a system that can automatically detect harmful birds and animals and can confirm their location via the Internet [2] [3]. In the system, images taken by the camera are sent to the cloud on the Internet and are analyzed to detect the wild animals on the images. However, in order to send the images with large data size to the server on the Internet and to detect the wild animals in real-time, a high-speed communication network should be prepared, hence the location where the system can be deployed is limited. In addition, the system should be able to work even at night when the wild animals frequently appear [4].

Therefore, in this study, we propose a new sensor network system that can estimate not only the presence of the wild animals but also their species and posture by analyzing images

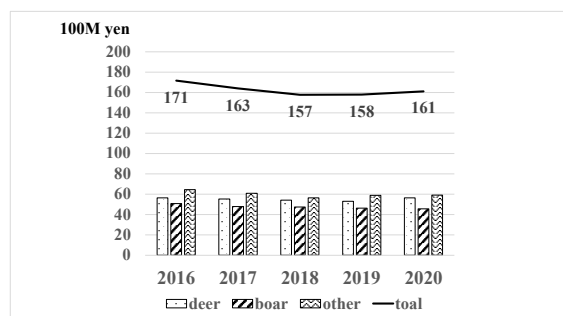


Fig. 1. Total amount of damage to crops caused by harmful animals and birds (from 2016 to 2020).

taken by a thermal camera that can measure the body temperature of living harmful animals day and night. The proposed system adopts an edge computing-based structure where a small computer installed in the field analyzes images taken by the thermal camera to estimate the type and posture of wild animals and sends only the results to the analysis server on the Internet. The system structure can greatly reduce the amount of data sent to the analysis server, making it possible to realize a system that can work even in a mountainous area where the high-speed communication network is not available. In addition, in order to ensure long-term operation even when the camera is installed in mountainous areas where power supply is difficult, the system is configured to activate the thermal camera only when the sensor with low power consumption detects moving objects, thereby realizing overall power saving of the system. Furthermore, this study considers a system for detecting the status of harmful birds and animals using 3D LiDAR which can accurately measure the distance, position, and shape of the target object.

## II. RELATED WORKS AND OBJECTIVES OF OUR STUDY

### A. Harmful wildlife detection system utilizing deep learning for radio wave sensing

In previous research by Ogami et al. (2018), a system for detecting harmful animals using radio wave sensing is proposed and developed [5]. The proposed system consists of a device that transmits and receives the radio beacons in multiple frequency bands (2.4 GHz, 920 MHz, and 429 MHz) that have different characteristics of reflection and diffraction between each other. This device is capable of measuring the received signal strength of the radio wave transmitted in multiple frequency bands. By analyzing the measured signal strength of radio waves using machine learning techniques, the system can capture the features related to the size and shape of the wild animals passing between transmitter/receiver devices and can estimate their species and the number. However, this method cannot estimate the posture of the animal that can be used to identify the detailed behavior of the target (e.g., feeding).

### B. Damage prevention systems by wild animals using image analysis

In the existing study by Kamesaka et al. (2018), a system for capturing harmful animals using image analysis and machine learning has been proposed [6]. In this proposed system, an RGB camera attached to a cage that lures a group of monkeys takes images and uploads them to the server. And then, the server analyzes the images by utilizing machine learning to determine whether a group of monkeys exist in the image. When a group of monkeys is detected, the person in charge operates the door of the cage via a web browser to capture them. However, since the system uses images with relatively large data size, it requires a high-speed communication network to transmit the data to the server via the Internet to detect the wildlife in real-time. Therefore, the system cannot be used in mountainous areas where wild animals live, but the cellular

network such as 4G/LTE is not sufficiently available for the system. Furthermore, since it is difficult to prepare the power supply of the system in the area, the power consumption of the device that detects harmful animals should be reduced so that the system becomes available for a long time.

### C. Objectives of our research

In this study, we propose and develop a harmful birds and animals condition detection system that can estimate not only the presence of wild animals but also their species and posture by analyzing images taken by a thermal camera that can measure the body temperature of living harmful animals day and night. The proposed system adopts a system structure of edge computing where images taken by the thermal camera are analyzed on a small computer installed in the field to estimate the type and posture of wild animals. The computer transmits only the result of the analysis to the server on the Internet so that the system can be used even in mountainous areas where a high-speed communication environment is not available. In addition, to ensure that the system can operate for a long time even when it is installed in mountainous areas where power supply is difficult, the system is configured so that the thermal camera with high power consumption is activated only when the doppler sensor with low power consumption detects moving objects.

## III. PROPOSED HARMFUL ANIMAL DETECTION SYSTEM

### A. Overview of the proposed system

The overall picture of the system proposed in this study is shown in Fig. 2. As shown in this figure, the proposed system consists of sensor nodes for detecting harmful birds and animals installed in the field and an analysis server deployed on the Internet. In order to save power consumption of the entire system, the sensor node activates the thermal camera for observing the detailed behavior of the target via a relay circuit only when detecting the approach or separation of the object by the doppler sensor. The sensor node then analyzes the image captured by the thermal camera using machine learning technology, estimates the type, number, and posture of the animals present in the image, and transmits the results to the analysis server. The analysis server accumulates the estimation results, visualizes the status of wildlife using a map application, and sends the e-mail to notify users (e.g., hunters, staff of local governments) of the appearance of wildlife.

### B. Device/function configuration of the proposed system

In this section, we describe the device and functional configuration of the sensor nodes and the analysis server computing the proposed system.

1) *Configuration of sensor nodes*: The sensor node is responsible for observing and analyzing data related to the ecology of the wildlife and consists of three devices divided by a function. The first device is a microcontroller that has the function of detecting the approach and departure of wild animals and controlling activation of the second device through relay circuits. In our proposed system, the Arduino



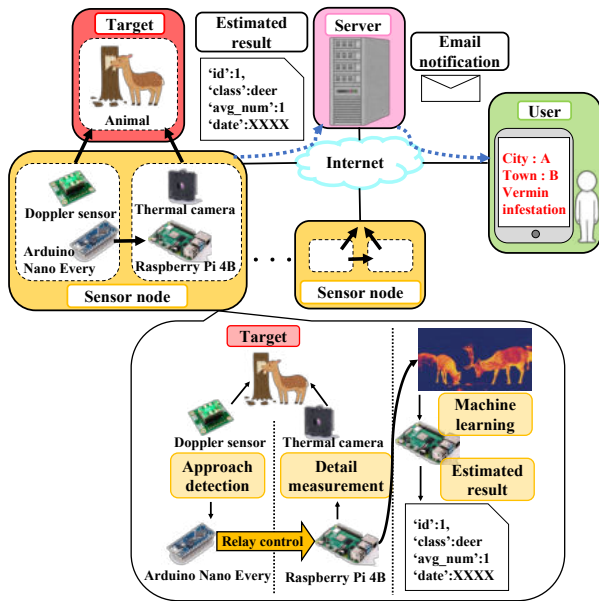


Fig. 2. Overview of proposed system.

Nano Every which is a power-saving microcomputer with a built-in CPU, memory, and an interface to accommodate a variety of sensors are used as the microcontroller. The microcontroller is connected to a doppler sensor for detecting the approach and separation of objects, and a relay circuit for turning on and off the electrical device. A 24 GHz microwave sensor module, NJR4265 J1, is used as the doppler sensor, and a solid-state relay which is a type of non-contact relay is used as the relay circuit. The doppler sensor has a detection distance of 10 m, a detection angle of 70° horizontally and 54° vertically and current consumption of 60 mA.

The second device is a small computer that controls a thermal camera to capture temperature images and analyzes them to identify objects in the images utilizing machine learning. In our proposed system, the small computer is a Raspberry Pi 4B that is a small single-board computer operated on a Linux-based OS and is configured to wake up with 5V power supplied when a relay circuit is connected to the microcontroller is activated. This small computer is connected to a thermal camera for photographing wild animals detected by the doppler sensor. In this study, the thermal camera is the ultra-compact Lepton 3.5 which is capable of capturing temperature images with a resolution of 160×120 pixels and is connected to a small computer via USB using an interface board, PureThermal 2.

The last one is a microcontroller for wireless communication (MAX32630FTHR) that controls the LTE-M (Long Term Evolution for machine-type-communication) module to send analysis results to the analysis server. The MAX32630FTHR is a platform for developing embedded devices that are equipped with a microSD card connector and 6-axis acceleration and gyro sensors. The proposed system assumes the use of LTE-M, a kind of LPWA (Low-Power Wide-Area), which enables

wide-area wireless communication with low power consumption, although the communication speed is slow. The LTE-M module of the proposed system is KYW01 which also has the function of positioning using GPS, and is connected to the microcontroller using a dedicated interface board [7]. Through the microcontroller for communication, the sensor data measured by the sensor node is sent to the analysis server on the Internet.

2) *Configuration of analysis server:* In this system, the analysis server is assumed to be a commercial PC, and a Mac Mini (2018) is used in this study. Here, all the functions of the analysis server are implemented using Python, and Elasticsearch is used as the database to store the data to be analyzed. Elasticsearch is a database supporting a full-text search engine developed by Elastic, which is suitable for real-time analysis of the time-series data. In addition, Kibana is used for a visualization of the data stored on Elasticsearch. The program in the analysis server stores the data received from the sensor node in folders created each date (year/month/day), determines the location of the origin of the data by referring to pre-registered location information of the sensor nodes by the sensor ID recorded in the received data, and stores the received data with the location information of the sensor node in the database in real-time.

### C. Linkage method of sensor devices in sensor nodes

In order to achieve overall power saving of the proposed system, the sensor device with high power consumption is activated only when the approaching of wild animals is detected by the sensor device with low power consumption, and the data used for estimating the type and the posture of the wild animal is acquired. In the sensor node in the proposed system, the microcontroller first receives the “approach” or “separation” signal from a doppler sensor, and when the type of received signal is “approach”, it activates the small computer by controlling a relay circuit. Next, the activated small computer estimates the type and posture of the approaching wildlife by applying the image analysis process described in the next section to the temperature images that are obtained from the connected thermal camera.

### D. Analysis method of temperature image in sensor node

In this research, LTE-M which is a type of LPWA is used as the communication method for sending data from sensor nodes to the analysis server, hence it is difficult to send large amounts of data such as images in real-time. Therefore, the proposed system adopts a system structure of edge computing configuration where the temperature images obtained from a thermal camera are analyzed in the sensor node, and only the estimated results of the type and the posture of the wild animal are sent to the analysis server. The data to be sent to the analysis server is summarized in Tab. I. As shown in this table, the data consists of the sensor ID to identify the location where the sensor node is installed, the estimated type/posture/the average number of wildlife, and the time when the wildlife is detected. In the following parts of this section, Section III-D1

TABLE I  
FORMAT OF DATA SENT FROM THE SENSOR NODE TO THE ANALYSIS SERVER.

Data name	Type	Example
Sensor ID	char	01
Estimated class	char	human_front
Average number of animals	float	2.2
Acquisition time (UNIX)	char	1605364660851.75

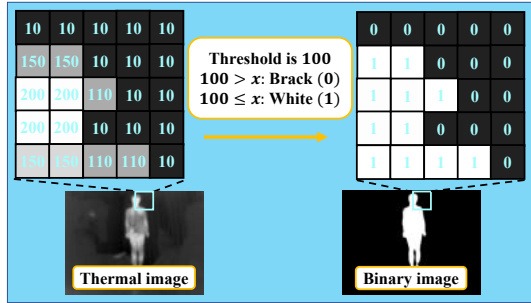


Fig. 3. Binarization method to convert thermal image to binary image (threshold value is 100).

describes the detailed process of extracting the contours of the wildlife from the temperature image as a pre-processing for machine learning, and Section III-D2 describes the detailed process of estimating the type and posture of the wildlife by the machine learning.

1) *Binarization process for thermal images*: As a pre-processing step for machine learning, a binarization is applied to the temperature images captured using the thermal camera to emphasize the contours of the wild animals in order to reduce noises on the temperature image and to focus on the part of the wildlife on the image for the analysis. The procedure for generating a binary image from the temperature image is shown in Fig. 3. As shown in this figure, a threshold value for the binarization is predetermined. If the grayscale value corresponding with the temperature in the pixel is smaller than the threshold, the value of the pixel is set to 0. If the value is greater than the threshold value, the value of the pixel is set to 1. In the binary image in Fig. 3, a pixel with a value of 0 is black, and a pixel with a value of 1 is white. In the proposed system, the different thresholds are selected among the different environments (i.e., 90 for indoors and 170 for outdoors) because the brightness that affects the grayscale value varies depending on the environment.

2) *Estimation process of the type and posture of wildlife using machine learning*: The small computer is a component of the sensor node that estimates the type and posture of the wildlife in the temperature image by applying the machine learning technology to the binary image generated by the process described in the previous section. In our proposed system, we use YOLO which is one of the object detection models using deep learning as a machine learning technology [8]. By inputting a binary image to the pre-built learning model

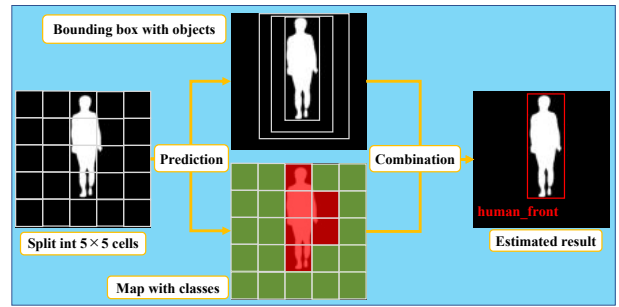


Fig. 4. Method for estimating animals by machine learning.

of YOLO, the location and the existing probability of each type of object in the image are estimated. Since we have not obtained temperature images of actual wildlife enough to construct the learning model yet, the analysis target of the proposed system is a human that is a kind of organism as shown in Fig. 4. As shown in this figure, the input image is firstly divided into  $5 \times 5$  cells. After that, the proposed method predicts the bounding rectangles where the object may exist and the map where the object may exist. Finally, it combines the bounding rectangles with the map and outputs the estimation result.

After waking up, the small computer starts the process of estimating the type and posture of wildlife every 10 seconds. First, all the binary images captured and generated during 10 seconds are input to the machine learning model to obtain the result in each image. And then, a majority vote is taken on all the estimation results based on the images, and the type, posture, and the average number of the most numerous wild animals are calculated. Finally, the results are sent to the analysis server.

#### IV. PRELIMINARY EXPERIMENTS TO DERIVE APPROPRIATE SETTINGS FOR MACHINE LEARNING

In this section, we describe the types of machine learning technology (YOLO) used in this study, and preliminary experiments to study the appropriate combination of libraries for implementing machine learning. In this experiment, we assume that the target organism is one type of human being and build a machine learning model to estimate three different postures to the camera: “facing front/facing right/facing left”.

There are two types of YOLO: version 3 (YOLOv3) which runs stably on the Raspberry Pi 4B, and YOLOv3-Tiny a lighter version. Each type is used together with TensorFlow which is a library for deep learning provided by Google or Keras which is an open-source library for building neural networks. In addition, TensorFlow version 2 (hereinafter, referred to as TF2) is used because YOLOv3 and YOLOv3-Tiny may not work with TensorFlow version 1.

##### A. Impact of the type of machine learning library

In order to clarify which library (TF2 or Keras) of the machine learning is appropriate for our proposed system, we evaluate the time taken to estimate and the estimation accuracy

TABLE II  
YOLOv3 + TF2 vs. YOLOv3 + KERAS

Pair	Detection time (sec)	Accuracy rate
YOLOv3 + TF2	31	1.0
YOLOv3 + Keras	43	1.0

TABLE III  
YOLOv3 + TF2 vs. YOLOv3-TINY + TF2

Method name	Detection time (sec)	Accuracy rate
YOLOv3 + TF2	31	1.0
YOLOv3-Tiny + TF2	0.7	1.0

for the combination of YOLOv3 and TF2, and the combination of YOLOv3 and Keras. In this evaluation, 32 images with the correct label of human\_front (human is facing front) are used as training data to build the learning model. For the constructed learning model, 10 images that are different from the training data are input to the learning model to estimate the posture of the target, and the average time required for estimation and the percentage of accuracy is estimated.

The experimental results are shown in Tab. II. From this table, we can see that the accuracy rate is 1.0 for all combinations but the time taken to estimate is 31 seconds for the combination of YOLOv3 and TF2 which is faster than the combination of YOLOv3 and Keras. Based on the result, TF2 is used as the machine learning library in the subsequent experiments.

### B. Impact of the version of YOLO

In order to clarify which version of YOLO is appropriate for the proposed method, we evaluate the performance in case that each version of YOLO (i.e., YOLOv3 or YOLOv3-Tiny) is used. The procedure for constructing and evaluating the learning model is the same as that in Section IV-A.

The experimental results are shown in Tab. III. From this table, we can see that the correct answer rate is 1.0 for both versions of YOLO, but the required time for YOLOv3-Tiny is about 44 times faster than that for YOLOv3. In general, the YOLOv3-Tiny is known to have lower detection accuracy than YOLOv3, but there is no degradation in estimation accuracy when using YOLOv3-Tiny because of the binarization preprocessing can emphasize the contours of the target. In addition, it is necessary to consider the improvement of estimation speed by using YOLOv3-Tiny rather than the estimation accuracy by using YOLOv3 because the proposed system needs to notify the administrator of the appearance of harmful birds and animals in real-time. Therefore, the YOLOv3-Tiny is adopted as the version of YOLO in the following evaluation in this study.

## V. DEMONSTRATION EXPERIMENT TO EVALUATE THE EFFECTIVENESS OF THE PROPOSED SYSTEM

In order to clarify the effectiveness of the proposed system, we evaluate the power consumption of the sensor nodes and

TABLE IV  
POWER CONSUMPTION OF EACH DEVICE

Device name	Electric current	Power consumption
Arduino Nano Every + NJR4265 J1 (Always-on startup)	0.104 A	0.52 W
Raspberry Pi 4B (Reading and writing images)	1.13 A	5.65 W
Raspberry Pi 4B (During object estimation)	1.40 A	7 W

the performance of detecting the type and posture of wildlife using machine learning.

### A. Evaluation of power consumption of sensor nodes

In this experiment, we measure the power constantly consumed by the microcontroller with the doppler sensor, as well as the power consumed by the small computer when the process of estimating the type and posture of wild animals. Based on the measurement result of the power consumption, the maximum time that the proposed system can operate continuously when using a battery with a capacity of 12 Ah is estimated.

The experimental results are shown in Tab. IV. As shown in this table, the current consumed by the microcontroller with a doppler sensor is 0.104 A. Based on the evaluation results of the current consumption, the proposed system can operate for only 60 hours even if the small computer is periodically activated three times per hour when using a battery with a capacity of 12 Ah. In the previous study, the wireless beacon device is used to detect the approaching of the moving object [5] and the power consumption of the device is 10 ~ 20 mA which is much smaller than the combination of microcontroller and doppler sensor used in the proposed system. This is mainly due to the fact that the current consumption of the doppler sensor is 60 to 70 mA.

### B. Evaluation of the performance of machine learning for object estimation

In this section, we evaluate the performance of the machine learning model described in Section IV. At present, it is difficult to install the prototype system in an actual field where wild animals live (e.g., mountainous areas). Therefore, we conduct an experimental evaluation that the detection target is a human on the campus of Ritsumeikan University, Japan.

The experimental results are shown in Tabs. V and VI. From these tables, it can be seen that the proposed system can estimate that the posture of the human with an accuracy of about 94% when the distance between the thermal camera and the subject is 3 m. However, when the distance between the thermal camera and the subject is 5 m, the accuracy decreases to about 45%. This is because, with the increase in the distance, it is more difficult to obtain the correct temperature of the surface of the subject. Therefore, in order to accurately estimate the type and posture of wildlife by the proposed system, it is necessary to build the learning model for each specific distance range.

TABLE V  
DISTANCE IS 3 M

Label of estimation \ Label of correct	human_front	human_right	human_left
human_front	161	-	-
human_right	-	138	2
human_left	-	7	163

TABLE VI  
DISTANCE IS 5 M

Label of estimation \ Label of correct	human_front	human_right	human_left
human_front	132	1	22
human_right	-	70	2
human_left	-	56	52

## VI. SUNSHINE HOURS REQUIRED FOR PERMANENT OPERATION OF THE NEW TYPE OF SENSOR NODES

Based on the measurement results in Section 5, we are considering a design of a new type of sensor node considering the microcontroller and the small computer. In the sensor node, the microcontroller detects the existence of the wildlife using the wireless beacon device to reduce the power consumption and sends a signal for activation once when detecting the wildlife. In addition, the small computer is equipped with the 3D LiDAR for observing the detailed shape of the target to identify the type and posture of the wildlife. In the proposed system, the microcontroller is an Adafruit Feather nRF52840 express, the 3D LiDAR is a Mid-70, and the small computer is a Jetson Nano.

When detecting the moving object in the vicinity, the microcontroller activates the small computer and the 3D LiDAR by transmitting a signal to the relay circuit. After that, the small computer wakes up from the suspend mode and acquires 3D point cloud data of the target object for 60 seconds from the activated 3D LiDAR.

In this experiment, we consider the sunshine hours required for the permanent operation of the sensor node consisting of the microcontroller and the small computer when using a power generation system consisting of a battery with a capacity of 50 Ah and a solar panel with a maximum output of 100 W. In the sensor node of this experiment, the microcontroller sends a signal for activation once an hour to the small computer and the relay circuit to which the 3D LiDAR is connected. After that, the small computer that has recovered from the suspend mode acquires 3D point cloud information of the target object from the activated 3D LiDAR. Here, it is assumed that the microcontroller detects the moving object and activates the small computer once per hour.

Here, the sunshine duration provided by the Japanese Meteorological Agency during the experiment in Otsu City, Shiga Prefecture, Japan, is shown in Fig. 5. As shown in this figure, if the total amount of sunlight in a week is 1127 minutes (about 18.9 hours), the system can operate semi-permanently because the amount of electricity generated by the power generation system exceeds the power consumption of the sensor nodes.

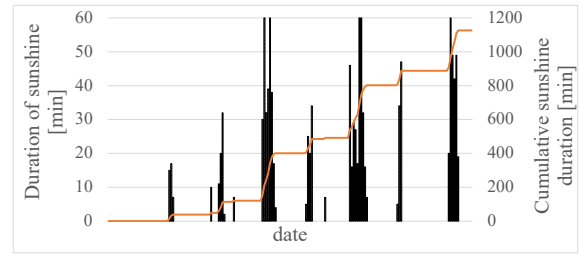


Fig. 5. Sunshine hours in Otsu City, Shiga Prefecture, Japan from July 8 to 15, 2021

## VII. CONCLUSION

In this study, we have proposed a new sensor network system that detects the presence of wild animals by analyzing images taken by the thermal camera. In addition, we have achieved power saving of the entire system by activating from the sensor with low power consumption to the sensor with high power consumption in stages. Furthermore, by analyzing the captured images taken by the thermal camera using a machine learning technology, we have shown that the system can distinguish not only the type and the number but also the posture of the animals. In the future, we will propose a new sensor network system that detects the presence of wild animals by analyzing point cloud data measured by the 3D LiDAR so as to accurately observe the type and posture of the wildlife. In addition, we will collect wild animals' data from sensor nodes installed in the field and analyze the data.

## REFERENCES

- [1] Ministry of Agriculture, Forestry and Fisheries of Japan, "Damage to crops caused by wild birds and animals in Japan", <https://www.maff.go.jp/j/press/nousin/tyozyu/attach/pdf/201223-2.pdf>, (accessed 2022-01-08).
- [2] Andy Rosales, "Where's The Bear? - Automating Wildlife Image Processing Using IoT and Edge Cloud", Internet-of-Things Design and Implementation (IoTDI), 2017 IEEE/ACM Second International Conference, April 2017.
- [3] Siddhanta Borah, R. Kumar, Subhradip Mukherjee, "Study of RTPPS algorithm in UWB communication medium for a surveillance system to protect agricultural crops from wild animals", 2020 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS), December 2020.
- [4] Keisuke ishikawa, "Invasion of mandarin orange fields by wild animals in the Mikkabi area, Shizuoka Prefecture", Bulletin of the Shizuoka Research Institute of Agriculture and Forestry, vol.10, pp.51-60, March 2017.
- [5] Ryota Ogami, Hiroshi Yamamoto, Takuya Kato, Eiji Utsunomiya, "Harmful Wildlife Detection System Utilizing Deep Learning for Radio Wave Sensing on Multiple Frequency Bands", 2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC), March 2019.
- [6] Ryoki Kamesaka, Yukinobu Hosino, "Prototype and study of animal damage prevention system for crops", Japan Society for Fuzzy Theory and Intelligent Informatics, The 34th Fuzzy System Symposium, FSS 2018 in Nagoya, September 2018.
- [7] Ministry of Internal Affairs and Communications in Japan, "Trends in wireless systems related to LPWA", [https://www.soumu.go.jp/main\\_content/000543715.pdf](https://www.soumu.go.jp/main_content/000543715.pdf), (accessed 2021-05-26).
- [8] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, "You Only Look Once: Unified, Real-Time Object Detection", The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.779-788, 2016.

# WiFi Positioning by Optimal k-NN in 3GPP Indoor Office Environment

Sung Hyun Oh  
Dept. Electronic Engineering  
Korea Polytechnic Univ.  
Siheung, Korea  
osh119@kpu.ac.kr

Jeong Gon Kim\*  
Dept. Electronic Engineering  
Korea Polytechnic Univ.  
Siheung, Korea  
jgkim@kpu.ac.kr

**Abstract**— The most important issue in the LBS(Location Based Service) industry is to accurately estimate the user's location to provide various location-based services. In the case of an outdoor environment, relatively high positioning accuracy may be provided through a GPS(Global Positioning System) or the like. However, the application of the GPS is limited in an indoor environment due to problems such as propagation loss. Therefore, in this paper, a technology for positioning a user using WiFi(Wireless Fidelity) communication applied to a general indoor environment is studied. First, a fingerprinting scheme that provides relatively high accuracy in combination with an RSSI(Received Signal Strength Indicator) is applied to perform user positioning. At this time, after arranging each RP(Reference Point) in the offline step, the RSSI value is measured to build a fingerprinting database. After, in the online step, the k-NN(k-Nearest Neighbor) algorithm, a technique of supervised learning, is applied by measuring the RSSI value of the fingerprinting database and the actual user's location. At this time, the initial search area of the PSO(Particle Swarm Optimization) algorithm is limited by deriving the closest RP from the actual user. After that, the particles are distributed in a limited area to finally determine the user's location. Through simulation, it can be confirmed that when k-NN and PSO are jointly used, improved positioning accuracy is obtained compared to the existing schemes.

**Keywords**—LBS(Location Based Service), Indoor Positioning, k-NN(k-Nearest Neighbor), PSO(Particle Swarm Optimization), Wi-Fi(Wireless Fidelity)

## I. INTRODUCTION

With the rapid development of mobile communication technology, the LBS(Location Based Service) industry is attracting attention. In general, LBS may be requested for personal or public purposes. The main examples are for customers to find the store they want by themselves in large and complex shopping malls, to support firefighters in case of a fire in a building, or to provide discount information at large marts. Conventionally, it is possible to provide relatively high positioning accuracy in an outdoor environment based on GPS (Global Positioning System) technology[1]. However, there is a limit to the application of GPS technology due to problems such as propagation loss due to the complicated radio wave environment in the indoor environment. Therefore, to solve this problem, technologies that provide high positioning accuracy based on communication technologies applicable in indoor environments such as WiFi(Wireless Fidelity), Bluetooth, and UWB (Ultra-Wide Band) became an important research subject[2].

In addition, technologies such as IoT(Internet of Things), Bigdata, and AI(Artificial Intelligence), which are the core

\*: Corresponding Author

technologies of the 4th industry, can be fused and applied to the LBS industry. Most people own a smartphone, an IoT device. Big data technology can store and use vast amounts of data. AI technology has the advantage of being able to quickly process complex calculations. When these three technologies are convergent applied, it is possible to provide each user with the optimal location accuracy in real-time.

As mentioned above, mobile communication technologies for indoor positioning that are currently generally used include WiFi, Bluetooth, UWB, and the like. And in the existing sensor positioning technology, there are techniques based on the range and methods that do not use the range. Among them, in general, most of the technology based on the range is applied. Among them, positioning technology based on RSSI(Received Signal Strength Indicator) is the most commonly used because it can obtain high positioning precision at a low cost when used in combination with a fingerprinting algorithm[3].

Existing studies related to these indoor positioning methods are as follows. In [4], the positioning accuracy was improved by limiting the MLE(Maximum Likelihood Estimation)-based PSO(Particle Swarm Optimization) scheme, and in [5], a method combining ANN(Artificial Neural Network) and PSO was proposed to estimate the user's location. In [6], a method of estimating the user's location through the re-sampling process using a particle filter was proposed, and in [7], a method that effectively converges RSSI fingerprinting and MF(Magnetic Field) fingerprinting was used. The above literatures proposed various approaches for indoor user positioning, but they did not consider both positioning accuracy and processing time at the same time.

Therefore, in this paper, we propose a method for estimating the user's location based on WiFi communication in the indoor environment suggested by 3GPP(Third Generation Partnership Project)[8]. The core of the proposed method is to effectively limit the initial search area of PSO through k-NN(k-Nearest Neighbor), an AI(Artificial Intelligence) technology. To this end, initially, RSSI values for a specific RP(Reference Point) are collected based on a fingerprinting technique. Then, k-NN techniques are applied between the RSSI value of the real user and the fingerprinting database value to derive k RPs closest to the user. The derived k points may be used to limit the initial search area of the PSO. At this time, we conduct the simulation by changing the k value and analyzing the positioning accuracy performance according to the k value.

The overall structure of this paper is as follows. Section 2 describes the system model. Section 3 describes the indoor positioning method proposed in this paper. Subsequent section 4 describes the parameter values and results used in the

simulation. Finally, Section 5 draws the conclusion of this paper.

## II. SYSTEM MODEL

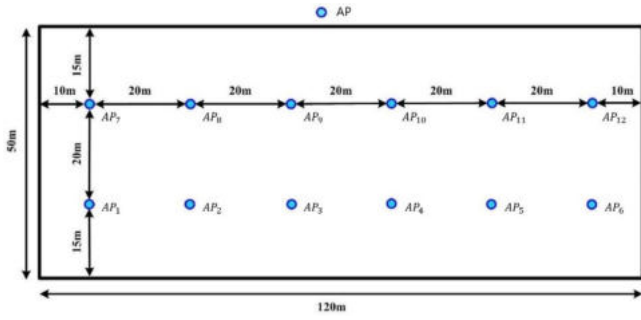


Fig. 1. Indoor environment suggested by 3GPP

The environment considered in this paper is shown in Fig.1[7]. Fig.1 shows the indoor environment suggested by 3GPP and has a size of  $120m \times 50m$ , with a total of 12 APs (Access Point) deployed. The distance between each AP is set to  $20m$ .

In the suggested environment, it communicates based on WiFi and locates the user by using the RSSI value between each AP and the UE (User Equipment). In this case, the RSSI value can be obtained as in (1) below.

$$RSSI_d = 10 \log(P_0) - 10 \log\left(\frac{d}{d_0}\right) + N \quad (1)$$

where,  $RSSI_d (dBm)$  and  $P_0 (dBm)$  are the received power between the AP and the user for each distance  $d$  and  $d_0$ .  $u$  is the path loss exponent, and  $N$  is the noise.

## III. PROPOSED POSITIONING SCHEME

The positioning method proposed in this paper is shown in Fig. 2. The proposed scheme locates the user's location by sequentially applying the fingerprinting scheme, k-NN algorithm, and PSO. At this time, the core idea is to limit the initial search area of PSO through a k-NN algorithm.

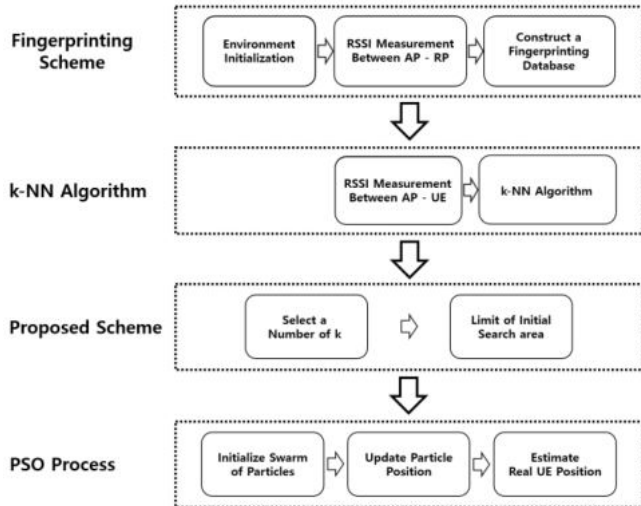


Fig. 2 Block diagram of the proposed scheme

As shown in Fig. 2, the proposed method uses three single algorithms fused. First, the fingerprinting scheme is performed in the offline step. The fingerprinting scheme measures the RSSI value from all APs in each RP. A

fingerprinting database is built based on the measured RSSI value. In the online step, RSSI between the UE and each AP is measured. The measured values apply the k-NN algorithm based on the fingerprinting database. By applying the k-NN algorithm, it is possible to derive k RPs closest to the UE. The derived k RPs are used to limit the initial search area of PSO. Limiting the initial search area of PSO is the core idea of this paper. If the initial search area is limited, the convergence time of the PSO process can be shortened and high positioning accuracy can be achieved.

PSO, performed in a limited search area, is an intelligent evolutionary computation algorithm that finds the location of a UE based on intelligent particles. PSO has advantages such as few parameters, simple implementation, and high positioning accuracy [8,9]. In PSO, particles share information and search for an optimal point. Since each particle determines the direction of movement based on shared information, the information of all particles must be updated periodically. The subsections below look at each step in detail.

### A. Fingerprinting Scheme

The fingerprinting scheme is a method of constructing a database by measuring the RSSI value at a specific location. Recently, indoor environments such as airports, large stadiums, high-rise buildings, and large department stores have become wider and more complex. Therefore, when building a fingerprinting database in an indoor environment, big data technology that can store a large number of RSSI samples is required. In this paper, a simulation-based fingerprinting technique was performed. First, the RP is placed in a specific location within the environment considered in the system model. Thereafter, each AP calculates an RSSI value for each RP based on (1). A fingerprinting database  $DB_F$  is built based on the calculated values. The constructed fingerprinting database  $DB_F$  can be expressed as (2) below.

$$DB_F = \begin{bmatrix} h_1^1 & \dots & h_1^m & \dots & h_1^M \\ \vdots & & \vdots & & \vdots \\ h_r^1 & \dots & h_r^m & \dots & h_r^M \\ \vdots & & \vdots & & \vdots \\ h_R^1 & \dots & h_R^m & \dots & h_R^M \end{bmatrix} \quad (2)$$

where,  $h_r^m$  represents an RSSI value between the  $m$ -th AP and the  $r$ -th RP. Thereafter, the  $DB_F$  value is used to estimate the real user's position in k-NN algorithm.

### B. k-NN Algorithm

The k-NN is one of the supervised learning algorithms that finds k closest data in feature space with random input. In other words, it is to find the k most adjacent RPs from the user's location. The k-NN algorithm is a method without a learning process, and when new data comes in, it selects neighbors by measuring the distance between existing data. Because k-NN does not build a model separately, it is also called Instance-based Learning. In k-NN, there are Euclidean Distance and Manhattan Distance as distance measurement methods. In this paper, we apply the method of deriving the adjacent RP based on the commonly used Euclidean Distance. A method of deriving the closest RP from the user is as follows.

First, the RSSI value for the user's location is measured in the online phase. The measured value can be expressed as follows.

$$H_u^{RSSI} = [h_u^1, h_u^2, h_u^m, \dots, h_u^M] \quad (3)$$

where,  $h_u^m$  is the RSSI value between the  $m$ th AP and the user UE  $u$ . Assuming the measured user's RSSI value as new data, the closest RP is derived by calculating the Euclidean Distance with the fingerprinting database value.

$$d_{u,r} = \|H_u^{RSSI} - DB_F\| = \sqrt{\sum_{m=1}^M (h_u^m - h_r^m)^2} \quad (4)$$

where,  $d_{u,r}$  denotes the Euclidean Distance between UE  $u$  and RP  $r$ , and the smaller the value, the closer UE  $u$  and RP  $r$ . Thereafter, the initial search area of the PSO algorithm is limited based on the  $k$  nearest RPs.

### C. PSO Algorithm

PSO is an intelligent evolutionary computational algorithm proposed by James Kennedy and Russell Eberhart in 1995 and can derive an optimal solution by distributing particles within the search area. The specific process of the PSO algorithm is as follows.

First, the particles in the swarm perform the initialization process. The initialized particles are randomly distributed in the search area, and the location of the UE is estimated. In the PSO, all particles repeat the process of finding the optimal solution estimated as the actual location of the UE. During the search, each particle shares its optimal position,  $pbest$ , and its optimal position,  $gbest$ , within the cluster. Particles are searched based on  $pbest$  and  $gbest$  to derive an optimal solution. The algorithm terminates when the maximum number of iterations is reached or the target accuracy is achieved. The parameter changes of each particle according to the repetition are shown below.

$$V_p(t+1) = wV_p(t) + c r [pbest_p(t) - x_p(t)] + c r [gbest(t) - x_p(t)] \quad (5)$$

$$X_p(t+1) = X_p(t) + V_p(t+1) \quad (6)$$

$$w = w_{max} - \frac{t}{T}(w_{max} - w_{min}) \quad (7)$$

where,  $V_p(t)$  is the velocity of the  $p$ -th particle in the  $t$ -th iteration,  $X_p(t)$  is the position of the  $p$ -th particle in the  $t$ -th iteration. In addition,  $c$  is an acceleration coefficient,  $w$  is an inertia coefficient, and  $r$  is an arbitrary coefficient of contraction.  $t$  represents the current number of iterations, and  $T$  is the total number of iterations of the PSO algorithm.

As above, while the particles perform repetitions, the PSO process is terminated when the preset target accuracy or the maximum number of repetitions is reached. After the PSO process is finished, the position of the particle having the most optimal solution becomes the estimated position of the UE. The process of the proposed technique is described in detail in Algorithm 1.

#### Algorithm 1: Proposed Positioning Algorithm

**Result:** Location of user  
 Environment Initialization  
 Distribute the RP at a specific location within the area for  $m = 1:M$   
 where,  $M$  is the total number of Wi-Fi APs  
 Measure Wi-Fi AP  $m$  and each RP's RSSI value  
 End

Fingerprinting database  $DB_F$  construction for  $m = 1:M$

Measure Wi-Fi AP  $m$  and UE's RSSI value

End

Construction  $H_u^{RSSI}$  with measured values

Weighted fuzzy matching with  $DB_F$  and  $H_u^{RSSI}$

Deriving the  $k$  RPs closest to the UE and obtaining a limited area

Randomly distribute particles over a limited area

For  $t = 1:T$

PSO algorithm implementation

End

Obtain the position of the particle with the most optimal fit and use it as the UE's estimated position

## IV. SIMULATION SETUP AND RESULTS

A simulation was performed based on MATLAB 2017b to evaluate the performance of the proposed scheme. The main parameters used in the simulation are summarized in Table I.

TABLE I. SIMULATION PARAMETER

Parameter	Value
Room size	120m × 50m
Distance between reference points	3, 6, 9 m
Number of iteration	10,000
Number of AP	12
Transmit Power of AP	20 dBm
Number of Particle	10
$c, r, w_{min}, w_{max}, T$	2, 0.3, 1, 0.4, 10
$k$ Value	3, 4, 5

As can be seen in Table 1, in this paper, user positioning is performed based on the indoor environment suggested by 3GPP. The size of the suggested indoor environment is 120m × 50m. To evaluate the positioning performance, the simulation is repeated a total of 10,000 times. The built environment is shown in Fig. 3.

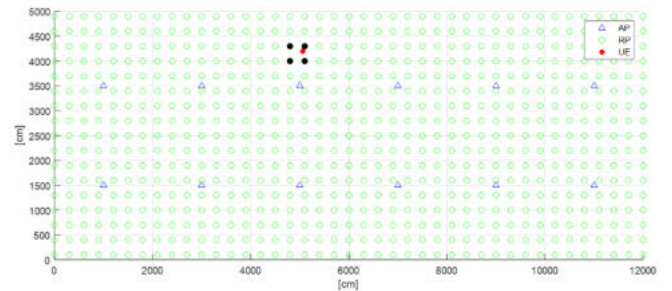


Fig. 3. Indoor environment and RP location realized by simulation

In Fig. 3, the blue triangle is a WiFi AP, and the red circle is a UE, and the green circle is an RP, and the black circles are the  $k$  RPs closest to the UE. A total of 12 APs are uniformly placed in the presented environment. The transmit power of each AP is set to 20dBm, and the interval between APs is 20m. At this time, to apply the fingerprinting scheme, the interval of the RP is arranged by changing it to 3, 6, and 9m. In the simulation, the UE is randomly placed in the suggested environment and the proposed positioning scheme is applied. The initial search area is limited for the PSO process.

Simulations are performed by changing the  $k$  value of the  $k$ -NN algorithm to 3, 4, and 5 to derive a limited area. Afterward, in the PSO process, the particles are initialized using the variables shown in Table 1. In the proposed scheme, the maximum number of repetitions of the PSO was set to a total of 10 times.

TABLE II. POSITIONING ERROR ACCORDING TO K-VALUE

Scheme	Positioning Error [m]		
	Distance between of RP [m]		
	3m	6m	9m
Best Match ( $k = 1$ )	2.380 m	3.134 m	4.416 m
$k$ -NN with $k = 3$	1.380 m	2.076 m	3.211 m
$k$ -NN with $k = 4$	1.227 m	2.019 m	3.102 m
$k$ -NN with $k = 5$	1.231 m	2.322 m	3.526 m

Table 2 shows the positioning error of each scheme according to the change in the RP interval. The best match is a scheme of estimating the coordinate value of the RP with the highest proximity as the coordinates of the UE. In this case, the best match set the value of  $k$  to one in the  $k$ -NN algorithm. In  $k$ -NN,  $k$  values were set to 3, 4, and 5, and simulations were performed. As can be seen from the results, it can be confirmed that the highest positioning accuracy performance is shown when  $k$  values are 3 and 4. This shows that the smaller the initial search area of the PSO algorithm, the higher the positioning accuracy performance.

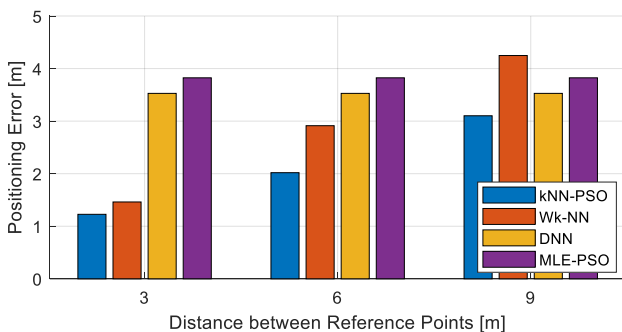


Fig. 4. Positioning Error vs Distance between Reference Points

Fig. 4 is the result of comparing the positioning accuracy between the proposed scheme and the existing scheme.  $k$ NN-PSO is a scheme proposed in this paper and based on the results of Table 2, the  $k$  value is 4. W(Weighted) $k$ -NN is a method of deriving  $k$  adjacent RP, assigning weights through Euclidean distance calculation, and estimating the user's location. MLE-PSO[4] is a method of improving the positioning accuracy by estimating the user's approximate location through the MLE method and then applying the PSO algorithm additionally. A DNN(deep neural network) is a method of estimating a user's location using a feed-forward neural network. As can be seen from the fig.4, it can be seen that MLE-PSO and DNN have a constant positioning error regardless of the change in the distance between RPs. This shows that the two schemes are a method of estimating the user's location without relying on RP. Unlike this, it can be seen that the positioning error of  $k$ NN-PSO and Wk-NN increases as the distance of RP increases. As the result shows, when the distance between RP is 3m, it could be verified that the scheme proposed in this paper achieves the highest

positioning accuracy. This improved the probability of particle convergence by limiting the area in which the actual user may exist to the PSO search area.

TABLE III. COMPARISON OF PROCESSING TIME OF EACH SCHEME

Scheme	Processing Time [s]
Wk-NN	0.06572
MLE-PSO	0.15314
$k$ -NN-PSO	0.10847
DNN	0.00144

Table 3 shows the algorithm processing time of each scheme. First, the  $k$ NN-PSO scheme and MLE-PSO scheme proposed in this paper have the longest processing time. It can be confirmed that this is related to the time required for the PSO algorithm to converge. In the case of Wk-NN, it can be confirmed that the PSO scheme achieves a shorter processing time than the above two methods in a way that is not applied. In the case of DNN, when learning is completed in the offline stage as one of the supervised learning models, it can be confirmed that the time required for the user's position is the shortest in the online stage. However, since all four schemes may complete the positioning within 1 second, it is judged that the positioning will not be a problem.

## V. CONCLUSION

In this paper, a study was conducted to improve the positioning accuracy of users in the indoor environment suggested by 3GPP. It is based on WiFi communication and uses a fusion of fingerprinting,  $k$ -NN, and PSO algorithms to locate the user. Through the simulation results, it was confirmed that the positioning accuracy varies according to the size of the fingerprinting database built in the offline stage and the  $k$  value in the  $k$ -NN algorithm. In this paper, it was confirmed that the highest positioning accuracy was achieved when the distance between RPs was 3 m and the  $k$  value was 4. In the future, research is planned to improve the positioning accuracy through optimization of the PSO algorithm.

## ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (NRF-2021R1F1A1063845).

This research was supported by the Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea government (MOTIE) (N0002429, The Competency Development Program for Industry Specialists).

## REFERENCES

- [1] S. C. Yeh, W. H. Hsu, M. Y. Su, C. H. Chen, K. H. Liu, "A Study on Outdoor Positioning Technology using GPS and WiFi Networks," 2009 International Conference on Networking, Sensing and Control, Mar. 2009.
- [2] B. H. Kim, M. C. Kwak, J. K. Lee, T. K. Kwon, "A multi-pronged approach for indoor positioning with WiFi, magnetic and cellular signals," 2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Oct. 2014.
- [3] N. Li, J. Chen, Y. Yuan, "A WiFi Indoor Localization Strategy Using Particle Swarm Optimization Based Artificial Neural Networks,"



- International Journal of Distributed Sensor Networks, vol. 12, pp. 1-9, Mar. 2016.
- [4] Z. Chong, W. Bo, "A MLE-PSO Indoor Localization Algorithm Based On RSSI," 2017 36th Chinese Control Conference (CCC), Jul. 2017.
  - [5] S. K. Gharghan, R. Nordin, M. Ismail, J. A. Ali, "Accurate Wireless Sensor Localization Technique Based on Hybrid PSO-ANN Algorithm for Indoor and Outdoor Track Cycling," IEEE Sensors Journal, vol. 16, pp. 529-541, Jan. 2015.
  - [6] Z. Yajun, W. Hao, W. Hongjun, "Indoor Navigation System Design based on Particle Filter," 2016 International Conference on Intelligence Transportation, Big Data & Smart City(ICITBS), pp.105-108, Dec. 2016.
  - [7] K. S. Kim, S. H. Lee, K. Huang, "A scalable deep neural network architecture for multi-building and multi-floor indoor localization based on Wi-Fi fingerprinting," Big Data Analytics, vol. 3, pp. 1, Apr. 2018.
  - [8] "Study on channel model for frequencies from 0.5 to 100 GHz (Release14)," 3GPP TR 38.901.
  - [9] H. K. Yu, S. H. Oh, J. G. Kim, "AI based Location Tracking in WiFi Indoor Positioning Application," 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC), Feb. 2020.

# A Study on the improvement of chinese automatic speech recognition accuracy using a lexicon

1<sup>st</sup> Min-Jeong Gu  
University of Science and Technology  
Daejeon, Republic of Korea  
gmj1130203@gmail.com

2<sup>nd</sup> Shin-Gak KANG  
ETRI  
Daejeon, Republic of Korea  
sgkang@etri.re.kr

**Abstract**—In this paper, in order to improve the error rate that occurs when Automatic Speech Recognition (ASR), one of the technologies widely applied in the field of speech recognition, is applied to Chinese, the study results on how to improve the Chinese recognition error rate by proposing a model to be added are described. As a result of testing by applying the xlsr-53 data set based on the Wav2vec2.0 model, it was confirmed that the Chinese recognition rate was improved.

**Keywords**—ASR Model, Word lexicon, Chinese speech recognition, Wav2vec 2.0

## I. INTRODUCTION

With the development and dissemination of speech recognition technology, much attention is paid to improving the accuracy of speech recognition results. Various methods have been proposed to improve the accuracy of speech recognition, but there are still errors that need to be improved, such as homonyms and misrecognition, in order to be introduced into an expert system. A recently widely used technology in the field of speech recognition is automatic speech recognition (ASR). ASR is a technology that automatically executes the process of receiving voice input and outputting it as text.

In the past, experts directly adjust each parameter of the acoustic model (AM), the lexicon (Lexicon), and the language model (LM) using HMM (Hidden Markov Model) models and perform voice recognition. However, deep learning technology, which has recently been attracting attention in speech recognition, has enabled end-to-end learning that learns a target result (output) from data (input) without a separate intermediary [3].

Despite the advantage of being able to output a model that optimizes all parameters from start to finish just by putting it into a deep learning model, this method also has limitations. This method showed better

performance when a large amount of data was given as input data, especially when labeled data was given as input data. However, it required a lot of labor, time, and work to create a labeled data set, which became a very difficult problem for researchers.

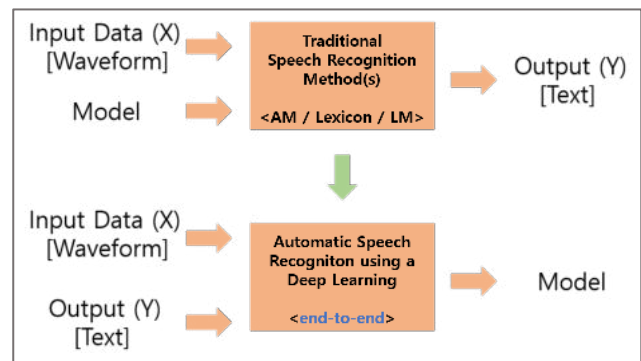


Fig. 1. Comparing with Traditional Method and Deep Learning Method

To solve the problem that requires a lot of labeled data, Facebook proposed a new voice recognition model, Wav2vec 2.0, in June 2020. The Wav2vec 2.0 model, which uses self-supervised learning and shows high accuracy even with a small amount of label data, is currently being actively used in the field of automatic speech recognition (ASR). It is mainly used in the form of fine-tuning the pre-trained wav2vec 2.0 model in the subject you want to apply.

This paper describes the research results on the improvement of the error rate that occurs when automatic speech recognition (ASR: Automatic Speech Recognition) is applied to Chinese. Basically, the wav2vec 2.0 model was used, and a method to improve the Chinese recognition error rate was proposed by adding Pinyin, the Chinese pronunciation symbol, to the Word Lexicon. And by applying the xlsr-53 data set, an experiment was performed to check whether the Chinese recognition rate was improved, and the experimental results were analyzed.

## II. OVERVIEW OF SPEECH RECOGNITION TECHNOLOGY

### A. Before emergency of the Deep learning

Traditional speech recognition technology has been developed based on statistics, and a representative model is the HMM-based model of Figure 2 using the Hidden Markov Model.

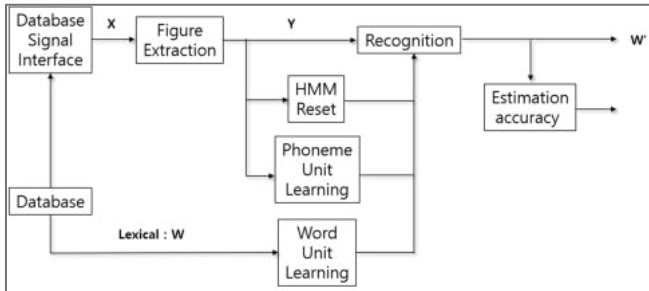


Fig. 2. Speech Recognition system structure using HMM Model.[1]

In the past, the speech recognition system was divided into three main parts: an acoustic model (AM), a lexicon (Lexicon), and a language model (LM). Unlike other data, voice data may include a variety of information in one input. This is a very attractive part because it can include various data such as the speaker's speech characteristics, intonation, surrounding environment, and speech space as well as the speech content, but it has a big difference in that it is continuous unlike other data.

### B. After emergency of the Deep learning

However, in modern times, a lot of data has been generated due to the development of the Internet and various electronic devices, and it was expected that voice patterns could be easily found based on a large amount of data. At that time, the deep learning technique advantageous for pattern matching became a big issue at the time, and we will try to apply it to the field of automatic speech recognition. Through several data preprocessing processes, it was possible to visualize the voice data with the characteristics of the waveform data and analyze the pattern. After applying the deep learning technique in the previous step, which obtained the loss in each of the existing three steps, the three processes are combined to make into one loss.

The data preprocessing process for converting voice data into visual data that can find patterns was somewhat complicated, but in the end, the input voice data was Fourier transformed and expressed as a frequency spectrogram, which is then weighted to a low frequency band similar to human speech frequency. It is possible to image using the vector value converted to Mel-spectrogram.

## III. DEEP LEARNING BASED ASR MODEL

Recently, along with the development of the deep learning technology mentioned above, research to apply the deep learning technology to automatic speech recognition is being actively promoted. The representative ASR basic structure that takes an end-to-end format by integrating the existing three steps into a decoder by applying deep learning technology is the 'Encoder-Decoder' model shown in Figure 3.

### A. Encoder – Decoder (Sequence to Sequence)

The basic structure of the encoder-decoder model is shown in same as Figure 3 The encoder compresses and expresses the input voice data, and the decoder plays a role in converting the compressed voice data into text. The encoder receives data and compresses information into a single vector. This compressed vector data is called a context vector, and the decoder converts the context vector back into text data format. Whenever an encoder receives a value, it is stored in the context vector. When new data is input, it is accumulated in the context vector and input and stored. After going through this process, at the end, the context vector accumulates and stores all input data before. If the input value is excessively generated in the context vector, the encoder compresses the input information and converts it into a fixed-length vector. Finally, the encoder result is stored in a context vector named  $h$ .

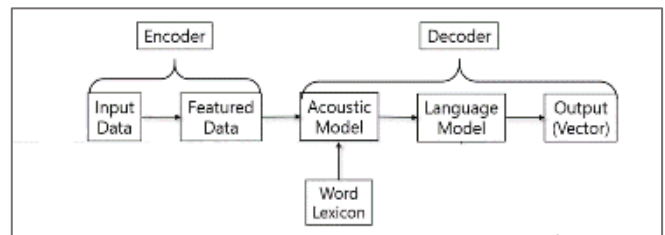


Fig. 3. ASR and basic structure of Encoder – Decoder Model.

The decoder converts the above context vector into an RNN-based language model (LM). Decoder is a value that comes out through the RNN-based model (usually LSTM) and the softmax function. (LSTM model - Affine layer - Softmax layer // Affine layer: This is a network that receives the hidden state as an input and outputs the number of classifications.) However, the encoder-decoder structure of seq2seq has a limitation in that the performance deteriorates when the input data becomes long, as mentioned in the above encoder structure. If the encoder does not effectively compress the input data, it may not include all of the key information, resulting in a decrease in accuracy. Therefore, the attention mechanism was used to compensate for this problem when compressing the input data, and it shows a very improved result.

### B. Self-supervised Learning

In the case of speech recognition, in particular, the amount of labeled correct answer data was very small, which caused many difficulties. In the case of voice data, the acoustic data and the language model match well, so that the text similar to the correct answer data as much as possible matches the align. The self-supervised learning method has solved this problem, and the initial model applied to the ASR field is the CTC

model. However, although CTC was meaningful in its early attempts in the speech domain, it did not deal with speech recognition as a major topic.[5] Subsequently, an initial version of wav2vec was introduced, which was a model that specialized in applying CTC to voice [6]. In another method, the VQ wav2vec model [6], BERT's MLM pre-training was introduced into the voice by adding a quantization module. The model based on the existing CTC method receives voice data continuously, but the biggest difference is that the model receives it discretely.

Lastly, the Wav2vec 2.0 model [7] is a voice recognition specialized model that applied deep learning technology announced in 2016 by Facebook. The Acoustic Model stage, which implements the wavelength data unique to voice as a Mel-Spectrogram that can be recognized by humans with the end-to-end technique, and the Language Model stage, which manages the decoded text data as an output result. It is a model implemented by combining . This model is similar to the VQ wav2vec model, but it is different from the existing method in that it supplements the quantization module and changes the structure to enable end-to-end learning. In the case of the Wav2vec2.0 model, which is based on self-supervised learning, it is a model that shows good learning performance even with a small amount of labeled correct answer data, unlike the models that previously required a large amount of labeled data. Labeled data is pre-trained, and unlabeled data is fine-tuned.

And in December 2020, Facebook announced XLSR-Wav2vec 2.0 Unsupervised Cross-lingual Representation Learning using a multilingual dataset called xlsr-53 on Facebook. [10] As a result of this study, the learning process was simplified by learning without using a language model (LM), but the error rate was increased. That is, in the case of the end-to-end learning method including the existing Wav2vec 2.0 model, unlike the traditional voice recognition, phoneme-mediated training and the Lexicon dictionary are omitted.

#### IV. PROBLEM STATEMENT AND PROPOSAL

Various studies related to speech recognition have been actively carried out, but research on 'English' is mostly. English is the most used language in the world, and since the language system has a relatively simple structure of 26 alphabets, it has the advantage of having a structure that is easy to directly experiment with theoretical models. In particular, in the case of a model using deep learning, a large amount of data is required, and since it is easy to experiment with a relatively simple language system, English is overwhelmingly more common than other languages. Therefore, even if the published speech recognition model is applied to other languages such as Korean or Chinese and attempts to verify the recognition result, the data set is insufficient. In particular, in the case of Korean and Chinese, the number and amount of publicly available labeled data that anyone can use for model validation are small. Although the multilingual data set called xlsr-53 provided by Facebook is insufficient, it provides data that can conduct speech recognition research on Chinese to some extent.

The Wav2vec 2.0 model is recognized as a good model that shows high performance in speech recognition research using deep learning, and is currently a universally used model in the automatic speech recognition (ASR) field. If this is not sufficient, there is a limit to the study of speech recognition for the corresponding language. In other words, although a small amount of labeled data shows good performance, the more high-quality labeled data is used, the better the speech recognition accuracy is. Considering this environment, this study proposes a model that can improve the error rate even when using a small amount of dataset in Chinese speech recognition, and analyzes the speech recognition results according to the proposed model.

Many studies have already been published confirming that the performance of automatic speech recognition is improved when the vocabulary dictionary is well constructed in speech recognition [8]. However, there are few studies that have actually investigated the correlation between recognition rates using official phonetic symbols approved by the state.

In the case of Chinese, since it has quite a lot of characters, there are not only officially designated phonetic symbols, but also the official Chinese pronunciation added to the Word Lexicon for Chinese, which is a language designated in the form of the English alphabet. We analyzed how pinyin, the symbol, affects the accuracy of speech recognition.

In this study, as described above, the Wav2vec2.0 model [4], which is the most widely used among the models applying deep learning in the automatic speech recognition (ASR) field, was used. The step of referring the Lexicon dictionary to the existing Wav2vec2.0 model was added as a post-processing concept after the acoustic model (AM). suggested. In order to verify the performance of the proposed model, it is necessary to compare the performance with the existing model under the same conditions, so the error rate was compared and analyzed using the pre-trained wav2vec2.0.

#### V. EXPERIMENT

##### A. Experiment Environments

In this study, 'Fairseq', that is, the Wav2vec2.0 model announced by Facebook, which was published on github, an open source code community site, is used as an automatic speech recognition model. In addition, the data set used for the experiment used 50 hours of Chinese data from the xlsr-53 multilingual package for 53 languages provided on github. The Chinese data set used contains 7,176 sentences and 169,193 words.

As the development environment for this experiment, two NVIDIA GPU RTX Quadra 8000 GPUs and 64 CPU cores were used. Also, as a result of performing some experiments using the virtualized GPU resources provided by Google's CoLab environment, there was no particular difference from the experiments in the local environment.

##### B. Establishment of experiment model

As shown in Figure 4, the model proposed in this paper to improve the speech recognition error rate is

based on the existing Wav2vec 2.0 model and presents an improved structure by adding Chinese phonetic symbols to the vocabulary dictionary. In addition to the vocabulary existing in the existing language model stage, as shown in Figure 6 below, In addition to the vocabulary existing in the existing language model stage, as shown in Figure 6 below, a total of 398 phonetic symbols that combine 23

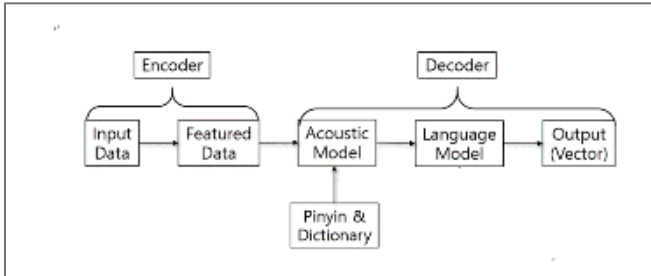


Fig. 4. Proposed Model – Add a pronunciation symbol ‘Pinyin’ put in ‘Language word dictionary’.

Chinese consonants and 35 vowels are included in the vocabulary dictionary. Chinese phonological rules can be broadly divided into falsification (變調), light tone (轻声) and ‘Er-hwa’ (儿化). In Chinese, each syllable is a unit with a meaning, so there is almost no phenomenon in which the sound changes as in Korean. (9) Therefore, it can be said that the method of putting all combinations following the alphabetical formula in the vocabulary is an effective method in reducing the error rate. An example of adding phonetic symbols to the vocabulary is shown in Figure 5.

Fig. 5. Chinese pinyin Consonant vowel combination table [12]

Figure 5 shows all Chinese alphabets with phonetic symbols, and syllables that cannot be pronounced or are not used are not marked. In the case of this material, it has the name ‘小學漢語拼音字母表’, and it is a pronunciation chart certified by the Ministry of Education, a national institution of China, and is actually a pronunciation symbol taught in the elementary education course.

In addition, considering the characteristic that tones are attached only to vowels as a characteristic of Chinese language itself, a combination table was created by inserting the numbers 1, 2, 3, 4, meaning 4 Chinese characters, after all vowels. Therefore, the entire combination was trained to refer to a total of 1,592

existing pronunciation possible combinations, 398 × 4 adults. Unlike Korean, which belongs to an agglutinative language, in which parts of speech are changed by adding an additional proposition, Chinese has the characteristic of language isolate that parts of speech change depending on location. In other words, there is less articulation change, which is a change in pronunciation, compared to English or Korean.

Data pre-processing and feature extracted vector values are obtained from the waveform form, which is the first raw data, and the vector value with the highest probability can be finally output as text by referring to Lexicon.

1	ba	17	pa	34	ma	53	fa	62	da	85	la	104	na	128	la
2	bo	18	po	35	mo	54	fo	63	de	86	te	105	ne	129	lo
3	bi	19	pi	36	me	55	fu	64	di	87	ti	106	ni	130	li
4	bu	20	pu	37	mi	56	fu	65	du	88	tu	107	nu	131	li
5	bai	21	pai	38	mi	57	fu	66	dai	89	tai	108	nu	132	lu
6	bei	22	pei	39	mai	58	fan	67	dai	90	tui	109	nai	133	lu
7	bao	23	pao	40	mei	59	fen	68	dui	91	tao	110	nei	134	lai
8	bie	24	pou	41	miao	60	fang	69	diao	92	lou	111	nou	135	lei
9	ban	25	pie	42	mou	61	feng	70	dou	93	tie	112	nou	136	lai
10	ban	26	pan	43	miu			71	diu	94	tan	113	nu	137	lou
11	bin	27	pen	44	mie			72	die	95	tun	114	nie	138	liu
12	bang	28	pin	45	man			73	dian	96	fang	115	nuan	139	lei
13	bang	29	ping	46	man			74	dian	97	fang	116	nian	140	lei
14	bang	30	peng	47	man			75	dian	98	fang	117	nian	141	lan
15	bian	31	ping	48	shang			76	dang	99	tong	118	min	142	lin
16	biao	32	pián	49	shang			77	dong	100	lian	119	niang	143	lian
		33	piao	50	ming			78	dong	101	biao	120	niang	144	liang
				51	mian			79	dong	102	tuan	121	niang	145	liang
				52	miao			80	diao	103	tuo	122	nong	146	liang
								81	dian			123	mian	147	long
								82	diao			124	niang	148	lia
								83	dian			125	miao	149	lian
								84	duo			126	nian	150	liang
												127	nuo	151	liang
												128	nuo	152	liang
												129	nuo	153	liang

Fig. 6. Example of Chinese pinyin Lexicon.

### C. Experiment result

In the previous study [10], when the wav2vec2.0 model was run for 1 hour using the xlsr-53 data set, the error rate (PER: Phoneme Error Rate) was reported to be 18.3%. In addition, as a result of running the existing model on its own using the given experimental environment, the PER was measured to be 18.8%, and the sub error rate generated by the synthesis was measured to be 12%. The result of running the model in which Pinyin, the Chinese phonetic symbol, is added to the lexical dictionary proposed in this paper in the same environment is shown in Table 1.

Table 1. Table of adding pronunciation symbol in dictionary result

SEMR	#Syn	#Word	Corr	Sub	Del	Ins	Err	S.Err
ORIG	7176	169198	83.8	8.2	8.8	2.6	11.8	91.8
Sum/Avg	7176	169198	83.8	8.2	8.8	2.6	11.8	91.8
Mean	7176	169198	83.8	8.2	8.8	2.6	11.8	91.8
SD	0	0	0	0	0	0	0	0
Median	7176	169198	83.8	8.2	8.8	2.6	11.8	91.8

As can be seen from the results of this experiment, the recognition error rate was measured to be 14.6%, confirming that the error rate was reduced. Also, the rate of sub error was reduced to 8.2%. This result means that the error rate due to misrecognition can be reduced when automatic speech recognition is executed by including phonetic symbols (that is, pinyin) in the lexical dictionary.

## VI. CONCLUSION

In this paper, we conducted a study on how to improve the error rate of Chinese speech recognition using the Wav2vec 2.0 model, which is a representative automatic speech recognition model published by Facebook. Based on the Wav2vec 2.0 model, a kind of post-processing was performed so that the lexicon can be additionally referenced after the acoustic model (AM).

As a result of the experiment, adding Chinese pronunciation information to the lexicon of the existing model reduced the speech recognition error rate. This was meaningful in the study. In this study, the xlsr-53 data set was used to analyze the improvement of the error rate of Chinese speech recognition, but the provided data set is relatively scarce compared to English, etc., so it is thought that there is a limit to improving the error rate. In the future, it is expected that more improved results can be obtained if a richer data set is secured and the proposed model of this study is applied to training.

#### ACKNOWLEDGMENT

1. This work was supported by Electronics and Telecommunications Research Institute(ETRI) grant funded by the Korean government [21YR 1200, Enhancement of ETRI Open Source Governance and Supporting Open R&D Activity (2021.09.01.-22.04.30)]

2. If you intend to utilize the contents of this paper, you must disclose that the research was funded by Electronics and Telecommunications Research Institute (ETRI)

#### REFERENCES

- [1] Park, Yoo-hyun. "Quantitative Analysis of Gartner's." *Journal of the Korea Institute of Information and Communication Engineering* 22.8 (2018): 1041-1048.
- [2] Lee, Suji, et al. "Korean speech recognition using deep learning." *The Korean Journal of Applied Statistics* 32.2 (2019): 213-227.
- [3] Davis, Ken H., R. Biddulph, and Stephen Balashek. "Automatic recognition of spoken digits." *The Journal of the Acoustical Society of America* 24.6 (1952): 637-642.
- [4] Lee, Gun-sang, "Speech Recognition", Hanyang Univ. , 2001, p.28, Figure 2.1
- [5] Oord, Aaron van den, Yazhe Li, and Oriol Vinyals. "Representation learning with contrastive predictive coding." arXiv:1807.03748 (2018).
- [6] Schneider, Steffen, et al. "wav2vec: Unsupervised pre-training for speech recognition." arXiv preprint arXiv:1904.05862 (2019).
- [7] Baevski, Alexei, Steffen Schneider, and Michael Auli. "vq-wav2vec: Self-supervised learning of discrete speech representations." arXiv preprint arXiv:1910.05453 (2019).
- [8] Baevski, Alexei, et al. "wav2vec 2.0: A framework for self-supervised learning of speech representations." arXiv preprint arXiv:2006.11477 (2020).
- [9] Jang, Tae-Yeoub, "Implementation of a non-native pronunciation dictionary for automatic recognition of utterances by Korean learners of English", (*Journal of humanities*) 56 pp.99~122, (2006).
- [10] Cho, Ara, " Finding ways to educate Chinese learners on Korean pronunciation(중국인학습자의 한국어 발음교육방안 모색)." , *Korean Language in China* (2), (2019): 50-63.
- [11] Conneau, Alexis, et al. "Unsupervised cross-lingual representation learning for speech recognition." arXiv preprint arXiv:2006.13979 (2020)
- [12] Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems*. 2017
- [13] <https://www.51wendang.com/doc/dd97ebf4e656d3a950a1c397e408710308690493>

# Addressing Data Sparsity with GANs for Multi-fault Diagnosing in Emerging Cellular Networks

A. Rizwan, A. Abu-Dayya, F. Filali  
Qatar Mobility Innovations Centre  
Qatar University, Doha, Qatar  
[arizwan, adnan, filali]@qmic.com

A. Imran  
University of Oklahoma  
Oklahoma, USA  
ali.imran@ou.edu

**Abstract**—Data-driven machine learning is considered a means to address the paramount challenge of timely fault diagnosis in modern and futuristic ultra-dense and highly complex mobile networks. Whereas diagnosing multiple faults in the network at the same time remains an open challenge. In this context, the data sparsity is hindering the potential of machine learning to address such issues. In this work, we have proposed a data augmentation scheme comprising Pix2Pix Generative Adversarial Network (GAN) and a customized loss function never used before, to address the data sparsity challenge in Minimization of Drive Tests (MDT) data. Our proposed unique augmentation scheme generates images of MDT coverage maps with Peak signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) values of 25 and 0.97 respectively, which are significantly higher than those achieved without our customized loss function. The performance of data augmentation scheme used is further evaluated with a Convolutional Neural Network (CNN) model for simultaneously detecting most commonly occurring network faults, such as antenna up-tilt, antenna down-tilt, transmission power degradation, and cell outage. The CNN applied on the data generated from the 1% of the MDT data with the proposed augmentation scheme has led to a gain of 550% in the detection of all classes, including the four faults and cell with normal behavior, as compared to when it is applied on the data generated without our customized loss function.

**Index Terms**—GAN, ZSM, Fault diagnosis, Automation, Machine Learning, Deep learning, Wireless cellular networks.

## I. INTRODUCTION

Network Performance Management(NPM) has always been a strenuous job highly dependent on skilled human resources. Network operators spend a significant share of their OPEX on NPM. But with the emergence of contemporary and futuristic technologies aka 5G, beyond 5G (B5G) and 6G, it has become essential to automate the functions of NPM. 3GPP introduced Self Organising Networks (SON) for automating network operations grouped in three main categories of self-healing, self-configuration, and self-optimization. The research on SON has led to significant progress made so far in automating network operations in 5G, and it also provides the ground base for the projects like Hexa-X and ETSI ZSM aiming on Artificial Intelligence (AI) driven automation in B5G and 6G [1]. Research work on self-healing function in SON has set the premises for the automated NPM by introducing the concept and solutions for the automation of fault detection and diagnosis. AI equipped SON heavily relies on data-driven machine learning for the automation of network operations.

An important and equally challenging task of NPM in emerging and futuristic networks is automating the detection and diagnosis of cells facing some type of technical issues. Thanks to 3GPP release 10 for introducing Minimisation Drive Test (MDT) that enabled network operators to collect key data from users' equipment, rather than conducting drive test incurring too much operational cost and unnecessary delays. MDT data and machine learning can help in devising solutions for the automation task here but the sparsity of MDT data limits its potential. There exist different approaches like inpainting techniques and machine learning based models that can help in addressing data sparsity. They all have their pros and cons. But one machine learning tool recently developed, Generative Adversarial Network (GAN), has gathered the much attention of the researchers for its performance and efficacy in applications like addressing data sparsity.

Different GAN architectures proposed in recent research mostly aim at image-based applications like image completion, image super-resolution, image transformation, etc. But these GAN are proved to be very effective for such tasks as addressing data sparsity in the form of image completion. The key advantage of GANs is that, instead of just creating copies or averaging out values, they learn the data distribution patterns and create the new samples from those distributions. The newly generated samples are similar but not the same as the original ones. This little variance in the GAN generated data samples can reflect the real-world variations in the outcomes of the same network deployment schemes.

In this article, we propose a novel scheme for automating the detection of multiple commonly occurring network faults from the sparse MDT data. To the best of our knowledge, it is the first time that GANs are used to address sparsity challenges in mobile networks MDT data, presented as images, for detecting multiple faults in the network. The contributions we have made in this study are summarised as follows:

- We have introduced a Pix2Pix GAN-based unique data augmentation scheme to address data sparsity in MDT reports. The proposed scheme can generate a complete coverage map from the 1% data.
- We have introduced our own customized perceptual loss function, never used before, that has increased the performance of the GAN model manifolds.
- We have introduced a CNN model that can successfully

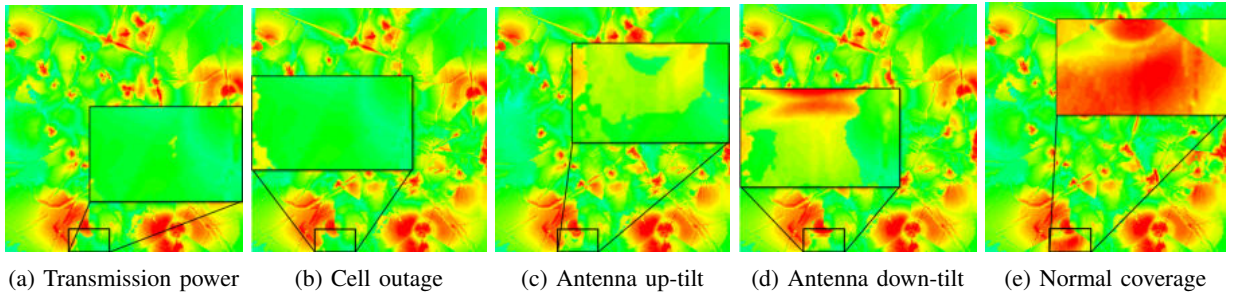


Fig. 1: Signatures of normal coverage and different network faults

detect multiple faults in the network even in sparse data and yields manifold gain for the detection of multiple faults on the data augmented with our proposed scheme.

The rest of the paper is organized as follows. The Section II offers a survey on state-of-the-art. Section III provides a brief description of data generated from the simulator. Section IV discuss the methodology adopted for the data enrichment and fault diagnosis. Section V presents discussion on the results and Section VI concludes the study.

## II. RELATED WORKS

There are many studies that propose schemes and solution for cell outage detection [2], [3] and many of them take it as anomaly detection problem. But, there are very limited studies that focus on the detection of other network issues than the cell outage, that may cause sub-optimal performance in specific cells in the network. The studies on the diagnosis of such faults are even rare [4]. Whereas the detection of multiple faults, instead of just one single fault, is still an open research issue. One of the such studies is [5], where the authors have applied a semi-supervised learning scheme on real network Call Detail Record (CDR) data for grouping cells into multiple classes based on their performance. They have detected and diagnosed cells with sub-optimal performance and identified reasons of sub-optimal performance. But this study used CDR data form real network and rely on the input from the expert for the labelling of cells based on their performance, and presents only broad level network performance issues.

In a recent study [4] authors have proposed solution for the diagnosis of commonly occurring multiple faults such as site outage, transmission power, antenna up-tilt and antenna down-tilt. In addition to issues addressed in [4] authors in study [6] have introduced solution for diagnosing even more advanced network issues like Too Late Handover (TLHO), Inter-Cell Interference (ICI), and Cell Overload (CO). But main bottleneck of these two studies is, they are using complete coverage map generated from simulators. Whereas in the real network data points are very sparse. But the aspect of addressing data sparsity for the NPM management tasks is missing in literature.

Recently GANs have been very popular as a tool for data augmentation and have been used for diverse applications. Most popular applications of GANs are image based, like for image generation, image to image translation, image

super-resolution, semantic segmentation etc., [7]. For images, two main application of GANs have been image quality enhancement and image completion. GANs have been used to successfully generate complete images from the incomplete images [8] and improve the resolution of the images [9]. The same concepts can be applied in our case for generating complete MDT network coverage map from the incomplete coverage map. This approach has potential to address the data sparsity issue in MDT reports.

## III. DATASET DESCRIPTION

In this study for the MDT data generation, we have used At-tol, an RF planning software capable to generate real scenario data. To make sure that the data generated is close to real-world network data, we have considered network topology and parameters from a real mobile network operator in Brussels. Besides that the network is also simulated over an area in Brussels, Belgium considering 15 types of clutters based on different terrain and environmental profiles. The simulated network comprises 24 sites (macrocells) and 72 transmission antennas (cells), with 3 antennas deployed on each base station, overall covering an area of  $15 \text{ km}^2$ . Detail about the network parameters used is listed in Table I.

Using that simulation environment, from the MDT reports we have generated SINR based network coverage map of 72 cells with 68 cells performing normally and four randomly selected cells are induced with any of the four faults listed below. Samples of the each possible status of the cells are shown in Figure 1. The color range from green to red present

TABLE I: Network parameters used for MDT data generation

Network Parameters	Values
Propagation Model	Aster Propagation Model (Ray-tracing)
Maximum transmission power	43 dBm
Cell individual offset (CIO)	0 dB
Antenna tilt	0
Antenna gain	18.3 dBi
Carrier frequency	2100 MHz
Network layout	24 Macrocell BS
Simulation area	15 km <sup>2</sup>
Transmitters (sectors) per BS	3
Base station height	Actual site heights
Clutter types	15 classes
Geographical information	Digital Terrain Model (Ground heights)
	Digital Land Use Map (clutter classes)



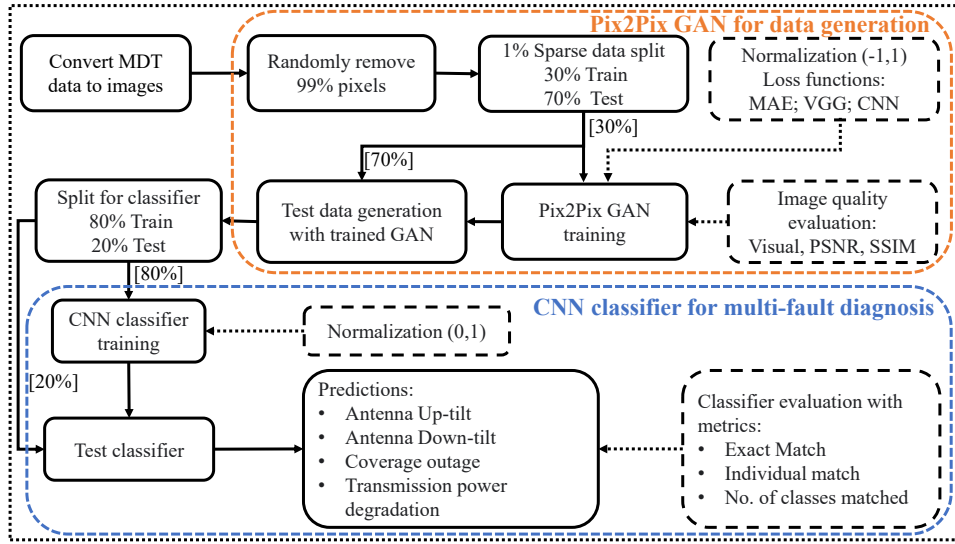


Fig. 2: Process flow diagram for MDT data augmentation and classification of multiple faults

poor to good SINR values in the network. In total 22,864 coverage map images are generated.

- Low Transmission Power (LTP): Maximum transmission power of a normally functioning cell is 43 dBm. But, based on common experience in the industry, here we have reduced it to 25 dBm.
- Cell Outage (CO): In this type of issue cell is not functional at all, it can be caused by transmitter deactivation.
- The antenna is tilted by an angle of  $-20^\circ$  from the standard normal antenna angle of  $0^\circ$ .
- This fault is induced by changing the tilt value of the simulation from  $0^\circ$  to  $20^\circ$ .

#### IV. PROPOSED SCHEME FOR DATA AUGMENTATION AND FAULT DIAGNOSIS

The important steps of the methodology adopted in this study to accomplish two main tasks, addressing data sparsity and diagnosing multiple network faults along with a preliminary step of generating relevant sparse data images, are listed in Figure 2 and discussed in this section.

##### A. Generating sparse data

One of the main objectives of this study is to address the challenge of data sparsity in mobile networks. The sparse data images are, therefore, generated from the complete images of simulated coverage maps already discussed in Section III. Since each pixel in the images is a representative of an SINR value from a UE, therefore, to create an extreme data sparsity scenario, for this study, we have removed 99% of the pixels from the images that left only 1% of as much information as present in original complete images. The original images are resized to 256 by 256 dimensions before pixel removal for convenience in image processing and consistency in implementation of deep learning. Since each pixel represents a user, therefore, a complete 256 by 256 image means, the  $15km^2$  coverage area in the image has 65,536 users and the number of users per cell is around 910. So when we remove 99% of

the pixels then the new incomplete images have around 655 users in the  $15km^2$  coverage area i.e around 9 users per cell or base station BS and around 43 users per  $km^2$ .

##### B. Generating enriched data using GAN

Next important task is developing a scheme for generating complete network coverage map from the incomplete coverage map images of 1% data points. For that purpose we have used a conditional GAN architecture, Pix2Pix-GAN proposed in [10] with customized perceptual loss function in [11] also used in [9] with VGG-19 model. Pix2Pix GAN is a conditional GAN that takes an input image  $x$ , like an incomplete network coverage map shown in Figure 3a, along with a noise vector  $z$ , to learn a mapping function from  $x$  to  $y$  the target image which is the complete coverage map as shown in Figure 3b and to generate the outcome image  $\hat{y}$  like shown in Figure 3c. We provide noise only in the form of dropout, applied on several layers of our generator at both training and test time. The dropout noise is observed to lead to minor stochasticity in the output of our model making generated images more close to the real network scenarios.

Like the standard architecture of GANs, Pix2Pix also comprises a generator and a discriminator. Where the main objective of the Pix2Pix proposed in [10] is:

$$G^* = \arg \min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L_1}(G) \quad (1)$$

Where  $\mathcal{L}_{cGAN}(G, D)$  presents the objective function of typical conditional GAN computed as follows:

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y} [\log D(x, y)] + \mathbb{E}_{x,z} [1 - \log D(x, G(x, z))] \quad (2)$$

Here the generator  $G$  tries to minimize this objective against an adversarial discriminator  $D$  that tries to maximize it with following approach: The authors in [10] included the other part, one of the commonly used loss function, the  $L_{L_1}$ , to

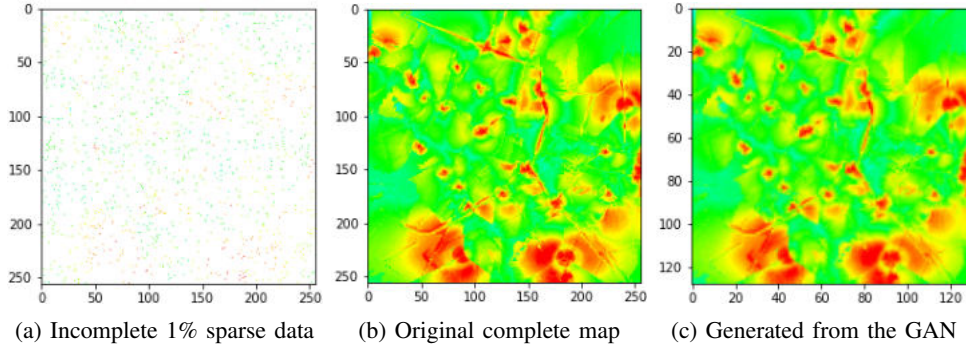


Fig. 3: Sample of images from input(left),target (middle),and output (right) in Pix2Pix GAN

compute the distance between the pixel values of generated and target image as follows:

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y,z} [\|y_i - \hat{y}_i\|_1] \quad (3)$$

However, the above MAE optimization can lead to satisfactorily high PSNR, but often it lacks high frequency content which results in perceptually unsatisfying solutions. We have, therefore, introduced a perceptual loss also known as feature reconstruction loss which is a type of content loss introduced in [11]. Rather than encouraging the pixel to pixel match of the output image  $\hat{y}$  and target image  $y$ , we encourage GAN to learn similar feature representations as computed by the loss network  $\phi$ . while processing the image  $x$  if the  $\phi_j(x)$  is the activation of the  $j$ th layer of the network  $\phi$  and  $j$  is a convolutional layer then  $\phi_j(x)$  results in a feature map of shape  $C_j \times H_j \times W_j$ . The feature reconstruction loss is the normalized, squared Euclidean distance between feature representations computed as follows:

$$\ell_{feat}^{\phi,j}(\hat{y}, y) = \frac{1}{C_j W_j H_j} \|\phi(\hat{y}_i) - \phi(y_i)\|_2^2 \quad (4)$$

So in this study we have used two network functions  $\phi$  to compute the above perceptual loss. One  $\phi$  is VGG-19 inspired from [9] and the other  $\phi$  is our own CNN network. We have computed the perceptual loss as an additional loss to the loss computed in equation IV-B. As a result the objective function in our case becomes:

$$G^* = \arg \min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L_1}(G) + \alpha \ell_{feat}^{\phi,j}(\hat{y}, y) \quad (5)$$

where  $\lambda = 100$  and  $\alpha = 10^{-3}$  as suggested in [10] and [9] respectively. So using the using the objective function in equation IV-B in Pix2Pix GAN architecture proposed in [10] we have generated the network coverage maps from the sparse data coverage map. We have generated images using the perceptual loss in IV-B with two network VGG and our own network CNN separately and results for the both are compared in the results section.

Once the images are generated from the GAN a crucial task is to evaluate the quality of the generated images. Common evaluation metrics measure the pixel to pixel euclidean distance which does not take the contextual or structural

information into consideration. Here we have therefore used two popular and relevant metrics, Peak Signal to Noise Ratio (PSNR) to evaluate the pixel to pix match and structural similarity index measure (SSIM) to evaluate structural similarity. PSNR and SSIM are good indicative of the image quality but better PSNR and SSIM values not necessarily mean that the images generated are true representation of original images. Apart form these evaluation tools, we have majorly relied on the performance of our classifier for the selection of images generated by the Pix2Pix GAN.

### C. Classifying multiple faults

In total around 16000 images are generated from Pix2Pix GAN, which makes 70% of the total images generated from the simulator. The generated images have all pixel values and visually look similar to the original images. After the enriched images generation, the next step is to identify the four faults present in the network using those enriched images. For that goal, we fine-tuned many of the popular pre-trained models and also applied a custom CNN architecture. For the development of the classifier, we split the GAN-generated data into train and test data, such that 80% of the data is used for training classifiers and 20% of the data is used for evaluation.

The classifiers are developed such that they not only predict the four BS having any issue and type of the issue present but also predict which BS performs normally. Hence the classifier has 360 predicates in total, 72 cells having five possible outcomes, performing normal, or having issues due to up-tilt, down tilt, transmission power degradation, or complete outage. Overall accuracy can not be the representative metrics to evaluate the performance of such a classification scheme. Since we are trying to predict multiple faults so it is important to know that in how many cases(images) status of all 72 cells are detected correctly from five possible outcomes. Here, therefore we have used the exact match as a metric to find in how many cases all 360 predicates are predicted correctly. Besides that, we have also calculated the class-specific accuracy to reflect the accuracy rate of identification of any individual fault or normal behavior. The third factor we evaluate is the performance of the classifier for detecting the number of faults.

## V. RESULTS AND DISCUSSION

In this section, first, we present the results produced from the Pix2Pix GAN data augmentation scheme applied for addressing the data sparsity. Later, the results of the multi-fault classification scheme are presented.

### A. Results of data enrichment with Pix2Pix GAN scheme

A representative sample of the images generated as the result of the Pix2Pix model-based augmentation scheme is shown in Figure 3c. In Figure 3, the image 3a on the left presents the sample from the sparse data input images for the Pix2Pix GAN scheme used for data augmentation. Images like 3a are produced by removing 99% of the pixels from the complete coverage maps. The image 3c on the right presents a sample from the generated images with our proposed data augmentation scheme. Even visually it can be seen that 3c looks almost the same as the original complete coverage map 3b in the middle.

Figures 4a and 4b present the PSNR and SSIM value for the five epochs where the data generated showed the best five performances on classifier as reflected from the Figures 5 and 6. So when we applied the VGG and our custom loss functions in our Pix2Pix GAN models, it not only improved the PSNR and SSIM values for data generated at each epoch as shown in the Figures 4a and 4b but the performance of the classifier also improved significantly as it can be seen from the Figure 6. The histograms in Figures 4c and 4d present PSNR and SSIM values of all images generated from the GAN model with our custom loss function trained for epoch 40. The data generated from this epoch yields the best performance in the classification for the multi-fault diagnosis. It can be seen that, here, for the majority of the images generated PSNR value is between 22 and 27 similarly the SSIM for the majority of images generated is between 0.96 and 0.98. It is an indicator that the GAN has learned for features for the majority of the

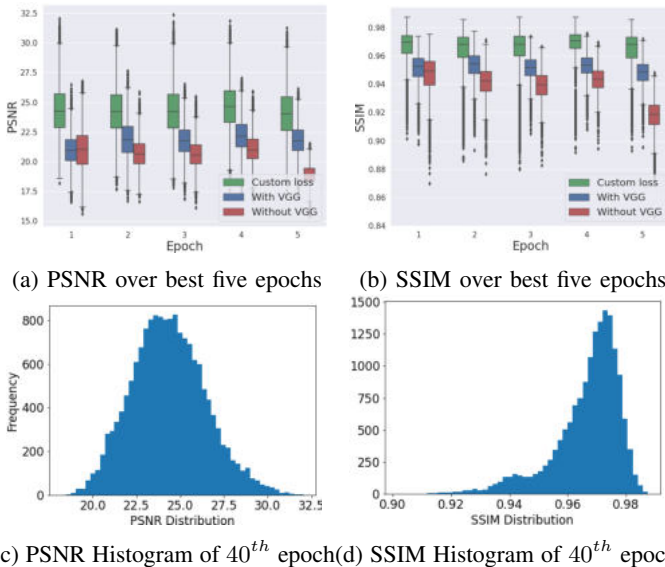


Fig. 4: SSIM and PSNR for data generated from Pix2Pix GAN

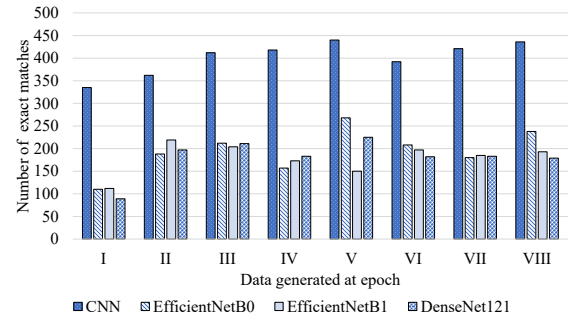


Fig. 5: Performance of classification models on VGG-GAN data

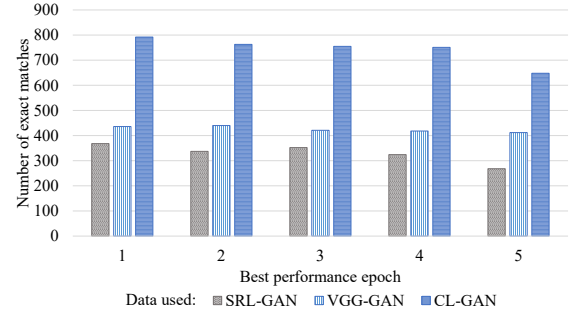


Fig. 6: Performance of CNN with different loss functions

images and there are fewer images for which GAN could not grab the desired information.

### B. Results of classification schemes

The performance of the classifiers is evaluated on the original complete map data, sparse data (with 1% data points), data generated from sparse data with Pix2Pix GAN using simple reconstruction loss function (SRL-GAN data), and data generated from Pix2Pix GAN with perceptual loss functions, one exploiting VGG model, labelled as VGG-GAN data, and the other with our Customized Loss Function (CL-GAN data) based on our own CNN classifier. The results of the some best performance predefined popular image-based classification models using transfer learning along with the results of our own CNN model, all on the data generated from VGG-GAN up to eight epochs, are presented in Figure 5. It can be seen that the CNN proposed outperforms predefined models. The performance of the predefined model is observed to be even more poor on the raw sparse data images, whereas CNN could identify all the classes in around 119 samples out of 3201 test samples when trained and evaluated on sparse data.

Since CNN has shown significantly better performance as compared to other models, therefore, further detailed results are presented for CNN only. Figure 6 shows the performance of CNN for correctly detecting all five classes when applied on the detests generated with Pix2Pix SRL-GAN, VGG-GAN, and CL-GAN. The results are presented for those five epochs where CNN has the best five performances, 1 for the first best and 2 for the second-best performance. Figure 6 clearly shows that the use of perceptual loss improves the quality of images and the performance of the classifier subsequently. By

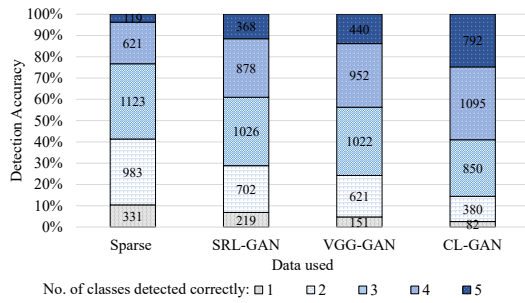


Fig. 7: Number of classes detected correctly by CNN on data generated from GAN with different loss functions

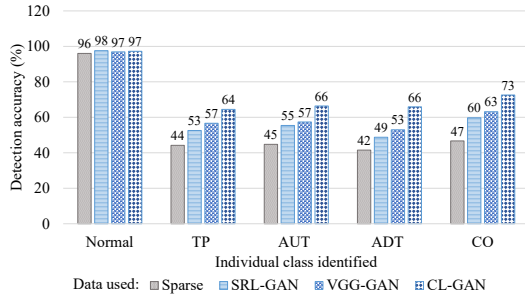


Fig. 8: Performance of CNN with different loss functions for the detection of individual class

detecting all five classes for 72 cells in 440 coverage maps, the use of the VGG-loss increased the exact match rate when compared to the data generated with simple SRL-GAN where for 368 coverage maps all classes were correctly predicted. Our customized loss function has lead to further significant improvement in the exact match rate by detecting all classes in 792 images out of 3201.

Figure 7 further reflects the potential of our augmentation and classification scheme. It shows that for the best quality data set generated with CL-GAN, the classifier could identify all classes including four faults in around 25% of the test samples, whereas it could identify four and three classes in almost 35% and 26% of the test samples. So overall it can be seen that in around 86% of cases at least three classes could be identified correctly. For the exact match detection rate for all classes only, it can be clearly seen that our proposed data-augmentation scheme comprising Pix2Pix CL-GAN, yields a gain of around 115% as compared to data generated by SRL-GAN and a gain of around 565% as compared to all fault detection rate on raw data.

The detection accuracy of each individual class is shown in Figure 8, where it can be seen that the normal cell behavior could be detected easily even when CNN is applied on the raw data leading to around 96% detection accuracy. The other comparatively easily detected fault is coverage outage which is detected with an accuracy of around 73% for the CL-GAN data. The rest of the faults have almost the same detection rate of around 65% for the CL-GAN data.

## VI. CONCLUSION

In this study, we have presented a comprehensive scheme comprising a unique data augmentation method for addressing data sparsity and a classification model to diagnose multiple faults in the network. As part of the data augmentation scheme, we have introduced a customized perceptual loss that helps a Pix2Pix-GAN model generate images of high quality with PSNR and SSIM values of around 25 and 0.97 respectively. We have evaluated the performance of our augmentation scheme using a CNN model that yields a gain of 550% in the detection of all five classes, including four faults as compared to when it is applied on the sparse data sample with 1% of the information available. Our proposed scheme can not only help in addressing the data sparsity challenge in MDT but also provides a solution for the multiple-fault diagnosis an important task in network performance management.

## ACKNOWLEDGMENT

This work is supported by the Qatar National Research Fund (QNRF) (a member of The Qatar Foundation) under Grant No. NPRP12-S 0311-190302. The statements made herein are solely the responsibility of the authors.

## REFERENCES

- [1] K. Koufos, K. Haloui, M. Dianati, M. Higgins, J. Elmirghani, M. Imran, and R. Tafazolli, "Trends in intelligent communication systems: Review of standards, major research projects, and identification of research gaps," *Journal of Sensor and Actuator Networks*, vol. 10, no. 4, p. 60, 2021.
- [2] A. Asghar, H. Farooq, H. N. Qureshi, A. Abu-Dayya, and A. Imran, "Entropy field decomposition based outage detection for ultra-dense networks," *IEEE Access*, 2021.
- [3] T. Zhang, K. Zhu, and D. Niyato, "Detection of sleeping cells in self-organizing cellular networks: An adversarial auto-encoder method," *IEEE Transactions on Cognitive Communications and Networking*, 2021.
- [4] J. B. Porch, C. H. Foh, H. Farooq, and A. Imran, "Machine learning approach for automatic fault detection and diagnosis in cellular networks," in *2020 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*. IEEE, 2020, pp. 1–5.
- [5] A. Rizwan, J. P. B. Nadas, M. A. Imran, and M. Jaber, "Performance based cells classification in cellular network using cdr data," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–7.
- [6] W. Zhang, R. Ford, J. Cho, C. J. Zhang, Y. Zhang, and D. Raychaudhuri, "Self-organizing cellular radio access network with deep learning," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2019, pp. 429–434.
- [7] L. Wang, W. Chen, W. Yang, F. Bi, and F. R. Yu, "A state-of-the-art review on image synthesis with generative adversarial networks," *IEEE Access*, vol. 8, pp. 63 514–63 537, 2020.
- [8] R. Wang, Z. Fang, J. Gu, Y. Guo, S. Zhou, Y. Wang, C. Chang, and J. Yu, "High-resolution image reconstruction for portable ultrasound imaging devices," *EURASIP Journal on Advances in Signal Processing*, vol. 2019, no. 1, pp. 1–12, 2019.
- [9] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690.
- [10] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.
- [11] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European conference on computer vision*. Springer, 2016, pp. 694–711.

# Edge-Computing based Secure E-learning Platforms

Sameer Ahmad Bhat<sup>1,2</sup>, Dalia Alyahya<sup>3</sup>, Muneer Ahmad Dar<sup>4</sup> and Saadiya Shah<sup>4</sup>

<sup>1</sup>Graduate Studies and Research, Gulf University for Science and Technology (GUST), Kuwait.

<sup>2</sup>Dept. of Multimedia Systems, Gdansk University of Technology, Pomerania Gdansk, Republic of Poland.

<sup>3</sup>Dept. of Instructional Technology, King Saud University (KSU), Riyadh, Saudi Arabia.

<sup>4</sup>National Institute of Electronics and Information Technology (NIELIT), Jammu & Kashmir, India.

Email(s): bhat.s@gust.edu.kw, dmalyahya@ksu.edu.sa, muneer@nielit.gov.in, shah.saadiya@gmail.com

**Abstract**—Implementation of Information and Communication Technologies (ICT) in E-Learning environments have brought up dramatic changes in the current educational sector. Distance learning, online learning, and networked learning are few examples that promote educational interaction between students, lecturers and learning communities. Although being an efficient form of real learning resource, online electronic resources are subject to threats and vulnerabilities on the internet. Authentication, access and storage of data is a major concern among many organizations implementing E-learning platforms. This study provides a literature review of past five-year research studies, and proposes Edge-computing based solution to the currently existing authentication and data access problems that prevail in the current E-learning management systems using cloud services for data storage. The study guides researchers towards enabling Edge-computing based E-learning platforms to support low power computing devices running Elliptic Curve Cryptography for secure access and authentication.

**Index Terms**—E-learning, Cryptography, Edge-Computing, Cloud Computing, Cryptography, ECC, ECDH, Instructional Technology

## I. INTRODUCTION

Rapid E-Learning technology progression is currently shaping the methodology of how the teaching and learning practices are carried out in today's educational environments [1]. The report by [28] states that "COVID-19 pandemic is expected to positively impact the growth rate of the e-learning market, owing to increase in adoption of digital technologies among various schools, colleges and universities across the globe and growing government support for improving e-learning platform across various developing nations of Asia-Pacific and LAMEA countries". Previously valued at \$197.00 billion in 2020, the global e-learning market size is projected to reach at level of \$840.11 billion by 2030, thereby registering a CAGR of 17.5% from 2021 to 2030. While several definitions of eLearning have been proposed, generally agreed definitions state that eLearning employs computers and several other instruments of information communication technology (ICT) to enable support to and facilitate in the ongoing teaching and learning processes [2]. With focus on augmented technologies and boom in information access, E-learning has been widely accepted as an essential platform of learning since it allows learners to acquire knowledge and skills ubiquitously, whilst in the physical absence of a mentor(s) or teacher(s) [3]. Typically, E-learning makes use of computing technologies, primarily connected over an intranet



Fig. 1. Asia-Pacific would exhibit the highest CAGR of 17.4% during 2021-2030 [28].

or Internet, to deliver information and instructions to individual learners [21]. While this broader definition is interesting, E-learning in this context is referred to as training delivered via network technology. Here the term 'training' refers to planned efforts that increase job-related knowledge and skills [4].

Academic or non-academic organizations, or even industry experts cover knowledge management and virtual collaboration in their definition [22]. E-learning in this case is described to broadly include any system that generates and disseminates information and is designed to improve learners' performance [5]. Alternatively, E-learning is also considered as a disruptive technology since it transforms practices of how learning is approached in an educational context [6]. In the times of CoVID-19, most of the education systems are in future envisioned to change entirely with the development of newer innovate E-Learning platforms, in particular the quality e-education services and supporting processes.

E-Learning systems mainly have five significant participants – Authors, Students, Managers, Teachers and System Developer (System Administrators). Apart from that, unauthorized users, basically hackers may attempt to gain illegal access to the E-learning systems to steal critical resources [7]. In present E-learning systems, managers or instructors communicate with learners, and share resources that require elevated security and

secure data communication methods and protocols, thereby prevent unauthorized and unprecedented access to services by unrecognized users. Hackers can alter or modify E-Learning resources such as learning materials, certificates, question papers, lecture materials, mark sheets etc. [8]. To leverage the IoT features in the E-Learning systems, the study by [24] highlights vital security issues that significantly influence the ways, how IoT layered architectures are structured, and how the vulnerabilities exhibited by the Cloud networking services severely pose threats data and information security. Apparently, vulnerabilities as such lead to inefficient and nonfunctional service thereby expose critical information to the malevolent users. This could pose a severe threat to the E-learning systems, and E-learning data management teams must highly secure the E-learning system resources [9].

From our literature review, we observe the different definitions of E-learning. For example, the definition by [10] finds four dimensions Fig. 2 to define the concept of E-learning:

- Technology-driven: Use of technology to deliver learning and training programs;
- Delivery-system-oriented: The delivery of a learning, training, or education program by electronic means;
- Communication-oriented: Learning facilitated by the use of digital tools and content that involves some form of interactivity, which may include online interaction between the learner and their teacher or peers; and
- Educational-paradigm-oriented: Information and communication technologies used to support students to improve their learning.

## II. CURRENT TRENDS – ENCRYPTION STANDARDS IN E-LEARNING

Though being efficient, online e-learning resources available on the internet are prone to threats and vulnerabilities on the internet. E-learning systems as such must satisfy the basic concepts of data security – integrity, confidentiality and availability [11]. Confidentiality – aims to ensure privacy to data and information. Data and information, both are kept secret and private, and prevented from unauthorized access by people, application processes or any hardware devices. Integrity – aims to ensure originality, correctness, and accuracy of data and information by preventing it from any accidental losses, or intended malicious attacks to update or modify the original content. Data and information must remain in original structure and format. Availability – aims to ensure access to reliable data, information, and communication in a timely manner to authorized users. Non-repudiation assures that user who carry out operations in a system, cannot deny their actions. For example, if a user deletes his/her learner's results, and then denies the act, then the system should provide necessary log files pertaining to operations carried out by the user, so as to back track or trace performed operations on the system. Moreover, tamper-proof and reliable log files must be retained, and system auditing allows to fulfil this requirement. A digital signature is a countermeasure for non-repudiation.

Typical encryption algorithms applied to secure file systems are: Data Encryption Standard (DES), Advanced Encryption Standard (AES), Blowfish, Chaos Approach. Based on review

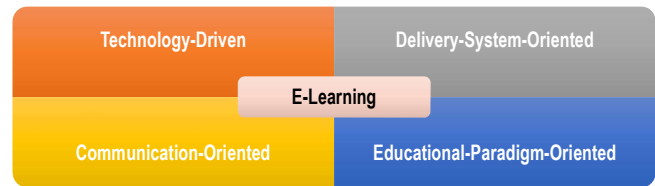


Fig. 2. Sangrà et al. [10]

of literature it is found that each of these algorithm is giving varying performance as context changes and there are some constraints observed among few of these algorithm. Various algorithms have been used to encrypt data in E-learning systems. For example, Ambalika Ghosh at al. [12] propose an object-oriented modelling approach, and implement International Data Encryption Algorithm (IDEA) to show how privacy and confidentiality of information, communicated between a teacher and student at the time of displaying marks scored for a course can be achieved. IDEA is a symmetric-key block cipher, and operates on 64-bit blocks using a 128-bit key and consists of a series of eight identical transformations. IDEA, as an encryption–decryption technique can protect roll number or marks for any changes made by the hackers.

Nikhilesh Barik et al. [13] propose framework implement unique authorization with a Two-Level Access Control (TLAC). Shared secret key encrypts a service request and public key infrastructure guarantees the confidentiality of message during transmission. If any document was altered by the malicious users, the receiver would get the different digests for the original message. So the integrity and non-repudiation is achieved by using digital signature. In case deletion of any documents from the managers' side, it is possible to trace back with archived log files. Tamper proof mechanism may be used to solve the problem.

Ahmad Baihaqi et al. [14] have implemented AES 256-bit for document encryption, RSA 2048-bit for digital signature, and SHA 256 bit for message digest in PHP and JavaScript programming language. The study designs a Secure Electronic Learning System (SELS) application that supports basic security features – confidentiality, integrity, authentication, and non-repudiation.

Chuyang Li et al. [8] propose Blockchain technology based solutions to problems in online education, wherein it is used for e-learning assessment and certification. The study also proposes a network structure using combined public and the private Blockchain. New network structure using combined public and private Blockchain eliminates limitation of a single node role in traditional single-Blockchain systems with high flexibility, and it also fully retains the security and credibility of the Blockchain technology.

Karima et al. [15] propose a plug-in named EL-Security checker that enable controlling, verifying and eliminating attacks in e-learning platforms. The study proposes a solution that analyzes, verifies and checks authentication attacks. A new layer, called El-Security Layer is formulated to control and verify authentication vulnerabilities, affecting the e-learning system. Module to module communication is established using the Request-Response model.

Jegatha Deborah L et. al [16], propose a simple mutual authentication dialogue between the students and the server. The online examination system develops a secure system to distribute and collect the question papers and answers scripts to and from the students, respectively. At the end of examination, each student submits its response back to the server. S. Kanimozhi et. al [17] have implemented a cloud-based e-learning system employing access control mechanism that prevents illegal user access to the resources of cloud. The study discusses key management schemes in combination with access control technique to secure share content and enable protection to e-learning environment. Findings show that cloud services are secure, flexible, and scalable in cloud-based e-learning.

Priyanka Saxena et al. [18] propose an intuitive authorization mechanism implementing customized set of rules and authorization key-based mechanisms that increases the level of security. The study develops a new system built with training machines to: a) identify unauthorized user accesses, and b) prevent theft or mal-processes if any user is authenticated by other means. Post to authentication, authorization is carried out using modified role-based access control. Key encryption uses SHA256 hash generation working one way only, and cannot be reversed by anybody in the system, though by users who leak the key.

We conducted a literature review of articles published over the past five years. From the review, we observe that most of the research studies have focused on developing frameworks and improving content organization in E-learning systems. However, just few research studies have attempt to provide an insight into the security issues prevailing in the E-learning systems.

#### A. Problem Statement

Qualitative analysis of recent articles referenced in the prior section shows that:

- the basic security mechanisms have been implemented mostly on server side, and clients are expected to own their devices with medium to high level of computational power, in order to access available E-learning resources online.
- Authentication mechanisms employ encryption algorithms to allow users gain access to the E-learning systems. Encryption algorithms typically used are computationally expensive in terms of required hardware resources. For example, algorithms, such as Rivest–Shamir–Adleman (RSA) may offer secure solutions that are often difficult to comprise, these in turn demand high speed computational devices for efficient operation.
- The power of Edge-computing [19] has been overlooked completely to process data locally, and mostly cloud servers are employed for access to data and storage.

### III. SECURE E-LEARNING SYSTEMS

In the following sub-sections, we propose solutions to the previously stated challenges. The first part addresses the need

to reduce the processing load on the authentication server. The second and the third part address the need to reduce computational complexity of security mechanism, as well as to ensure local availability of data.

#### A. Light weight Cryptography

Despite more appealing features, researchers, techno-savvy and educated class of smartphone users are highly concerned about the security of data stored in either smartphones or other mobile devices such as laptops or Tablet computers, and the way how confidentiality is ensured when such mobile devices exchange data with each other or communicate over a network susceptible to intruders. Typically, learners employ such low power computational devices to gain access to E-learning resources, and this requires encryption of authentication and authorization data. Symmetric encryption algorithms such as DES, 3DES, ADES, and others, and asymmetric encryption algorithms which include RSA, Diffe-Hellman, ECC, and others, are traditional encryption algorithms [20]. Data encryption by these algorithms show low operability, posing obstacles in subsequent data processing.

To ensure, low power computational devices are supported, we propose Elliptical Curve Cryptography (ECC) as the best possible solution since it offers the same level of encryption/decryption with just a key size of 210 bits compared to the level of security offered by RSA that uses a long key size of 2048 bits (see Table I). Encryption key used in ECC conceals secret data so prevent an unidentified user to decipher its contents. The enciphering process of hides the plain text data, and the encryption process results cipher text as its output. While in engaged in communication, the encipherer decodes the message to be communicated and generates a cryptogram. The cryptogram is transmitted and sent to the recipient.

Elliptic Curve Cryptography (ECC) as an asymmetric, public key cryptographic technique, allows communicating devices to generate two keys – a public key and a secret key called the private key. The public key is distributed to all the devices, and the private key is hidden and kept secret by the client encrypting or decrypting the message [23].

**Definition 1 (Elliptic Curves).** Let  $P$  represents the field of characteristic  $\neq 2, 3$ , then an elliptic curve  $\mathbb{G}_P$  defined over the  $P$  consists of the set of elements  $(x, y) \in P^2$ , that satisfy the eq(1), which is the short Weierstraß equation of an elliptic curve.

An Elliptic curve  $\mathbb{G}$  over  $\mathbb{G}_P$  is a set of all solutions  $(x, y) \in \mathbb{P} * \mathbb{P}$  to an equation

$$y^2 = x^3 + cx + d \quad (1)$$

where  $(a, b) \in P$  satisfy the relation  $-16(4a^3 + 27b^2) \neq 0$ , represents quantity  $\Delta$ , the discriminant of eq(1).  $\Delta \neq 0$  denotes the singularity of the curve  $\mathbb{G}_P$ .

The elliptic curves  $\mathbb{G}_P$  consists of elements called points. Fig. 3 and Fig. 4 show the elliptic curves defined over the two finite fields  $F_{1021}$  and  $F_{16381}$ .

### IV. EDGE-COMPUTING ENABLED AUTHENTICATION AND DATA ACCESS

Edge computing (EC) is an extension of cloud computing with its own characteristics that are different than the

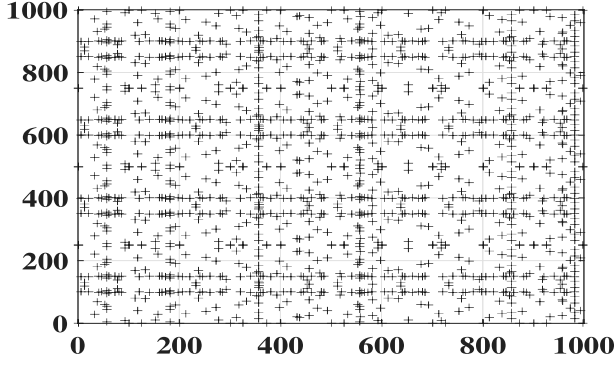


Fig. 3.  $G_{P_{1021}} : y^2 = x^3 - 3x + 3$  [23]

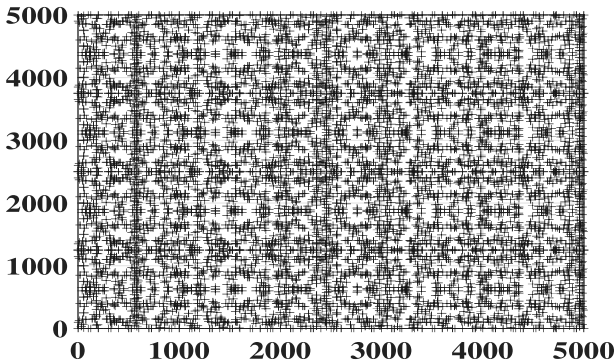


Fig. 4.  $G_{P_{16381}} : y^2 = x^3 - 3x + 3$  [23]

characteristics of cloud computing. Cloud servers process large amounts of data, offer in-depth analysis on data, and even support real-time or non-real times solutions in business decision making processes. EC enables storage of data on locally available devices, mostly close to the information source, and thus the need to upload data to Cloud servers is eliminated. Bandwidth, delay, and jitter are easily controllable and improve due to the reduction of network size. In the case of E-learning, network bandwidth is significantly improved with reduced network size. EC is employed in small-scale intelligent analysis and local services, while Cloud computing supports centralized processing of large-scale data. EC extends a small-scale role compared to Cloud services and allows real-time intelligent analysis and processing of data locally on devices supporting computing at Edge networks. While being close to the source and feasible, at the edge of a networks, EC systems enable platforms that allow storage to consumer data and access features. Healthcare improvisation, Network optimization processing, and data transportation are some of services managed by EC devices currently.

Digital transformation needs intuitive solutions to drive the current processes and methodologies implemented in Industry Revolution 4.0 (IR). IR demands data sensing mechanisms implementing Industrial Internet of Things (IIoT), Blockchain, Cyber Physical Systems (CPS), Digital Twin, and high speed reliable access to internet provided by state-of-art technolo-

TABLE I  
COMPARISON OF KEY LENGTH (IN BITS)

RSA- key	ECC - key	Proportion of RSA/ECC
512	106	5:1
768	132	6:1
1024	160	7:1
2048	210	10:1

gies, such as 5G or 6G. These are basically the core drivers of digital transformation in industries looking forward to developing Smart systems. However their role in academics is still at infancy stage, though exceptions are there, for instance Edge Computing has significantly impact the ways in which academia and industries have emerged in the recent times, especially during CoVID-19 pandemic.

The Internet of Things (IoT), virtual worlds, and vehicle-to-vehicle interactions are mostly realized to rely on the Cloud computing infrastructures as these enable access to end users at ease and ubiquitously [25]–[27]. EC systems basically integrate the advanced capabilities of IoT and 5G networking infrastructure. As such EC systems reveal features, such as low network bandwidth requirement, portability and safety of edge devices. While functional and operative at network's edge, EC systems may expose various computational, storage, and networking services to the end users. Consumers of EC system services can access data and other system serves, whereas the edge software can be developed and managed in a short span of time compared to cloud services that often need longer times to ensure highly efficiency.

As from the literature review, typical authentication processes in E-learning systems require sending users' data to the server for authentication, and then servers respond with a validation response. With prolific users connected to a system, large authentication processes overload the server, thereby resulting in slow response times or even denial of requests by the server. Though in usual user authentication processes, the current systems may run without showcasing any serious implications, however when it comes to conducting events, such as large scale online exam, demand reliability and exhibit serious concerns.

EC offers the optimized solution to overcome this problem. The resources in the EC and users of edge network are in the close proximity (e.g., location), this lead EC to enable personalized services for users, as per the current scenarios, such as location-based services. We propose a two stage authentication process Fig 5, wherein Edge manager first locally authenticates users, and then communicates authentication statistics to the main Cloud server, or other servers used in the authentication process. Many trust domains can coexist since edge computing serves as a source of distributed interactive computing environment. Assignment of an identity to each entity is essential within the trust domains and even for mutual authentication, when different trust domains coexist. Cross-domain authentication and handover authentication technologies offer support to data and privacy security of users in different trust domains and heterogeneous network



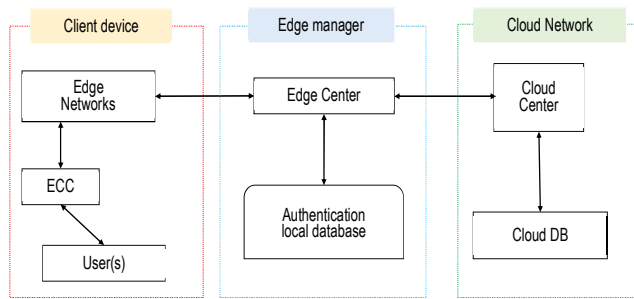


Fig. 5. Authentication process using Edge-Computing.

environments [19]. EC group specific databases can be stored locally on EC networks and then users assigned to those groups can be authenticated without the need from the actual authentication server. This would significantly reduce the load on the authentication server, and response times from the main authentication server can improve drastically.

## V. CONCLUSIONS

This study provides a review of the recent security methods, prevailing in the existing E-learning management systems. Research studies of past five years are evaluated to determine the current trends and techniques that are employed to enable security in online learning platforms. Edge-computing as an emerging solution is proposed in this study, to address the challenges of authentication and data access methods in E-learning systems. We envision that Edge networks can act as proxy servers enabling local authentication and authorization to resources existing on the actual Cloud server. This would enable dual authentication as well as add a new local security layer to the existing learning management networks. In our future studies, we aim to implement the concept in a real time scenario to observe the real time characteristics of the proposed system.

## REFERENCES

- [1] H. Lucas and J. Kinsman, "Distance- and blended-learning in global health research: potentials and challenges," *Glob. Health Action*, vol. 9, no. 1, p. 33429, Dec. 2016, doi: 10.3402/gha.v9.33429.
- [2] R. Phillips, G. Kennedy, and C. McNaught, "The role of theory in learning technology evaluation research," *Australas. J. Educ. Technol.*, vol. 28, no. 7 SE-Articles, Aug. 2012, doi: 10.14742/ajet.791.
- [3] A. Moubayed, M. Injadat, A. Shami, and H. Lutfiyya, "Student engagement level in an e-learning environment: Clustering using k-means," *Am. J. Distance Educ.*, vol. 34, no. 2, pp. 137–156, 2020.
- [4] E. T. Welsh, C. R. Wanberg, K. G. Brown, and M. J. Simmering, "E-learning: emerging uses, empirical results and future directions," *Int. J. Train. Dev.*, vol. 7, no. 4, pp. 245–258, Dec. 2003, doi: <https://doi.org/10.1046/j.1360-3736.2003.00184.x>.
- [5] M. J. Rosenberg and R. Foshay, "E-learning: Strategies for delivering knowledge in the digital age." Wiley Online Library, 2002.
- [6] D. R. Garrison, *E-learning in the 21st century: A community of inquiry framework for research and practice*. Taylor & Francis, 2016.
- [7] R. Bansal, A. Gupta, R. Singh, and V. K. Nassa, "Role and Impact of Digital Technologies in E-Learning amidst COVID-19 Pandemic," in 2021 Fourth International Conference on Computational Intelligence and Communication Technologies (CCICT), 2021, pp. 194–202.
- [8] C. Li, J. Guo, G. Zhang, Y. Wang, Y. Sun, and R. Bie, "A blockchain system for E-learning assessment and certification," in 2019 IEEE International Conference on Smart Internet of Things (SmartIoT), 2019, pp. 212–219.

- [9] A. A. Keshlaf, A. A. Alahresh, and M. K. H. Aswad, "Factors Influencing the Use of On-Line Meeting Tools," in 2021 IEEE 1st International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering MI-STA, 2021, pp. 908–912.
- [10] A. Sangrá, J. E. Raffaghelli, and M. Guitert-Catasús, "Learning ecologies through a lens: Ontological, methodological and applicative issues. A systematic review of the literature," *Br. J. Educ. Technol.*, vol. 50, no. 4, pp. 1619–1638, 2019.
- [11] S. Kausar, X. Huahu, A. Ullah, Z. Wenhao, and M. Y. Shabir, "Fog-Assisted Secure Data Exchange for Examination and Testing in E-learning System," *Mob. Networks Appl.*, 2020, doi: 10.1007/s11036-019-01429-x.
- [12] A. Ghosh and S. Karforma, "Object-oriented Modeling of IDEA for E-learning Security," in *Intelligent Computing and Applications*, Springer, 2015, pp. 105–113.
- [13] N. Barik and S. Karforma, "Secure e-Learning Framework (SeLF) BT - Information Systems Design and Intelligent Applications," 2015, pp. 691–698.
- [14] O. C. Briliyant and A. Baihaqi, "Implementation of RSA 2048-bit and AES 128-bit for Secure e-learning web-based application," in 2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA), 2017, pp. 1–5, doi: 10.1109/TSSA.2017.8272903.
- [15] K. Aissaoui and M. Azizi, "El-Security: E-learning Systems Security Checker Plug-in," in *Proceedings of the 2nd international Conference on Big Data, Cloud and Applications*, 2017, pp. 1–6.
- [16] J. D. L. K. R., V. P., B. S. Rawal, and Y. Wang, "Secure Online Examination System for e-learning," in 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), 2019, pp. 1–4, doi: 10.1109/CCECE43985.2019.9052408.
- [17] S. Kanimozhi, A. Kannan, K. Suganya Devi, and K. Selvamani, "Secure cloud-based e-learning system with access control and group key mechanism," *Concurr. Comput. Pract. Exp.*, vol. 31, no. 12, p. e4841, 2019.
- [18] P. Saxena and H. Sanyal, "Improved Rules and Authorization Key Processing for Secured Online Training," in 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), 2020, pp. 690–694, doi: 10.1109/ICECA49313.2020.9297463.
- [19] K. Cao, Y. Liu, G. Meng, and Q. Sun, "An overview on edge computing research," *IEEE access*, vol. 8, pp. 85714–85728, 2020.
- [20] Dar, M. A., Askar, A., Alyahya, D., & Bhat, S. A. (2021). Lightweight and Secure Elliptical Curve Cryptography (ECC) Key Exchange for Mobile Phones. *International Journal of Interactive Mobile Technologies (ijim)*, 15(23), pp. 89–103., doi: 10.3991/ijim.v15i23.26337.
- [21] M. A. Dar and S. A. Bhat, "Evaluation of mobile learning in workplace training," 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2016, pp. 1468–1473, doi: 10.1109/ICACCI.2016.7732255.
- [22] Askar, A. Mobile Electronic Performance Support System as a Learning and Performance Solution: A Qualitative Study Examining Usage, Performance, and Attitudes. *Turkish Online Journal of Educational Technology*, 17(2), (2018), pp. 76–88.
- [23] Djath, L., 2021. RNS-Flexible hardware accelerators for high-security asymmetric cryptography (Doctoral dissertation, Université de Bretagne occidentale-Brest).
- [24] Shah JL, Bhat HF, Khan AI. Integration of Cloud and IoT for smart e-healthcare. In *Healthcare Paradigms in the Internet of Things Ecosystem 2021 Jan 1* (pp. 101-136). Academic Press.
- [25] Bhat SA, Dar MA, Elalfy H, Matheen MA, Shah S (2021). A Novel Framework for Modelling Wheelchairs under the Realm of Internet-of-Things, *International Journal of Advanced Computer Science and Applications(IJACSA)*, 12(2), (2021). <http://dx.doi.org/10.14569/IJACSA.2021.0120293>
- [26] Belchior R, Vasconcelos A, Guerreiro S, Correia M. A survey on blockchain interoperability: Past, present, and future trends. *ACM Computing Surveys (CSUR)*. 2021 Oct 4;54(8):1-41.
- [27] Rover DT, Mina M, Herron-Martinez AR, Rodriguez SL, Espino ML, Le BD. Improving the Student Experience to Broaden Participation in Electrical, Computer and Software Engineering. In *2020 IEEE Frontiers in Education Conference (FIE) 2020 Oct 21* (pp. 1-7). IEEE.
- [28] Allied Market Research. <https://www.alliedmarketresearch.com/e-learning-market-A06253>, 2021. (accessed on: 20.01.2022).

# Efficient classification of human activity using PCA and deep learning LSTM with WiFi CSI

Sang-Chul Kim

Department of Computer Science  
Kookmin University  
Seoul, Rep of Korea  
sckim7@kookmin.ac.kr

Yong-Hwan Kim

Department of Computer Science  
Kookmin University  
Seoul, Rep of Korea  
brightface@kookmin.ac.kr

**Abstract:** Currently, we are entering the wearable internet era. It is easy to find network access points (APs) in today's world. APs can be useful for more than just connecting to the internet. The waveform of WiFi signal changes when there is a person or human action between the two APs. In previous studies, we described how changes in waveforms affect the channel state information (CSI) of a signal and how machine learning can use this information to recognize and predict human behavior. In this study, we present points that can be improved on the previous study and the solution to filter the information. Preprocessing is one method for increasing performance. We found that using principal component analysis (PCA) reduced the processing time by 50% when classifying some classes. General-purpose PCA is useful for classifying data by reducing its dimensions.

**Keywords:** LSTM; CSI; PCA; Wearable device; RNN; Smart Home; Smart device;

## I. INTRODUCTION

With recent advancements in the Internet of Things technology, mobile devices are receiving WiFi signals in many places moving around the city.

It is not a good idea to waste Wifi access point (AP) signals, which could be useful for many things. Network AP transmits channel state information (CSI), which contains information from Wifi signals. We can use and advance this information to develop many skills. This CSI shows the distribution channel path of transmitted signals. They can be reflected or diffracted by objects such as buildings and vehicles.

Therefore, there is a problem with hardware noise or weak signal fading of the signal. Using off-the-shelf network interface cards (NICs) that were not designed to measure CSI introduces hardware-induced noise. Additionally, predicting through these signals takes a significant amount of time and money, which can save

time and money by reducing dimensionality using principal component analysis (PCA), which is one of many data filtering techniques.

We used two Intel WiFi Link 5300 wireless NICs to extract CSI data and the transceiver used three antennas to receive the WiFi signal. If there is a person between the sending and receiving locations, the transmitted WiFi signal will be reflected and the information will be extracted differently. Previously, the signal was transmitted over a new path from a transmission location. The CSI of the receiving antenna has unique characteristics in the following structure.  $A \in C^{(N_{sa} \times N_{sc} \times N_{tx} \times N_{rx})}$ , where  $N_{sa}$  and  $N_{sc}$  are the number of sampled WiFi packets and the number of subcarriers, respectively.  $N_{tx}$  and  $N_{rx}$  are the numbers of transmitting and receiving antennas, respectively. In our setup,  $N_{sc} = 30$ ,  $N_{tx} = 3$  and  $N_{rx} = 3$ .

At each sampling timestamp, we can use the CSI as a multichannel tensor ( $a_i \in C^{(30 \times 3 \times 3)}$ ,  $i \in [1, N_{sa}]$ ) to extract features from the data for machine learning. In this study, we use PCA filtering to develop a model capable of recognizing human behavior using long short-term memory (LSTM).

Unfiltered CSI data contains noise that is not required for human behavioral perception. Because the globe contains many objects in various environments, noise occurs because of the absence of hardware responsibility. We must increase the performance so that we can use these technologies in a practical environment using data preprocessing and filtering to ensure that the model has more performance for changes in surrounding objects.

Chapter 2 of this study describes the results and limitations of previous study topics. Because real-world WiFi channel state information-based human activity recognition and prediction (reference) experiments are conducted in a controlled environment, accuracy and

recognizing of activity speed cannot be guaranteed in various situations.

Chapter 3 covers the PCA used and how it reduces time to recognize human activity and compares the previous research performance.

Finally, Chapter 4 explains technical limitations and a discussion.

## II. PREVIOUS RESULT AND LIMITATION

A previous study (Kim, 2021) demonstrated that deep learning methodology is used for human behavior recognition. [1] Figure 1 shows a misclassification table indicating that the model is greater than 95% accuracy. The advantage of using deep learning over traditional machine learning is that deep learning can discriminate between a wider range of activities and is more changes-resistant or surroundings.

However, using LSTMs without preprocessing may cause the machine unable to train noise in real life or learn properly when a noise occurs.

In a large building's room rather than a small building's room, the signal dataset was not sufficient to extract features. Furthermore, it cannot capture noise properly or generate CSI that captures and removes unnecessary data because the NIC used in this experiment was not originally designed to collect CSI. There are two methods for increasing the accuracy of the deep learning model and the training speed of the data. The first step is to increase the size of the training dataset. The second is to preprocess the data to transform the data before training. In this study, we focus on filtering the dataset to minimize noise and extracting new features to improve the training speed.

	Walk	Empty	Sit down
Accuracy	93(%)	100(%)	97(%)

Figure 1

## III. METHODOLOGY

### 3.1. Principle Component Analysis

PCA is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables known as principal components. Therefore, it can be used to reduce dimension or for noise filtering. If there are  $n$  observations with  $p$  variables, the

number of distinct principal components is  $\min(n-1, p)$ . This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors form an uncorrelated orthogonal basis set. PCA is sensitive to the relative scaling of the original variables.

The PCA method has been mentioned in many studies, such as Keystroke Recognition using Wifi signals [2], understanding and modeling of WiFi signal-based human activity recognition [3], and wireless signal denoising model for human activity recognition [4]. Specially, we get data CSI filtering technic from Wifi related data with PCA in this study [5].

Figure 2 plots the amplitudes of CSI time series of four different subcarriers. The figure shows that all subcarriers show correlated variations in their time series. The subcarriers that are closely spaced in frequency show identical variations, whereas the farther subcarriers in frequency show nonidentical changes. Despite nonidentical changes, a strong correlation still exists even across the subcarriers with vastly different in frequencies. Human activity recognition system leverages this correlation and calculates the principal components from all CSI time series.

It then chooses those principal components representing the most common variations among all CSI time series.

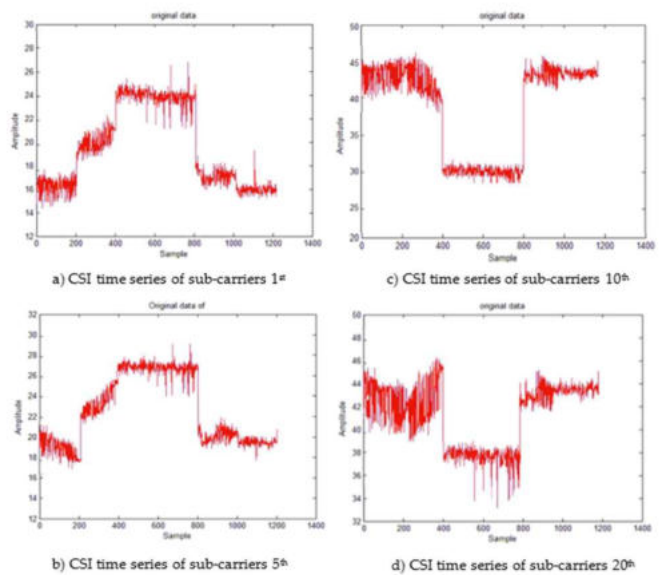


Figure 2. Correlated variations in subcarriers.

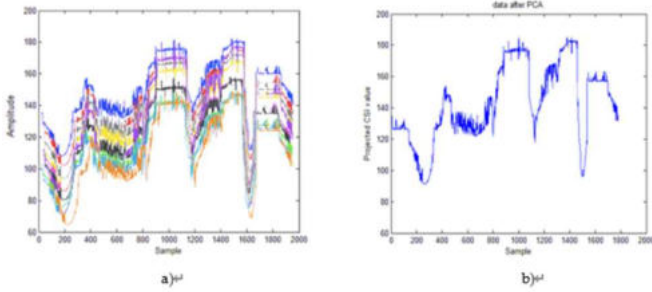


Figure 3. Sitdown

Figure 3 shows how the PCA is working visually by reducing dimensions and extracting the principal component. Figure 3(a) shows an original dataset and Figure 3(b) shows a principal component after processing. To get the principal component, from dataset  $X$ , let us define the principal component  $W_1$ , such that, we must maximize the variance.

$$W_1 = \arg \max_{\|W\|=1} E\{(W^T X)^2\} \quad (1)$$

$W^T$  is the matrix of basis vectors, one vector per column, where each basis vector is an eigenvector of the covariance matrix.

In Figure 4, the linear function is a basis vector, and blue spots are the dataset  $X$ , so  $W^T X$  means the variance of the projected vector, which finds the basis vector that maximizes the variance, then projects the dataset into a basis vector and is the principal component.

To calculate further components, the  $k$ th component can be found by subtracting the first  $k - 1$  principal component from  $X$ :

$$\hat{X}_k = X - \sum_{i=1}^{k-1} W_i W_i^T X \quad (2)$$

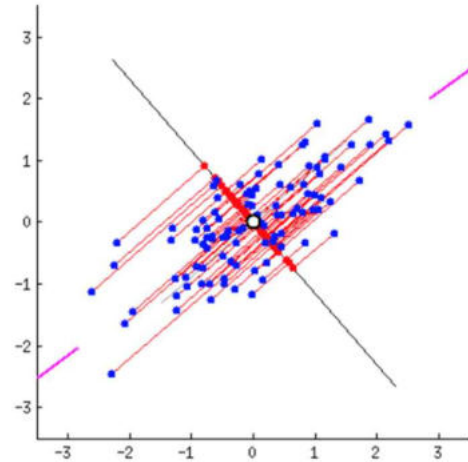


Figure 4. PCA

subtract this value from the dataset and then find the principal component again.

$$W_k = \arg \max_{\|W\|=1} E\{(W^T \hat{X}_k)^2\} \quad (3)$$

From the dataset  $X$ , subtract fractures that are already extracted, and find a  $k$ th component out of it. Then, the  $k$ th component can be found.

The result below shows that applying just PCA in a general case might cause significant loss of information in the wave and cause poor accuracy.

However, there is one benefit to using PCA, which is decreasing computational complexity. In a situation such as intrusion, which only requires two classes can keep the accuracy high enough.

	<i>default</i>	<i>PCA processed</i>
<i>Accuracy (4classes)</i>	90	72
<i>Accuracy (2classes)</i>	90	87

Figure 5. Accuracy Result

#### IV. CONCLUSION

PCA is a feature extraction algorithm that uses orthogonal transforms. The orthogonal transform reduces the dimensions to obtain the principal components. The main benefit of using PCA is reduced processing time. However, this method has a significant disadvantage in terms of accuracy. However, this method can be useful for recognizing some classes in hardware-constrained environments. For example, you can quickly check whether there is a person in a certain space, or whether action can be classified.

#### ACKNOWLEDGMENTS

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2022-2018-0-01396) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation

#### References

- [1] Y. Kim., "Recognizing human activity using deep learning with WiFi CSI and filtering," ICAIIC, 2021
- [2] L. A. W. W. Ali K., "Keystroke recognition using WiFi signals," ACM MobiCom, 2015.
- [3] L. A. S. M. Wang W., "Understanding and modeling of WiFi signal based human activity ecognition," ACM MobiCom, 2015.
- [4] H. S. K. Y. Chun-xiang WU, "A wireless signal denoising model for human activity recognition," AICS, 2016.
- [5] N. V. Steven Weber., "PCA-based statistical anomaly detection of reactive jamming in WiFi networks," IEEE-SP17\_Posters.

# MARL-based Optimal Route Control in Multi-AGV Warehouses

Ho-Bin Choi  
*Future Convergence  
Engineering*  
Korea University of Technology  
and Education  
Cheonan, South Korea  
chb3350@koreatech.ac.kr

Ju-Bong Kim  
*Future Convergence  
Engineering*  
Korea University of Technology  
and Education  
Cheonan, South Korea  
rlawnqhd@koreatech.ac.kr

Chang-Hoon Ji  
*Future Convergence  
Engineering*  
Korea University of Technology  
and Education  
Cheonan, South Korea  
koir5660@koreatech.ac.kr

Ullah Ihsan  
*Advanced Technology Research  
Center*  
Korea University of Technology  
and Education  
Cheonan, South Korea  
ihsan@koreatech.ac.kr

Youn-Hee Han  
*Future Convergence  
Engineering*  
Korea University of Technology  
and Education  
Cheonan, South Korea  
yhhan@koreatech.ac.kr

Se-Won Oh  
*Dept. of Knowledge-converged  
Super Brain Convergence  
Research*  
ETRI  
Daejeon, South Korea  
sewonoh@etri.re.kr

Kwi-Hoon Kim  
*Dept. of Artificial Intelligence  
Convergence Education*  
Korea National University of  
Education  
Cheongju, South Korea  
kimkh@knue.ac.kr

Cheol-Sig Pyo  
*Dept. of Knowledge-converged  
Super Brain Convergence  
Research*  
ETRI  
Daejeon, South Korea  
cspyo@etri.re.kr

**Abstract**—Automated guided vehicles (AGVs) are an essential component for automation fulfillment centers, a kind of warehouse. Efficient control of the AGV leads to easier management of inventory in the fulfillment center. To increase the productivity of various warehouses including fulfillment centers, we propose a Multi-Agent Reinforcement Learning (MARL)-based algorithm for cooperative control of AGVs. The proposed algorithm is based on a popular cooperative MARL algorithm, and utilizes an additional technique for path control of AGVs to distinguish the sacrifices of each agent and compensate them accordingly. We evaluate the proposed algorithm in comparison with a basic MARL algorithm on two fulfillment center layouts and provide further insight via the visualization of the results.

**Keywords**—AGV, Warehouse, Optimal Route, Deep Learning, Reinforcement Learning, MARL

## I. INTRODUCTION

The growth of e-commerce has caused many changes in the logistics market, and many companies have introduced fulfillment services to meet customer needs. In the online distribution industry, fulfillment service is a series of processes of picking, packing, and shipping products from warehouses to the customers according to their orders. These processes not only satisfy the needs of customers quickly, but also have many advantages for companies, such as delivery agency, inventory management, security, and fire insurance. Within the fulfillment center, systematic management is essential because numerous inventories must be moved in real-time.

The tasks performed in the fulfillment center such as picking, packing, and shipping can only be done by humans. However, simple transport of inventory is not dependent on humans, as automated guided vehicles (AGVs) can easily move heavy

inventory. Therefore, they are essential components for automation fulfillment centers, and their coordinated control leads to efficient management of inventory and increases warehouse productivity.

Meanwhile, most real-world problems including the multi-AGV warehouses occur when many entities cooperate or compete with each other rather than a single entity [1]. Multi-agent reinforcement learning (MARL) has achieved good results in various fields as in [2-4] and has three major frameworks: (1) fully centralized learning, which is a general framework utilized in single-agent reinforcement learning, but it has a fatal problem that the action space increases exponentially as the number of agents increases, (2) conversely, fully decentralized learning, which does not have the disadvantage of increasing the action space, but the non-stationarity problem is further exacerbated by the lack of communication between agents, and (3) centralized training with decentralized execution (CTDE) [5], which combines these two frameworks well, eliminates those two drawbacks and is suitable for decentralized systems.

In this paper, we adopt the CTDE framework-based MARL to increase productivity by systematically controlling the path of numerous individual AGVs moving autonomously within the AGV warehouse. To this end, we formulate the multi-agent environment modeled by the AGV warehouse as a decentralized partially-observable markov decision process (Dec-POMDP) [6]. In addition, we present state and observation representation, action representation, and reward function along with a description of the modules constituting the system and an overall scenario. The algorithm proposed in this paper has the ability to recognize the contributions or sacrifices of individual agents. Experiments are presented with results for three metrics and we provide further insight via visualization of the results.

## II. RELATED WORK

### A. Automated Guided Vehicles in Warehouses

The Amazon Robotics, formerly Kiva Systems, deals with resource allocation problems including decision making under uncertainty, robot path planning, and scheduling in the AGV warehouse. Fortunately, they provide a natural multi-agent AGV warehouse scenario for coordinated autonomy and decentralized decision making in [7]. The insights in [7] have encouraged research across various fields, among them we focus on deep reinforcement learning. The path planning problem of multi-AGV can be interpreted in various ways and a lot of algorithms have been proposed as in [8-10]. Recently, studies to control the path of multi-AGV in real time by applying reinforcement learning are being attempted, achieving good results that outperform traditional algorithms [11-12]. To the best of our knowledge, there are no papers that have studied path control or path planning of multi-AGV using multi-agent reinforcement learning.

### B. Multi-Agent Deep Reinforcement Learning

In general, it is natural to use Q learning with DQN to single agent RL [13-14]. This Q learning can be simply extended to Independent Q-learning (IQL) to be applied to multi-agent RL [15]. However, a better approach is needed for Q-learning to achieve good performance in multi-agent RL with stronger non-stationarity. The value deposition networks (VDN) estimate the joint action value of the multi-agent through  $Q^{tot}$  calculated by adding  $Q^a$  of all agents [16]. This additivity constraint is relaxed by QMIX as a monotonicity constraint [17]. QMIX estimates  $Q^{tot}$  in a complex non-linear method using MLP rather than in a simple summation method. In addition, QMIX can represent a richer joint action value because it allows extra state information to be used in the mixing network. The mixing network estimating this joint action value  $Q^{tot}$  guarantees the monotonicity constraints via non-negative weights. Since the factorization of VDN and QMIX facilitates the decentralized execution of multi-agents, they are suitable as marl algorithms to deal with real-world problems.

## III. SYSTEM MODEL

We consider a multi-agent environment to solve a task that requires collaboration between agents. This can be formulated as a Decentralized Partially-Observable Markov Decision Process (Dec-POMDP) [6], represented by a tuple  $\langle A, S, O, Z, U, P, R_g, R_l, \gamma \rangle$ . At each time step  $t$ , the environment outputs a global state  $s_t \in S$ , and observations  $o_t^a = O(s_t, a) \in Z$ , where  $S$  denotes the global state space,  $Z$  denotes the observation space for each agent  $a \in \{1, \dots, n\} \equiv A$ , and  $O: S \times A \rightarrow Z$  denotes the observation function respectively. Each agent uses its own observation to select an action  $u_t^a \in U$ , where  $U$  denotes the action space for each agent  $a \in \{1, \dots, n\}$ . The environment performs the joint action  $\mathbf{u}_t \in \mathbf{U} \equiv U^n$  of all agents and results in a transition according to the state transition function  $P(s_{t+1}|s_t, \mathbf{u}_t): S \times \mathbf{U} \times S \rightarrow [0, 1]$ . This transition is completed with a global reward  $\mathbf{r}_t = R_g(s_t, \mathbf{u}_t)$  and individual rewards  $r_t^a = R_l(s_t, u_t^a)$ , where  $R_g: S \times \mathbf{U} \rightarrow \mathbb{R}$  denotes the global reward function and  $R_l: S \times U \rightarrow \mathbb{R}$  denotes the individual reward function respectively.

In this process, each agent will try to increase its own individual reward which is aggregated into the team reward, ultimately aiming to maximize the discounted cumulative global reward  $\sum_{i=0}^{\infty} \gamma^i r_{t+i}$ . Each agent generates its own observation-action history  $\tau^a \in T \equiv (Z \times U)^*$  with a stochastic policy  $\pi^a(u^a|\tau^a): T \times U \rightarrow [0, 1]$  as a condition, where  $(Z \times U)^*$  represents the set of all possible observation-action histories. The joint policy  $\pi$  consisting of each agent's policy  $\pi^a$  has a joint action-value function, which is formulated as:

$$Q^\pi(s_t, \mathbf{u}_t) = \mathbb{E}_{s_{t+1: \infty}, \mathbf{u}_{t+1: \infty}} [\sum_{i=0}^{\infty} \gamma^i r_{t+i} | s_t, \mathbf{u}_t] \quad (1)$$

### A. Problem Statement

In this section, we present the RL environment that models the AGV warehouse required by the fulfillment center. Fig. 1(a) is a snapshot of an layout example, and each cell denotes a module represented by a specific color as shown in Fig. 1(b).

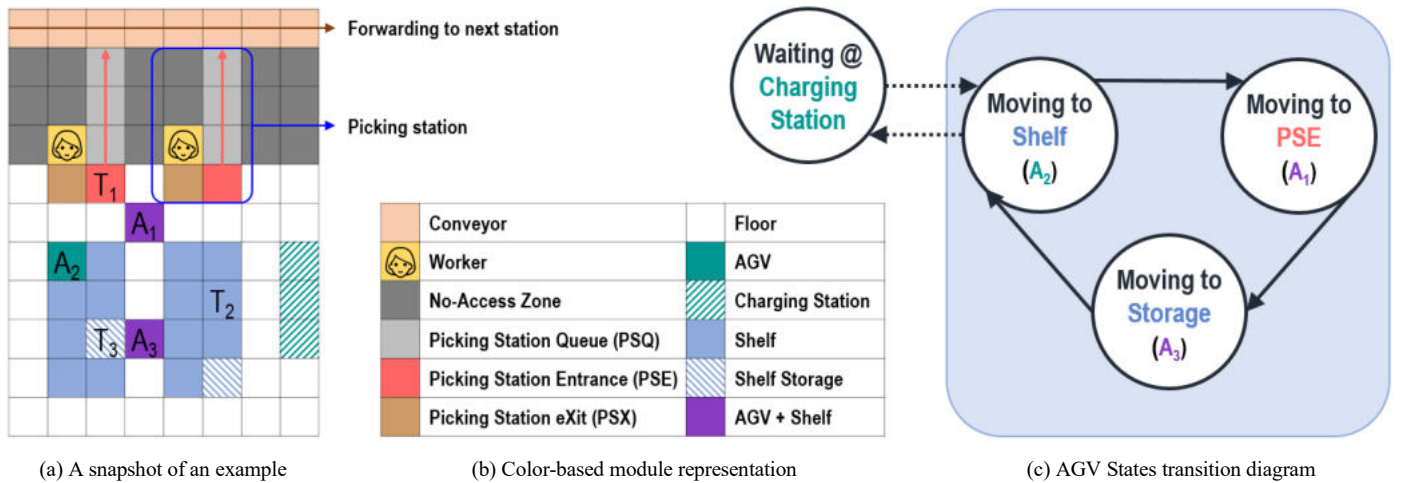


Fig. 1. Description of the RL environment in which the AGV warehouse is modeled, where the target position of agent  $A_i$  is  $T_i$ .

The overall scenario is carried out as: (1) When the system starts operating, all AGVs wait for assignment at the charging station. (2) Each picking station's worker allocates the required shelf to a random AGV. (3) The assigned AGV moves to the requested shelf and lifts it. (4) The AGV moves to the PSE of the picking station. (5) When the work at the picking station is finished, the AGV moves to a random shelf storage. (6) The AGV lays it down and repeats this from (2). In summary, each AGV can be in one of the 4 states: Waiting, Moving to Shelf, Moving to PSE, and Moving to Storage. Fig. 1(c) shows the circulation of AGV states in this process as a state transition diagram. Note that AGV, shelf, and shelf storage are not randomly selected in the actual AGV warehouse, so the environment we present is a more difficult problem to solve because they are randomly selected.

### B. State and Observation Representation

The observation of each agent consists of two-dimensional information normalized as a two-channel image for the surrounding  $9 \times 9$  area centered on itself. The shape of the observation is  $2 \times 9 \times 9$ , and the values of each channel represent the information of the corresponding position in the layout. The first channel denotes whether the agent can move to the corresponding location, and the second channel denotes the remaining Manhattan distance from the corresponding location to the target location. Meanwhile, the shape of the state is  $1 \times 20 \times 20$ , which consists of a normalized cumulative number of visits to each module in the layout within the episode.

### C. Action Representation

At the beginning of the episode, each agent's looking direction is randomly selected, thereafter it is determined by the action it performs. The action is performed to move one cell based on this looking direction, and the action space is defined as: {Stop, Moving Forward, Moving Right, Moving Left, Moving Back}. For this, the observation is rotated with respect to the looking direction of the agent.

The environment informs agents of available actions so that collisions can be avoided. By choosing these actions, it is guaranteed that they won't hit a wall or collide between agents. In other words, experiences related to the collision are not generated through Invalid Action masking, only high-quality experiences are accumulated in the replay buffer.

### D. Reward Function

The environment returns individual rewards for each agent. After the agents perform their chosen action, each agent's individual reward is positive if the Manhattan distance to their target position gets closer or the agent arrives at their target position, otherwise it is negative. It may be considered harsh to receive a negative reward even when the agent takes a stop action, but we expect the  $Q$  value to be updated to prevent such situations. The absolute value of the individual reward is  $1/\text{The number of agents}$ , and the global reward is the sum of all individual rewards. Accordingly, the range of the global reward that can be output in one time step from the environment is  $-1.0$  to  $+1.0$ .

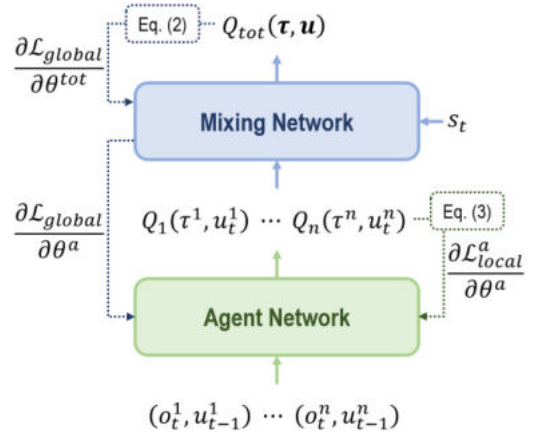


Fig. 2. The overall model architecture of the proposed algorithm.

## IV. PROPOSED ALGORITHM

We propose an algorithm that applies a new technique to QMIX [17] while maintaining the centralized training with decentralized execution framework. QMIX uses the Mean Square Error between  $Q^{tot}$  and its target generated by inputting the  $Q$  value of each agent into the mixing network as loss:

$$L_{global} = \sum_{i=1}^b \left[ (y_i^{tot} - Q_i^{tot}(\tau, \mathbf{u}, s; \theta^{tot}))^2 \right], \quad (2)$$

where  $y^{tot} = r + \gamma \max_{\mathbf{u}} Q^{tot}(\tau', \mathbf{u}', s'; \bar{\theta}^{tot})$ . For this, each agent extracts  $u^a$  and  $Q^a(\tau^a, u^a)$  according to the epsilon greedy method. QMIX restricts the relationship between  $Q^{tot}$  and  $Q^a$  to the monotonicity constraint. To enforce this monotonicity constraint, the weights of the mixing network are constrained to be non-negative. After the loss of the mixing network is calculated as in (2), it is backpropagated to a feed-forward network consisting of Mixing network and Agent network. In this process, the gradient is backpropagated to the agent network, but the effect can be considered insignificant. Although monotonicity is guaranteed by the constraint of the relationship between  $Q^{tot}$  and  $Q^a$ , it is harsh to allow only the mixing network to judge the contribution of individual agents to sacrifice. Thus, we propose an additional local loss for the agent network formulated as:

$$L_{local}^a = \sum_{i=1}^b \left[ (y_i^a - Q_i^a(\tau^a, u^a; \theta^a))^2 \right], \quad (3)$$

where  $y^a = r^a + \gamma \max_{u_a} Q^a(\tau_a', u_a'; \bar{\theta}^a)$ .

The global loss  $L_{global}$  is calculated using  $Q^{tot}$  of the mixing network, while the local loss  $L_{local}^a$  is calculated using  $Q^a$  of the agent network for each agent. In this study, we adopted that the agent network is shared but it is not essential. Note that each agent's observation-action history  $\tau^a$  is not shared and local loss  $L_{local}^a$  is calculated separately. Its gradient is only backpropagated to the agent network regardless of the mixing network. Backpropagation through global loss as in (2) also affects the agent network, but backpropagation through local



loss as in (3) can clarify the feedback on the actions of the individual agents. Hence, the final loss equation is defined as:

$$\mathcal{L} = \mathcal{L}_{global} + \sum_{a=1}^n \mathcal{L}_{local}^a \quad (4)$$

Fig. 2 briefly shows the overall network structure along with the direction in which the gradient of the proposed loss in (4) is backpropagated. The agent network is composed of GRU [18] and MLP, whereas the mixing network consists only of MLP, but its weights and biases are obtained from the output of hyper-networks to ensure the monotonicity constraints. The state and observation composed of two-dimensional information are first converted into one-dimensional features through CNN and then inputted to the MLP.

Each agent interacts with the environment and stores experiences in the replay buffer, performing distributed execution using only its own observations. The mixing network is used to update the agent network by estimating the  $Q$  values of all agents as total  $Q$  values. In other words, the mixing network does not directly interact with the environment, but only requires the state and  $Q$  values of each agent sampled from the replay buffer. Note that the mixing network, the main idea of centralized training, is not used in testing after training is complete.

Even though  $\mathcal{L}_{global}$  adopted in QMIX [17] is used for update, the influence of the newly added  $\mathcal{L}_{local}$  is so strong that it can be considered that the influence of  $\mathcal{L}_{global}$  is relatively weak. So, it is a reasonable doubt that there will be no significant difference from IQL which uses only  $\mathcal{L}_{local}$  as the final loss. In Section 5 we conducted an experiment comparing the algorithm proposed in this paper with IQL and also verify its validity by analyzing the results.

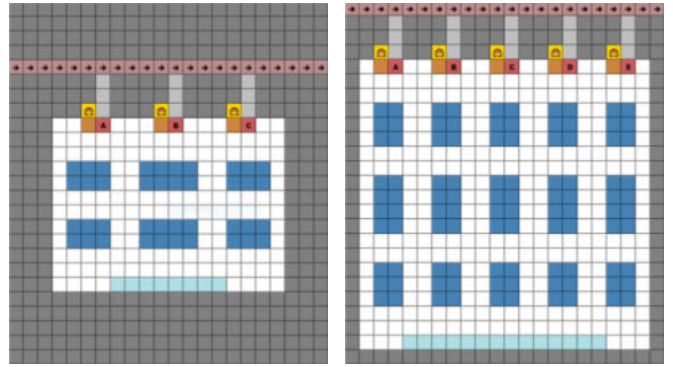


Fig. 3. Layout configuration: 'Small' layout (left) and 'Large' layout (right).

## V. EXPERIMENTS

In this section, the performance of the algorithm proposed in this paper is presented in comparison with IQL, which is considered the most basic marl algorithm. The experiments are performed in 'Small' layout and 'Large' layout as shown in Fig. 3. The 'Small' layout consists of 8 AGVs and 3 picking stations in 12x16 grid size, while the 'Large' layout consists of 14 AGVs and 5 picking stations in 20x20 grid size. For detailed evaluation we adopted three metrics: episode rewards, average path lengths, and number of shelves arrived at PSEs. During training, for every 10 training steps each metric is measured as the average value of 5 episodes in which all agents took a greedy action. We plot the average of 5 runs with 95% confidence intervals for every metric on both layouts. Note that 1 training step means a single model update by the final loss that aggregates the global loss and the local losses as in (4), and it is calculated using the experience sampled from the replay buffer.

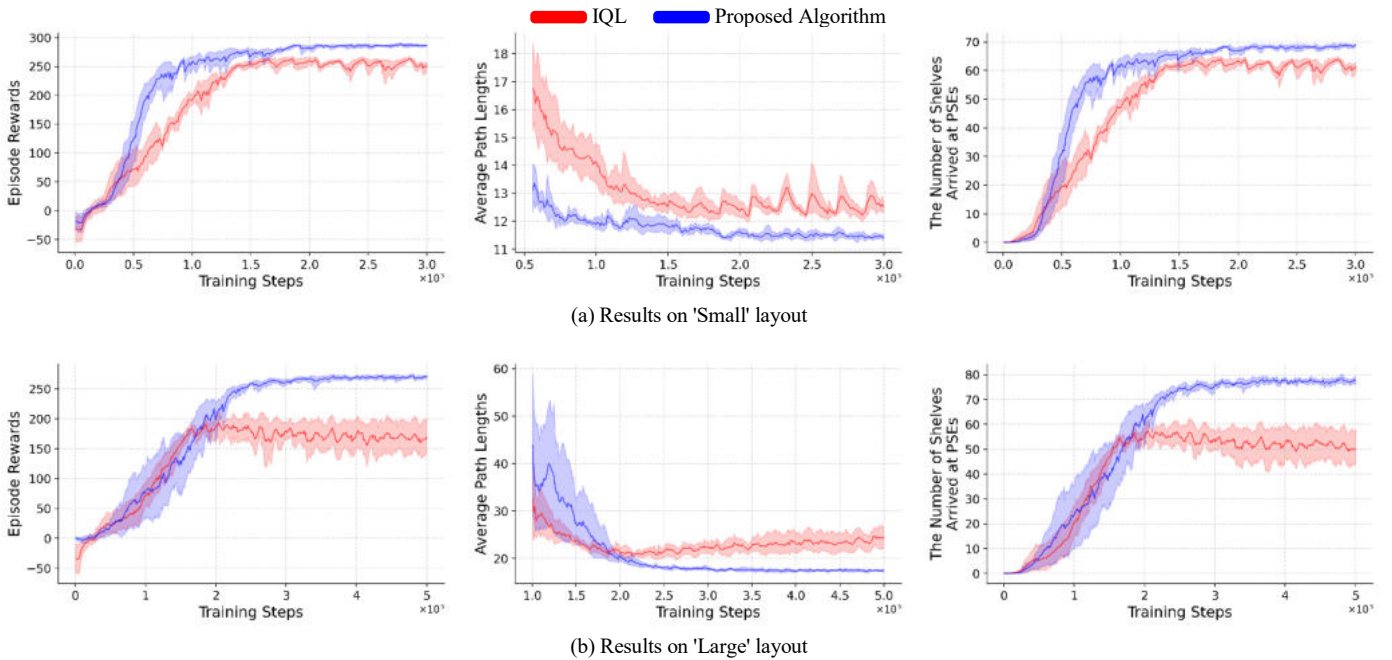


Fig. 4. Performance comparison for three metrics. Note that the front part of the graphs about average path lengths is omitted for the readability of the graph.

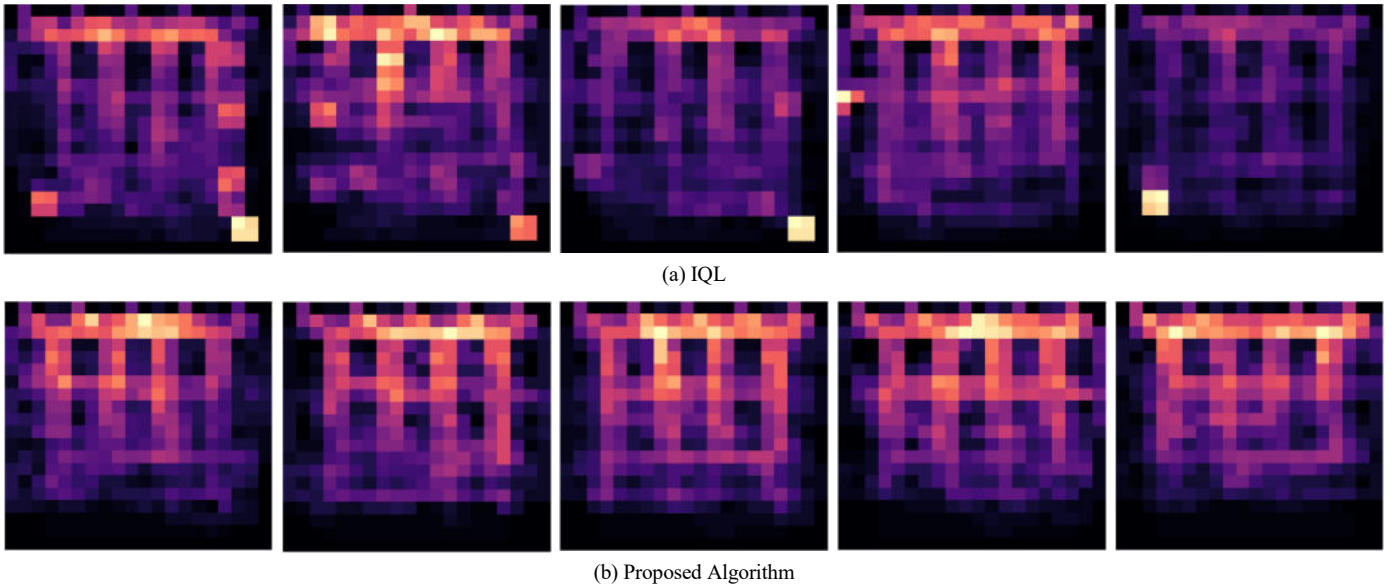


Fig. 5. Heat maps on 'Large' layout.

#### A. Performance Comparison

Fig. 4 shows the performance of the algorithm proposed in this paper compared with IQL as the change of three metrics according to the training steps, and Fig. 4(a) and 4(b) are the results on 'Small' layout and 'Large' layout, respectively. As shown in the graphs about the two metrics, episode rewards and number of shelves arrived at PSEs, we can know that our algorithms learn faster and more consistently. More specifically, our algorithms achieve unshakable convergence at better values, while IQL tends to be unstable, shaking even at worse values. This contrast is more evident in 'Large' layout than in 'Small' layout. Since IQL has no information exchange between agents, the performance of IQL decreases sharply as the number of agents increases. On the other hand, even if the number of agents increases, our algorithm proves their robust performance through efficient information exchange and cooperation between agents. These results can be easily confirmed from the graphs about the metric, average path lengths. For our algorithm, it becomes shorter and more stable as training progresses, but for IQL it does not converge and exhibits a very unstable appearance, and even in 'Large' layout it becomes longer. As a result, we can infer that the more complex the structure of the layout (i.e., the more AGVs), the more important is the collaboration between agents through communication.

#### B. Heat Maps

In order to analyze the actual behavior patterns of agents, we visualize the cumulative number of visits to the movement of AGVs as shown in Fig. 5. Episodes during 500 time steps were drawn five times each for our algorithm and IQL on the 'Large' layout. The relatively larger the cumulative number of visits, the brighter the corresponding position. From this analysis, we found a very interesting fact about IQL that certain AGVs are in an idle state at arbitrarily inappropriate positions. Therefore, the

heat map of IQL has a very bright specific position, not the main path of the entire system, and can be interpreted that the absence of communication between agents develops into a fatal problem. In contrast, our algorithm has been confirmed that, as we expected, all AGVs mainly move between picking stations and shelves, the main paths of the entire system. When comparing these results, we emphasize that all AGVs learned with our algorithms move primarily within a limited range, which is the main path of the entire system, but do not cause congestion. Accordingly, state circulation of all AGVs occurs smoothly and high performance follows.

## VI. CONCLUSION

In this paper, we proposed a CTDE-based MARL algorithm that can efficiently control the routes of AGVs which are essential components to increase the productivity of fulfillment centers. To evaluate the proposed algorithm, we presented state and observation representation, action representation, and reward function along with a description of the modules constituting the system and an overall scenario. The evaluation results are that the proposed algorithm outperforms IQL for three metrics on two different 'Small' and 'Large' layouts. We also provide insight into the importance of communication between agents via the visualization of the results.

#### ACKNOWLEDGMENT

This work was supported by the National Research Council of Science & Technology (NST) grant by the Korea government (MSIP) (No. CRC-15-05-ETRI), and also supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. NRF-2020R111A3065610).

## REFERENCES

- [1] L. Buşoniu, R. Babuška, and B. Schutter, Multi-agent reinforcement learning: An overview, *Innovations in multi-agent systems and applications-1*, pp. 183–221, 2010.
- [2] J. Cui, Y. Liu, and A. Nallanathan, "Multi-agent reinforcement learning-based resource allocation for UAV networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 2, pp. 729–743, 2019.
- [3] X. Li, J. Zhang, J. Bian, Y. Tong, and T. Liu, "A cooperative multi-agent reinforcement learning framework for resource balancing in complex logistics network," *In Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, 2019.
- [4] X. Li, X. Hu, W. Li, and H. Hu, "A multi-agent reinforcement learning routing protocol for underwater optical sensor networks," *In Proceedings of IEEE International Conference on Communications*, 2019.
- [5] F. A. Oliehoek, M. T. J. Spaan, and N. Vlassis, "Optimal and approximate Q-value functions for decentralized POMDPs," *Journal of Artificial Intelligence Research*, vol. 32, pp.289–353, 2008.
- [6] F. A. Oliehoek and C. Amato, *A concise introduction to decentralized POMDPs*, SpringerBriefs in Intelligent Systems, Springer, 2016.
- [7] J.J. Enright and P.R. Wurman, "Optimization and coordinated autonomy in mobile fulfillment systems," *In Proceedings of the AAAI workshop on automated action planning for autonomous mobile robots*, pp. 33–38, 2011.
- [8] J. Bae and W. Chung, "A heuristic for a heterogeneous automated guided vehicle routing problem," *International Journal of Precision Engineering and Manufacturing*, vol. 18, no. 6, pp. 795–801, 2017.
- [9] Z. Han, D. Wang, F. Liu, and Z. Zhao, "Multi-AGV path planning with double-path constraints by using an improved genetic algorithm," *PloS one*, vol. 12, no. 7, 2017.
- [10] Y. Lian and W. Xie, "Improved A\* multi-AGV path planning algorithm based on grid-shaped network," *In 2019 Chinese Control Conference*, 2019.
- [11] R. Kamoshida and Y. Kazama, "Acquisition of automated guided vehicle route planning policy using deep reinforcement learning," *IEEE International Conference on Advanced Logistics and Transport (ICALT)*, 2017.
- [12] Y. Yang, J. Li, and L. Peng, "Multi-robot path planning based on a deep reinforcement learning DQN algorithm," *CAAI Transactions on Intelligence Technology*, vol. 5, no. 3, pp. 177–183, 2020.
- [13] C.J.C.H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, pp. 279–292, 1992.
- [14] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
- [15] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," *In Proceedings of the Tenth International Conference on Machine Learning*, pp. 330–337, 1993.
- [16] P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. F. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls, and T. Graepel, "Value-decomposition networks for cooperative multi-agent learning based on team reward," *In Proceedings of 17th International Conference on Autonomous Agents and Multiagent Systems*, Stockholm, Sweden, 2018.
- [17] T. Rashid, M. Samvelyan, C. S. de Witt, G. Farquhar, J. Foerster, and S. Whiteson, "QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning," *In Proceedings of the 35th International Conference on Machine Learning*, Stockholm, Sweden, 2018.
- [18] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *In NIPS 2014 Workshop on Deep Learning*, 2014.

# DDPG-Edge-Cloud: A Deep-Deterministic Policy Gradient based Multi-Resource Allocation in Edge-Cloud System

Arslan Qadeer, and Myung J. Lee  
Department of Electrical Engineering  
The City College of New York of CUNY  
New York, USA

{aqadeer000@citymail, mlee@ccny}.cuny.edu

**Abstract**—5G and beyond is the key enabler for extreme mobile-broadband (xMBB), Massive and Ultra-reliable machine-type communication (mMTC, uMTC). To handle such large-scale and real-time traffics, Edge Cloud (EC) plays a critical role to minimize the latency and provide compute power at the edge of the network for Internet of Things (IoTs). However, an EC endures limited compute capacity in contrast with the back-end cloud (BC). Intelligent resource management techniques become imperative in such resource constrained environment. This paper studies the problem of compute and wireless resource allocation in an integrated EC and BC environment. Machine learning-based techniques are emerging to solve such optimization problems. However, it is challenging to adopt traditional discrete action space-based methods since they do not scale well in large-scale environments. To this end, to overcome the bottlenecks of wireless bandwidth and compute capacity in resource constrained EC and BC, we propose continuous action space-based DDPG-Edge-Cloud, a deep deterministic policy gradient-based multi-resource allocation (MRA) framework with a pruning principle. The proposed agent is equipped with a Conv1D residual block, gated recurrent unit (GRU) layer and an attention layer for local and long-term temporal feature extraction. We validate the proposed framework by comparing with two alternative agents. Experimental results demonstrate that our proposed agent converges fast and achieves up to 55% and 86.5% reduction in operational cost and rejection rate, and achieves up to 115% gain in the quality of experience on average.

**Index Terms**—Edge cloud computing, deep deterministic policy gradient, resource allocation, smart city, IoT.

## I. INTRODUCTION

Next generation mobile applications (e.g. Augmented Reality (AR), Virtual Reality (VR)) and streetscape applications (e.g. swift control-response for emergency vehicles and situation-aware traffic/pedestrian signaling) possess resource-hungry and real-time constraints [1]. Edge-cloud (EC) architecture is a stepping stone to meet the above compute and real-time constraints by reducing the network latency and providing the computational resources at the edge of the network [2]. Furthermore, A three-tier hierarchical EC system integrated with the back-end cloud (BC) provides support for a broad-range of applications with varying QoS requirements in greater extent [3].

Edge clouds possess a limited amount of computational resources [3] and back-end clouds experience the same in the case of pay-as-you go model [4]. 5G supports dynamic

Radio Access Network (RAN) and a wider frequency spectrum landscape [5]. However, in the presence of excessive amount of connected devices in the EC environment, a large amount of concurrent traffic can be anticipated. Thus, communication resources, which connect the devices with EC and BC, also become a bottleneck for the system. This multi-resource allocation (MRA) and system cost reduction challenge is manifold: First, handling the user requests from a wide range of applications at large scale with different QoS requirements. Second, the computational complexity pertaining to optimal resource allocation in a large system particularly in dynamic traffic patterns requires scalable solution techniques.

The proven success of Machine Learning (ML) based techniques has spurred the adoption of ML algorithms to solve control and management problems for IoTs in clouds and 5G wireless networks [6]–[8]. Cheng *et al.* [9] utilized the Deep Reinforcement Learning (DRL) to train the Deep Q-Network in order to solve a resource provisioning and task scheduling problem in a cloud-based environment under the strict QoS requirement. Wei *et al.* [10] proposed a natural actor-critic reinforcement learning framework to jointly solve the problem of content caching, computation offloading and radio resource management with the goal of minimizing the end-to-end delay. Deep deterministic policy gradient (DDPG) based methods, which provide superior state representation in high-dimensional space, are also adopted to solve the resource allocation problems. Peng *et al.* [11] leveraged the DDPG and hierarchical learning architectures to jointly solve the spectrum, computation and storage allocation problem in an EC based system. Recent studies [12], [13] solved the computation offloading and resource allocation problem for multiple mobile users in EC based systems by utilizing the DDPG-based framework and proposing the state-of-the-art algorithms. Nevertheless, all of the above consider either EC or BC based environments separately, and lack the three-tier hierarchical architecture integrated with the BC.

Motivated from the above discussion, we aim at presenting a scalable solution which can also be easily implemented in real-world scenarios like COSMOS testbed. The COSMOS is a National Science Foundation (NSF) sponsored project with many academia partners including The City College of New York [14]. Our recent work [15] proposed a novel resource allocation framework to solve the bandwidth allocation problem in COSMOS based environment. The presented results are encouraging, which stimulated to extend the current framework [15] and comprehensively solve the multi-resource allocation (MRA) problem with continuous action-space. This is the rationale behind the introduction of DDPG-Edge-Cloud.

Identify applicable funding agency here. If none, delete this.

The proposed framework takes into account the wireless and computation resource allocation problem equitably at the Edge-cloud (EC) and back-end cloud (BC). To the best of our knowledge, none of the existing works applied DDPG with local and temporal feature learning networks, and pruning principle to solve the MRA problem jointly in EC and BC with the goal of minimizing overall system cost for providers, meeting the strict QoS requirements of applications and fast self-learning capability.

The main contribution of our work is summarized as follows:

- We present a simple user job model which takes into consideration both the deadline of the job and data to be processed at the same time. Thereafter, to aptly process these jobs, we present a multi-resource allocation (MRA) model under an integrated wireless communication, EC and BC environment to handle a large-scale of user requests under constrained resources.
- We formulate the MRA problem for user requests into DDPG-based Actor-Critic framework. The reward maximization objective for resource allocation with wireless bandwidth, EC and BC compute resources considers to optimize three fundamental bench-marking points; 1) Minimize system cost; 2) Minimize rejection rate for enhanced reliability; and 3) Minimize round-trip time of a user request for better Quality of Experience (QoE).
- Instead of using fully-connected networks (FCNs), both the actor and critic networks in the DDPG-based framework consist of convolution (Conv1D) residual block, gated recurrent unit (GRU) and an attention layer. Conv1D residual block aims at learning the correlations among local features of each input state, and GRU and attention layers capture the temporal features. Further, our proposed pruning principle [15] helps to minimize the rejection rate by efficiently offloading the service requests to servers and base stations.
- The performance of DDPG-Edge-Cloud is evaluated in terms of convergence efficiency, average operational cost, rejection rate and QoE. Compared with other DDPG-based agents, our proposed method achieves better convergence and loss results during training. In addition, our proposed approach outperforms two DDPG-based methods in all of the test scenarios. Overall our proposed approach achieves better performance for the MRA problem.

The remainder of this paper is organised as follows: The multi-resource allocation (MRA) in EC and BC based smart streetscape system is modeled in Section II. Section III introduces our core DDPG-based framework and pruning principle for our MRA system. Section IV presents the performance evaluation and discusses the experimental results, followed by the conclusion and future directions in Section V.

## II. SYSTEM MODEL AND PROBLEM FORMULATION

### A. System Description

We consider the Edge-cloud (EC) based streetscape system as shown in Fig. 1. Our proposed system provides 5G based wireless radio access (including sub-6 GHz and mmWave) to sensors, street signals, vehicles, security cameras, mobile devices and other Internet of Things (IoT) via software defined radios (SDRs) herein called virtual base stations (BSs). In 5G architecture, one SDR/BS covers a small cell; therefore, multiple BSs can connect to one nearby edge-computing infrastructure via high speed and low-latency software defined fiber links [14]. For the beyond 5G deployments, mobile network operators (MNO) rely on connected EC and BC for



Fig. 1: System model and Structure of Edge-cloud based Streetscape System.

scalability and enhanced services [16]. Therefore, it is vital to consider EC along with BC in order to propose a realistic system. In our proposed system, EC is interconnected with the BC via high speed fiber links to leverage abundant and always available resources at public clouds (AWS, Google Compute Engine, Microsoft Azure) in order to offload delay-tolerant multimedia tasks or to save data for future uses.

Context aware control of resources is main ingredient of the smart streetscape environment such as smart control of pedestrian signal for elderly people and traffic signal control for emergency situations (e.g. fire on a building) or in a logistics unloading case, traffic can automatically be diverted to a safer and smoother street with the help of data sent by the IoTs [17].

Multi-media (AR, VR, Video) applications are bandwidth hungry and resource intensive, at the same time require low end-to-end latency [1]. In such scenarios, proposed EC infrastructure plays an important role to supply uninterrupted services to both streetscape and multi-media applications. Upon arrival of a service request, based on its exigency, the system decides whether to run it on EC or BC, and based on the availability of resources how much bandwidth and compute resources have to be allocated in order to execute the task within the deadline.

### B. Users and Jobs Model

In the proposed framework, we consider all the devices which are connected to the system as EC users. Each user offloads its computational task in the form of a distinct service request. The entire workload of the system is a set of  $J$  jobs from  $U$  users i.e.  $J = \{j_1(dl_1, d_1), j_2(dl_2, d_2), \dots, j_U(dl_U, d_U)\}$ . A job  $j_u(dl_u, d_u)$  is a tuple of two variables where  $dl_u$  means the hard deadline in milliseconds and  $d_u$  represents the data in bytes to be processed in job  $j_u$  offloaded by user  $u$ . The user job can demand both CPU and Memory for a successful execution but processing vitally involves CPU; therefore, we only consider CPU cycles as a job processing source as proposed by Cheng *et al.* [9]. Suppose  $C$  represents the number of CPU cycles required to process 1 Byte of data, then  $L_u$  is given as the total CPU cycles required to compute data  $d_u$  ( $L_u = d_u \times C$ ). A similar task computation model (with CPU cycles) is proposed in [12], [13] as well.

### C. Wireless Bandwidth Model

As shown in Fig. 1, an EC system owns  $W$  base stations (BSs)  $\{1, 2, 3, \dots, W\}$ , each of which makes meshed wireless connections with actuators/relays to provide wireless access capacity, load-balancing and failover. The total wireless bandwidth that is available on all BSs is represented as  $B$ . Each BS  $w$  can support  $H$  wireless bandwidth channels  $\{ch_1^w, ch_2^w, ch_3^w, \dots, ch_H^w\}$ , and each wireless bandwidth channel  $ch_h^w$  ( $h \in [1, H]$ ) provides a variable amount of bandwidth (data rate in bps) and costs  $c_h^w$ . The directional antenna array in mmWave cellular networks of the COSMOS testbed [14] is capable of exploiting beamforming, which compensates the increased path-loss at mmWave frequencies and overcomes the additional noise due to the large transmission bandwidth [5]. Interference isolation is also achieved in directional beamforming, which, as a result, reduces the adjacent-cell interference. Therefore, in our case, we ignore path-loss, channel noise and interference factors, and manage resources at the application layer through well-defined APIs [3], which provides a fine-grained control of the wireless bandwidth.

### D. Computational Model

1) *Edge Cloud*: An Edge-cloud (EC) owns  $M$  servers  $\{1, 2, 3, \dots, M\}$ . Each server  $m$  processes an offloaded job via a set of virtual machines (VMs). Let  $K = \{vm_1^m, vm_2^m, vm_3^m, \dots, vm_K^m\}$  be the set of VMs that can be assigned by server  $m$  and each VM  $vm_k^m$  provides a variable amount of compute capacity (CPU cycles in Hz) to process a job and costs  $c_k^m$ . The total compute capacity on all EC servers is given by  $C_{ec}$  CPU cycles. Chen *et al.* [13] proposes a similar computational model; however, they consider EC with unlimited compute resources, which may not be valid for practical scenarios.

In practice, a fine-grained control of the compute resources can be achieved at the application layer by processing user jobs in a Docker container, which uses cgroups to limit the system resources. However, for the simplicity of the model, we will use the term VM in this study.

2) *Back-end Cloud*: Previous work [18] considered back-end cloud (BC) as a source of unlimited compute capacity. However, we take into account a realistic model to minimize the overall cost of the system by adopting pay-as-you go model. Therefore, just like an EC, we consider  $N$  BC servers  $\{1, 2, 3, \dots, N\}$ , which execute a job via a set of  $K = \{vm_1^n, vm_2^n, vm_3^n, \dots, vm_K^n\}$  VMs, each with a variable amount of compute capacity (CPU cycles) and costs  $c_k^n$ .  $C_{bc}$  denotes the total number of CPU cycles available on all BC servers. This model can be easily extended to infinite resource model by relaxing  $C_{bc}$  and  $K$  sufficiently large.

### E. Delay Model

We define round-trip time (RTT) as the total time that it takes for a job to upload to the EC or BC via a wireless channel, process the job and then send the result back. This involves propagation, transmission (2-way) and processing time. A similar delay model is also used in [10].

1) *Propagation Time*: Propagation time through fiber or air media is negligible and assumed to be constant. Therefore, we consider a constant delay  $t_u(prop_{ec}) = 5ms$  for EC and  $t_u(prop_{bc}) = 50ms$  for BC depending on where the resources are allocated for job execution.

2) *Transmission Time*: This includes the time that a job takes to upload to the EC or BC and the time to send the result back successfully. It depends upon the amount of data and wireless channel that is allocated. The transmission time of a job to the EC can be calculated as:  $t_u(trans_{ec}) = \frac{d_u}{ch_h^w} + \frac{R_u}{ch_h^w}$ ,

where  $R_u$  is the result which is generated after the job execution and sent back. In general, the result is a control signal and contains only a few kilobytes of data [13]. Nonetheless, the result for AR/VR applications can be substantially large, therefore, we incorporate it in our framework.

According to Fig. 1, EC is connected with BC via a high capacity fiber link and considered to guarantee  $b$  bandwidth [14]. Thus, the transmission time between a device and BC can be given as:  $t_u(trans_{bc}) = t_u(trans_{ec}) + \frac{d_u}{b} + \frac{R_u}{b}$ .

3) *Processing Time*: This depends upon the number of required CPU cycles  $L_u$  and the allocated VM  $vm_k^m$ ,  $vm_k^n$  at EC or BC, respectively, and given as:  $t_u(proc_{ec}) = \frac{L_u}{vm_k^m}$  for the EC, and  $t_u(proc_{bc}) = \frac{L_u}{vm_k^n}$  for the BC.

To summarize, when a job  $j_u$  is offloaded to the EC, the total round-trip time is given as:  $r_{tt_{ec}}(j_u) = t_u(prop_{ec}) + t_u(trans_{ec}) + t_u(proc_{ec})$ , similarly, when the job is offloaded to the BC then:  $r_{tt_{bc}}(j_u) = t_u(prop_{bc}) + t_u(trans_{bc}) + t_u(proc_{bc})$ .

### F. Utility Model

The total usage of the system resources at any given time  $t$  is the sum of the occupied resources by all the jobs which are being processed. Therefore, the bandwidth utility rate of all base stations  $W$  is given as:

$$Ur_W(t) = \frac{\sum_{w=1}^W (\sum_{h=1}^H ch_h^w \cdot \mu_h^w(t))}{B}, \quad (1)$$

where  $\mu_h^w(t)$  is the total number of  $ch_h^w$  channels which are serving the jobs at time  $t$ . Similarly, the utility rate of a server at EC and BC can be measured as:

$$Ur_M(t) = \frac{\sum_{m=1}^M (\sum_{k=1}^K vm_k^m \cdot \mu_k^m(t))}{C_{ec}}, \quad (2)$$

and

$$Ur_N(t) = \frac{\sum_{n=1}^N (\sum_{k=1}^K vm_k^n \cdot \mu_k^n(t))}{C_{bc}}, \quad (3)$$

respectively.

## III. THE PROPOSED DDPG-EDGE-CLOUD AGENT

In this section, we present DDPG-Edge-Cloud to solve the MRA problem for mobile and streetscape based applications. Like RL agent in [15], the proposed DDPG-Edge-Cloud agent also runs on the EC for better accessibility of the environment and faster training. In our DDPG-based framework, the agent contains actor and critic networks whose architectures are same. The actor, i.e., a policy function, observes the state  $s_t$  by interacting with the environment and takes a continuous action  $a_t$  via a deterministic policy and receives an immediate reward  $r_t$ . On the other hand, the critic uses the action-value function  $Q(s_t, a_t)$  to update the policy parameters. At each state, different resource allocation actions yield different rewards. The goal of the agent is to maximize the long-term reward by finding an optimal resource allocation policy. We define state and action space followed by our unique reward model below.

1) *State Space*: The state of the system at any time  $t$  is the observation of utility rates of all the base stations, EC and BC servers, and the current job  $j_u$  which has to be scheduled either at EC or BC. The state contains five parameters, i.e.  $s_t = \{Ur_W, Ur_M, Ur_N, j_u(d_u, d_u)\}$ . Based on these sequences, the agent learns optimal resource allocation strategies in each decision epoch.

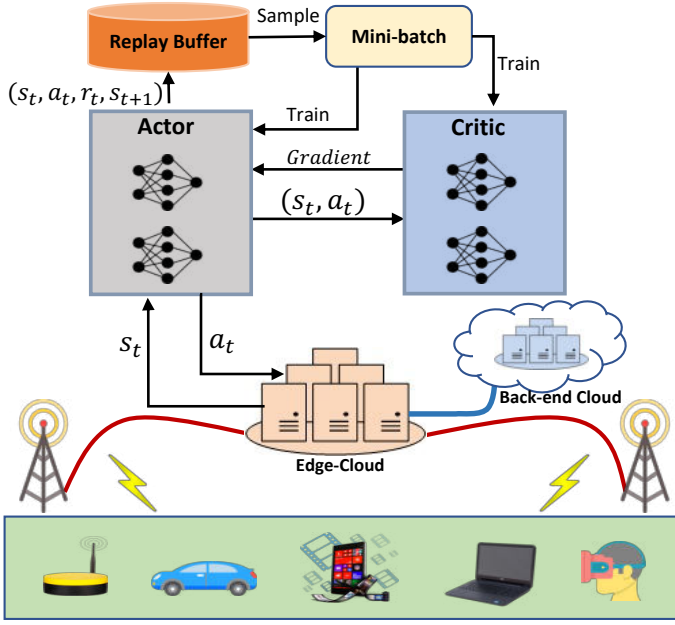


Fig. 2: The architecture of the proposed DDPG-Edge-Cloud framework.

2) *Action Space*: In each state  $s_t$ , the agent decides how much bandwidth and VM resources (on EC or BC) have to be allocated in order to successfully execute the job within the deadline. The continuous action space,  $a_t = \{ch_h, vm_k, cloud\}$ , has three parameters.  $ch_h$  is the amount of bandwidth,  $vm_k$  is the amount of CPU cycles and  $cloud$  has value either 0 or 1 which means task is offloaded to EC or BC, respectively.

3) *Reward*: In our model, the reward can be calculated as the sum of lump income and cost of resource procurement in the MRA system:

$$r_t(s_t, a_t) = I_u(s_t, a_t) - cost_u(s_t, a_t) \times rtt(j_u), \quad (4)$$

where  $I_u(s_t, a_t)$  is the net income earned by executing the job  $j_u$  and  $cost_u(s_t, a_t)$  is the cost of resource procurement for  $rtt(j_u)$  time (time it takes to process the job) for selecting action  $a_t$  at state  $s_t$ . In the definitions of  $I_u(s_t, a_t)$  and  $cost_u(s_t, a_t)$ , we take into account the completion time of the job and the utility of wireless and computation resources.  $I_u(s_t, a_t)$  is defined as,

$$I_u(s_t, a_t) = \begin{cases} (dl_u - rtt_{ec}(j_u)) \cdot \delta, & \text{if } cloud_t = 0 \\ (dl_u - rtt_{bc}(j_u)) \cdot \delta, & \text{if } cloud_t = 1. \end{cases} \quad (5)$$

In Eq. (5),  $\delta$  represents the revenue that the service provider generates by successfully executing the user job. It can be set on-the-fly which can differ depending on the job completion time. Note that, if the completion time of job ( $rtt(j_u)$ ) exceeds the deadline ( $dl_u$ ), then the income becomes negative which impacts the overall reward. This effect encourages the system to take the allocation decisions that can complete the jobs before deadlines.

In contrast to the income,  $cost_u(s_t, a_t)$  describes the cost of resource procurement per unit time by allocating wireless bandwidth channel and a VM at EC or BC and is given as,

$$cost_u(s_t, a_t) = \begin{cases} Ur_W(t) \cdot c_h + Ur_M(t) \cdot c_k, & \text{if } cloud_t = 0 \\ Ur_W(t) \cdot c_h + Ur_N(t) \cdot c_k, & \text{if } cloud_t = 1, \end{cases} \quad (6)$$

where  $c_h$  and  $c_k$  represent the cost of wireless bandwidth channel and VM allocation per unit time, respectively.

As compared to [18], we calculate the reward for each individual job, so as the cost and the income. This approach is more meaningful in a way that the QoS requirement in each job may vary and calculating reward for every individual job can better assist the agent to derive an optimal resource allocation policy.

The optimization problem of wireless bandwidth and VM allocation for the user jobs at different base stations and servers, while considering the varying QoS requirements and constrained resources is formulated as below:

$$\text{maximize } \sum_{t=1}^T r_t(s_t, a_t) \quad (7)$$

subject to the constraints  $Ur_W(t) \leq 1$ ,  $Ur_M(t) \leq 1$  and  $Ur_N(t) \leq 1, \forall t \in T$ , which describes that the utility of bandwidth and EC and BC servers does not exceed from its total available capacity, respectively. The constraint  $rtt_{ec}(j_u) \leq dl_u, rtt_{bc}(j_u) \leq dl_u, \forall u \in U$  guarantees the hard deadline requirement of the job offloaded at EC or BC, respectively.

#### A. Actor-Critic Framework

In traditional DDPG-based framework, fully-connected networks (FCNs) are mostly used as both actor and critic networks [19], which have huge trainable weights and capture only global discriminative features of the task sequences. However, the computation-intensive tasks have complex temporal variations in nature. For high-quality state representation and better function approximation of MRA system, we propose a network to capture the local and temporal features in sequential data. Inspired from [13], the first part of our proposed agent contains Conv1D residual block structure to learn the correlations among local features of each input state. The second part contains GRU to learn the temporal dependencies and an attention mechanism to capture meaningful information at certain moments that has decisive effect on prediction. GRU model has fewer parameters and controls the flow of the information without using a memory unit, resulting in less complicated structure with the performance on par with LSTM [20]. This is why we prefer GRU over LSTM.

To break the undesired temporal correlations of training samples and reduce variance, an experience replay (ER) is used to store all the experience  $((s_t, a_t, r_t, s_{t+1}))$  to train the agent on more independent samples. Uniform sampling is used to randomly select a mini-batch of transitions from the ER buffer to train the actor and critic networks.

1) *Pruning Principle*: Our proposed agent is responsible to select appropriate amount of bandwidth, VM units and cloud to execute a job. We introduce pruning principle to further select the base station and server with the least utility. The BS is given as  $w_t = \arg \min(Ur_w(t)), \forall t \in T, \forall w \in W$ , the EC server is given as  $m_t = \arg \min(Ur_m(t)), \forall t \in T, \forall m \in M$ , and the BC server (if task offloaded to BC)  $n_t = \arg \min(Ur_n(t)), \forall t \in T, \forall n \in N$ . The major contribution of our proposed pruning principle is the reduction of action space at every state by a significant amount. It also helps to balance the load among all the base stations and servers; which means, it does not lead to overload a single base station or server which could potentially result in higher rejection rate for future jobs.

The training process of DDPG-Edge-Cloud agent with pruning principle to obtain an optimal MRA policy is summarized in Algorithm 1. The target networks of actor and critic are clone of their respective online networks.

---

**Algorithm 1:** Training process of DDPG-Edge-cloud agent with Pruning Principle

---

**Input :** User jobs with varying QoS requirements

- 1 Initialize replay memory  $\Delta$  to capacity  $\Omega$ ;
- 2 Initialize the actor and critic online and target networks with random weights;
- 3 **for**  $episode = 1$  to  $E$  **do**
- 4     Reset the environment;
- 5     **for**  $t = 1$  to  $T$  **do**
- 6         Predict an action  $a_t$  using the actor network;
- 7         Apply pruning principle to select base station and server
- 8         Execute action  $a_t$ , observe next state  $s_{t+1}$  and receive reward  $r_t$
- 9         Store transition sample  $(s_t, a_t, r_t, s_{t+1})$  in  $\Delta$
- 10         Sample random mini-batch of transitions from  $\Delta$
- 11         Train the critic and actor on sampled mini-batch
- 12         Update the weight vectors of online networks in actor and critic
- 13         Update the weight vectors of target networks in actor and critic
- 14          $s_t \leftarrow s_{t+1}$
- 15     **end**
- 16 **end**

**Output:** Optimal Multi-Resource Allocation policy

---

#### IV. EXPERIMENTAL RESULTS

We develop our MRA framework and proposed DDPG-Edge-Cloud agent with pruning principle in the Python 3.8.10 to simulate a near real-world environment. The simulation code will be made public after the community release of COSMOS testbed [14]. We run all the experiments on Dell Desktop Machine with 2.9 GHz Intel Core i7 processor, 128GB memory and Windows 10 Pro 64-bit OS, and discuss the advantages of our proposed algorithm over the alternative methods.

##### A. Experiment Setup

The two baselines we use to compare with the DDPG-Edge-Cloud agent are described below:

- **DDPG-NN:** Existing DDPG [19] with two fully-connected network layers used in the actor and critic networks. Same uniform sampling replay buffer is used for a fair comparison.
- **DDPG-CNN:** In a DDPG-based actor and critic networks, the fully-connected layers are replaced by identical Conv1D residual blocks introduced in Section III-A. As opposed to pruning principle in our proposed method, base stations and servers are randomly selected in the case of these agents.

We perform the experiment on three different sizes of environments, small, medium and large. Small-scale environment contains 4 wireless base stations, 10 EC and 10 BC servers. Medium-scale environment contains 12 base stations, 30 servers at the EC and 30 at the BC. The large-scale environment consists of 20 base stations, 50 EC and 50 BC servers. We set each base station to provide 1Gbps of total bandwidth, each server with 18 cores and each core with 2GHz (total 36GHz) both in the EC and BC servers. The bandwidth (b) between EC and BC set to 1Gbps. The configurations of the proposed system can be customized; however, here our objective is to compare the performance of three algorithms.

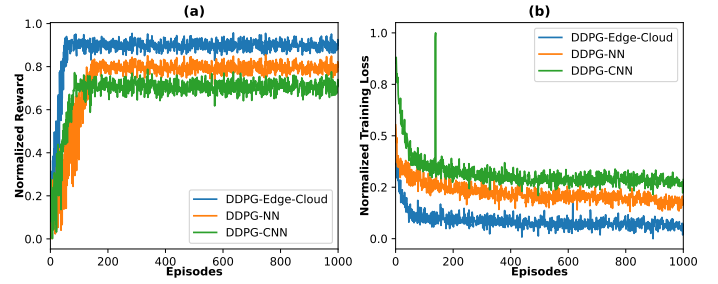


Fig. 3: Convergence comparison of three agents in small-scale environment. (a) Normalized Reward. (b) Normalized training loss.

We conduct experiments on small-scale, medium-scale and large-scale environments with 10,000, 100,000 and 1,000,000 jobs, respectively. The wireless bandwidth and VM per unit rental/usage cost is normalized to 1 cent per second. The value of revenue  $\delta$  is set to 10 cents. Mobile users and other IoT devices generate tasks of different sizes with varying QoS demands. We group these user tasks/jobs into two groups; 1) Type-1 tasks are critical tasks with comparably small size (bytes) and deadline. We set the deadline of such tasks to be in the range of [200ms, 800ms], and the size to be in the range of [100KB, 500KB]; 2) Type-2 tasks are multi-media tasks with large size and relaxed deadline as compared to the Type-1 tasks. The deadline of such tasks is set between [1000ms, 2000ms], and the size between [1MB, 2MB]. CPU cycles required to process 1-byte of data is randomly generated between [1000, 4000] and the resultant data is also randomly generated between [500, 1000] in KBs.

The default learning rate is set to 0.0001 and 0.001 for actor and critic networks, respectively, and the discount factor  $\gamma = 0.99$ . The Adam optimizer is used to optimize the loss function during training. The learning iterations ( $T$ ) per episode are set to 1000.

##### B. Convergence

We compare the convergence rate of DDPG-Edge-Cloud with DDPG-NN and DDPG-CNN in a small-scale environment to get the intuition of the performance of the agents. Fig. 3(a) and Fig. 3(b) depict the convergence rate in terms of reward and training loss which are normalized using max-min normalization method. Initially, DDPG-CNN grows up quickly as compared to DDPG-NN; however, due to the nature of local feature extraction only, the DDPG-CNN agent is trapped in local optima. Our proposed agent learns both local and long-term features, which results in efficient training and superiority in convergence.

##### C. Performance Analysis

Performance comparison is based upon three key performance indicators (KPIs) in the MRA system: operational cost, rejection rate and QoE. The operational cost is calculated using Eq. (6). A job is considered rejected if its completion time exceeds the given deadline or no more resources are available for allocation. Both the cost and rejection rate are averaged over the total number of accepted jobs in the respective scale. In our environment, QoE is inversely proportional to the round-trip time (RTT) of the job. This means, smaller RTT of a job will induce better QoE for the users. Fig. 4(a) illustrates the average operational cost, our proposed agent, on average, achieves 30.5%, 42% and 55% reduction in cost in small, medium and large-scale environments, respectively. Fig. 4(b) shows that our proposed strategy consistently gives rejection rate a multiple of  $10^{-3}$  even in the large scale environment.



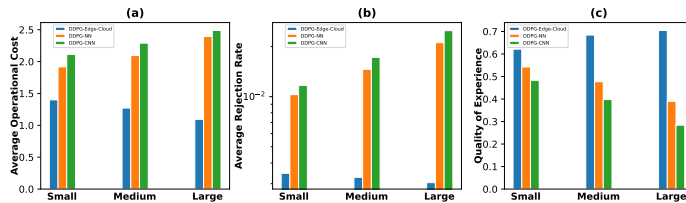


Fig. 4: Performance comparisons of three agents in small, medium and large-scale environments. (a) Average operational cost, (b) Average rejection rate, (c) Quality of Experience.

Moreover, our proposed agent, on average, achieves 68%, 79% and 86.5% reduction in rejection rate in small, medium and large-scale environments, respectively. The improvement of this magnitude is due to the virtue of our proposed pruning principle, which evenly distributes the load among all base stations and servers, and minimizes the probability of rejection for the future jobs. In view of Fig. 4(c), our proposed agent achieves nearly 22%, 60% and 115% gain in QoE in small, medium and large-scale environments, respectively.

To summarize, our proposed DDPG-Edge-Cloud adaptively determines the dynamic selection of VMs, wireless bandwidth channels and cloud for joint optimization of resources in the EC and BC. Moreover, our proposed pruning principle helps reduce the rejection rate significantly. Overall, our proposed framework outperforms the alternative agents in all three environments.

## V. CONCLUSION AND FUTURE WORK

We present DDPG-Edge-Cloud, a deep deterministic policy gradient-based multi-resource allocation (MRA) framework. The MRA system is utilized to optimize the problem of compute and wireless resources for the IoTs and mobile users in a smart streetscape based edge-cloud (EC) and back-end cloud environment. The proposed DDPG-Edge-Cloud agent is equipped with Conv1D residual block, gated recurrent unit (GRU) layer and an attention layer. The agent runs in the EC to make dynamic resource allocation decisions for the user tasks. The presented algorithm learns the local and long-term temporal features from the sequential data and outperforms the alternative methods in convergence speed. DDPG-Edge-Cloud achieves up to 55% reduction in operational cost on average. The proposed pruning principle helps our agent to achieve up to 86.5% reduction in rejection rate on average. Further, the proposed agent achieves up to 115% gain in the QoE of the users.

In future, we plan to explore priority-based replay buffer techniques where priority will be calculated using a heuristic function, which will help further boost up the convergence rate. Moreover, to cope with the overwhelming volume of accumulated data from numerous IoTs, we plan to investigate on data parallelism techniques for training in a distributed fashion to accelerate the learning process.

## ACKNOWLEDGMENT

This work is supported by NSF PAWR (#1827923) and NSF IRNC (#2029295).

## REFERENCES

- [1] Y. Liu *et al.*, "Toward Edge Intelligence: Multiaccess Edge Computing for 5G and Internet of Things," in *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 6722-6747, Aug. 2020, doi: 10.1109/JIOT.2020.3004500.
- [2] Z. Ning *et al.*, "Green and Sustainable Cloud of Things: Enabling Collaborative Edge Computing," in *IEEE Communications Magazine*, vol. 57, no. 1, pp. 72-78, January 2019, doi: 10.1109/MCOM.2018.1700895.

- [3] O. -M. Ungureanu *et al.*, "Collaborative Cloud - Edge: A Declarative API orchestration model for the NextGen 5G Core," 2021 IEEE International Conference on Service-Oriented System Engineering (SOSE), 2021, pp. 124-133, doi: 10.1109/SOSE52839.2021.00019.
- [4] S. Gong *et al.*, "Adaptive Resource Allocation of Multiple Servers for Service-Based Systems in Cloud Computing," 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC), Turin, 2017, pp. 603-608.
- [5] M. A. Habibi *et al.*, "A Comprehensive Survey of RAN Architectures Toward 5G Mobile Communication System," in *IEEE Access*, vol. 7, pp. 70371-70421, 2019, doi: 10.1109/ACCESS.2019.2919657.
- [6] L. Lei *et al.*, "Deep Reinforcement Learning for Autonomous Internet of Things: Model, Applications and Challenges," in *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1722-1760, thirdquarter 2020, doi: 10.1109/COMST.2020.2988367.
- [7] X. Wang *et al.*, "Convergence of Edge Computing and Deep Learning: A Comprehensive Survey," in *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 869-904, Secondquarter 2020, doi: 10.1109/COMST.2020.2970550.
- [8] A. Feriani *et al.*, "Single and Multi-Agent Deep Reinforcement Learning for AI-Enabled Wireless Networks: A Tutorial," in *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1226-1252, Secondquarter 2021, doi: 10.1109/COMST.2021.3063822.
- [9] M. Cheng *et al.*, "DRL-cloud: Deep reinforcement learning-based resource provisioning and task scheduling for cloud service providers," 2018 23rd Asia and South Pacific Design Automation Conference (ASPDAC), 2018, pp. 129-134, doi: 10.1109/ASPDAC.2018.8297294.
- [10] Y. Wei *et al.*, "Joint Optimization of Caching, Computing, and Radio Resources for Fog-Enabled IoT Using Natural Actor-Critic Deep Reinforcement Learning," in *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2061-2073, April 2019, doi: 10.1109/JIOT.2018.2878435.
- [11] H. Peng *et al.*, "Deep Reinforcement Learning Based Resource Management for Multi-Access Edge Computing in Vehicular Networks," in *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2416-2428, 1 Oct.-Dec. 2020, doi: 10.1109/TNSE.2020.2978856.
- [12] S. Nath *et al.*, "Dynamic Computation Offloading and Resource Allocation for Multi-user Mobile Edge Computing," *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020, pp. 1-6, doi: 10.1109/GLOBECOM42002.2020.9348161.
- [13] J. Chen *et al.*, "A DRL Agent for Jointly Optimizing Computation Offloading and Resource Allocation in MEC," in *IEEE Internet of Things Journal*, vol. 8, no. 24, pp. 17508-17524, 15 Dec.15, 2021, doi: 10.1109/JIOT.2021.3081694.
- [14] D. Raychaudhuri *et al.*, 2020. Challenge: COSMOS: A city-scale programmable testbed for experimentation with advanced wireless. Proceedings of the 26th Annual International Conference on Mobile Computing and Networking. Association for Computing Machinery, New York, NY, USA, Article 14, 1-13. DOI:https://doi.org/10.1145/3372224.3380891
- [15] A. Qadeer *et al.*, "Flow-Level Dynamic Bandwidth Allocation in SDN-Enabled Edge Cloud using Heuristic Reinforcement Learning," 2021 8th International Conference on Future Internet of Things and Cloud (FiCloud), 2021, pp. 1-10, doi: 10.1109/FiCloud49777.2021.00009.
- [16] Arno, H. Van, Mazur, M.: Telecom infrastructure the open source way. Technical report, Ubuntu, Canonical, United Kingdom (October 2021)
- [17] J. Yang *et al.*, "Regional Smart City Development Focus: The South Korean National Strategic Smart City Program," in *IEEE Access*, vol. 9, pp. 7193-7210, 2021, doi: 10.1109/ACCESS.2020.3047139.
- [18] Y. Liu *et al.*, "Adaptive Multi-Resource Allocation for Cloudlet-Based Mobile Cloud Computing System," in *IEEE Transactions on Mobile Computing*, vol. 15, no. 10, pp. 2398-2410, 1 Oct. 2016, doi: 10.1109/TMC.2015.2504091.
- [19] Chen, Z. *et al.*, "Decentralized computation offloading for multi-user mobile edge computing: a deep reinforcement learning approach." *J Wireless Com Network* 2020, 188 (2020). https://doi.org/10.1186/s13638-020-01801-6
- [20] S. Gao *et al.*, "Short-term runoff prediction with GRU and LSTM networks without requiring time step optimization during sample generation," *Journal of Hydrology*, Volume 589, 2020, 125188, ISSN 0022-1694, https://doi.org/10.1016/j.jhydrol.2020.125188.

# A Study on Update Frequency of Q-Learning-based Transmission Datarate Adaptation using Redundant Check Information for IEEE 802.11ax Wireless LAN

Kazuto Yano<sup>†</sup> Kenta Suzuki<sup>†</sup> Babatunde Segun Ojetunde<sup>†</sup> Koji Yamamoto<sup>‡</sup>  
<sup>†</sup>Wave Engineering Laboratories, Advanced Telecommunications Research Institute International  
2-2-2 Hikaridai, Seika, Souraku, Kyoto 619-0288, Japan  
<sup>‡</sup>Graduate School of Informatics, Kyoto University  
Yoshida-honmachi, Sakyo-ku, Kyoto 606-8501, Japan  
{kzyano, kenta-suzuki, ojetunde}@atr.jp, kyamamot@i.kyoto-u.ac.jp

**Abstract**—The authors have proposed a practical transmission datarate adaptation (TDA) scheme using Q-learning applicable to IEEE 802.11ax wireless local area networks (WLANs). In the proposed scheme, each basic service set (BSS) selects an appropriate transmission datarate according to the buffer statuses of adjacent BSSs which are periodically collected and the transmission results of DATA frames in the BSS. Then, the BSS conducts underlay transmissions based on the framework of spatial reuse defined in IEEE 802.11ax. This paper compares two methods of Q-value update. One method is the frame-by-frame update taking account for the payload length and the transmission datarate of each DATA frame. The other method is the periodic update based on the total throughput in a BSS. The performance of the proposed scheme is evaluated through system-level computer simulation based on IEEE 802.11ax WLAN assuming downlink and uplink full-buffer traffic. It is confirmed that the proposed scheme can achieve better average area throughput than conventional Robust Rate Adaptation Algorithm (RRAA) and adaptive modulation and coding (AMC) in most of cases. It is also confirmed that the frame-by-frame Q-value update can achieve better area throughput than the periodic update based on the total throughput.

**Index Terms**—Wireless LAN, IEEE 802.11ax, Q-learning, transmission datarate adaptation, underlay transmission

## I. INTRODUCTION

IEEE 802.11 wireless local area networks (WLANs) [1] have been widely and densely deployed, and their demand is still growing. The current WLAN generally employs a distributed and autonomous channel access mechanism based on carrier sense multiple access with collision avoidance (CSMA/CA). Therefore, increase of traffic demand in the WLAN brings severe contention among multiple basic service sets (BSSs) sharing the same radio channel. It causes frequent collision and resultant failure of frame transmission.

Since IEEE 802.11 WLANs support multiple transmission datarates, each transmitter needs to adjust its transmission datarate to a suitable one. When the signal-to-noise ratio (SNR) of a transmitted frame is insufficient, its transmitter should lower the transmission datarate. On the other hand, when transmission failure is caused by collision, the transmis-

sion datarate should be raised so as to shorten the length of the transmitted frame and to reduce the probability of collision consequently.

Therefore, in general, the transmission datarate is adjusted according to one or more metrics calculated from the results of frame transmission [2]. For example, Automatic Rate Fallback (ARF) [3] raises/decreases the transmission datarate if the number of successful/failed frame transmissions reaches predetermined a threshold. Robust Rate Adaptation Algorithm (RRAA) [4] raises/decreases the transmission datarate if a frame error rate (FER) becomes larger/less than a given FER threshold. SampleRate [5] measures the average transmission time on an other WLAN channel other than the current operating channel, and selects the WLAN channel on which the average transmission time becomes minimum.

However, IEEE 802.11 WLANs have no way to know directly whether a failure of frame transmission is caused by collision or by other reasons such as insufficient SNR because the transmitter recognizes its transmission failure by occurring timeout of ACK frame reception. Several studies were recently conducted to estimate the factor of transmission failure using one or more frame sniffers to utilize the estimated cause to adjust the transmission datarate [6], [7]. In these studies, frame sniffers are employed to collect information of frame transmissions and judge whether multiple transmitters transmit their frames simultaneously and cause a collision. The aim of these studies is to use the estimation result (i.e. the cause of transmission failure) to determine the proper transmission datarate on the operating channel.

Furthermore, a transmission datarate adaptation (TDA) scheme was studied to select an appropriate transmission datarate using Q-learning with the aid of side information called “redundant check information” about the frame transmission in adjacent basic service sets (BSSs) [8], [9]. This scheme collects, as the redundant check information, the information whether or not adjacent BSSs will transmit their frames in near future, and then learns and selects the best action

(i.e., selects the best transmission datarate or defers its frame transmission). If the best transmission datarate is selected, each BSS makes underlay transmission against its adjacent BSSs, and thus this scheme can improve the throughput.

However, the performance evaluation conducted in [8], [9] assumes slotted channel access. On the other hand, IEEE 802.11 WLANs employ random backoff based on CSMA/CA [1], and thus it is difficult to precisely know when each node will transmit its frame. Hence, the authors have proposed a practical scheme to apply the concept of transmission datarate adaptation (TDA) in [8], [9] to IEEE 802.11ax WLAN [10] which defines a mechanism of spatial reuse for underlay transmissions [11], [12]. The proposed scheme adjusts the transmission datarate of each BSS based on Q-learning at an adaptation interval. The buffer status of each BSS is collected as the redundant check information, and it is obtained using buffer status report (BSR) defined in the IEEE 802.11ax standard.

For further study of the proposed TDA scheme, this paper introduces and compares two methods of Q-value update in the Q-learning. One method is frame-by-frame update taking account for the payload length and the transmission datarate of each DATA frame. The other method is the update based on the total throughput in a BSS. The performance of the proposed scheme is evaluated through system-level computer simulation based on IEEE 802.11ax WLAN assuming downlink and uplink full-buffer traffic with different payload length.

The remainder of this paper is as follows. Section II explains our proposed TDA scheme. Section III introduces the two methods of Q-value update. Section IV explains the configuration of the system-level computer simulation and its results. Finally, conclusion is given in Sect. V.

## II. TDA SCHEME USING REDUNDANT CHECK INFORMATION

### A. Basic Concept of TDA using Redundant Check Information and Q-Learning

Figure 1 shows the concept of the TDA presented in [8], [9] applicable to slotted channel access. We focus on BSS 0 as the target BSS of TDA, and other BSSs are adjacent BSSs. BSS 0 collects the information whether adjacent BSSs will transmit their frames or not in the next slot. As shown in Fig. 1, this information is encoded as “state.” BSS 0 selects a transmission datarate randomly with a certain probability  $P_{\text{rand}}$  (hereafter, this probability is called as “random selection probability”) for searching the appropriate action. Otherwise, the transmission datarate with the maximum Q-value on the state using the knowledge obtained through learning. BSS 0 makes its DATA frame transmission at the selected transmission datarate, and then updates the Q-value according to the result of frame transmission.

The above process is conducted slot-by-slot (in other words, frame-by-frame). This scheme can improve throughput comparing with the conventional slotted ALOHA because it can select an appropriate transmission datarate even when collision is expected.

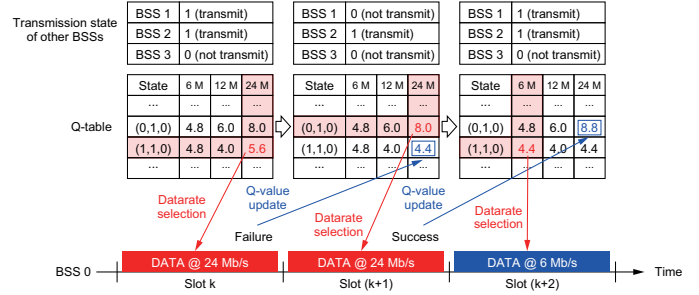


Fig. 1. Basic concept of TDA using redundant check information and Q-learning in [8], [9].

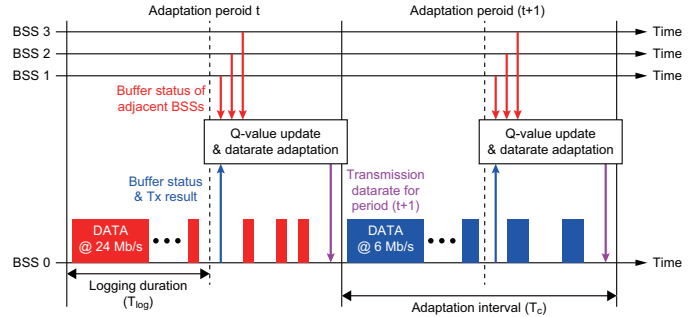


Fig. 2. Concept of our proposed TDA for IEEE 802.11ax WLAN [11], [12].

### B. Proposed TDA for IEEE 802.11ax WLAN

Figure 2 shows the concept of our proposed TDA scheme [11], [12]. In the proposed scheme, the transmission datarate is adjusted at an interval ( $T_c$ ). Since BSS 0 cannot know when adjacent BSSs will transmit their frames exactly due to the random backoff, it collects the buffer statuses of adjacent BSSs, and selects the transmission datarate that will be used in the next adaptation period. Here, an access point (AP) in each BSS obtains the buffer status of its associating stations (STAs) by BSR. Each node checks whether the received frame comes from the BSS that is expected to transmit frame(s) in the current adaptation period by using BSS Color [10] defined in the IEEE 802.11ax standard. (The state of such BSS is denoted by “1” in Fig. 2.) If the node detects a frame from such BSS, the CCA level at the node is raised so that the frame from the BSS is not detected using the framework of spatial reuse, which is also defined in the IEEE 802.11ax standard, in order to enable underlay transmissions.

The Q-value is updated using the transmission results of DATA frames in a logging duration ( $T_{\text{log}}$  from the beginning of the adaptation period). The detail of Q-value update is explained in the next section.

## III. Q-VALUE UPDATE METHODS

Since our proposed TDA scheme uses the transmission results of DATA frames in the logging duration, we can take two ways of Q-value update. One is updating the Q-value

frame-by-frame using the transmission result of each DATA frames in the logging duration as follows [11], [12]:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \{r_{t+1} - Q(s_t, a_t)\}, \quad (1)$$

where  $\alpha$  is the learning rate,  $s_t$  is the state of buffers in adjacent BSSs, and  $a_t$  is the selected action of the target BSS (i.e., BSS 0 in Fig. 2). The reward  $r_{t+1}$  of each DATA frame is calculated by

$$r_{t+1} = \begin{cases} (\delta_{X,1}W_{\text{suc}} - \delta_{X,0}W_{\text{fail}})RD & \text{if } R \geq 0 \\ r_{\text{notx}} & \text{if } R = -1, \end{cases} \quad (2)$$

where  $\delta_{a,b}$  denotes Kronecker delta,  $X$  is the transmission result (“1” means SUCCESS, and “0” means FAILURE) of the DATA frame,  $D$  [kbyte] is the payload length of the DATA frame,  $W_{\text{suc}}$  and  $W_{\text{fail}}$  are the weights of reward for successful and failed frame transmissions, respectively.  $R$  is the used transmission datarate [Mb/s]. Here,  $R = -1$  denotes that frame transmission is pended in the corresponding adaptation period, and the reward is set to  $r_{\text{notx}}$  in this case.

Since the reward (and the Q-value) highly depends on the payload length in the above method, it may not work well if multiple traffic flows with different payload sizes coexist in a BSS. Hence, we also introduce another method to update the Q-value by aggregating the transmission results of DATA frames in an adaptation period (i.e., period-by-period update) as follows:

$$r_{t+1} = \begin{cases} \sum_i (\delta_{X_i,1}W_{\text{suc}} - \delta_{X_i,0}W_{\text{fail}}) D_i & \text{if } R \geq 0 \\ r_{\text{notx}} & \text{if } R = -1, \end{cases} \quad (3)$$

where  $X_i$  is the transmission result of the  $i$ th DATA frame in the logging duration,  $D_i$  [kbyte] is the payload length of the  $i$ th DATA frame. In this method, the transmission datarate  $R$  is not accounted for in the reward calculation because the rewards corresponding to the throughput of the BSS in the adaptation period by taking summation of the payload length of every DATA frame.

#### IV. COMPUTER SIMULATION

##### A. Simulation Configurations

The performance of the proposed TDA scheme is evaluated through system-level computer simulation based on IEEE 802.11ax WLAN. We compare four schemes: the proposed scheme with two different methods of the Q-value update, RRAA, and adaptive modulation and coding (AMC) based on the received power.

Table I shows the simulation parameters. The area size is assumed to be  $80\text{ m} \times 80\text{ m}$ , and it is segmented into  $4 \times 4$  (thus, the segment size is  $20 \times 20\text{ m}$ ). Each BSS is located in different segment as shown in Fig. 3. In the proposed scheme, each BSS collects the buffer statuses of the BSSs whose frame is received with the received power equal to or greater than  $-88\text{ dBm}$  which is same as the frame detection limit in this simulation. The random selection probability of action  $P_{\text{rand}}$  at the  $t$ th adaptation period is set by

$$P_{\text{rand}} = 1/(1 + t/133), \quad (4)$$

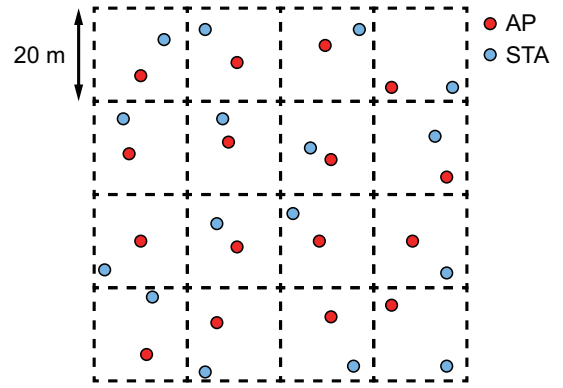


Fig. 3. Simulation area.

which gives  $P_{\text{rand}} = 0.1$  at 2 minutes, because it gives high performance in the preliminary simulation.

In RRAA, the transmission datarate is adjusted every  $T_c$  if the number of transmitted DATA frames after the previous datarate adjustment is equal to or greater than 50. If frame error rate (FER) is greater than 0.1, the modulation and coding scheme (MCS) index is decremented by one, and it is incremented by one if the FER is less than 0.05. Otherwise, the current MCS index is retained. In AMC, the maximum MCS index at which the received power satisfies the minimum input sensitivity defined in the IEEE 802.11ax standard [10].

In this evaluation, downlink and uplink full-buffer traffic is assumed. Four configurations of the payload length of the DATA frames are assumed as follows.

- 3 kbytes (downlink), 1 kbyte (uplink)
- 3 kbytes (downlink), 3 kbytes (uplink)
- 15 kbytes (downlink), 5 kbytes (uplink)
- 15 kbytes (downlink), 15 kbytes (uplink)

It should be noted that the MAC efficiency improves, and the number of DATA frames used for updating the Q-value reduces as the payload becomes longer.

##### B. Simulation Results

Figures 4 and 5 show the average area throughput and the frame delivery rate of the proposed scheme, AMC, and RRAA, respectively when the number of BSSs is 4. In addition, the average number of DATA frames transmitted at each MCS index when the payload lengths for the downlink and uplink are 3 kbytes and 1 kbyte is shown in Fig. 6. The average number of DATA transmitted frames when the payload length is 15 kbytes for both the downlink and uplink is shown in Fig. 7.

The average area throughput of RRAA is quite low because it selects low transmission datarates (mainly, MCS indexes 0 and 1), and resultantly the number of DATA frame transmissions is smaller than other schemes. The average area throughput of AMC is lower than the proposed scheme even though it selects higher transmission datarates (mainly, MCS index 7). This is because it does not care frame collision, and resultantly the frame delivery rate is lower than other

TABLE I  
SIMULATION SETTINGS

Evaluation duration	20 minutes
Area size	80 m × 80 m (20 × 20 m segment, 4 × 4 segments)
Number of BSSs	4, 16
Number of STAs per BSS	1
Supported MCS index	0–9 for IEEE 802.11ax [10]
Transmission power	20 dBm
Signal bandwidth	20 MHz
Noise level	−94 dBm (including 7 dB noise figure)
Minimum signal detection level	−88 dBm
Propagation model	IEEE 802.11 TGax Residential scenario [13]
Frequency channel	Ch 1 in the 2.4 GHz band
Retransmission limit	7 times
RTS/CTS exchange	Not in use
Interval of datarate adaptation ( $T_c$ )	100 ms
Logging duration ( $T_{\log}$ )	80 ms
Learning rate ( $\alpha$ )	0.1
Weights of reward	$(W_{\text{suc}}, W_{\text{fail}}) = (1, 0.1)$
Reward for pending transmission ( $r_{\text{notx}}$ )	−2

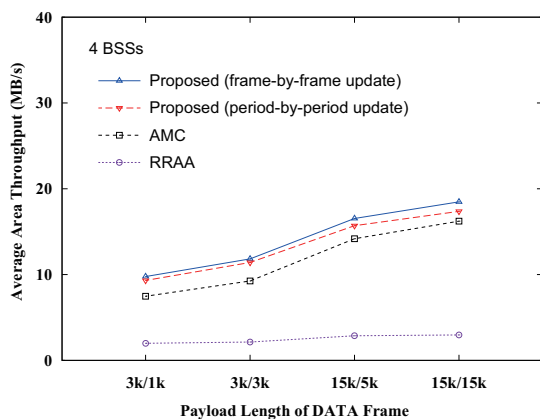


Fig. 4. Average area throughput performance (4 BSSs).

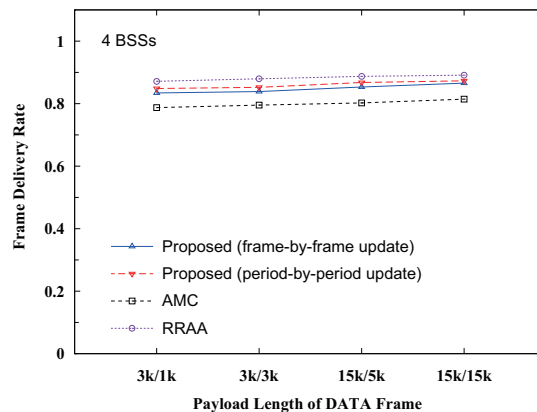


Fig. 5. Frame delivery rate performance (4 BSSs).

schemes. The proposed scheme achieves better average area throughput than the conventional two schemes because it selects higher transmission datarates and can keep the frame delivery rate relatively high even though underlay transmission is introduced. In the proposed scheme, the frame-by-frame Q-value update tends to select higher transmission datarates than the period-by-period Q-value update does. It implies that it is better to explicitly take the transmission datarate account for the reward calculation.

Figures 8 and 9 show the average area throughput and the frame delivery rate of the proposed scheme, AMC, and RRAA, respectively when the number of BSSs is 16. In addition, the average number of DATA frames transmitted at each MCS index when the payload lengths for the downlink and uplink are 3 kbytes and 1 kbyte is shown in Fig. 10. The average number of DATA transmitted frames when the payload length is 15 kbytes for both the downlink and uplink is shown in Fig. 11.

The proposed scheme with the period-by-period Q-value update likely selects middle transmission datarates (around MCS index 4), and the throughput degradation from the frame-by-frame Q-value update becomes larger than the 4-BSS case. The proposed scheme with the frame-by-frame Q-value update also less selects higher transmission datarates than the 4-BSS case, and the average area throughput is worse than AMC when the payload length is 15 kbytes. It is because transmission failure occurs more frequently as the payload becomes longer, especially when there are many interfering nodes. Therefore, further tuning of operation parameters will be necessary for heavily-congested situations. For example, the frame delivery rate will be affected by the magnitude of the weight for transmission failure  $W_{\text{fail}}$ . If we can set appropriate  $W_{\text{fail}}$ , the proposed scheme can further improve the average area throughput.

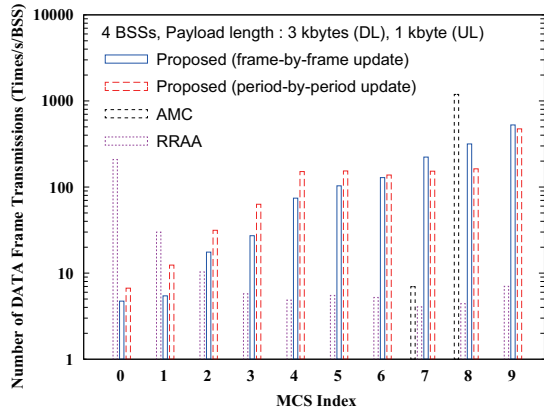


Fig. 6. Average number of transmit DATA frames per BSS (4 BSSs, 3 kbytes/1 kbyte payload).

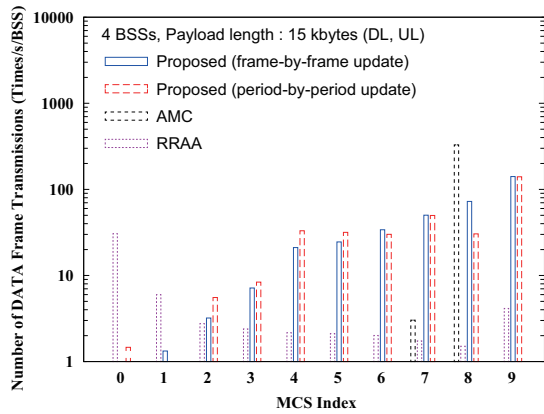


Fig. 7. Average number of transmit DATA frames per BSS (4 BSSs, 15 kbytes payload).

## V. CONCLUSION

This paper introduced two methods of Q-value update for our proposed TDA scheme using Q-learning applicable to IEEE 802.11ax WLAN. One method updates the Q-value frame-by-frame with taking account for the payload length and the transmission datarate of each DATA frame. The other method updates the Q-value period-by-period based on the total throughput in a BSS. The performance of the proposed scheme with two different Q-value updating methods was evaluated through system-level computer simulation based on IEEE 802.11ax WLAN assuming four configurations of down-link and uplink full-buffer traffic. It was confirmed that the proposed scheme can achieve better average area throughput than conventional RRAA and AMC schemes except when there are many interfering nodes and the payload of the DATA frames is long. It was also confirmed that the frame-by-frame Q-value update can achieve better area throughput than the period-by-period update. It implies that it is better to explicitly take the transmission datarate account for the reward calculation.

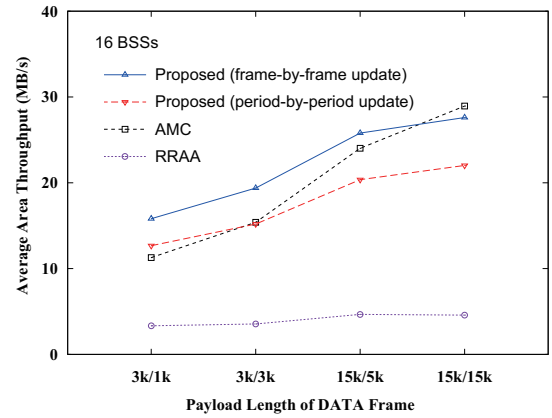


Fig. 8. Average area throughput performance (16 BSSs).

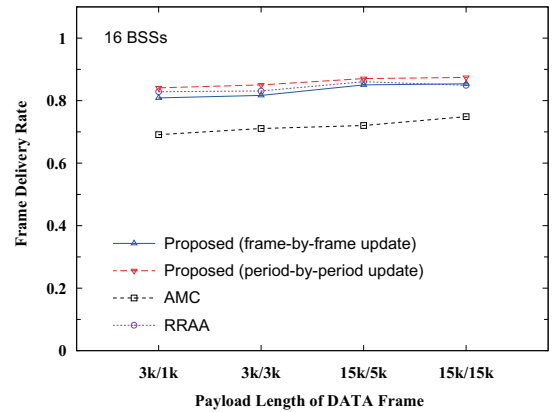


Fig. 9. Frame delivery rate performance (16 BSSs).

## ACKNOWLEDGMENT

This research and development work was supported by the MIC/SCOPE #JP196000002.

## REFERENCES

- [1] IEEE Std 802.11-2020, Feb. 2021.
- [2] W. Yin, P. Hu, J. Indulska, M. Portmann, and Y. Mao, "MAC-layer rate control for 802.11 networks: Lesson learned and looking forward," arXiv:1807.02827v1, July 2018.
- [3] A. Karmerman and L. Monteban, "WaveLAN-II: A high-performance wireless LAN for the unlicensed band," Bell Labs Technical Journal, vol. 2, no. 3, pp. 118–133, 1997. DOI:10.1002/bltj.2069
- [4] S. H. Y. Wong, H. Yang, S. Lu, and V. Bharghavan, "Robust rate adaptation for 802.11 wireless networks," Proc. ACM MOBICOM'06, pp. 146–157, Sept. 2006. DOI:10.1145/1161089.1161107
- [5] J. C. Bicket, "Bit-rate selection in wireless networks," Ph.D. thesis, MIT Master's Thesis, Feb. 2005.
- [6] H. Senda, O. Takyu, A. Kamio, M. Ohta, and T. Fujii, "Discrimination of communication quality deterioration utilizing retransmission flag and modulation and coding scheme(MCS) in wireless LAN," IEICE Technical Report, SR2020-7, pp. 45–49, June 2020.
- [7] K. Yamamoto, M. Mieda, S. Kondo, T. Nishio, A. Taya, and K. Yano, "Interference source determination based on history of transmissions in WLANs," IEICE Tech. Rep., RCS2021-141, pp. 122–125, Oct. 2021.
- [8] K. Yamamoto, Y. Kihira, Y. Koda, T. Nishio, and M. Morikura, "Factor analysis of communication quality using redundancy-check information in wireless LANs," Proc. IEICE Gen. Conf 2020, B-5-147, March 2020.

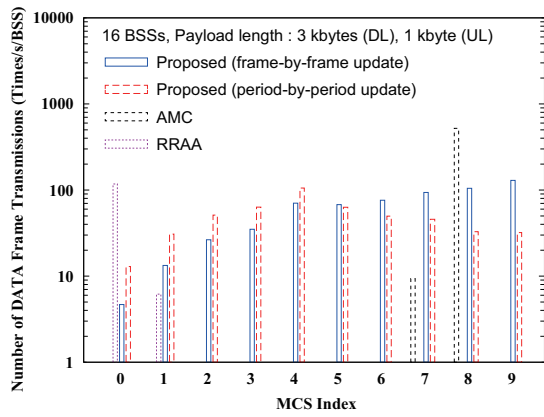


Fig. 10. Average number of transmit DATA frames per BSS (16 BSSs, 3 kbytes/1 kbyte payload).

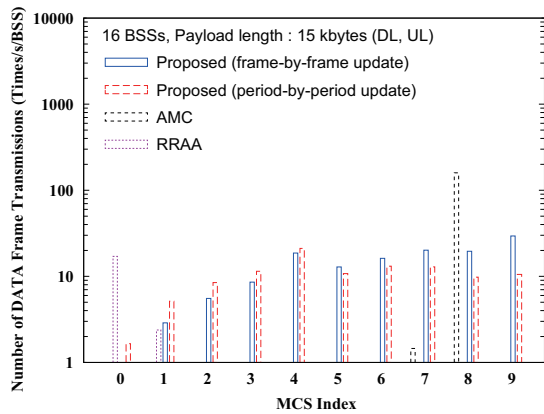


Fig. 11. Average number of transmit DATA frames per BSS (16 BSSs, 15 kbytes payload).

[9] Y. Kihira, Y. Koda, K. Yamamoto, T. Nishio, and M. Morikura, "Adversarial reinforcement learning-based robust access point coordination against uncoordinated interference," Proc. the 92nd IEEE Vehicular Technology Conference (VTC2020-Fall), Online, Nov. 2020. DOI: 10.1109/VTC2020-Fall49728.2020.9348462

[10] IEEE Std 802.11ax-2021, May 2021.

[11] K. Yano, K. Suzuki, B. Ojetunde, and K. Yamamoto, "Transmission datarate adaptation using redundant check information for IEEE 802.11ax wireless LAN," submitted to IEICE ComEX.

[12] K. Yano, K. Suzuki, B. Ojetunde, and K. Yamamoto, "Performance evaluation of access control and transmission datarate adaptation using redundant check information for IEEE 802.11ax wireless LAN," IEICE Tech. Rep., SR2021-81, pp. 103–110, Jan. 2022.

[13] R. Porat, *et al*, "11ax Evaluation Methodology," doc. IEEE 802.11-14/0571r12, Jan. 2016.

# User Coverage Maximization for a UAV-mounted Base Station Using Reinforcement Learning and Greedy Methods

Adhitya Bantwal Bhandarkar  
*Comm. & Info. Sciences Lab (CISL)*  
ECE Department,  
University of New Mexico,  
Albuquerque, USA  
abhandarkar@unm.edu

Sudharman K. Jayaweera  
*Comm. & Info. Sciences Lab (CISL)*  
ECE Department,  
University of New Mexico,  
Albuquerque, USA  
jayaweera@unm.edu

Steven A. Lane  
*Air Force Research Laboratory (AFRL)*  
Space Vehicles Directorate,  
Kirtland Air Force Base,  
Albuquerque, USA

**Abstract**—This paper proposes two methods to maximize the coverage of distinct ground users by an unmanned aerial vehicle (UAV) -mounted mobile base station: a deep reinforcement learning (DRL) approach, and a reward-based greedy approach. The performance of the proposed methods are evaluated based on two aspects: the number of distinct ground users covered, and the delay experienced by a user until it first receives coverage. The distribution of ground users is modelled as a Gaussian Mixture Model (GMM) with fixed and time-varying means with the latter scenario mimicking the mobility of users. Simulation results show that both proposed methods lead to efficient coverage and latency performance with the DRL based approach significantly outperforming the rewards-based greedy algorithm, especially when ground users are allowed to be mobile.

**Index Terms**—Optimal trajectory learning, Unmanned Aerial Vehicles (UAVs), wireless user coverage, Deep Reinforcement Learning (DRL), Deep Q-Network (DQN), greedy algorithm

## I. INTRODUCTION

The use of Unmanned Aerial Vehicles (UAVs) for civilian applications, in particular, using UAVs as portable Mobile Base Stations, has gained traction in the past few years [1] [2]. For this, UAVs are fitted with transceivers and are flown over areas where setting up traditional Base Stations (BSs) may not be feasible [3] [4]. For example, areas that are affected by natural (e.g., floods) or man-made disasters (e.g., terrorist attacks). Unlike traditional BSs, the portability of UAVs allows for choosing a flight path that covers as many distinct ground users as possible.

Several approaches have been proposed over the past years to provide better coverage to ground users. For example, the authors of [5] discuss deployment of single and multiple UAVs to provide coverage to ground users in a way that maximizes fairness and reduces interference. Using Reinforcement Learning (RL) based methods, the authors were able to significantly increase the fairness compared to baseline methods. However, [5] assumed the distribution of users to be static, which may not be the case in practice. Authors of [6] formulated the UAV trajectory optimization problem as a Mixed Integer Linear Programming Problem and proposed

an algorithm that finds an optimal trajectory iteratively. The authors assumed that the users are non-stationary and periodically send their location information to the UAV. Based on this information, the algorithm determines an optimal path. In this case, however, the performance and the optimality of the trajectory thus obtained is contingent upon the accuracy of location information relayed to the UAV by the users. Authors of [7] discussed techniques for optimal placement of a UAV in 3D space for maximal coverage of users. An algorithm was developed to determine the optimal location to place the UAV to cover maximum number of users at the required signal to noise ratio (SNR). However, owing to the limited coverage radius of the UAV, it is unlikely that all users can be covered from a single location, especially if users are allowed to be mobile. Placement of a UAV to optimize the user coverage by taking user mobility also into account was discussed in [8]. The authors modeled the movement of users as a random-walk. The distance travelled by a user in each transition of the random-walk is modeled to be Rayleigh distributed with a fixed shape parameter. The position of the UAV is updated periodically, which is solved iteratively based on the temporal coverage probability that the authors derived analytically. However, when the number of users is large, finding the optimal location to place the UAV for each update instance iteratively may not be feasible nor desirable. The authors of [9] proposed an algorithm to minimize the average distance between users and a UAV. When a few users are located outside of a group or a cluster, however, the resulting deployment might result in reduced signal quality for rest of the users due to the outliers.

Most of the papers in literature assess the performance of the system based solely on the number of users covered [6] [7]. This may not be the best metric because it does not take into account the coverage fairness; i.e., it is not possible to determine if a user, or a group of users, have higher uptime compared to other users. In addition, the delay experienced by a typical user until it first received the UAV coverage might also be important. The authors of [5] and [8] take the



fairness into account, but latency experienced by users is still not considered. In this paper, we propose two methods to find a UAV trajectory that maximizes the coverage for distinct users while also ensuring fairness in coverage: a deep reinforcement learning (DRL) -based approach, and a reward-based greedy method. We use the complimentary cumulative distribution functions (ccdfs) of number of distinct users covered, as well as the delay experienced by a user until it has coverage for the first time to gain deeper insights on the performance beyond averages. In addition, the performance of our methods are investigated for two different types of user distributions. In the first case, we assume that the users are clustered around regions of high user density called hotspots. In the second case, we allow these clusters to be mobile to model more realistic application scenarios.

The remainder of the paper is organized as follows. Section II discusses our system model and assumptions. Section III discusses the proposed trajectory learning methods. Section IV discusses the performance of the proposed approaches compared to other alternatives in simulated application scenarios. Finally, section V concludes the paper.

## II. SYSTEM MODEL AND THE PROBLEM FORMULATION

### A. System Model

We consider a square geographical area of side-length  $L$  divided into  $M$  square cells of equal size as shown in Figure 1. The number of square cells into which the area is to be divided is determined by the application context (which will be dependent on parameters such as the communications radius of the UAV). An arbitrary but known maximum number of users are assumed to be distributed in this area. We assume that the UAV maintains a fixed altitude precluding the actions of moving up or down. In many situations this may make sense in order to provide uniform coverage while simplifying interference management among multiple UAVs and subscribers. For simplicity, the UAV is restricted to hover only over the center and corners of the cells, so that there are  $2\sqrt{M}(\sqrt{M} + 1) + 1$  possible hovering points. If we assume that the coverage radius of the UAV base station is  $l/2$ , where  $l$  is the side-length of a square cell, then the UAV base station will be able to provide coverage to any user from one of the possible hovering points. This means that with the proposed model, the area of interest need not necessarily be a square. It can be of any arbitrary shape, but with sufficient number of hovering points so that any location can be covered by the UAV from one of the hovering points. With this model, the movement of the UAV can be restricted to nine directions namely: North, North-East, East, South-East, South, South-West, West, North-West or be stationary.

Since UAVs are powered by batteries, their flying time is limited. We divide the total flying time of the UAV into slots of equal duration. During each slot, the UAV is expected to hover over one of the possible hovering points providing connectivity to the users inside its coverage radius. The objective of this paper is to design methods that find a flight path for the UAV-

mounted mobile BS to provide coverage to as many *distinct* ground users as possible before exhausting the UAV's energy.

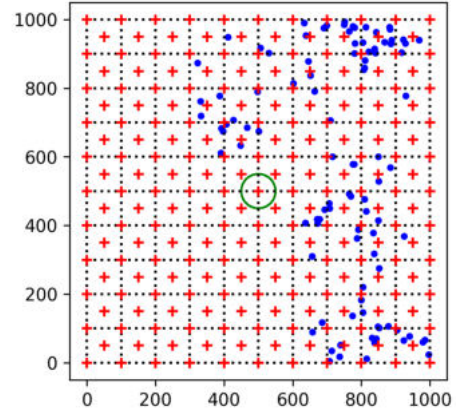


Fig. 1. A UAV hovering at location (500,500). The UAV coverage radius is represented by the green circle. The blue dots indicate the users and the 221 red crosses represent the possible hovering points.

### B. User Distribution

The authors of [5] assumed that the users are distributed uniformly over the square area. In reality, however, this may not be the case, since users are usually clustered around places like offices, universities and residential areas. The authors of [10] used the map of downtown San Francisco for planning the path of a UAV. Although this approach is realistic, it may not necessarily generalize well to other places.

In this paper, we use two approaches to model how users are distributed over the area of interest. In the first method, we assume that the users are clustered around certain fixed hotspots. In the second approach, we assume that the users are clustered and the cluster means are time-varying. In both approaches, we model the clusters as Gaussian mixture models (GMMs). Each component or cluster in the mixture is parameterized by cluster weight,  $\pi_k$ , cluster mean,  $\mu_k$ , and variance,  $\sigma_k^2$ . Thus, the location distribution of the users can be written as:

$$(X_n, Y_n) \sim \left( \sum_{k=1}^K \pi_k \mathcal{N}(\mu_x, \sigma_x^2), \sum_{k=1}^K \pi_k \mathcal{N}(\mu_y, \sigma_y^2) \right)$$

where  $K$  is the number of clusters and  $\sum_{k=1}^K \pi_k = 1$ . The cluster weights represent the probability of users being in each cluster. For example, Figure 1 corresponds to a user distribution with  $K = 4$  clusters.

The cluster mean,  $\mu_k = [\mu_x, \mu_y]$ , represents the mean  $X$  and  $Y$  coordinates of a user in each cluster  $k$  while the cluster variances,  $\sigma_k^2 = [\sigma_x^2, \sigma_y^2]$ , represents the location spread along  $X$  and  $Y$  coordinates. A smaller variance represents a tightly packed cluster whereas a larger variance means a cluster in which users are more spread out. Figure 1 corresponds to a distribution in which all clusters have the same variance of 100.

In the second approach, we allow the cluster means to move in a fixed direction and magnitude. This can be appropriate when the UAV is providing coverage to on-the-move ground

troops or people commuting to work from their homes. Note that, the methods developed do not require the user mobility to have a fixed direction and/or magnitude. This is assumed for the purpose of training simplicity. Future work will investigate the scenarios with arbitrary mobility directions and magnitudes.

### III. PROPOSED UAV TRAJECTORY LEARNING METHODS

#### A. Proposed Method 1: RL Approach Using DQN

According to our model, the total flying time of the UAV is divided into time slots and during each time slot, a decision to be made with regard to the UAV's movement. This setup can be modelled as a Markov Decision Process (MDP). In an MDP, we have an environment and an agent that interacts with the environment. At each time step,  $t$ , the agent observes the state of the environment at that time denoted by  $S_t$ , takes an action appropriate for that state and time given by  $a_t$  and receives a scalar reward  $r_t$  depending on the action and the environment's transition to a new state  $S_{t+1}$ . The reward function is used to measure an action qualitatively. An action with favorable impact on the performance will have a higher reward whereas an action with an adverse affect on the performance will have a lower reward. The reward function for the  $t$ -th time instance is defined as:

$$r_t = (|N_t| - |N_{t-1}|) + g * V + F_t + p_t \quad (1)$$

where  $N_t$  is the set of users covered until  $t$ -th time instant. Thus,  $|N_t| - |N_{t-1}|$  is the number of new users who are not previously covered,  $V \in \{0, 1\}$ , denotes whether the current hovering point at  $(x_t, y_t)$  was visited previously, and  $g$  is a scaling factor. The Jain's fairness function  $F_t$  at time step  $t$  for  $N$  ground users is defined as [11]:

$$F_t = \frac{(\sum_{n=1}^N \sum_{i=0}^t Cov_n^i)^2}{N(\sum_{n=1}^N (\sum_{i=0}^t Cov_n^i)^2)}$$

where  $Cov_n^t \in \{0, 1\}$ , for  $n \in \{1, \dots, N\}$ , denotes whether the  $n$ -th ground user has the coverage at time instant  $t$  or not:

$$Cov_n^t = \begin{cases} 1, & \text{if the user is inside coverage area of UAV} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

The penalty function,  $p_t$ , is non-zero only if the UAV attempts an action that will cause it to fly out of the designated area:

$$p_t = \begin{cases} 0, & \text{if } 0 \leq x_{t+1}, y_{t+1} \leq L \\ P, & \text{otherwise} \end{cases} \quad (3)$$

In Eq. (3),  $(x_{t+1}, y_{t+1})$  denotes the resulting coordinates of the UAV if action  $a_t$  is taken at the  $t$ -th time step and  $P < 0$  is the penalty for selecting an action that results in flying outside the desired coverage area. A large magnitude for the penalty  $P$  discourages the UAV from taking actions in future that would result in UAV flying out of the desired area.

Note that, if only the number of users covered was taken as the reward function in Eq. (1), there is a possibility that the UAV may get stuck at a hovering point and not move further,

because any movement may result in a reduced reward. The reward function, Eq.(1), avoids this because if the UAV hovers at the same location twice, then the first and the second term of the reward function will be zero, thus reducing the reward for those time instants. The proposed reward function encourages the UAV to visit new hovering points, thereby increasing the probability of covering new users.

For any given state  $S_t$ , the agent takes an action  $a_t$  that maximizes the sum of discounted expected future rewards [12]. In the case of Q-Learning, this is done by maintaining a Q-table, whose entries represent these expected future rewards if a particular action was taken when in a particular state. Hence, the agent selects an action that has the highest expected future rewards as:  $a_t = \operatorname{argmax}_a Q(S_t, a)$ .

Note that,  $Q(S_t, a)$  is a function of the two variables  $S_t$  and  $a$ . A Q-table approach to learning the function  $Q(S_t, a)$  works if both  $S_t$  and  $a$  were to take values on finite sets. Even then, unless the number of actions and possible states are relatively small, the Q-table approach may not be convenient in practice. The Deep Q-Network (DQN) approach for RL proposed in [13], instead uses a deep neural network (DNN) as a function estimator to learn  $Q(S_t, a)$ . The DQN makes use of an experience replay [13] where a replay memory is used to store tuples of transitions consisting of the state,  $S_t$ , the action  $a_t$  taken at that state, reward  $r_t$  obtained by taking that action, and the resulting new state,  $S_{t+1}$ . At every time step, the DQN is trained by sampling a random batch of experiences from the experience replay memory. The Algorithm 1 details the DQN-based RL approach.

At any given time instant, the DQN makes a decision based on the current state,  $S_t$ . Hence, the state  $S_t$  must be selected carefully to incorporate vital details of the environment that can impact the performance. The state  $S_t$  for time step  $t$  in the proposed method consists of the  $X$  and  $Y$  coordinates of the location of the UAV at that time instance, the users covered at that location, a binary value indicating whether the current hovering point was visited previously and the remaining UAV energy at that time instant along with the same information for a finite number of previous time instances.

#### B. Proposed method 2: An AI based greedy UAV trajectory learning algorithm

For a time instant,  $t$ , and corresponding state,  $S_t$ , the proposed greedy algorithm chooses an action  $a_t$  that yields immediate maximum reward  $r_t$  defined in Eq. (1). The greedy algorithm does not take into account whether the selected action will have unfavorable results in the long run. It should be noted that if there are two or more actions that yield the same reward, an action is selected at random out of the multiple actions resulting in the identical reward.

Although the algorithm is relatively simple compared to the DQN-based approach proposed in the previous section, it can nevertheless be effective in many situations. Indeed, as we will show in the next section, it was able to perform reasonably well compared to the DQN method with significantly lower computational complexity.

---

**Algorithm 1** DQN Algorithm

---

```
1: Initialize policy network  $Q(s, a)$  with random weights
2: Set target network  $Q'(s, a)$  with the same weights
3: Set  $\varepsilon = 1$ 
4: Create a replay memory
5: for episode = 1 to M do
6:   for time = 1 to N do
7:     Generate a random number  $w$  between 0 and 1
8:     if  $\varepsilon > w$  then
9:       Take a random action  $a_t$ 
10:    else
11:      Take action:  $a_t = \operatorname{argmax}_a Q(S_t, a)$ 
12:    Store transition:  $(S_i, a_i, r_i, S_{i+1})$  into buffer
13:    Sample random batch of transitions from the buffer
14:    if Episode ends at  $i + 1$  then
15:      Set  $y_i = r_i$ 
16:    else
17:      Set  $y_i = r_i + \gamma \max_{a'} Q'(S_{i+1}, a')$ 
18:    Fit the policy network on the data:  $(S_t, y_t)$ .
19:    Reduce the value of  $\varepsilon$ 
20:    Every K steps, set weights of target equal to policy
    network
```

---

#### IV. SIMULATION RESULTS

We consider an area of  $1\text{ km} \times 1\text{ km}$  divided into  $M = 100$  square cells of similar size resulting in 221 hovering points as was shown in Figure 1. The users were distributed over this area according to a Gaussian Mixture Model (GMM) with either  $K = 4$  or  $K = 8$  clusters. The parameters of the GMMs are given in tables I, II and III.

We use three metrics to assess the performance of our proposed methods: (1) the average number of distinct users covered in an episode, (2) the complementary cumulative distribution function (ccdf) of the number of users covered, and (3) the average delay experienced by a user until it receives coverage for the first time (in an episode) where the average delay is computed as:

$$\text{Average Delay} = \frac{\sum_{t=1}^T (t \times U_t)}{\sum_{t=1}^T U_t}$$

where  $U_t$  denotes the number of users covered during time step  $t$ , and  $T$  denotes the time step when the UAV has exhausted its energy.

Note that, sometimes the ccdf may give better insight on the performance than just comparing the averages. Specifically, the ccdf shows the probability of covering more than a given number of users during a UAV flying episode, which cannot be deduced by knowing only the average number of users covered.

##### A. Stationary user distributions

In this setup, during each epoch, 100 users are independently and identically distributed (iid) over the desired area according to a GMM with  $K$  clusters. An epoch is defined as one run of the UAV from the beginning to exhaustion of its

energy. It should be noted that, the number of clusters, and the means and variances of the GMM are fixed throughout each simulation.

With stationary distributions, two scenarios are discussed: (1) clusters with larger variances representing user distributions that are spread out, and (2) clusters with smaller variances representing densely packed users. The parameters of the GMM used in each of the cases are tabulated in Table I and Table II, respectively for the case  $K = 4$ . Table IV and Table V give additional parameters about the environment and specifications of the Convolutional Neural Network (CNN) used.

---

**TABLE I** Gaussian Mixture Parameters for Larger Variance

---

Number of clusters (K)	4
Cluster Weights ( $\pi_k$ )	0.25,0.25,0.25 and 0.25
Cluster Means ( $\mu_k$ )	(500,800) (800,0) (800,400) (850,900)
Cluster Variances ( $\sigma_k^2$ )	100, 100, 100 and 100

---

---

**TABLE II** Gaussian Mixture Parameters for Smaller Variance

---

Number of clusters (K)	4
Cluster Weights ( $\pi_k$ )	0.25,0.25,0.25 and 0.25
Cluster Means ( $\mu_k$ )	(300,200) (800,400) (200,600) (500,800)
Cluster Variances ( $\sigma_k^2$ )	10, 10, 10 and 10

---

---

**TABLE III** Gaussian Mixture Parameters for Time Varying Means

---

Number of clusters (K)	4
Cluster Weights	0.25,0.25,0.25 and 0.25
Cluster Means	(300,200) (800,400) (200,600) (500,800)
Cluster Variances	100, 100, 100 and 100
Direction	$\pi/4$ radians
Magnitude	$1m$

---

The DQN algorithm learns iteratively by means of trial and error. It takes about 500 training epochs for the algorithm to find an optimal path as can be inferred from Figure 2.

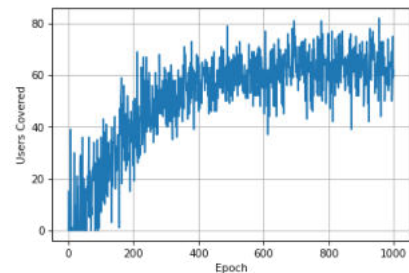


Fig. 2. Epoch as a Function of the Number of Users Covered

Figure 3 shows the number of distinct ground users covered by the reward-based greedy, the trained DQN and random action selection algorithms when users are distributed according

to GMMs with  $K = 4$  and  $K = 8$ . Note that, the latter takes random decisions at each time step. From Figure 3, it can be seen that compared to random action selection method, a significant number of users are covered by the proposed methods. Moreover, it is seen that the proposed rewards-based greedy algorithm performance is only about 15% - 20% less than that of the DQN-based approach. It is also interesting to notice that the performance of both learning methods slightly deteriorates when the number of clusters increases while the opposite is true for the random action selection algorithm. This may be attributed to the fact that more clusters make systematic learning more difficult while helping random guesses to be more successful.

**TABLE IV** Simulation parameters

Area	1000m x 1000m
Number of Blocks (M)	100
Size of Blocks	100m x 100m
Number of Hovering points	221
Number of users (N)	100
Number of time slots	67
Radius of coverage of UAV	50m
Penalty (P)	-1
g	0.2

**TABLE V** Specifications of the Convolutional Neural Network used as the DQN

Input Shape	10x5x1
Kernel Size	2 x 2
Padding	Same
Layers	3
Number of filters in each layer	40, 45 , 55
Activation function	Sigmoid
Optimizer	Adam
Loss Function	Huber
Gamma	0.9
Learning rate	0.0001

Let  $U$  denote the number of distinct users covered by the UAV in an epoch. In Figure 3 we show the ccdf of  $U$  given by  $Pr(U > n)$ . It is worth observing from Figure 3 that the maximum number of users that can be covered with a non-zero probability is also larger with the DQN based trajectories compared to that with the greedy trajectories.

Figure 4 shows the cumulative distribution function (cdf) of average delay until a user is first covered in an episode. Clearly, both proposed learning algorithms result in much smaller average delays than with the random actions. Perhaps surprisingly, the average delay with the greedy algorithm is not too far from the DQN-based learning algorithm, again showing that the proposed rewards-based greedy algorithm is a good alternative to the DQN-based learning at much less computational complexity and learning duration.

In some practical scenarios, users may be densely concentrated around mobile hotspots like offices, universities, or

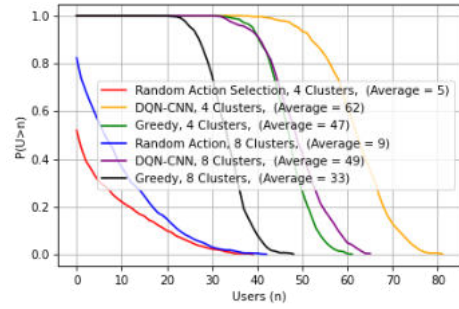


Fig. 3. Complementary Cumulative Distribution Function Showing Probability of Covering More Than  $n$  Users

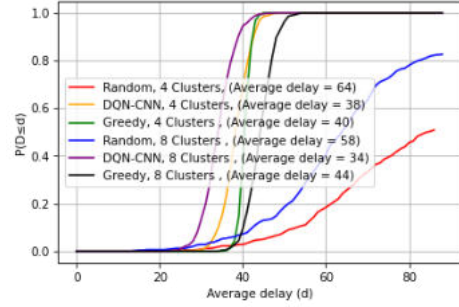


Fig. 4. Cumulative Distribution Function Showing Probability of the Average Delay  $< d$

residential areas. Figure 5 and Figure 6 show the ccdf of number of users covered and the cdf of average delay seen by a user for the first time coverage during an epoch for this case, respectively. It can be seen from Figure 5 that the proposed DQN and the rewards-based greedy methods are at par with each other, covering more than 70 and 50 users on average in four and eight cluster cases, respectively. Moreover, Figure 5 shows that DQN-based learning offers additional desirable performance traits. For example, it can be seen from Figure 5 that the greedy algorithm is not able to find and provide coverage to all 100 users with non-zero probability and, in fact, the maximum number of users covered with non-zero probability can be far less than 100. However, the DQN-based algorithm is able to find most of the users with non-zero probability. In the case of  $K = 4$  clusters, Figure 5 shows that it is able to find all 100 users with more than 0.4 probability!

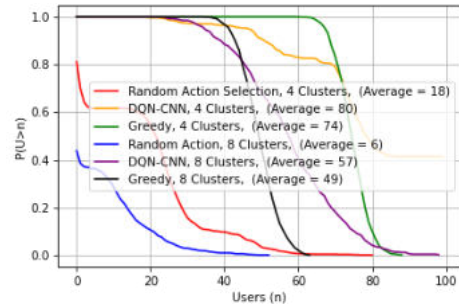


Fig. 5. Complementary Cumulative Distribution Function Showing Probability of Covering More Than  $n$  Users

### B. Non-stationary user distributions

In scenarios where ground users are mobile, like troops on the move or users commuting to work, it makes sense to model

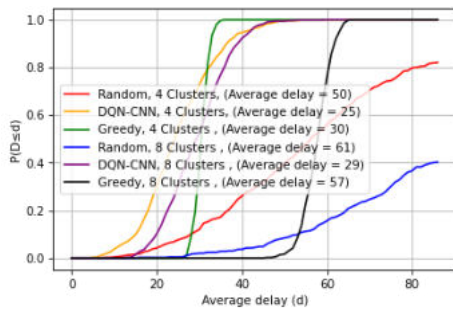


Fig. 6. Cumulative Distribution Function Showing Probability of the Average Delay  $< d$

the users as clusters, but with moving means. The parameters for the GMM in this case is given in Table 3. For simplicity, the directions and magnitudes of movements are assumed to be fixed.

Since the distribution of users at a time instance is correlated to distribution of users at previous time instances, for this scenario we may expect a Recurrent Neural Network (RNN) to be a better candidate for the DQN over CNNs, since RNNs are known for their ability to capture the temporal correlations in the input.

Figure 7 and Figure 8 illustrate the performance of the proposed and random action selection methods in this scenario. It can be inferred from Figure 7 that the proposed DQN method covers significantly more users than the reward-based greedy method. Moreover, the proposed DQN method with RNN as the function estimator is slightly better than the one with CNN, which can also be inferred from cdf plots shown in Figure 7. Furthermore, the average delay experienced by users was significantly lower when using the proposed DQN method in contrast to random action selection method and reward-based greedy method, which can be seen in Figure 8. In fact, the DQN method with RNN resulted in fewer delays in the case of  $K = 4$  clusters.

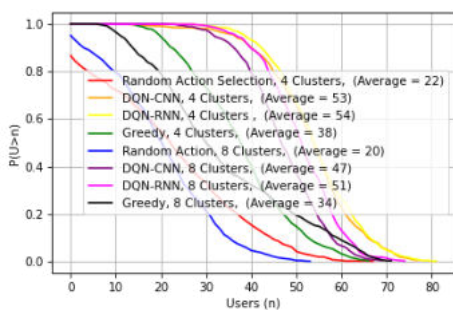


Fig. 7. Complementary Cumulative Distribution Function Showing Probability of Covering More Than  $n$  Users

## V. CONCLUSION

This paper proposed DRL and reward-based greedy methods to maximize the coverage of distinct ground users by a UAV-mounted base station. The performance of the proposed methods were analyzed based on the number of users covered and the delay experienced a by user until it first receives coverage while assuming two scenarios for the distribution of users:

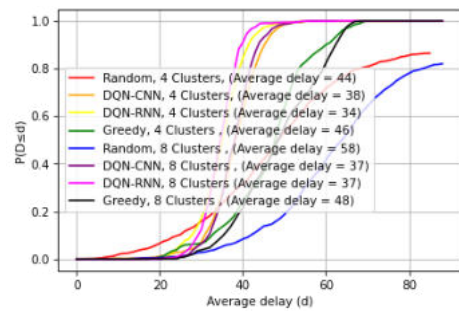


Fig. 8. Cumulative Distribution Function Showing Probability of the Average Delay  $< d$

(1) a GMM with fixed and (2) time-varying means. It was observed that the proposed methods were able to cover a significantly more number of users compared to random actions in both cases. Computationally lightweight greedy algorithm was observed to perform very close to the DRL method when user distributions were stationary. However, the DRL was shown to outperform the greedy algorithm significantly, especially when users were allowed to be mobile. Further work is needed to incorporate complex mobility models for user distribution and also to take into account wireless channel effects.

## REFERENCES

- [1] M. Byk, R. Duvar, and O. Urhan, "Deep Learning Based Vehicle Detection with Images Taken from Unmanned Air Vehicle," in *2020 Innov. Intell. Syst. Appl. Conf. (ASYU)*, 2020, pp. 1–4.
- [2] S. Hayat, E. Yanmaz, and R. Muzaffar, "Survey on Unmanned Aerial Vehicle Networks for Civil Applications: A Communications Viewpoint," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2624–2661, 2016.
- [3] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless Communications with Unmanned Aerial Vehicles: Opportunities and Challenges," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 36–42, 2016.
- [4] R. I. Bor-Yaliniz, A. El-Keyi, and H. Yanikomeroglu, "Efficient 3-D Placement of an Aerial Base Station in Next Generation Cellular Networks," in *2016 IEEE Int. Conf. Commun. (ICC)*, 2016, pp. 1–5.
- [5] H. V. Abeywickrama, Y. He, E. Dutkiewicz, B. A. Jayawickrama, and M. Mueck, "A Reinforcement Learning Approach for Fair User Coverage Using UAV Mounted Base Stations Under Energy Constraints," *IEEE Open J. Veh. Technol.*, vol. 1, pp. 67–81, 2020.
- [6] G. Li, C. Zhuang, Q. Wang, Y. Li, X. Xu, and W. Zhou, "A UAV Real-time Trajectory Optimized Strategy for Moving Users," in *2019 11th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, 2019, pp. 1–6.
- [7] N. Cherif, W. Jaafar, H. Yanikomeroglu, and A. Yongacoglu, "On the Optimal 3D Placement of a UAV Base Station for Maximal Coverage of UAV Users," in *GLOBECOM 2020 - 2020 IEEE Global Commun. Conf.*, 2020, pp. 1–6.
- [8] M. Peer, V. A. Bohara, A. Srivastava, and G. Ghatak, "User Mobility-aware Time Stamp for UAV-BS Placement," in *2021 IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, 2021, pp. 1–6.
- [9] A. V. Savkin and H. Huang, "Deployment of Unmanned Aerial Vehicle Base Stations for Optimal Quality of Coverage," *IEEE Wireless Commun. Lett.*, vol. 8, no. 1, pp. 321–324, 2019.
- [10] V. Saxena, J. Jaldn, and H. Klessig, "Optimal UAV Base Station Trajectories Using Flow-level Models for Reinforcement Learning," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 4, pp. 1101–1112, 2019.
- [11] R. Jain, D. M. Chiu, and H. WR, "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems," *CoRR*, vol. cs.NI/9809099, 01 1998.
- [12] R. S. Sutton and A. Barto, *Reinforcement learning: An Introduction*. MIT Press, 2018.
- [13] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller, "Playing Atari with Deep Reinforcement Learning," *CoRR*, vol. abs/1312.5602, 2013. [Online]. Available: <http://arxiv.org/abs/1312.5602>

# SVR-based Blind Equalization on HF Channels with a Doppler Spread

Soon-Young Kwon  
 Dept. Electronics Engineering  
 Pusan National University  
 Busan, Republic of Korea  
 ysk1680@pusan.ac.kr

Ji-Hyeon Kim  
 Dept. Electronics Engineering  
 Pusan National University  
 Busan, Republic of Korea  
 kjihyeon@pusan.ac.kr

Hyoungh-Nam Kim  
 Dept. Electronics Engineering  
 Pusan National University  
 Busan, Republic of Korea  
 hnkim@pusan.ac.kr

**Abstract**—A transmission signal through a high-frequency (HF) channel is usually reflected by the ionospheric layers and become a multipath signal, resulting in inter-symbol interference (ISI). To remove ISI, a receiver recovers the multipath-faded signal by using channel equalization. Among various channel equalization methods, blind equalization that does not use training sequences draws an interest because it may increase bandwidth efficiency. The HF signal needs to be equalized with a small number of symbols due to a Doppler spread. Therefore, to equalize the HF channel signal, a batch method based on support vector regression (SVR) can be used. In this respect, we applied an SVR-based batch blind equalization to HF channels and then analyzed its performance.

**Keywords**—blind equalization, support vector regression, high frequency channel

## I. INTRODUCTION

In a high frequency (HF) band digital communication channel, the signal reception performance is deteriorated due to inter-symbol interference (ISI), which makes it hard to recover the signal and causes a symbol error. To remove ISI, various channel equalization techniques have been studied recently [1],[2].

Channel equalization method is divided into two types depending on whether a training sequence is used or not. A representative method using a training sequences is the least mean squared (LMS) algorithm, which is easy to implement and widely used due to its low complexity [3]. However, using a training sequence reduces the transmission rate. Therefore, to efficiently transmit a signal, blind equalization methods that do not use a training sequence have been studied [4].

Blind equalization can be divided into online method and batch algorithm. The online blind algorithms are based on stochastic gradient descent (SGD) minimization of a cost function. The most representative method is constant modulus

algorithm (CMA) [5]. On the other hand, the batch algorithms use a block of data and iteratively minimize a cost function based on support vector regression (SVR) or cumulant [6]. Batch algorithm can achieve good equalization performance by using fewer symbols than online one [7].

In the HF channel with a Doppler spread, since the channel changes with time, the received signal should be equalized with a small number of symbols. Therefore, the batch method is better than online ones. In this paper, the equalization performance of the HF channel in the mid-latitude region is analyzed by using the SVR-based batch equalization algorithm according to the channel conditions.

## II. PROBLEM FORMULATION

The overall block diagram of the blind equalization including the signal process is shown in Fig. 1. We consider baseband representation of the digital communication system. A sequence of independent and identically distributed (i.i.d) symbols  $s(k)$  is sent through a channel with coefficients  $h(k)$ . The resulting channel output can be expressed as

$$x(k) = \sum_{n=0}^{L-1} h(n)s(k-n) + v(k), \quad (1)$$

where  $v(k)$  is an additive white Gaussian noise (AWGN) and  $L$  is channel coefficient length.

The objective of blind equalization is to remove the ISI caused by the channel. The equalization output can be expressed as

$$y(k) = \sum_{n=0}^{M-1} w(n)x(k-n) = \mathbf{x}^T \mathbf{w}, \quad (2)$$

where  $\mathbf{w}$  is the vector of filter coefficients and  $M$  is filter coefficient length.

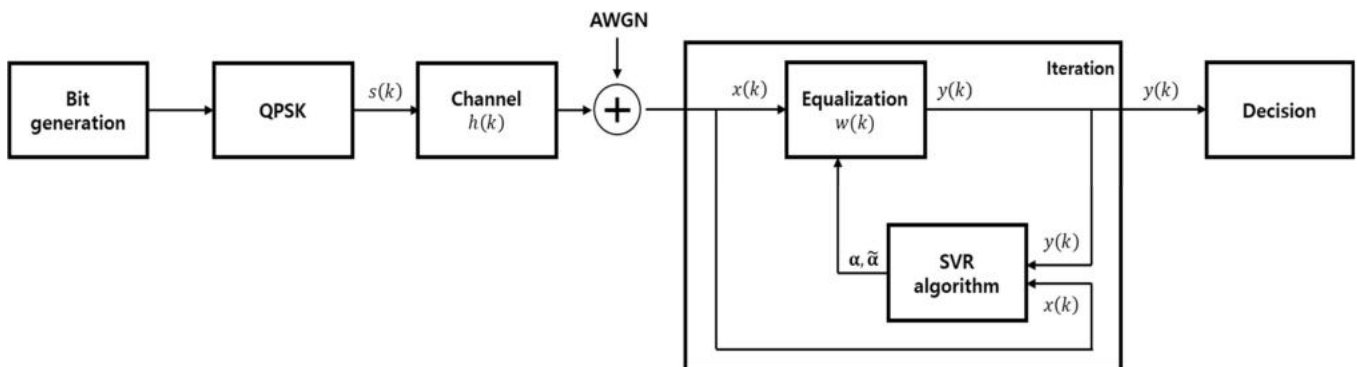


Fig. 1. Block diagram of signal transmission/reception including blind equalization.

### A. SVR-based Batch Blind Equalization

When the data block size is  $N$ , batch equalization minimizes the following SVR-based cost function that exploits the constant modulus (CM) property of the signal [8].

$$J(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N |1 - (\mathbf{w}^T \mathbf{x}_i)^2|_\epsilon \quad (3)$$

where  $C$  is penalty value,  $\mathbf{x}_i = (x_i, \dots, x_{i-M+1})^T$ , and

$$|1 - (\mathbf{w}^T \mathbf{x}_i)^2|_\epsilon = \max\{0, |1 - (\mathbf{w}^T \mathbf{x}_i)^2| - \epsilon\} \quad (4)$$

the so-called Vapnik's  $\epsilon$ -insensitive loss function.

If there are training error, by using a set of positive slack variables  $\xi_i$  and  $\tilde{\xi}_i$ , the optimization equation can be expressed as: To minimize

$$L(\mathbf{w}, \boldsymbol{\xi}, \tilde{\boldsymbol{\xi}}) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N (\xi_i + \tilde{\xi}_i) \quad (5)$$

subject to

$$\begin{aligned} (\mathbf{w}^T \mathbf{x}_i)^2 - 1 &\leq \epsilon + \xi_i \\ 1 - (\mathbf{w}^T \mathbf{x}_i)^2 &\leq \epsilon + \tilde{\xi}_i \\ \xi_i, \tilde{\xi}_i &\geq 0 \end{aligned}$$

for all  $i = 1, 2, \dots, N$ .

To transform the quadratic inequality of the constraint into linear one and make it a quadratic programming (QP) problem, we can use

$$y_i = \mathbf{w}^T \mathbf{x}_i. \quad (6)$$

If the transformed optimization problem is solved through Lagrange dual, the solution is as follows.

$$\mathbf{w}^* = \sum_{i=1}^N (\tilde{\alpha} - \alpha) y_i \mathbf{x}_i \quad (7)$$

where  $\alpha$  and  $\tilde{\alpha}$  are Lagrange multipliers, and can be obtained by minimizing the following quadratic form:

$$\begin{aligned} W(\boldsymbol{\alpha}, \tilde{\boldsymbol{\alpha}}) &= \epsilon \sum_{i=1}^N (\alpha_i + \tilde{\alpha}_i) - \sum_{i=1}^N (\alpha_i - \tilde{\alpha}_i) \\ &+ \frac{1}{2} \sum_{i,j=1}^N (\alpha_i - \tilde{\alpha}_i)(\alpha_j - \tilde{\alpha}_j)(y_i y_j) \langle \mathbf{x}_i, \mathbf{x}_j \rangle \end{aligned} \quad (8)$$

### B. HF channel modeling

The HF channel uses a frequency band of 3 to 30 MHz. Since long-distance communication can be performed with a low power, the HF channel is widely used despite the poor communication channel.

The HF channel signal is reflected by the ionosphere of the atmosphere and can propagate far away by repeated reflections of the ionosphere and the Earth's surface. Because of these advantages, HF channels are widely used in international broadcasting or amateur radio. The ionosphere is divided according to the distribution of ions, and the channel state varies according to various conditions such as weather and latitude. HF channels generally have Doppler spread due

Table 1. High frequency channel parameters in mid latitude regions according to channel conditions.

Parameters	Channel Condition		
	<i>Quiet</i>	<i>Moderate</i>	<i>Disturbed</i>
Differential time delay	0.5 ms	1 ms	2 ms
Doppler spread	0.1 Hz	0.5 Hz	1 Hz

to fine tremor of ionospheric ions, and each multipath signal has a Rayleigh distribution.

Channel parameters according and channel conditions are modeled as representative values in ITU-R (international telecommunication union - radiocommunication) F.1487 [9]. The mid-latitude HF channel parameters according to the channel conditions are summarized in Table 1.

### III. SIMULATION RESULT

In simulation, we use  $C = 1, \epsilon = 0.01$ . The modulation scheme is quadrature phase shift keying (QPSK). The number of equalizer taps is  $M = 17$  and signal-to-noise ratio (SNR) is 30 dB. The central tap of equalizer is initialized to 1, and the remaining equalizer taps are initialized to 0.

The performance evaluation criteria are the residual ISI and probability of convergence. The residual ISI is defined as

$$\text{ISI} = 10 \log_{10} \frac{\sum_k |\theta_k|^2 - \max_k |\theta_k|^2}{\max_k |\theta_k|^2} \quad (9)$$

where  $\theta = \mathbf{h} * \mathbf{w}$ . In simulation, convergence means that the final residual ISI value is less than -5 dB.

Fig. 2 shows the convergence probability according to the data block size. When the channel condition is "Quiet," it converges to 100% when the data block size is 200 or more. When the channel condition is "Moderate," it almost converges to 100% at data block size 200. However, when it is larger than 200, the convergence probability is reduced to 96% because the channel is time-varying. When the channel condition is "Disturbed," it converge to 94% when data block sized is 200. The better the channel condition, the higher the convergence probability.

Fig. 3 shows the residual ISI according to the data block size. When the channel condition is "Quiet," the residual ISI

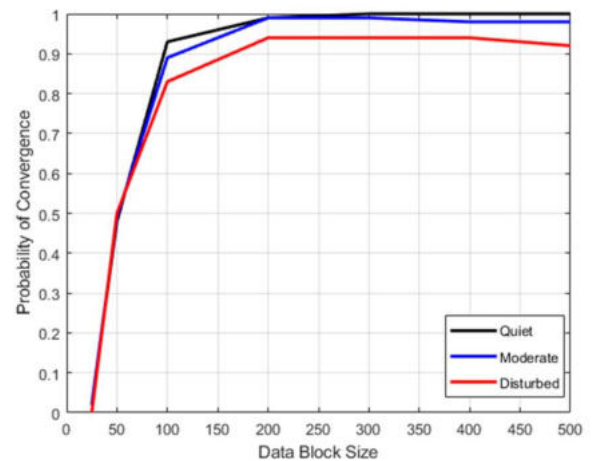


Fig. 2. Probability of convergence according to data block size using SVR-based blind equalization

decreases as the data block size increases, and it converges to about -23 dB. However, When the channel condition is “Moderate” or “Disturbed,” the residual ISI decreases and then increases again as the data block size increases. This is because, as the Doppler spread increases, the channel changes rapidly. When the channel condition is “Moderate” and block data size is 200 or 300, the residual ISI is -17 dB and the performance is the best. When the channel condition is “Disturbed” and block data size is 200, the residual ISI is -13 dB and the performance is the best.

#### IV. CONCLUSION

In this paper, SVR-based blind equalization performance is analyzed for the HF channel in ITU-R F.1487. In the case of a fixed channel, the larger the data block size, the better the equalization performance. However, In the case of a time-varying channel, residual ISI decreases at first but then increases when the block size increases. In simulation, when the channel condition is “Quiet,” as the data block size increases, the equalization performance gets better. However, when the channel condition is “Moderate” or “Disturbed,” we need to find an appropriate batch size by taking a trade-off.

#### ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2021R1F1A1060025).

#### REFERENCES

[1] Y. Wang, L. Yang, F. Wang and L. Bai, "Blind Equalization Using the Support Vector Regression via PDF Error Function," 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2016, pp. 212-216.

[2] X. Liu, Y. L. Guan and Q. Xu, "Support Vector Machine-Based Blind Equalization for High-Order QAM With Short Data Length," in IEEE Signal Processing Letters, vol. 28, pp. 259-263, 2021.

[3] N. Sireesha, K. Chithra and T. Sudhakar, "Adaptive filtering based on least mean square algorithm," 2013 Ocean Electronics (SYMPOL), Kochi, India, 2013, pp. 42-48.

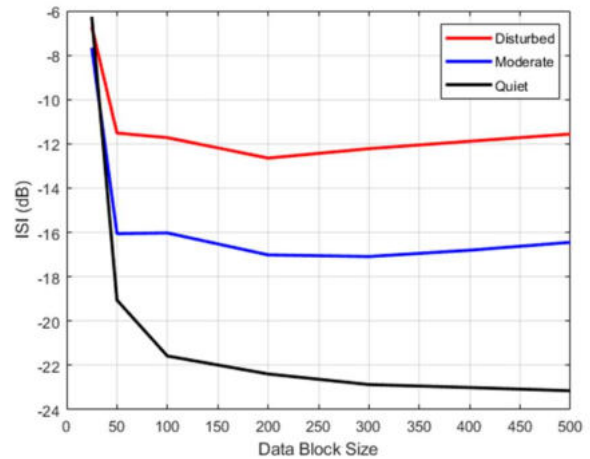


Fig. 3. Residual ISI according to data block size using SVR-based blind equalization

[4] F. Wang, L. Yang, Y. Wang and L. Bai, "The Multimode Blind Equalization Algorithm Based on Gaussian Process Regression," 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2016, pp. 208-211.

[5] D. N. Godard, "Self-recovering equalization and carrier tracking in two dimensional data communication systems," IEEE Trans. Commun., vol. COM-28, no. 11, pp. 1867-1875, Nov. 1980.

[6] O. Shalvi and E. Weinstein, "Super-exponential methods for blind deconvolution," IEEE Trans. Inform. Theory, vol. 39, pp. 504-519.

[7] M. Lázaro, I. Santamaría, J. Vía and D. Erdogmus, "Blind equalization of multilevel signals using support vector machines," 2004 12th European Signal Processing Conference, 2004, pp. 41-44.

[8] I. Santamaria, C. Pantaleon, L. Vielva and J. Ibanez, "Blind equalization of constant modulus signals using support vector machines," in IEEE Transactions on Signal Processing, vol. 52, no. 6, pp. 1773-1782, June 2004.

[9] Recommendation ITU-R F.1487 (05/2000).



# Resolving Camera Position for a Practical Application of Gaze Estimation on Edge Devices

Linh Van Ma, Tin Trung Tran, Moongu Jeon  
School of Electrical Engineering and Computer Science  
Gwangju Institute of Science and Technology  
Gwangju, South Korea  
{linh.mavan, ttrungtin, mgjeon}@gist.ac.kr

**Abstract**—Most Gaze estimation research only works on a setup condition that a camera perfectly captures eyes gaze. They have not literarily specified how to set up a camera correctly for a given position of a person. In this paper, we carry out a study on gaze estimation with a logical camera setup position. We further bring our research in a practical application by using inexpensive edge devices with a realistic scenario. That is, we first set up a shopping environment where we want to grasp customers gazing behaviors. This setup needs an optimal camera position in order to maintain estimation accuracy from existing gaze estimation research. We then apply the state-of-the-art of few-shot learning gaze estimation to reduce training sampling in the inference phase. In the experiment, we perform our implemented research on NVIDIA Jetson TX2 and achieve a reasonable speed, 12 FPS which is faster compared with our reference work, without much degradation of gaze estimation accuracy. The source code is released at <https://github.com/linh-gist/GazeEstimationTX2>.

**Index Terms**—Gaze estimation, Few shot learning, Edge devices, Customers' Attention, Triangulation.

## I. INTRODUCTION

Businesses are now benefiting from computer vision applications. One of the well-known businesses in the USA, Amazon Go [1] has successfully applied deep learning, sensor fusion, and computer vision for checkout, purchase, and proceed for payment automatically without any human interactions. Hence, understanding customers' attention/behavior/preferences during shopping is crucial to increase avenues. For example, a customer who buys milk usually looks for bread. We can place milk and bread next to each other. Gaze is an individual's perception and awareness of human visual attention. We can track the gaze to know which products customers prefer the most and their shopping behaviors.

Many eye commercial trackers perfectly measure the motion of an eye relative to the head. For example, Tobii [2] can estimate and track eye gaze without requiring recalibration. However, it is expensive and hard to upgrade once a business decides to buy those physical devices. More importantly, up-

This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2019R1A2C2087489), and Ministry of Culture, Sports and Tourism(MCST) and Korea CreativeContent Agency(KOCCA) in the Culture Technology(CT) Research & Development (R2020070004) Program 2021.

gradeable, inexpensive, and easy-to-deploy are the key points to leverage business avenues.

In this paper, we introduce research on nearly real-time eye gaze estimation. We use low-priced RGB cameras to capture eyes gaze frames. Those frames are processed by economical edge devices to find out where a person stares at. More specifically, we first reflect a shopping environment where a customer looks at a shelf to buy goods, groceries. We then find an optimal position inside the shelf to set up an RGB camera. This position supports the camera appropriately grasping the customer's eyes gaze. Afterward, we build a deep eye gaze estimation network for edge devices. In this network, images captured from an RBG camera are first fed into a face detection module. Subsequently, facial landmark detection is employed to find key points on facial images. Finally, gaze estimation is accurately estimated from those obtained key points. We use few-shot learning to reduce the number of samples require to fine-tune a deep gaze estimation network. This few-shot learning also makes our work more easily to be deployed in a real-world application because we can fine-tune our deep network within a few samples. In the experiment, we perform our implemented system on NVIDIA Jetson TX2. We achieve a reasonable speed, 12 FPS which is faster compared with our reference work, without much degradation of gaze estimation accuracy. We also prove that the camera should be set up in an optimal position to increase gaze estimation accuracy.

This paper is organized as follows. Section II presents our based research. In Section III, we logically find an optimal camera position that supports the camera to accurately estimate a person's eyes gaze. Subsequently, we demonstrate our method with several experiments in Section IV. Finally, we conclude this paper with a future research direction.

## II. RELATED WORKS

In gaze estimation, we first need to detect faces using face detectors [3]–[6]. MTCNN proposed in [4] is a fast and efficient facial detection model. MTCNN composes of deep cascaded networks, Proposal Network (P-Net), Refinement Network (R-Net), and Output Network (O-Net). These three networks exploit the properties correlation between face alignment (in R-Net) and face detection (in P-Net) to increase the performance of face detection (in O-Net). Furthermore, the authors propose a new online hard sample mining strategy

leading to an improvement during the training process with fewer manual factors.

Facial landmarks [7], [8] to find key points (68 landmarks) on detected faces, is the next crucial step for gaze estimation. Facial landmark detectors essentially try to label and localize the seven facial regions as follows: (1) Right eyebrow, (2) Left eyebrow, (3) Right eye, (4) Left eye, (5) Nose, (6) Mouth, and (7) Jaw. More specifically, the authors [7] propose a general framework based on gradient boosting for learning an ensemble of regression trees that optimizes the sum of square error loss and naturally handles missing or partially labeled data. In short, an ensemble regression trees is trained to estimate the face’s landmark positions directly from a sparse subset of pixel intensities. Notably, their result can achieve face alignment in milliseconds for a single image. This result brings us a chance to implement facial landmark detection on edge devices where we do not have many computational resources. Fortunately, this module was well implemented in Dlib [9]. In contrast, in [8] the authors propose a HRNetV2, a modification on HRNet [10], to work high-resolution representation learning. This learning leads to stronger representations and higher landmark localization accuracy. Hence, we use this HRNetV2 to detect facial landmarks while fine-tuning our gaze estimation network. It has high accuracy but not fast compared to [7].

Gaze estimation [11], [12] is a process to predict where a person is gazing at given an image with the person’s full face. Similar to our approach [13] argued that each person has a distinct gaze, a person-specific gaze. It leads to limit the accuracy of person-independent gaze estimation networks. Hence, they propose personalizing gaze networks. This network encodes face appearance, gaze direction, and head rotation into latent space by using a disentangling encoder-decoder architecture. Their method allows the network to learn person-specific gaze within a few samples (less than nine samples).

Thanks to a real-time detecting eye blink algorithm proposed in [14], we can determine whether eyes are widely opened or slightly closed. They first use landmark positions to calculate the eye aspect ratio (EAR). A Support Vector Machine [15] classifier is subsequently employed to determine whether eyes are blinking or non-blinking pattern based on EAR value. In our approach, this EAR allows us to check whether our camera position set up on the shelf is optimal or not because eyes are slightly narrower when a person’s eyes look downward. Given a camera position, we can check this EAR value to determine the position can avoid the above effect or not.

In [16], [17], the authors argue that the limitation of accuracy from facial landmark detection comes from the training process, which is lack of quality and quantity of annotated face databases. The databases mostly were manually annotated by a trained expert and the fatigue factor is hard to avoid which lead to the error during the works. Hence, Sagonas et al. [16], [17] propose a unified annotation pipeline with a semi-automatic annotation system. They use Active Orientation Models (AOMs) generative models [18] to train their network

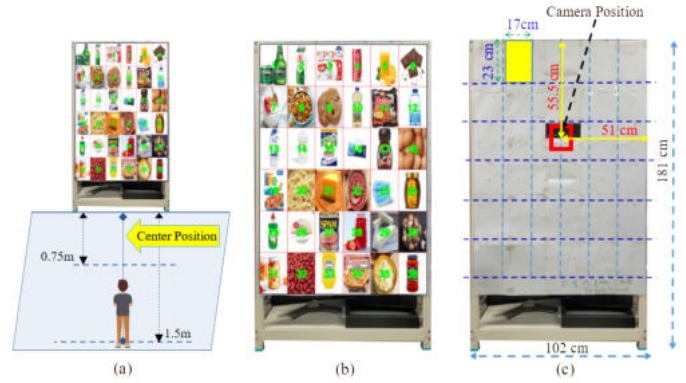


Fig. 1. (a) We ideally model a shelf in a store, (b) each item is mapped to a labeled rectangle number from the top-left to the bottom-right, (c) a simplified version of a shelf in a store.

with an image from mixed expression and viewing angle. The resulted train model can generate accurate annotation in different conditions and can be generalized to unseen images. Along with the problem of the database, the authors [19] further introduce a data normalization method to combine the images and gaze direction to a normalized space, which can cancel out the varieties of head and eye pose positions.

### III. SYSTEM OVERVIEW

We ideally model a physical store in our research environment with a shelf as shown in Fig. 1. It has goods and groceries placed separately in different positions on the shelf. In Fig. 1 (a), a user stands closely 0.75 meters and far at most 1.5 meters from the shelf. In Fig. 1 (b), we divide the shelf into 36 (6x6) rectangles labeled from 1 at the top-left to 36 at the bottom right. Each rectangle has a size of 17 centimeters in width and 23 centimeters in height. Each item is mapped/encoded to one or two rectangles. To simplify, in Fig. 1 (c), we remove groceries and put a large paper with printed 36 rectangles onto the shelf. The camera is located somewhere inside the shelf behind the large white paper. If a unit measurement is not specified, we use centimeters throughout our paper.

Fig. 2 illustrates a side view projection of Fig. 1 (a) (projection from left to right). We name  $B$  is at the top,  $C$  is at the bottom (of the large white paper, not the floor),  $D$  is at the camera location on the shelf.  $A$  locates at the person’s eyes (or middle of them). From the eyes to  $B$ ,  $D$ , and  $C$ , we have two angles  $\alpha_1 = \widehat{BAD}$ ,  $\alpha_2 = \widehat{DAC}$ . The camera should be in the optimal position so that when a person’s head rotates vertically (up and down, or pitch in Euler angle),  $AD$  should equally divide the angle  $\widehat{BAC}$  created when the person looks at the top and bottom of the shelf as shown in Fig. 2. In other words, the camera should make this equality  $\alpha_1 = \alpha_2$  happens. A person looks at the shelf while standing in the range 750 to 1500 centimeters far from the shelf.

Fig. 3 shows an example of nonoptimal camera position (24.5 cm from the top of the shelf) where  $\alpha_1 \neq \alpha_2$ . A person who looks downward with his open eyes. However, the camera determines his eyes are mostly closed. We (humans) tend to

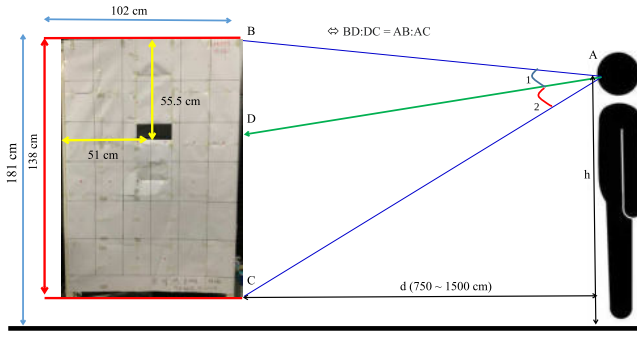


Fig. 2. A shelf with 181 (cm) height and width of 102 (cm). A person stands away from the shelf with a distance ranging from 750 to 1500 (cm). The camera places optimally at the center width and 55.5 (cm) from the top. (A) represents the person’s eyes (or middle of them), (B) top of the shelf, (C) bottom of the large white paper, not floor, (D) is the optimal camera position on the self.



Fig. 3. A person looks downward to the bottom of the shelf. His eye is widely opened, but the camera position is set up inappropriately (about 24.5 cm from the top of the shelf) resulting in the camera looks at him with mostly closed eyes. We experimentally detect facial landmarks [16], [17], [20] to calculate Eye Aspect Ratio (EAR) proposed in [14]. This EAR characterizes how eyes are largely or small opened. In [14], the authors determine that eyes are widely opened with EAR above 0.2. If we inappropriately set up the camera, we can obtain EAR 0.0667 with open eyes as shown on the left side.

open our eyes wider while looking upward and oppositely narrower while looking downward. This unwanted effect leads to our eye gaze algorithm (deep learning algorithm) misunderstands that his eyes are closed. The reason is that we train our deep learning model with a dataset with eyes straightly look at a camera. It has no idea to determine whether eyes look downward and are widely opened. We experimentally detect facial landmarks [16], [17], [20] to calculate Eye Aspect Ratio (EAR) proposed in [14]. This EAR characterizes how eyes are largely or small opened. In [14], the authors determine that eyes are widely opened with EAR above 0.2. If we inappropriately set up the camera, we can obtain EAR 0.0667 with open eyes as shown on the left side.

In Fig. 2,  $BC = 138$  cm is the height of the white large paper,  $750 \leq d \leq 1500$  cm is the distance between the shelf and a person,  $h$  is the height of a person minus 4.8 cm (approximation distance from eyes to the top of head),  $b = 181$  cm is the height of the shelf. We apply one property of bisection, if  $\alpha_1 = \alpha_2$  then  $BD : DC = AB : AC$ . In this equality, we have already known  $BC$ , while  $AB$  and  $AC$  can be calculated by using Pythagorean theorem. Given  $h$  is the

TABLE I

CAMERA IS PLACED 55.5 CM FROM THE TOP OF THE SHELF. A PERSON WITH 1800 CM HEIGHT IS RECOMMENDED TO STAY AWAY FROM THE SHELF AT 1212.1 CM.

Item	Person Height	Distance from the shelf
1	1500	851.453
2	1550	928.174
3	1600	996.515
4	1650	1058.101
5	1700	1114.053
6	1750	1165.182
7	1800	1212.100

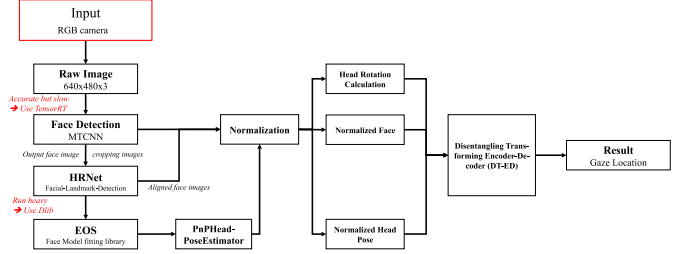


Fig. 4. Workflow of our gaze estimation system. It is largely inspired by FAZE proposed in [13]. We use MTCNN [4] for face detection and HRNetv2 [8] for facial landmark detection in fine-tuning. In inference, these two tasks are computationally expensive for edge devices. Hence, we replace the MTCNN module with a similar one but implemented in TensorRT [22], TensorRT MTCNN. Though HRNetv2 has high accuracy, we replace it with a landmark detection module in Dlib [7], [9].

height of a person, we look to find a position  $d$ . The height follows Gaussian distribution with the global mean height for 159 cm for women and men 171 cm [21]. From  $\frac{DC}{BD} = \frac{AC}{AB}$ , we have  $\frac{AC}{AB} + 1 = \frac{BC}{DB}$ . Using this fact, we put (1), (2) altogether and have equation (3). It has three variables  $d, h$  and  $DB$ , ( $b - BC = 43$ ). We randomly generate  $h$  with its Gaussian distribution (mean is 165, standard deviation is 6), and  $d$  is uniformly distributed in the range  $[750, 1500]$  and obtain  $DB$  is optimal at 55.5 cm from the top of the shelf.

$$AB = \sqrt{d^2 + (b - h)^2}. \quad (1)$$

$$AC = \sqrt{d^2 + (h - (b - 138))^2}. \quad (2)$$

$$DB = \frac{138\sqrt{d^2 + (h - 43)^2}}{\sqrt{d^2 + (b - h)^2} + \sqrt{d^2 + (h - 43)^2}}. \quad (3)$$

If the camera optimally places at 55.5 cm from the top of the shelf, Table I depicts that a person with a specific height must stay far from the shelf to increase gaze estimation accuracy. For example, a person with 170 cm height (we ignore a space from eyes to the top of the human head for simplifying explanation) is recommended to stay away from the shelf at 1114.053 cm.

We use few shot learning techniques to reduce time to fine-tune our deep gaze estimation network. These techniques make our gaze estimation module applicable in real-world

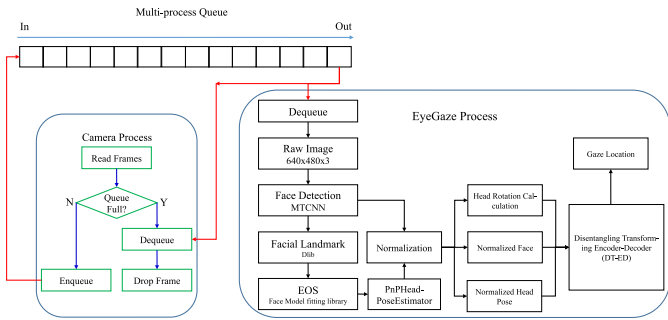


Fig. 5. Working with multiprocessing. Our algorithm cannot process to find gaze location in real-time (30 FPS). We use a queue with two processes. One process named “Camera Process” puts images captured from a camera into the queue. Another process named “EyeGaze Process” obtains the latest recent images in that queue and starts processing to estimate gaze location

applications where we do not have much data to train. It also supports us utilize existing eye gaze research in our configured environment. We intensively inherit FAZE proposed in [13] to build our gaze estimation network. Fig. 4 shows the overall workflow of FAZE. They use MTCNN [4] for face detection and HRNet [8] for facial landmark detection. These two tasks are computationally expensive. Hence, we replace the MTCNN module with a similar one but implemented in TensorRT [22], TensorRT MTCNN Face Detector. Though HRNetv2 has high accuracy, we replace it with a landmark detection module in Dlib [7], [9]. Facial landmark detection in Dlib is fast, which is fairly real-time without degrading accuracy compared with HRNetv2. To maintain high accuracy of inference, we still use the original HRNetv2 and MTCNN to finetune FAZE model. We only use these two replacement modules in inference on edge devices. We use data normalization proposed in [19] to cancel out the significant variability in head pose such as head rotation with respect to the camera.

Despite our great effort, we cannot achieve real-time processing (30 FPS) in edge devices such as Jetson TX2. It can only run 10 to 15 FPS. Hence, we use multiprocessing and queue to ignore frames from the camera. Suppose our gaze estimation algorithm works on frame  $f_i$ , it moves to work on the next latest frame capture from the camera  $f_j$  ( $i > 0, j > i + 1, i, j \in \mathbb{N}$ ). All frames  $\{f_{i+1}, \dots, f_{j-1}\}$  captured during processing frame  $f_i$  are discarded. This procedure ensures that we always capture changes in a person’s head such as rotation. Fig. 5 illustrates our idea of using a queue. We use a queue with two processes. One process named “Camera Process” puts images captured from a camera into the queue. Another process named “EyeGaze Process” obtains the latest recent images in that queue and starts processing to estimate gaze location.

#### IV. EXPERIMENTS

We put a large white paper into the shelf to simulation our real-world shopping environment as shown in Fig. 1 and Fig. 2. It has a width of 102 cm and a height of 138 cm. We divide it into 36 equal rectangles size 17 cm in width and 23 cm in

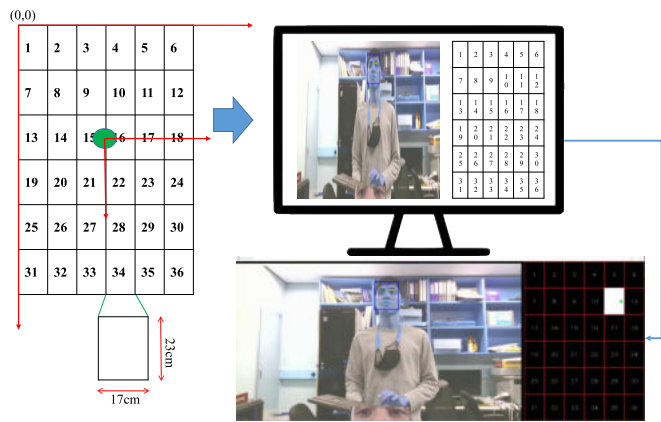


Fig. 6. Mapping from the physical world to screen monitor for visualization. We use the coordinates axis (0,0) at the top-left corner of the shelf to label location of each point center at each rectangle. However, our algorithm only works with the coordinates axis (0,0) at the camera pinhole location (large green dot).

height. The camera is always at the center of width and down from the top at 55.5 cm (calculated optimal position in Section III) as shown in Fig. 6. We use the coordinates axis (0,0) at the top-left corner of the shelf to label the location of each point center at each rectangle. However, our algorithm works with the coordinates axis (0,0) at the camera pinhole location. We obtain our custom dataset to fine-tune our gaze estimation module by asking a user constantly looking at each point centered at each rectangle. In this experiment, we only have one participant. The paper (size 102x138 cm) is mapped to a screen monitor and still maintains the width height ratio. This mapping supports us to visualize our process on the monitor while collecting dataset.

More specifically, we make a Python script to output one video file and one Pickle file [23] contains ground truth (with the origin of the coordinates axis (0,0) at the top-left corner of the shelf). A person needs to look at 36 points (centered at each rectangle) to collect a calibration dataset. We select ten frames at each point. These ten frames contain images that specify a person constantly looks at one point (out of 36). Hence, we have a total of 360 frames in one calibration video. However, to fine-tune our deep learning model, we only use three random images for training and one image for validation from those ten collected images of a given point. In Fig. 6, there have thirty-six labeled squares ranging from 1 to 36. We experimentally find that fine-tuning and validation locations should be equally distributed in the shelf in order to avoid bias in any region (left, right, top, down) respect to camera location. Hence, We use squares with number  $\{8, 11, 26, 29\}$  for validation. Those points location are equally located with respected to camera location. The rest 32 points are used in different sets of fine-tuning as shown in Table II.

Our implementation is intensively based on a few shot learning [13]. We ignore the training phrase since their model was fairly trained with the following number of calibration points  $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,$

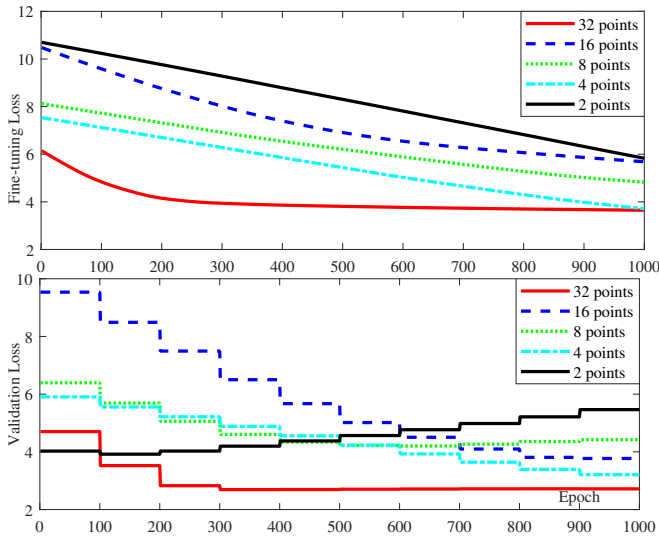


Fig. 7. We fine-tune FAZE network using five different sets of points (located at each rectangle) shown in Table II. If we fine-tune with two points, validation loss increases compared to other sets. Hence, each person needs at least four samples to fine-tune FAZE network. As we increase the number of samples (points) to fine-tune the gaze estimation network, validation loss reduces accordingly.

TABLE II

DIFFERENCE TRAINING (FINE-TUNING) SETS OF POINTS. EACH POINT CENTERED AT EACH RECTANGLE.

Item	Tuning points	Point indexes
1	2	6, 31
2	4	3, 13, 18, 33
3	8	1, 3, 6, 13, 18, 31, 33, 36
4	16	1, 3, 4, 6, 13, 15, 16, 18, 19, 21, 22, 24, 31, 33, 34, 36
5	32	1, 2, 3, 4, 5, 6, 7, 9, 10, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 27, 28, 30, 31, 32, 33, 34, 35, 36

17, 18, 32, 64, 128, 256}. We do not re-train their network and only use their pre-trained weight on inference. Logically, gaze estimation accuracy can be improved if we increase the number of calibration points. However, we do not have enough time and labor resources to label our custom dataset. Hence, we use different sets of points to find the reasonable number of calibration points shown in Table II. As mentioned earlier, we fix four points to validate the fine-tuning process.

Fig. 7 illustrates fine-tuning loss (training and validation) with respect to an epoch. If we fine-tune with two points, validation loss increases compared to other sets. Hence, each person needs at least four samples to fine-tune FAZE network. As we increase the number of samples (points), to fine-tune the gaze estimation network, validation loss reduces accordingly.

Fig. 8 visually shows the output while fine-tuning our gaze network. RGB images are feed into our system. Our system outputs (1) face bounding box [4], (2) facial landmark [8], (3) head pose, (4) eyes patch after normalization [19]. We also output (5) the location of where the user gazes at. For

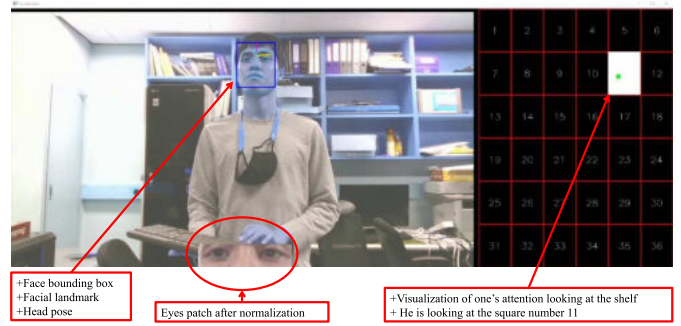


Fig. 8. We display the appearance of a person with (1) face bounding box [4], (2) facial landmark [8], (3) head pose, (4) eyes patch after normalization [19]. We also output (5) the location of where the user gazes at. For example, a user attentively looks at the square number eleven.

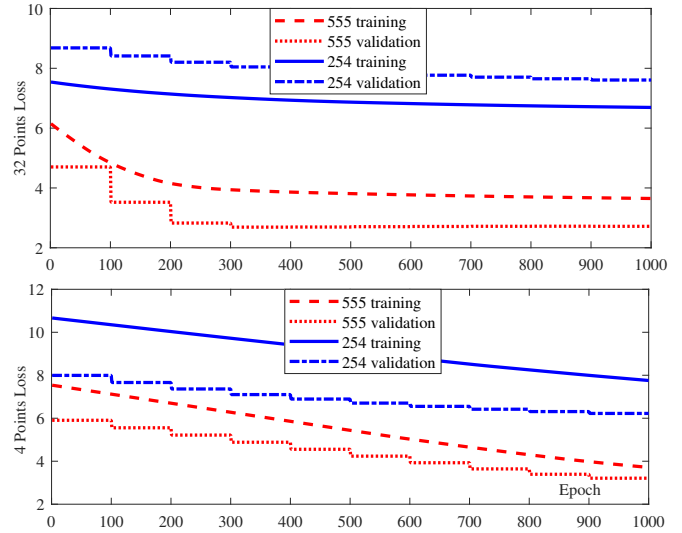


Fig. 9. We set up an RGB camera to capture frames from two different locations centered along the width. (1) optimal position 55.5 cm, and non-optimal position 24.5 cm from the top of the shelf. Fine-tuning results with two sets of 32 and 4 points from Table II. It shows that the camera at position 25.4 cm cannot reduce training and validation loss even though we increase the number of samples from 4 to 32.

example, a user attentively looks at the square number eleven.

If we set up a camera to capture frames inappropriately, we cannot archive high-accuracy gaze estimation. As shown in Fig. 9, we set up an RGB camera to capture frames from two different locations. We compare training and validation loss of two camera positions which are 55.5 and 25.4 cm from the top of the shelf. It shows that the camera at position 25.4 cm cannot reduce training and validation loss even though we increase the number of samples from 4 to 32. Especially, fine-tuning with 32 points even increases validation loss and maintains at 8. Oppositely, the camera at position 55.5 cm logically reduces training and validation loss to 2. Despite our hard efforts in finding a camera position, we still face the eye-looking effects shown in Fig. 10. Though, we mathematically find the optimal camera position 55.5 cm from the top. The eye-looking effect still prevents us to obtain equally opened



Fig. 10. We (humans) tend to open our eyes wider while looking upward and oppositely narrower while looking downward. We have found the optimal position of the camera (55.5 cm from the top) but this effect prevents us to equally capture opened eyes when looking upward and downward. However, we have overcome the effect and estimate eyes gaze more accurately compared to what we can obtain in Fig. 3.



Fig. 11. Our experiment with the optimal camera position at 55.5 cm from the top of the shelf. A user gazes at rectangle number 19 (leftmost side). Our gaze estimation module correctly identifies the user's gazing (shown in a white rectangle). It skips one to five frames in order to process real-time images captured from the camera. The processing speed is about 10 to 15 FPS.

eyes when looking upward and downward but it is much better compared to what we can obtain in Fig. 3.

Fig. 11 shows the final output of our experiment with the optimal camera position at 55.5 cm from the top of the shelf. A user gazes at rectangle number 19 (left side). Our gaze estimation module correctly identifies the user's gazing (shown in a white rectangle, right side). It skips one to five frames to process the latest images captured from the camera. A few captured images are discarded to ensure that our gaze estimation algorithm is not stuck or keeps processing past frames. Every change of personal appearance such as head rotation, eyes blink is constantly processed. The processing speed is about 10 to 15 FPS.

## V. CONCLUSION

In this paper, we bring research in gaze estimation to a real-world application. Gaze estimation has been done intensively in the literature, but it has many limitations and far beyond real-world constraints. We bridge the gap to find the optimal position for the camera. This effort maintains high accuracy grasped from our reference research. Then, we implement to run state-of-the-art gaze estimation on edge devices. Our experiment proves that setting the camera at an appropriate position supports us to obtain eyes gaze correctly. It results in maintaining high accuracy compared to the results that have been done in the research environment. However, we only consider a scenario in which only one person statically

standing in front of the camera and has no movement such as walking. Furthermore, the 3D shape of the face around the eyes strongly affects the gaze estimation accuracy so that the optimal camera position may be different among different people. In the future, we intend to experimentally and quantitatively confirm our optimal camera position with data from multiple participants. Additionally, we plan multiple cameras to avoid constraints from our current research environment making it more realistic. We can also extend it to a scenario where many people move around while looking at the camera.

## REFERENCES

- [1] K. Wankhede, B. Wukkadada, and V. Nadar, "Just walk-out technology and its challenges: A case of amazon go," in *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*. IEEE, 2018, pp. 254–257.
- [2] A. Gibaldi, M. Vanegas, P. J. Bex, and G. Maiello, "Evaluation of the tobii eyeX eye tracking controller and matlab toolkit for research," *Behavior research methods*, vol. 49, no. 3, pp. 923–946, 2017.
- [3] P. Viola and M. J. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.
- [4] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.
- [5] S. S. Farfadi, M. J. Saberian, and L.-J. Li, "Multi-view face detection using deep convolutional neural networks," in *Proceedings of the 5th ACM International Conference on Multimedia Retrieval*, 2015, pp. 643–650.
- [6] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *CVPR*, 2015, pp. 815–823.
- [7] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *CVPR*, 2014, pp. 1867–1874.
- [8] K. Sun, Y. Zhao, B. Jiang, T. Cheng, B. Xiao, D. Liu, Y. Mu, X. Wang, W. Liu, and J. Wang, "High-resolution representations for labeling pixels and regions," *arXiv preprint arXiv:1904.04514*, 2019.
- [9] D. E. King, "Dlib-ml: A machine learning toolkit," *The Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.
- [10] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *CVPR*, 2019, pp. 5693–5703.
- [11] T. Fischer, H. J. Chang, and Y. Demiris, "Rt-gene: Real-time eye gaze estimation in natural environments," in *ECCV*, 2018, pp. 334–352.
- [12] S. Park, A. Spurr, and O. Hilliges, "Deep pictorial gaze estimation," in *ECCV*, 2018, pp. 721–738.
- [13] S. Park, S. D. Mello, P. Molchanov, U. Iqbal, O. Hilliges, and J. Kautz, "Few-shot adaptive gaze estimation," in *ICCV*, 2019, pp. 9368–9377.
- [14] T. Soukupova and J. Cech, "Real-time eye blink detection using facial landmarks," *21st Computer Vision Winter Workshop*, pp. 1–8, 2016.
- [15] W. S. Noble, "What is a support vector machine?" *Nature biotechnology*, vol. 24, no. 12, pp. 1565–1567, 2006.
- [16] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: The first facial landmark localization challenge," in *ICCV Workshops*, 2013, pp. 397–403.
- [17] —, "A semi-automatic methodology for facial landmark annotation," in *CVPR workshops*, 2013, pp. 896–903.
- [18] G. Tzimiropoulos, J. Alabort-i Medina, S. P. Zafeiriou, and M. Pantic, "Active orientation models for face alignment in-the-wild," *IEEE transactions on information forensics and security*, vol. 9, no. 12, pp. 2024–2034, 2014.
- [19] X. Zhang, Y. Sugano, and A. Bulling, "Revisiting data normalization for appearance-based gaze estimation," in *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, 2018, pp. 1–9.
- [20] C. Antonakos, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: Database and results," *Image and Vision Computing*, vol. 47, pp. 3–18, 2016.
- [21] M. Roser, C. Appel, and H. Ritchie, "Human height," *Our world in data*, 2013.
- [22] H. Vanholder, "Efficient inference with tensorsrt," 2016.
- [23] M. Pilgrim and S. Willison, *Dive Into Python 3*. Springer, 2009, vol. 2.

# Throughput Prediction by Radio Environment Correlation Recognition Using Crowd Sensing and Federated Learning

Satoshi Nakaniida, Takeo Fujii  
Advanced Wireless Communication Research Center (AWCC)  
The University of Electro-Communications (UEC)  
1-5-1, Chofugaoka, Chofu, Tokyo, 182-8585 Japan  
E-mail: {nakaniida,fujii}@awcc.uec.ac.jp

**Abstract**— We propose an approach using federated learning for predicting Wi-Fi and LTE transmission control protocol (TCP) throughput to reduce the delay between the output of prediction results and the problem of security risks by sharing the datasets, which is a problem with conventional machine learning methods. The proposed method collects measurement datasets such as received signal strength index from distributed edge devices. Then, a shared learning model is created using the measured dataset on the server. The created model is retrieved by edge devices at any time and used to predict TCP throughput. To evaluate the effectiveness of the proposed method, we perform the emulation evaluation using measured datasets obtained in a real environment. The emulation results reveal that the proposed method can skillfully predict the TCP throughput in the realistic communications. Additionally, the prediction accuracy of the TCP throughput can be improved by creating a learning model for each network area.

**Keywords**— Federated Learning, Deep-Neural-Network, TCP Throughput, Crowd Sensing, Android

## I. INTRODUCTION

In recent years, methods that directly implement machine learning models on edge devices such as smartphones to perform everything from training to prediction have been attracting attention [1]-[5]. In conventional methods, the learning models are often placed in cloud services due to the advantages of scalable storage of large amounts of training data and the availability of high-performance processing functions [6][7]. However, if we try to complete the entire process from training to prediction in the cloud, it will take a long time for the edge devices to get the prediction results due to the delay caused by communication time. In addition, the exchange of data used for learning itself requires a large amount of communication, which poses a cost issue [8]. On the other hand, if learning and prediction are completed only by the edge device, prediction results can be obtained quickly, but the data used for learning is limited to the device itself. Federated Learning [1] is a method that leverages both the advantages of the cloud, where multiple devices can share their learning status, and the advantages of prediction on edge devices. In this paper, we focus on an approach to predict transmission control protocol (TCP) throughput using federated learning. In federated learning, learning and prediction is done by edge devices, while the learning status of the learning model is collected and shared on a server (we do not mention federated learning where edge devices share the learning model directly with each other). A device with a small amount of observation data can obtain highly accurate prediction results by downloading the weight parameters of the learning model learned by other devices.

A feature of this system is that it does not share the training data, but only the weight parameters of the training model. This solves the problems of security, communication volume, and implementation cost that conventional machine learning methods have faced [9].

## II. PROPOSED METHOD

### A. Federated Learning

In the federated learning used in this study, the weight parameters of each device are shared with other devices by uploading the data to one dedicated servers for sharing. All the machine learning models used in this study are Deep-Neural-Network (DNN), and the network used for prediction is Wi-Fi. TABLE I shows the list of features used for training. The movement speed shown here is calculated from the temporal variation of location information, and is obtained using a library provided in advance for Android.

### B. System Model

To build training models, we use Keras (in TensorFlow), a high-level neural network development library provided by Google [10]. We use TensorFlow because it can be easily converted into learning models optimized for mobile environments such as smartphones using existing APIs [11]. The hyper-parameters of the learning model are set to the values shown in TABLE II. To determine the hyper-parameters, we used a dataset obtained with the same equipment and methods as those used to create the dataset in "III. THROUGHPUT PREDICTION" described below. The dataset used for parameter tuning is not used for any other purpose than tuning. Since it is not practical to adjust the learning models of all the devices one by one, the values in TABLE II are fixed. When recording the learning status of the learning model to be used in this study, the file size for recording one machine learning model is always about 10kByte, regardless of the measurement time. This is smaller than the data size of about 2000kByte for 24 hours of measurement of the values in TABLE I at one per second. Keeping the file size small is an important consideration because it leads to a reduction in operational costs and communication time.

### C. Implementation of the Federated Learning System

We implemented a federated learning system on a computer that operates using the measured data. In the following, the device that manages the learning model (shared model) shared by multiple terminals is called the "global node," and the terminal that collects data and performs learning and prediction, corresponding to an edge device, is called the "local node."

TABLE I. INPUT/OUTPUT PARAMETERS TO THE MACHINE LEARNING MODEL

Feature	Description	Type
Network ID	Wi-Fi router identification number	Input
RSSI	Received Signal Strength Indicator	
Latitude	Location info. measured by smartphone	
Longitude	Location info. measured by smartphone	
Week	Parameters assigned to days of the week from 0 to 6.	
Hour	Hour is expressed as a number from 0 to 23.	
Speed	Smartphone movement speed	Output
Throughput	Downlink TCP Throughput	

TABLE II. CONFIGURATION PARAMETERS OF THE MACHINE LEARNING MODEL

Variable	Parameter
Epoch number	500 (Introduce Early-Stopping with patience=10)
Neurons count of Hidden layer	32
Number of hidden layers	1
Initial learning rate	0.001
Normalization rate	0.01 (L2 Normalization)
Dropout rate	(Not used)
Data ratio for cross-validation	20%
Loss function	Mean Square Error (MSE)
Activation function	ReLU

Fig.1 shows the system model of the implemented federated learning. In this study, we do not consider the communication time and communication errors between nodes, and local nodes do not communicate directly with each other. In addition, we did not use any public libraries to implement the federated learning itself, but implemented it ourselves.

#### D. Updating of Shared Model

The basic flow of federated learning is to iterate the process of updating the shared model until the learning converges. First, the global node sends a request to the local node to send the learning results to the global node. Next, the local node that responds to the request learns with its own data and shares only the weight parameters that are the results of learning. Finally, the shared weight parameters are processed based on the Federated Averaging algorithm to update the weight parameters of the shared model. This algorithm is expressed in Equation (1) [1].

$$\mathbf{W} = \sum_{k=1}^K \frac{n_k}{n} \mathbf{W}_k \quad (1)$$

Where  $\mathbf{W}$  is the weight matrix of the shared model,  $K$  is the number of local nodes,  $n_k$  is the number of records of training data used by local node  $k$ ,  $n$  is the number of records of training data used by all local nodes, and  $\mathbf{W}_k$  is the weight matrix of the training model at local node  $k$ .

TABLE III. THE EXPERIMENT SPECIFICATIONS

Number	Type	Overview
(1)	Device	Android11, Sharp Aquos R3
(2)	Device	Android6, Huawei MediaPad
(3)	Device	Android11, GalaxyS10
(4)	Router	Buffalo, WSR-1166DHP6 (Router A)
		NEC, AtermWR8370N (Router B)
(5)	PC	Lenovo, ThinkCenterM715qTiny Router A is connected with a wired LAN cable.
		Lenovo, ThinkCenterM715qTiny Router B is connected with a wired LAN cable.

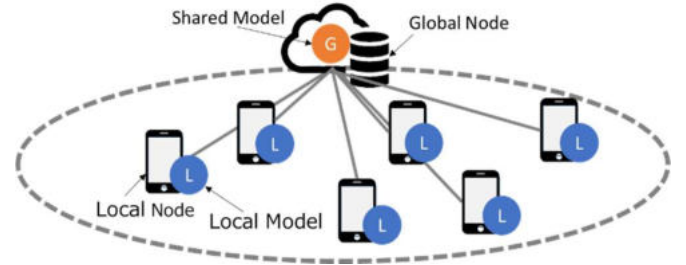


Fig. 1. System model (The global node is assumed to be operated by a serverless cloud service, and the local node is considered to be a smartphone with the Android OS. In this verification, the nodes are pseudo-split on the computer.)

Each local node reacquires the shared model when it is updated. At this stage, it is not yet possible to derive an appropriate value for the frequency with which the global node updates the shared model. In the literature [1], it is said that it is desirable to update the model at noon when many terminals are gathered and at night when terminals are not operated.

### III. THROUGHPUT PREDICTION

#### A. Dataset Preparation

Using the actual measured data, we evaluate the prediction accuracy of the throughput.

To obtain measured datasets, we used our own Android application and the open-source-software "iPerf3". iPerf3 is a popular software for measuring the maximum throughput. The reason why we chose to use a home-grown application is that there is no application that can measure the values that we put into the input features of the training model. However, by creating a home-made application, we can be flexible when we want to change the features we want to measure. In addition, application development will be necessary when we try to train and predict machine learning models on Android in the future, so we decided to create our own application. The data measured by the two software are merged in a time series and treated as a single dataset.

Table III shows the experiment specifications. To create the free Wi-Fi environment in the city, the two routers were placed on the second floor of a building. The building exists in a suburban area of Tokyo. We obtained measured dataset while walking around the communication area. In the measurement, the smartphones were held in our hands.



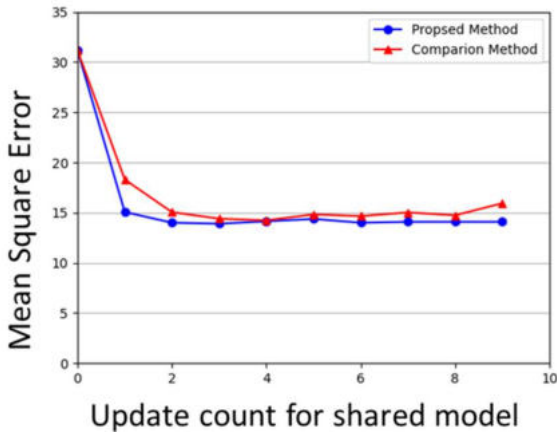


Fig. 2. Predicted mean square error of the proposed method using federated learning and comparison method (comparison method does not use federated learning)

The results presented here have been validated on three datasets. All datasets were created with three smartphones, and one dataset contains only the data measured by one smartphone.

The three smartphones started and ended their measurements almost simultaneously. The datasets used were also pre-processed. Among the features used for training, "week" and "time" were calculated from time stamps. The instantaneous values of throughput were then averaged every 5[s] in order to build an accurate learning model. The data was shuffled in uniform distribution so that the time series would be randomized. This preprocessing is used in many machine learning methods to construct an accurate learning model.

### B. Details

In the performance evaluation, the two local nodes and one global node are used. In the evaluation, the two datasets are used as the training data in the two local nodes. Another dataset is utilized as the test data. The validation results are recorded with different combinations of datasets, and the average prediction error is calculated. The assumption is that the combination pattern will not be changed until the shared model converges. To speed up the training of the shared model, only one dataset is assigned to each local node, and the same data is always used for training.

This paper uses DNN as a comparison method. This method inputs all the data into one DNN training model for one update. The structure and hyper-parameters of the learning model of the comparison method are the same as those of the proposed method.

### C. Results

Fig. 2 shows the error (Mean Squared Error, MSE) between the prediction results of the federated learning and comparison methods and the measured throughput values when the shared model is updated a total of nine times. The results show that federated learning, which updates only the weights, provides the same level of prediction accuracy as the comparison method. Here, the convergence of learning using coalitional learning is a little faster. However, we do not compare the convergence speed this time because the assumption that the same data is always used for training

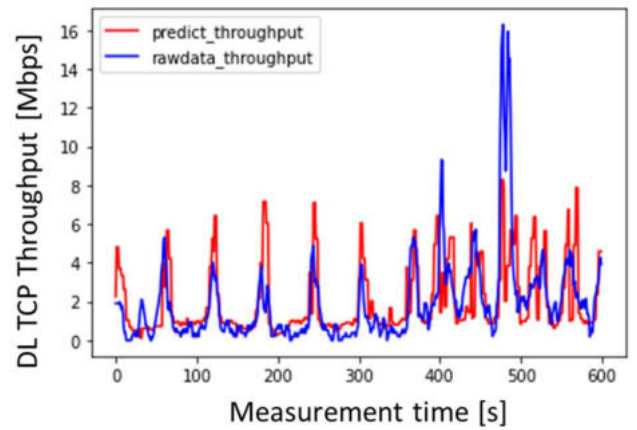


Fig. 3. Comparison of measured and predicted maximum TCP throughput values for DownLink (DL) (the prediction was made using a shared model that was updated seven times with minimal error)

the local model was introduced to make the learning converge faster. Fig. 3 shows the prediction results of the throughput in federated learning when the prediction error is the smallest for a single data pattern. A shared model that has been updated seven times is used to output the predictions shown in Fig. 3. This result shows that it can be seen that once the learning converges, the throughput can be predicted approximately even if parameters other than throughput are given as input. However, there is room for improvement in the prediction accuracy.

### D. Discussion

From the results of the verification, it can be seen that the prediction of the points where the throughput increases or decreases drastically is almost complete, but it does not fully follow the scale. For example, we switched the router to which we connected 360 seconds after the start of the measurement, and although the average throughput tends to be lower for the router connected first than for the router connected later, the prediction result shows that the average throughput is constant regardless of the access point connected. This result shows a trend that is not good for prediction. The learning model has not learned that the throughput value may change largely even if the input features change little. However, this unfavorable trend was expected before the verification. Therefore, we included location information and base station IDs (Network IDs) as input parameters so that the learning model could determine the differences in the characteristics of each access point. However, as the result shows, the expected results were not obtained.

In the next chapter, we propose a solution to this problem by narrowing down the target of the learning model.

## IV. ADDITIONAL VERIFICATION OF THROUGHPUT PREDICTION

This paper has only considered one shared model. However, the prediction error remains in the proposed method as described in Sect. III-C. Thus, we attempt to improve the prediction accuracy of throughput by creating a learning model for each area.

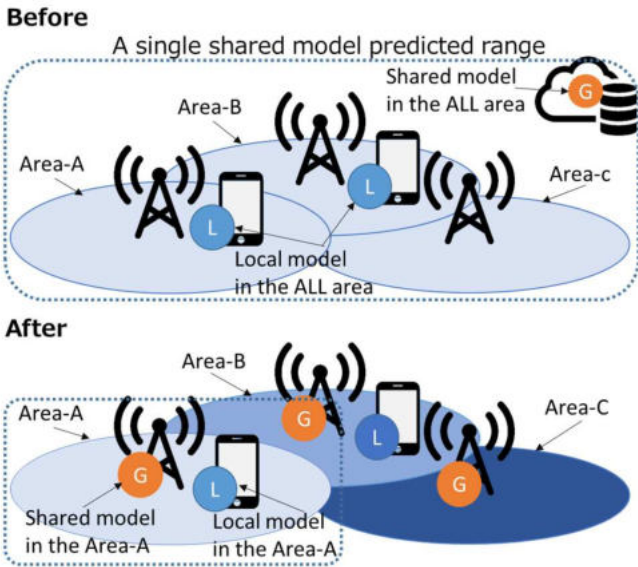


Fig. 4. Before/after comparison when changing the shared model's prediction target area

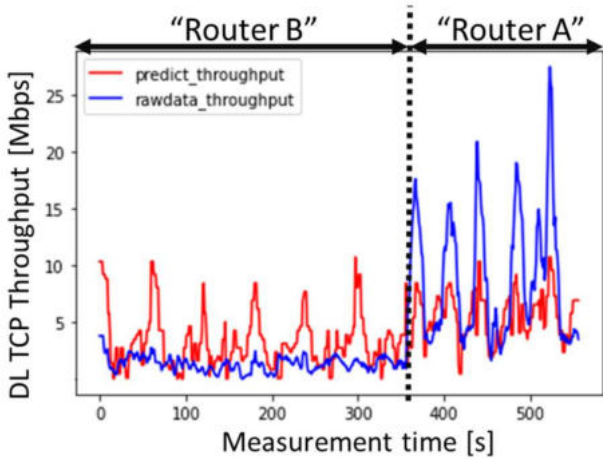


Fig. 5. Prediction results for measured values with large scale change in throughput using the shared model created before narrowing the area for prediction.

#### A. Details of Additional Validation

Fig. 4 shows an improvement of the proposed method. In the above figure that is represented as 'Before', a same shared model is used in all areas. Meanwhile, in the below figure, different models are utilized among areas. The local node switches the machine learning model for each access point. Using different models according to the area, the prediction accuracy of throughput can be improved.

This section verifies how much the prediction accuracy improves when the target of the learning model is narrowed down. In the additional validation, we use the same dataset as in Sect. III so that the prediction accuracy can be easily compared. For the sake of explanation, we name the routers used as connection points as Router A and Router B, respectively. In order to prepare a local node learning model and a shared model for each area, the datasets used for prediction must be only those for the same area. For example, if we want to predict the throughput of Router A, we should use only the dataset for Router A. As it is, the dataset used in Sect. III is recorded in a single file regard-

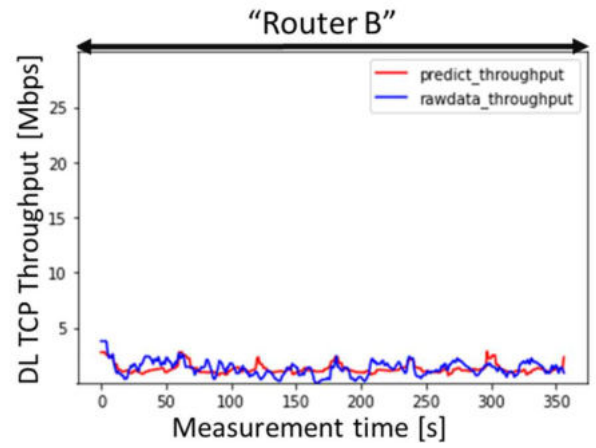


Fig. 6. Throughput prediction results in the same area by a learning model that predicts the communication-capable area of Router B.

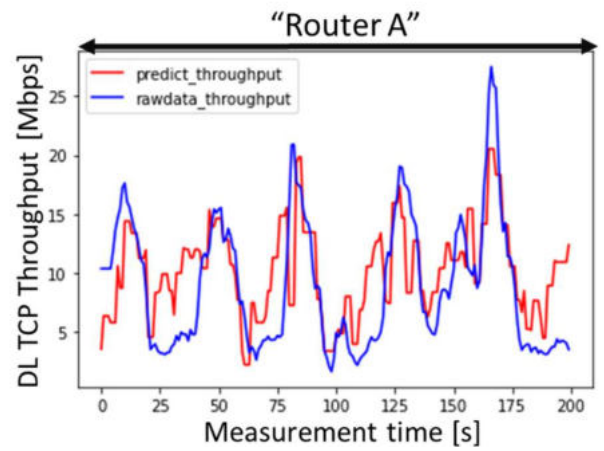


Fig. 7. Throughput prediction results in the same area by a learning model that predicts the communication-capable area of Router B.

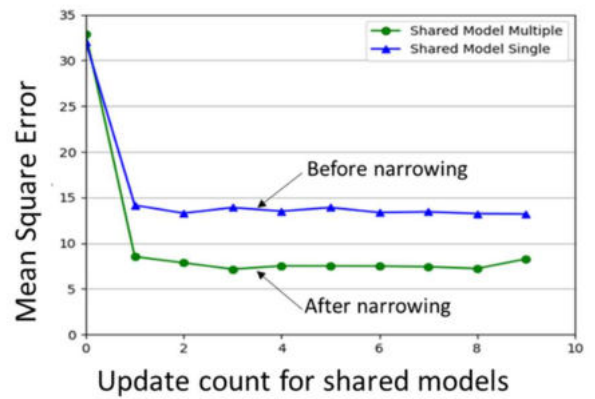


Fig. 8. Before/After narrowing the area for prediction

less of the number of access points to be connected. Therefore, the dataset was divided into separate files for each connection point before this verification. In other words, each of the three datasets was split into two files, one for Router A and one for Router B, and then the prediction accuracy was evaluated for each area. In other words, each of the three datasets will be split into one for Router A and one for Router B.

## B. Results of Additional Validation

Fig. 5 shows the results of predicting the average throughput of each access point with the shared model before narrowing down the prediction target. Fig. 6 and Fig. 7 show the results of prediction using the same measured values as in Fig. 5, but with different learning models for each router.

When comparing Fig. 8 shows the number of updates of the shared model and the error between the throughput prediction results of the shared model and the actual measured values. The errors in this Fig. are averaged over three patterns, each with different test data. From the results, we can expect a much better prediction accuracy.

## C. Discussion of Additional Validation

We have confirmed that narrowing down the target of prediction has certain advantages in terms of improving prediction accuracy, but we will discuss the possible disadvantages. Among several possible problems, one that we should pay particular attention to is that it takes a long time for the learning of a particular shared model to converge. The assumption is that each shared model can only collect local models from users in its own area. By subdividing the prediction target into smaller and smaller areas, it will be easier for users not to gather in the target area, or for the observation time in the target area to be shorter than before. If a local model that has not been sufficiently trained is shared, and a shared model with low prediction accuracy is created, it may adversely affect the convergence speed of training and prediction accuracy of all terminals that receive the shared model.

## V. FUTURE WORK

### A. Shared Model Transfer

We believe that transfer of training models can be effective in dealing with the problem of local nodes not congregating in a particular area, and we show a simple example of transferring a training model in Fig. 9. We will not go into the details of the validation results here, but by comparing the results of training a trained shared model with an untrained shared model, and training the model with a dataset under the same conditions, we found that the accuracy and convergence time were better when the trained shared model was transferred. The verification we have done to date is limited to only those cases where the surrounding environment of the source and destination are similar. Therefore, depending on the surrounding environment and communication method, we may not always obtain good results. As a future work, we would like to conduct additional discussions along with the increase of the datasets.

### B. Learning on Smartphones

Understanding the workload of smartphones is also an important issue. This is because smartphones have a limited drive time and can only process a limited number of tasks at a time. Even if the proposed method can reduce the security risk, it is difficult to share the weight parameters as a local node if the load on the smartphone is too high. Therefore, we are considering the use of TensorFlow Lite [11], which is expected to reduce the processing load of

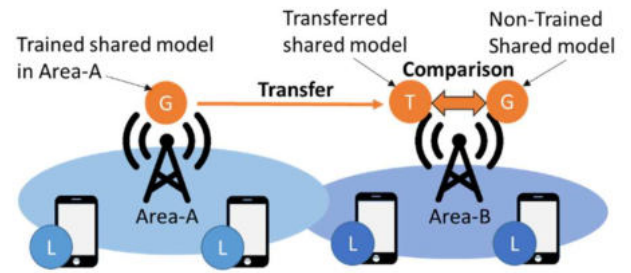


Fig. 9. A simple example of transferring a learned shared model to another area (comparing the method of learning from the transferred model with the method of learning from the unlearned model, and examining the difference in learning speed and prediction accuracy)

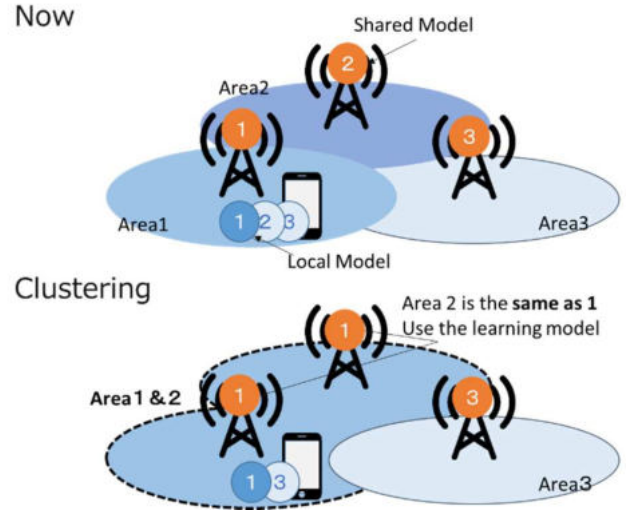


Fig. 10. Image of a clustering method that supports multiple areas with a single shared model.)

smartphones. This is expected to reduce the size of the model, thereby reducing not only the processing load but also the load on the device's storage. We plan to develop TensorFlow Lite models while confirming their performance in questionnaires to the extent that they do not impose a burden on users.

### C. Clustering of Shared Models

It is a good approach to reduce the total number of shared models, because if we prepare training models for each area, there will be less information sharing from local nodes, and it may not be possible to perform sufficient training. The number of training models is a tradeoff between prediction accuracy and learning difficulty. If there are fewer training models, the number of local nodes that provide data to the shared model increases, making learning easier, but the prediction accuracy becomes worse because the area to be predicted is too large. On the other hand, if the number of training models is many, prediction accuracy can be expected to improve because only local areas need to be predicted, but the number of local nodes providing data to one shared model will decrease, and in the worst case, no one will provide data. These tradeoffs can be seen from the additional validation described earlier. Therefore, if the prediction accuracy can be maintained to some extent, we should cluster the areas. Specifically, if the performance of the surrounding environment and router devices are similar, and if there is no significant difference in prediction accuracy, the approach is to integrate them.

Fig. 10 shows an image of a shared model that is clustered. In the upper part of the figure, a training model is needed for each area before clustering, but if area 1 and area 2 are merged as shown in the lower part of the figure, the area that one training model is responsible for can be expanded. The clustering method will be an issue to be discussed carefully.

## VI. CONCLUSION

We proposed an approach using coalition learning to solve the problems of security risk, communication volume, and operational cost in throughput prediction using conventional machine learning. Using real data measured by smartphones, coalition learning was conducted, and it was confirmed that the prediction accuracy was equivalent to that of conventional methods. In addition, the case where the prediction accuracy is improved by narrowing down the prediction target was also examined, and its merits and demerits were discussed.

This study was supported by JSPS Grants-in-Aid for Scientific Research JP18H01439 and 18KK0109.

## REFERENCES

- [1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson and B. Aguera, "Communication-efficient learning of deep networks from decentralized data," *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 1273-1282, 2017.
- [2] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," *Proc. IEEE ICC*, Shanghai, China, May 2019.
- [3] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *CoRR*, abs/1902.04885, 2019.
- [4] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," *In Proceedings of the 36th International Conference on Machine Learning*, pp. 634-643, 2019.
- [5] T. Yang, G. Andrew, H. Eichner, H. Sun, W. Li, N. Kong, D. Ramage, and F. Beaufays, "Applied federated learning: Improving Google keyboard query suggestions," *arXiv preprint 1812.02903*, 2018.
- [6] D. Phaneekham, S. Nair, N. Rao and M. Truty, "Predicting throughput of cloud network infra-structure Using Neural Networks," *IEEE INFOCOM 2021-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 1-6, 2021.
- [7] H. Jha and V. Vijayarajan, "Mobile internet throughput prediction using machine learning techniques," *2020 International Conference on Smart Electronics and Communication (ICOSEC)*, 2020, pp. 253-257, 2020.
- [8] A. Panghal, K. Govindan and K. Subramaniam, "Transmission time estimator for social and cloud applications in smartphones," *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1-6, 2017.
- [9] S. K. Das and S. Beborita, "Heralding the future of federated learning framework: architecture, tools and future directions," *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pp. 698-703, 2021.
- [10] TensorFlow, URL <https://www.tensorflow.org/>
- [11] TensorFlow Lite Converter, URL <https://www.tensorflow.org/lite/convert/>
- [12] Android Studio, URL <https://developer.android.com/>

# Thermal Array Sensor Resolution-Aware Activity Recognition using Convolutional Neural Network

Goodness Oluchi Anyanwu, Cosmas Ifeanyi Nwakanma, Adinda Riztia Putri, Jae-Min Lee, and Dong-Seong Kim

\*\*JeongHan Kim, and Gihwan Hwang

*Department of IT Convergence Engineering, Kumoh National Institute of Technology Gumi, South Korea*

\*\*TLBIZ Company Limited, Korea

(anyanwu.goodnes, cosmas.ifeanyi, adindariztia, ljmpaul, dskim)@kumoh.ac.kr

**Abstract**—Human activity detection and classification (HADC) has become a growing research issue due to the development of sensor technologies, deep learning models, and the need for the safety of people in smart spaces such as buildings and factories. Various researchers have employed sensors with different resolutions for HADC. However, the impact of sensor resolution on the sensor data quality and the accuracy of the deep learning algorithm in this field has been little discussed. In this work, the impact of three different thermal sensor resolutions was investigated while proposing a convolutional neural network (CNN). The results showed that the proposed CNN displayed a resolution-aware performance, being able to contain the impact of a change in thermal sensor resolutions. Although the CNN model had lower accuracy as the resolution was changed from 32x32 to 4x4H and 4x4, respectively. However, it was able to reduce the type errors and maintain an average accuracy of 82.74%.

**Index Terms**—Activity Recognition, Deep Learning, Thermal Sensor, Sensor Resolution.

## I. INTRODUCTION

Detecting human activities using sensor-based methods has attracted widespread attention in homes, factory shop floors and smart building sectors [1]. Currently, Human activity detection and classification (HADC) often comprises sensors (environmental, visual, and wearables), combining capabilities to provide and enable monitoring in different fields [2]. Existing technologies such as camera-based detection schemes expose privacy from a camera, thus invading the privacy of people whose activities are being monitored [3]. Also, conventional wearable approaches cause discomfort to users and sometimes require the cooperation of users for efficient and successful deployment.

Compared to the approaches above, a thermal sensor detects activity by concentrating the infrared (IR) energy radiated by an object onto the photo-detectors [4]. The photo-detector, in turn, transforms that energy into an electrical signal relative to the IR energy emitted by the object with developed technology for improved and accurate activity detection. Thermal sensors generally have distinct image pixel characteristics due to the intentional distinction between their image generation processes [5]. In recent years, some authors have proposed diverse methodologies for detecting and recognizing objects in thermal infrared imagery, particularly in sensing and human activity (dynamic and static) recognition.

The authors in [5] provided insights on a wide range of sensors used for HADC. The following thermal imaging sensors' resolutions were outlined and described: 4x16, 8x8, and 16x16. Emphasis was also placed on positioning and right sensor placement as thermal array sensors have a narrow field of view (FoV). In [6], sensor data from an 8x8 resolution thermal sensor was extracted and processed using the J-Butterworth and the Kalman filter. In [7] to reduce noise in the raw data and achieve an enhanced HADC mechanism, the Long-Short-Term Memory algorithm was the model used in [6] to extract the data feature. However, the authors failed to explain the environmental setup for their testbed.

In [8], two 4x16 low-resolution thermal sensors were used to collect activity data. The sensors were stationed halfway along a wall, and six different algorithms were trained. The feed-forward neural network gave the best activity detection accuracy, justifying the detection capability of Deep Learning (DL) models for the purpose of HADC [9]. However, there is no major evidence of existing work where the impact of thermal array sensor resolutions has been fully investigated to justify the impact or not of the resolution of thermal sensors on the overall performance of HADC systems. Although, [10] carried out similar modeling and achieved an accuracy of 80% and 90% performances for vision-based and classification-based workflows using the feature classification-based model. However, their model was based on predicting occupancy only.

Lower resolution thermal sensors have a narrow FoV and are usually implemented in pairs to widen their FoV [5]. Their advantage is that they introduce low noise due to little interference during data gathering. On the other hand, a higher resolution thermal sensor has a wider FoV and does not need to be paired up with other thermal sensors. However, they are subjected to noise interference [11]. There is a need, therefore, to design a DL model that can handle the introduction of noise interference as the resolution of sensors increases, as DL models are also efficient in recognizing human dynamic activities in line of sight and non-line of sight scenarios.

This study aims at exploring the resolution aware capacity of three (3) different thermal sensor resolutions to train a DL model for activity recognition by collecting data from these three different sensors. The approach adopted in this work is different from recent research, which only focuses on detection

or classification of human activity and not resolution-aware capacity. The main contribution of this work, with respect to the state of the art, is the implementation of a Convolutional Neural Network (CNN) applied to three different resolutions of Omron Thermal Sensors (4x4, 4x4H, and 32x32) for HADC and evaluating the resolution aware ability of each sensor using the DL approach.

The contributions of this work are as outlined below:

- 1) To design a non-intrusive and low-cost activity recognition system test bed setup using three different thermal sensors with different resolutions as the data acquisition device.
- 2) To develop a CNN model that is resolution-aware and able to handle the noise introduced by the increased resolution of the thermal sensors.
- 3) To investigate the impact of sensor resolution and prove that the model is resolution-aware, i.e., sensor data and resolution do not affect the ability of the algorithm to effectively detect and classify human activities.

The paper arrangement is thus: Section II is the proposed framework, architecture, hardware design, and testbed. Section III was devoted to experimental results and Section IV concluded the paper.

## II. SYSTEM MODEL

This section provided the background of the developed framework, description of the sensor and hardware testbed, software configuration, and the evaluation criteria on various sensor prediction models. The stages of the HADC model include raw sensor data collection in form of heatmaps, data preprocessing to temperature values and data segmentation, feature extraction and engineering, and the DL model.

### A. Sensors Testbed and Setup

The system was designed by placing the sensor on a ceiling located 2.5m meters away from ground level as shown in Fig.1, measuring the heat received from the object and converting it to temperature data. The choice of the ceiling is to achieve a good FoV in line with the product data-sheet hence proper placement of the sensor was established to ensure appropriate FoV of the sensors to capture the object effectively [12]. Two activity detection scenarios were designed namely; activity (class 1) and no activity (class 0). The Omron sensors were used because of their low noise, stable temperature values, and easy-to-use properties.

The OMRON thermal sensor was developed with thermal fire technology and which include a low noise amplifier, a cap with a silicon lens, MEMS thermopile sensor chips, and a microcontroller unit for converting analog signals to digital signals. The thermal sensor technology recognizes the combination of thermopile elements and ASICs into a unified platform, resulting in a super-compact footprint and high resolution. The silicon lens captures radiated heat from an object and directs it to the module's thermopile. An electromotive force is generated within this module and used to calculate the temperature value via an analog circuit made available

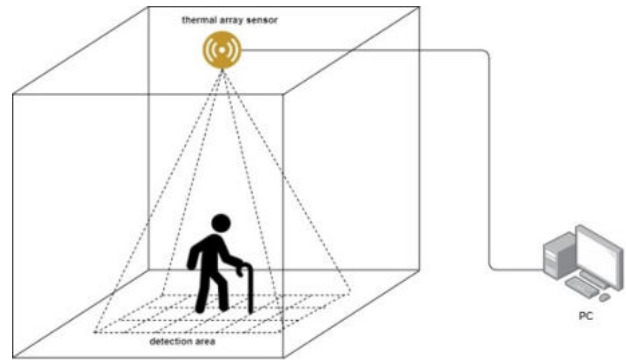


Fig. 1. Sensor Placement in the Smart Space at 2.5m from the ceiling to ensure target object is within FoV

through the use of the I2C protocol [13]. The description and specifications of the sensors used are shown in Table I.

TABLE I  
DESCRIPTION OF SENSORS USED

Sensor Name	D6T-44L-06	D6T-44L-06H	D6T-32L-01A
Sensor Elements (in Pixels)	16 (4 x 4)	16 (4 x 4)	1024 (32 x 32)
Sensors Standpoint: X-direction Y-direction	X = 44.2° Y = 45.7°	X = 44.2° Y = 45.7°	X = 90.0° Y = 90.0°
Distance: X & Y	X = 1.6m Y = 1.69m	X = 2.44m Y = 2.53m	X = 6m Y = 6m
Temperature Detection Range	5 – 50°C	5 – 200°C	0 – 200°C
Current Consumed	5 mA	5 mA	19 mA

The sensors capture multiple frames within one second because they are built to update temperature data every 300 ms or less. Evaluation was performed by connecting the sensors to the evaluation boards and Arduino with a harness cable as shown in Fig. 2. Output specifications of the sensors are digital values (binary codes) corresponding to the reference temperature (known as heatmap). Data capture was performed and processed using a low-cost desktop computer with an Intel(R) Core(TM) i5-8500 CPU 3.00GHz processor, 8GB RAM, and an NVIDIA GeForce GTX 1050 GPU running on Windows 10. The OMRON sensor was linked to an Arduino board- a microcontroller- to carry out this process. The Arduino board reads sensor data and transmits it to the sensor evaluation board via the UART interface.

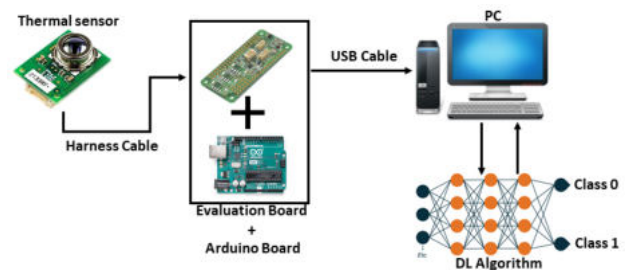


Fig. 2. Set up and Hardware Configuration

### B. Hardware Requirements, Assemble and Software Configuration

Hardware requirements and functionality are as follows: Arduino Board (MKR WiFi 1010), Omron Sensor Evaluation Board (2JCIE), Omron Sensors (D6T-44L-06, D6T-44L-06H, D6T-32L-01A), Omron Harness Cable (2JCIE-HARNESS-01 for Arduino), USB and PC. The process of the hardware assembly is shown in Fig. 2 and Fig. 3 are:

- Solder the evaluation board and connect it to the Arduino board.
- Connect the USB cable to the Arduino board.
- Connect the omron harness cable to the omron evaluation board.
- Connect the other end of the harness to the Omron sensor.
- Place the thermal sensor in the human detection area.
- Download Arduino IDE and install the package and set up the driver.
- Connect the Arduino board to PC using the USB cable.

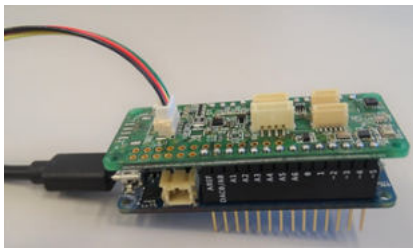


Fig. 3. Connecting the Sensor to the Evaluation Board prior to Placement

The steps for software configuration, visualization and data acquisition are as follows:

- Download Arduino IDE from their official site ([www.Arduino.cc](http://www.Arduino.cc)).
- Run the installation package and wait until the installation is finished.
- Plug the assembled Arduino device, if there is a package installation notification, click to install them.
- Arduino IDE is ready to use, write the code and upload it using the upload button to run the code to the sensor.
- Add Arduino SAMD Core Library to Arduino IDE > Tools menu > Boards > Boards Manager.
- Navigate to Device Manager in your Windows, make sure that Arduino MKR WiFi 1010 is available in COM & LPT ports.
- Set up the board and ports to be used in Arduino IDE > Tools > Board > Arduino SAMD (32 Bits) Boards > Arduino MKR WiFi 1010, and also the port > Tools > Port > Arduino port.
- Select the thermal sensor type and Upload the thermal data sketch to the Arduino interface as Zip file > Sketch > Include Library > Add .ZIP library
- Compile and upload the code to generate the sensor data from the thermal sensor
- Download the processing application to update temperature values generated from the Arduino interface.

### C. Proposed Convolutional Neural Network (CNN)

Due to growing computational resources, CNN has been widely adopted as a result of its full potential. [14], [15]. CNN extracts feature maps from a given dataset and learns patterns from the convolutional operations of such a dataset. The CNN model implemented in this work was built using the popular deep learning platform - Keras library on a Jupyter notebook. Standard Python libraries were used to provide optimized implementations of each sensor data. Although CNN is commonly used for image datasets, by using the 1D-CNN, its benefits were exploited. Furthermore, with little or no feature engineering, 1D CNNs have demonstrated to offer state-of-the-art performance on HADC tasks. As shown in Figure 4, the generated heat map, converted to temperature data is captured by the CNN model for the purpose of the HADC.

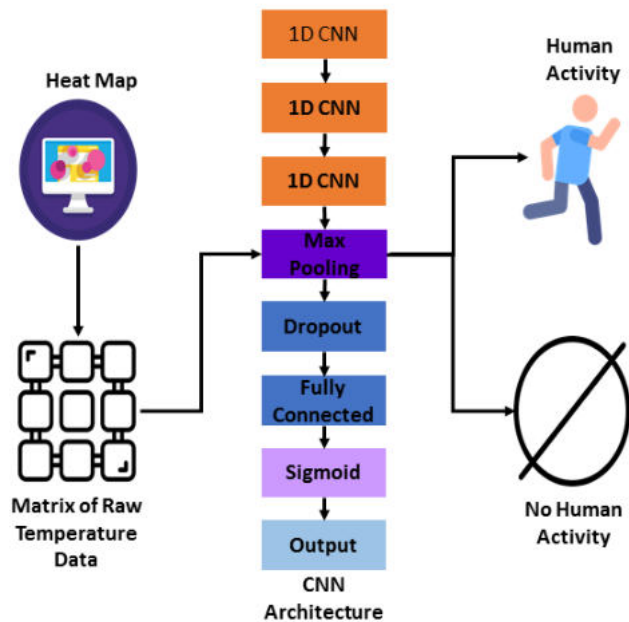


Fig. 4. Overall architecture of the HADC model showing the proposed 1D CNN model and its layers

TABLE II  
PARAMETER SETTINGS FOR PROPOSED CNN MODEL

CNN Layer	Filter Number	Kernel Size	Activation Function
1st Layer	64	5	ReLU
2nd Layer	64	7	ReLU
3rd Layer	256	7	ReLU
Max Pooling	2		
Dropout	0.5		
Loss Function	Binary Cross Entropy		
Optimizer	Adam		

The proposed 1D CNN consists of three (3) stacked convolutional layers with optimal parameter selection on each layer. The set up of each layer and criteria is shown in Table II. The convolutional layers summarize the presence

of the input features by creating a feature map and applying learned filters to the input dataset. The max-pooling layer down samples these feature maps by summarizing the features in patches [16]. To suppress over-fitting and co-dependency among neurons during the training process, a drop-out of 0.5 neurons is added before the fully connected layer. The Adam optimizer was used, given its benefits. The generated dataset is split into 80% for training and 20% for testing. Up-sampling was performed to deal with imbalanced data as activity data was far greater than non-activity data. Finally, the dataset is scaled using the standard scaler.

### III. PERFORMANCE EVALUATION AND RESULT

This section presents the experimental results for the proposed model on the three different sensor resolution types. As shown in Fig. 5, the proposed 1D CNN showed resilience in attaining appreciable linear accuracy as well as reduced errors despite the impact of sensor resolution. The limiting factor of using a larger thermal sensor array is noise. However, in this work, there is no analysis of the amount of noise received from the higher resolution sensor. Although, the aim of proposing the DL model implemented in this work is to ensure that the performance of the HADC model is not affected as the resolution of sensors increases. The proposed model achieved accuracy of 78.47%, 78.72% and 91.02% for 4x4, 4x4H and 32x32 thermal resolutions, respectively. It is agreeable that a change in resolution affects accuracy. However, the proposed CNN model maintained a reasonable average performance of 82.74%, thus resolution-aware.

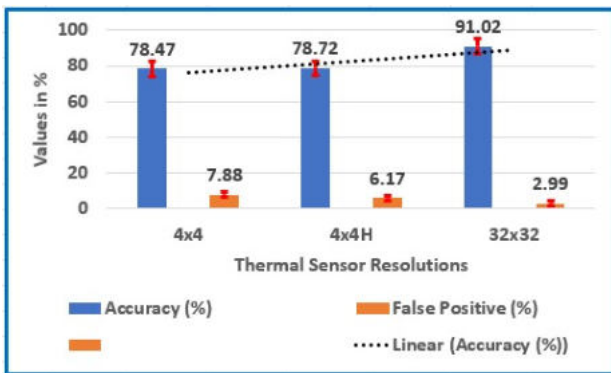


Fig. 5. Accuracy and False Positives of the Thermal Sensor Resolutions for Activity Recognition

Also, the result is presented with confusion matrix showing the true positive, true negative, false positive and false negative of the sensors in Fig. 6, Fig. 7, and Fig. 8 respectively. Since an increase in resolution tends to increase the noise in the sensor data, the goal of the CNN is to take care of the impact and ensure that type I error (false positive) is kept to a minimum no matter the increase in the resolution of the sensor and data. From the confusion matrix, it can be observed that the false positive or type I error of the proposed CNN for 4x4, 4x4H, and 32x32 resolutions are 7.88%, 6.17% and 2.99%

respectively. In the same vein, the type II error was effectively managed too as shown in Fig. 9.

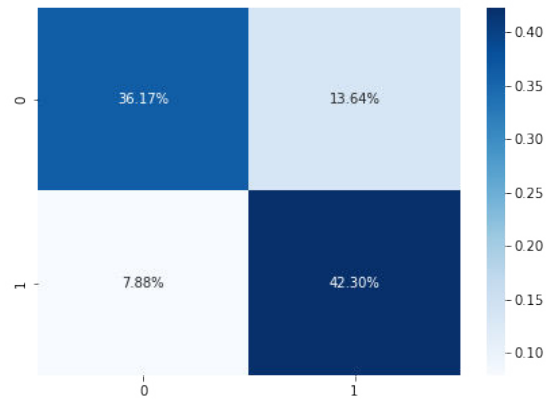


Fig. 6. Confusion Matrix of the proposed CNN using 4x4 Resolution Thermal Sensor Data

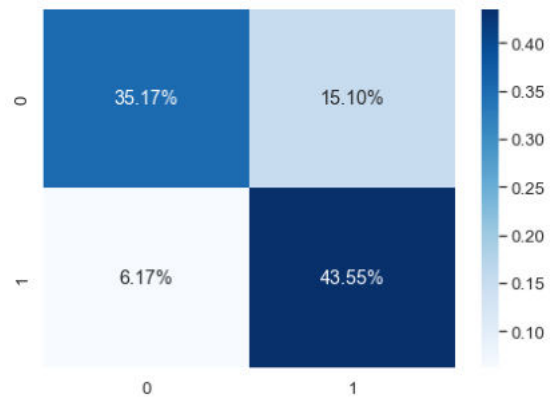


Fig. 7. Confusion Matrix of the proposed CNN using 4x4H Resolution Thermal Sensor Data

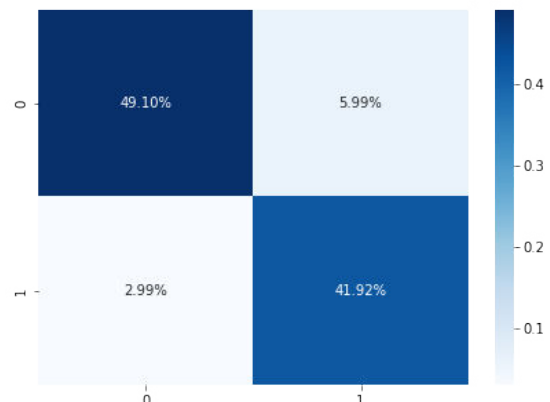


Fig. 8. Confusion Matrix of the proposed CNN using 32x32 Resolution Thermal Sensor Data

### IV. CONCLUSION

In this work, a CNN model was developed for detecting and classifying human activities using different thermal sensor



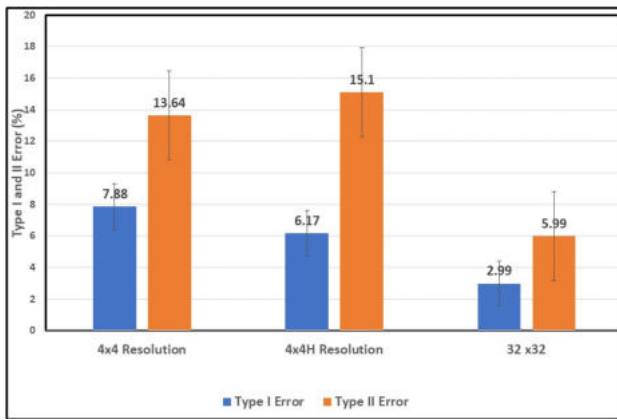


Fig. 9. CNN effectively handled the Type I and II errors despite the increase in thermal sensor resolution

resolutions. The result shows that irrespective of the resolution of sensors used in capturing the data, the proposed CNN showed resilience in its ability to effectively detect and classify the activities. It is a future research direction to explore other DL and ensemble learning candidates as a way of investigating their capabilities in comparison to the proposed CNN. The proposed CNN showed appreciable accuracy performance using thermal sensor resolutions of 4x4, 4x4H, and 32x32 respectively.

#### ACKNOWLEDGMENT

This research work was supported by Priority Research Centers Program through NRF funded by MEST (2018R1A6A1A03024003) and the Grand Information Technology Research Center support program (IITP-2021-2020-0-01612) supervised by the IITP by MSIT, Korea.

#### REFERENCES

[1] C. I. Nwakanma, F. B. Islam, M. P. Maharani, J.-M. Lee, and D.-S. Kim, "Detection and Classification of Human Activity for Emergency Response in Smart Factory Shop Floor," *Applied Sciences*, vol. 11, no. 8, 2021. [Online]. Available: <https://www.mdpi.com/2076-3417/11/8/3662>

[2] L. Tao, T. Volonakis, B. Tan, Y. Jing, K. Chetty, and M. Smith, "Home Activity Monitoring using Low Resolution Infrared Sensor," *CoRR*, vol. abs/1811.05416, 2018.

[3] Y. Wu, H. Liu, B. Li, and R. Kosonen, "Prediction of Thermal Sensation using Low-cost Infrared Array Sensors Monitoring System," *IOP Conference Series: Materials Science and Engineering*, vol. 609, p. 032002, October 2019.

[4] F. Riquelme, C. Espinoza, T. Rodenas, J.-G. Minonzio, and C. Taramasco, "eHomeSeniors Dataset: An Infrared Thermal Sensor Dataset for Automatic Fall Detection Research," *Sensors*, vol. 19, no. 20, 2019.

[5] B. Fu, N. Damer, F. Kirchbuchner, and A. Kuijper, "Sensing Technology for Human Activity Recognition: A Comprehensive Survey," *IEEE Access*, vol. 8, pp. 83 791–83 820, 2020.

[6] C. Yin, J. Chen, X. Miao, H. Jiang, and D. Chen, "Device-Free Human Activity Recognition with Low-Resolution Infrared Array Sensor Using Long Short-Term Memory Neural Network," *Sensors*, vol. 21, no. 10, 2021.

[7] A. A. Trofimova, A. Masciadri, F. Veronese, and F. Salice, "Indoor Human Detection Based on Thermal Array Sensor Data and Adaptive Background Estimation," *Journal of Computer and Communications*, vol. 5, no. 4, 2017.

[8] S. Shelke and B. Aksanli, "Static and Dynamic Activity Detection with Ambient Sensors in Smart Spaces," *Sensors*, vol. 19, no. 4, 2019. [Online]. Available: <https://www.mdpi.com/1424-8220/19/4/804>

[9] V. B. Semwal, A. Gupta, and P. Lalwani, "An Optimized Hybrid Deep Learning Model using Ensemble Learning Approach for Human Walking Activities Recognition," *The Journal of Supercomputing*, 2021.

[10] B. Sirmacek and M. Riveiro, "Occupancy Prediction Using Low-Cost and Low-Resolution Heat Sensors for Smart Offices," *Sensors*, vol. 20, no. 19, 2020.

[11] D. Jahier Pagliari and M. Poncino, "On the Impact of Smart Sensor Approximations on the Accuracy of Machine Learning Tasks," *Heliyon*, vol. 6, no. 12, p. e05750, 2020.

[12] A. Naser, A. Lotfi, and J. Zhong, "Adaptive Thermal Sensor Array Placement for Human Segmentation and Occupancy Estimation," *IEEE Sensors Journal*, vol. 21, no. 2, pp. 1993–2002, 2021.

[13] "Omron Electronic Components: White Paper on 'D6T MEMS Thermal Sensors: Non-Contact Thermal Sensor for Contact-less Measurement'." 2018. [Online]. Available: <https://www.omron-ecb.co.kr/product-detail?partNumber=D6T>

[14] T. Huynh-The, C.-H. Hua, Q.-V. Pham, and D.-S. Kim, "MCNet: An Efficient CNN Architecture for Robust Automatic Modulation Classification," *IEEE Communications Letters*, vol. 24, no. 4, pp. 811–815, 2020.

[15] S.-H. Kim, J.-W. Kim, W.-P. Nwadiugwu, and D.-S. Kim, "Deep Learning-Based Robust Automatic Modulation Classification for Cognitive Radio Networks," *IEEE Access*, vol. 9, pp. 92 386–92 393, 2021.

[16] D. Xiao, Y. Chen, and D. D.-U. Li, "One-Dimensional Deep Learning Architecture for Fast Fluorescence Lifetime Imaging," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 27, no. 4, pp. 1–10, 2021.

# An Investigation on Deep Learning-Based Activity Recognition Using IMUs and Stretch Sensors

1<sup>st</sup> Nguyen Thi Hoai Thu

School of Electronic and Electrical Engineering  
Kyungpook National University  
Daegu, Republic of Korea  
thunguyen@knu.ac.kr

2<sup>nd</sup> Dong Seog Han

School of Electronic and Electrical Engineering  
Kyungpook National University  
Daegu, Republic of Korea  
dshan@knu.ac.kr

**Abstract**—With the advancement and ubiquitousness of wearable devices, wearable sensor-based human activity recognition (HAR) has become a prominent research area in the healthcare domain and human-computer interaction. Inertial measurement unit (IMU) which can provide a wide range of information such as acceleration, angular velocity has become one of the most commonly used sensors in HAR. Recently, with the growing demand for soft and flexible wearable devices, mountable stretch sensors have become a new promising modality in wearable sensor-based HAR. In this paper, we propose a deep learning-based multi-modality HAR framework which consists of three IMUs and two fabric stretch sensors in order to evaluate the potential of stretch sensors independently and in combination with IMU sensors for the activity recognition task. Three different deep learning algorithms: long short-term memory (LSTM), convolutional neural network (CNN) and hybrid CNN-LSTM are deployed to the sensor data for automatically extracting deep features and performing activity classification. The impact of sensor type on recognition accuracy of different activities is also examined in this study. A dataset collected from the proposed framework, namely iSPL IMU-Stretch and a public dataset called w-HAR are used for experiments and performance evaluation.

**Index Terms**—activity recognition, IMUs, stretch sensors, wearable sensors, deep learning

## I. INTRODUCTION

Wearable sensor-based human activity recognition has become an emerging research topic that uses body-worn motion sensors to understand human behaviors, detecting abnormal activities (e.g., fall, gait disorder), hence it helps to encourage people to have healthier lifestyles and provides quick response to emergency situations. Recent advances in micro electro-mechanical system (MEMS) have enabled these sensors to be shrunken to extremely small sizes and widely embedded into smart wearable devices such as smartphones and smart-watches. Therefore, in view of ubiquitousness and user privacy preserving, this wearable sensor-based approach is preferred in healthcare applications over other approaches that use radio frequency signals or vision sensors.

One of the most commonly used wearable sensors in HAR is the inertial measurement unit (IMU) which can contain a variety of sensors such as motion sensor (accelerator), rotation

sensor (gyroscope) and magnetometer. Bächlin *et al.* [1] use two sensors, which provide 3-D acceleration and attach them to the Parkinson disease patients' leg (*i.e.*, shank and thigh) for detecting freezing of gait events. Authors in [2] and [3] use data collected from accelerator and gyroscope embedded in smartphones to recognize a wide range of daily living activities and detect fall events. Although a growing body of literature has successfully applied the inertial sensors to HAR, these body-worn sensors are made of inflexible and rigid materials that can be obtrusive, interfere with the user's natural movements, and lead to discomfort wearing experience [4]. This limitation has urged researchers to take further action on developing the next generation of wearable sensors.

In recent years, there has been considerable interest in designing soft, textile wearable sensors and they have shown a high potential in a variety of applications from the healthcare domain to human-computer interaction. Prominent among these sensors is the fabric stretch sensor also known as strain sensor. The sensor can be stitched to the clothes or directly attached to human skin to measure a wide range of motion from articulation bending-straightening to respiration and heartbeat. Chander *et al.* [5] have used a stretchable soft robotic sensor to detect the kinematics of the ankle joints during slip and trip perturbations. Bhat *et al.* [6] share a dataset that contains data collected from a wearable accelerometer and stretch sensor. Handcrafted feature extraction methods such as discrete wavelet transform and fast Fourier transform are applied to the collected sensor data before a neural network-based classifier is deployed for activity recognition. A full-body sensing system consisting of IMU, knitted piezoresistive fabric strain sensors, EMG electrodes, goniometers and force sensors is proposed by Klaassen *et al.* [7] for monitoring stroke patients in a home environment.

In this paper, a study on the potential of the stretch sensors in combination with IMU sensor in wearable sensor-based human activity recognition task is conducted with the use of deep learning algorithms. An investigation on different sensor combinations is carried out to evaluate the sensor combination effectiveness and impact of sensor position on activity recognition accuracy. The rest of the paper is organized as follows. Section II introduces the overall framework of multi-modality HAR and gives details on the data collection

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2021R1A6A1A03043144).

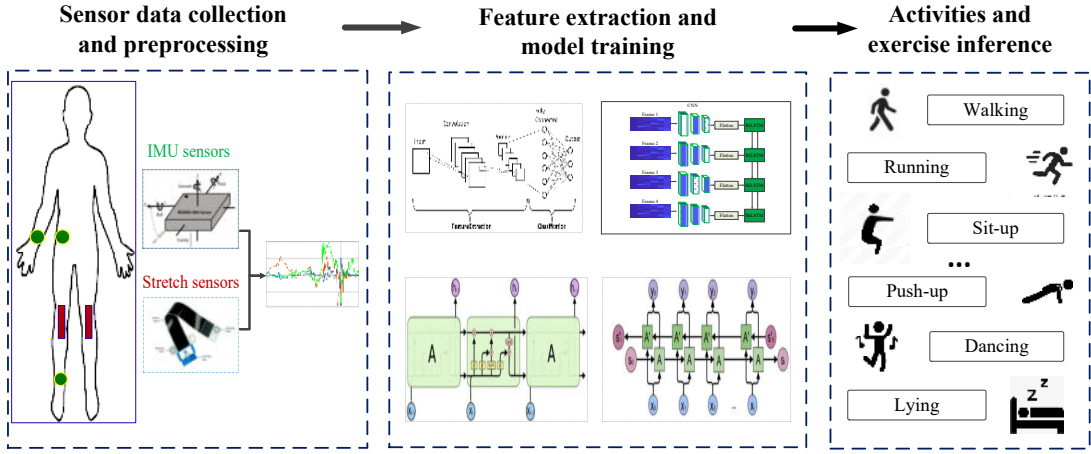


Fig. 1. Overall framework of the deep learning-based human activity recognition system using multi-modal wearable sensors.

method and structure of different deep learning-based HAR models. A set of experiments and results are presented in Section III with some discussion. Finally, our conclusions and some notes on future research are drawn in Section IV.

## II. METHODOLOGY

The deep learning-based multi-modality HAR framework considered in this study contains three main components: i) data collection and preprocessing, ii) feature extraction and model training, and iii) activities and exercise inference. Firstly, data collected from wearable sensors (*i.e.*, IMUs and stretch sensors) is transmitted to the server and pre-processed here. Secondly, the training dataset which contains pre-processed data with labels is then input into a deep learning model for the training process and automatic feature extraction. Finally, the trained model is used on sensor data in the testing phase or real-life scenarios for inferring human activities. The overall structure of this multi-modality HAR framework is illustrated in Fig. 1.

### A. Data Collection and Pre-processing

In this framework, three programmable WiFi 9-axis absolute orientation sensors are used and mounted at the user's wrist, waist and ankle. 3-axial acceleration, angular velocity and linear acceleration information are collected from each sensor, thus, there are 9 attributes at each time step. The fabric stretch sensors used in this study are flexible capacitors. Each stretch sensor is constructed from 5 layers: two outer layers are ground electrodes followed by two dielectric layers and the middle layer is a signal electrode. A coaxial cable is connected to the signal electrode and ground electrodes to collect the capacitance data. The capacitance will increase and decrease when the sensor extends and contracts. Because of this working principle, we stitched the two stretch sensors at the knees of the user garment in order to catch the bending and stretching activities of the user's legs.

Data collected from the IMU sensors is transmitted to the server using the message queuing telemetry transport (MQTT)

protocol and the Mosquitto broker [8] while the stretch sensor data is collected by using the Bluetooth low energy (BLE) technology. For a simple sensor data combination and to reduce the complexity, both types of sensors use the same sampling frequency of 25 Hz.

Data loss that happened during the wireless transmission is handled by using a linear interpolation method. Data collected from all the sensors are then synchronized and split into small windows with a fixed size of 2 seconds (50 data samples). An overlap of 50% between every two adjacent windows is applied to avoid missing data and increase the recognition granularity. Single window data of an IMU sensor can be described as

$$\mathbf{W}_{\text{IMU}} = (\mathbf{a}_x \ \mathbf{a}_y \ \mathbf{a}_z \ \mathbf{g}_x \ \mathbf{g}_y \ \mathbf{g}_z \ \mathbf{l}_x \ \mathbf{l}_y \ \mathbf{l}_z) \in \mathbb{R}^{N \times K} \quad (1)$$

where  $L$  is the window length,  $\mathbf{a}_x, \mathbf{a}_y, \mathbf{a}_z, \mathbf{g}_x, \mathbf{g}_y, \mathbf{g}_z, \mathbf{l}_x, \mathbf{l}_y,$  and  $\mathbf{l}_z$  are column vectors contain  $L$  data samples of 3-axial acceleration, 3-axial angular velocity and 3-axial linear acceleration, respectively. Because stretch sensor provides only the stretching-shrinking information so its window data can be expressed as a column vector  $\mathbf{w}_s \in \mathbb{R}^{L \times 1}$ . In this study,  $L$  is set to 50. When data of more than one sensor is used, the windows of all sensors are concatenated by joining all the column vectors, and finally a window data  $\mathbf{W}_{\text{fusion}} \in \mathbb{R}^{L \times N}$  is obtained, where  $N$  is the total number of sensor attributes.

### B. Deep Learning Models

With the capability of automatically extracting deep features during training and the fast growth in computing power and data source, deep learning has been widely applied to HAR applications. Therefore, our study on the effectiveness of IMU and stretch sensors in HAR is carried out with a special focus on the deep learning approach. Three deep learning networks which are long short-term memory (LSTM), 1-dimensional convolutional neural network (1D-CNN) and hybrid CNN-LSTM are deployed into the HAR model with different sets of input data. Detailed structures of the three deep learning-based HAR models are described in Fig. 2.

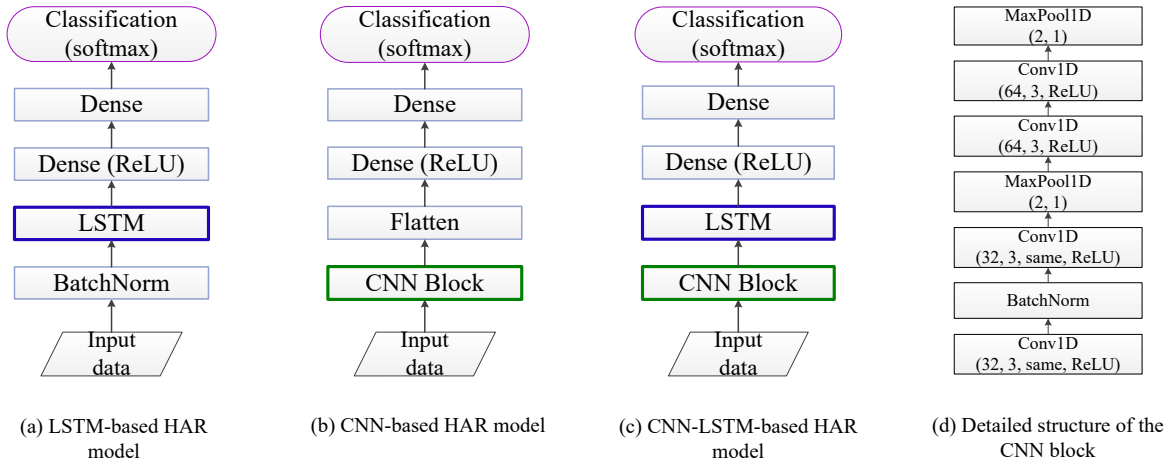


Fig. 2. Detailed structure of three deep learning models considered in the study.

1) *LSTM-based HAR*: Long short-term memory network (LSTM) [9], with the advantage in being able to connect the information in the past and present, has been widely applied to a variety of problems that relate to sequential data such as speech recognition and neural machine translation. In wearable sensor-based HAR, data is collected through time and managed as a set of time series, therefore an LSTM-based deep learning model is implemented for classifying human activity. Details of this LSTM-based HAR model is shown in Fig. 2(a).

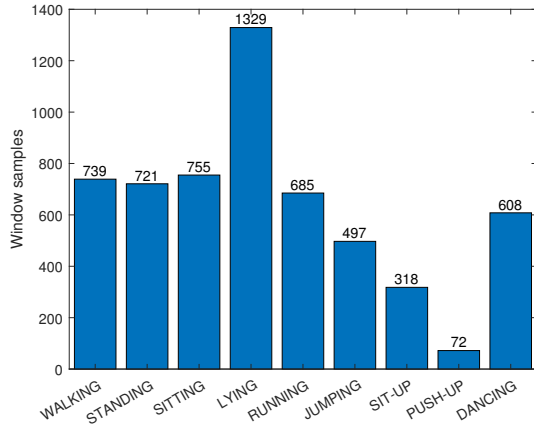
A batch normalization layer [10] is directly applied to the input data as a regularization method for reducing internal covariate shift and obtaining a faster convergence. The normalized sensor data is then forwarded to an LSTM layer where features are extracted along the time direction recurrently. Output hidden state of the final time step is fed into a dense layer (a.k.a fully-connected layer) with a ReLU activation function [11] for further analysis. Finally, another dense layer and a softmax activation function are deployed to normalize the outputs to a probability distribution over the predicted activities. During training, these normalized outputs are used to calculate the categorical cross-entropy loss, while in the testing phase, the activity with the highest probability is considered as the final prediction result.

2) *CNN-based HAR*: LeNet [12], a convolutional neural network and its variants have been successfully applied to not only imagery data in computer vision but also to time series in other fields such as natural language processing and time series forecasting. The operating of sliding several convolution kernels throughout the input feature maps helps CNNs to reduce the number of parameters by using a parameter sharing scheme and utilizing parallel computing techniques. In this research, a 1-dimensional CNN model whose kernels slide along the time direction of the window sensor data  $\mathbf{W}$  of length  $L$  and width  $N$  to extract local temporal features.  $N$ , number of attributes at a time step is treated as the depth of

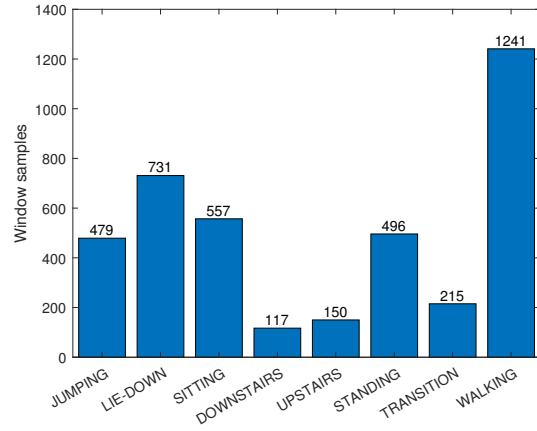
the input data, thus, kernels of the first convolutional layer have the depth equal to  $N$ .

The CNN-based HAR model contains a CNN block followed by a set of flatten and dense layers for activity classification as illustrated in Fig. 2(c). The CNN block is constructed from multiple convolutional, batch normalization and max-pooling layers. More details on this CNN block is described in Fig. 2(d). A description (32, 3, same, ReLU) of a convolutional layer can be understood as followed: there are 32 kernels whose length equals to 3; zero values are padded to the head and tail of the input in order to make the output has the same length with the input; a ReLU activation function is applied to the output.

3) *Hybrid CNN-LSTM-based HAR*: Recent research on HAR has proven that models consisting of CNN and LSTM can achieve better performance than single DL-based models [13]. Despite the power of LSTM in processing sequential data, LSTM-based HAR models cannot utilize parallel computing techniques as it has to process input sequence time step by time step. Furthermore, long sequences can easily lead to gradient vanishing and exploding problems during the training process with the use of backpropagation through time. Therefore, in the proposed hybrid CNN-LSTM-based HAR model, instead of applying LSTM directly to the window data  $\mathbf{W}$ , a 1D-CNN block is deployed to distill the input sequence, extract local temporal features and represent them in the depth dimension. Output feature map of this block has a size of  $(L' \times N')$ , where  $L'$  is the reduced length and  $N'$  is the new depth and equals to the number of kernels of the last convolutional layer. This new sequence of features is then fed into the LSTM layer for extracting long-term features. Finally, a group of dense layers with ReLU and softmax activation functions are used for further feature extraction and activity classification.



(a) iSPL IMU-Stretch Dataset



(b) w-HAR Dataset

Fig. 3. Window data distribution on different activities of the two datasets.

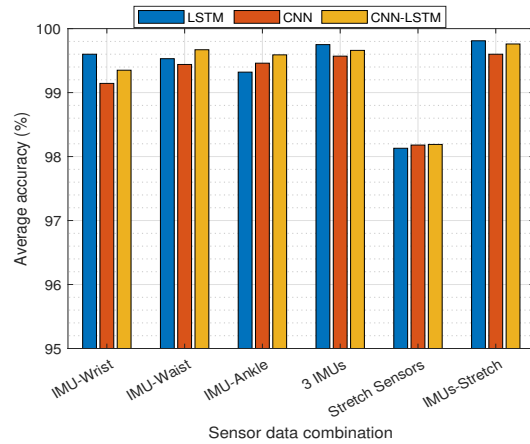
### III. EXPERIMENT RESULTS

#### A. Datasets

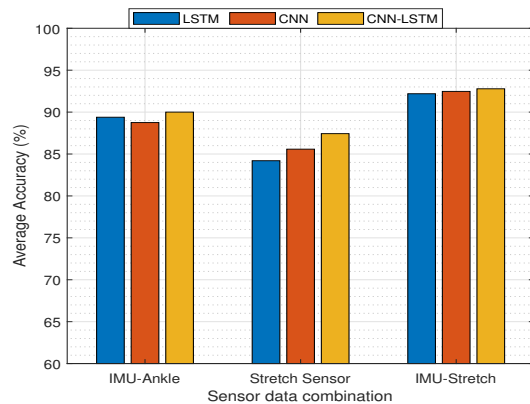
With the use of the proposed multi-modal HAR framework, we collected a dataset, namely iSPL IMU-Stretch, which contains data of 9 different daily-life activities and exercises: walking, standing, sitting, lying, running, jumping, sit-up, push-up and dancing. In addition, a public dataset called w-HAR [6] is also used for a comprehensive investigation on the two types of sensors. The w-HAR dataset was collected from an IMU attached at the ankle and a stretch sensor mounted at the knee. Eight activities including jumping, lying down, sitting, walking downstairs, walking upstairs, standing, walking and transition are collected from 22 users. Acceleration and angular velocity obtained from the IMU sensor are recorded at a sampling frequency of 250 Hz while stretch data is recorded at two frequencies of 100 Hz and 25 Hz. In this study, only the trials whose stretch sensor's sampling frequency is 25 Hz are used. The IMU sensor data are, then downsampled from 250 Hz to 25 Hz for reducing the computational cost. After the preprocessing step, there are 5724 and 3986 windows are obtained from iSPL IMU-Stretch and w-HAR datasets, respectively. The frequency distribution of all the activities in the two datasets is shown in Fig. 3.

#### B. Experiment Settings

The window data is randomly divided into training and testing sets with a proportion of 70% and 30%, respectively. All the deep learning models are implemented using the deep learning API Keras and are trained from scratch using the machine learning platform TensorFlow. The categorical cross-entropy loss function and adaptive moment optimizer (Adam) are used for the optimization process. For a fast convergence, the learning rate is initially set to 0.001 and reduced by half whenever the loss has stopped decreasing for 10 epochs. Each configuration (*i.e.*, model type and input data type) is trained and tested 10 times, and the average accuracy and F1 score are used as performance metrics. The iSPL IMU-Stretch dataset and implementation codes are available at [https://github.com/thunguyenth/HAR\\_IMU\\_Stretch](https://github.com/thunguyenth/HAR_IMU_Stretch).



(a) iSPL IMU-Stretch Dataset



(b) w-HAR Dataset

Fig. 4. Average classification accuracy of different deep learning models on different combinations of sensor data.

TABLE I  
F1 SCORE (%) OF DIFFERENT TYPES OF SENSOR DATA ON DIFFERENT ACTIVITIES USING HYBRID CNN-LSTM MODEL

	iSPL IMU-Stretch						w-HAR		
	IMU-Wrist	IMU-Waist	IMU-Ankle	3 IMUs	Stretch sensors	IMU-Stretches	IMU-Ankle	Stretch-Knee	IMU-Stretch
Walking	99.57	99.79	99.82	99.84	<b>99.29</b>	<b>99.93</b>	<b>94.33</b>	<b>95.03</b>	94.81
Standing	99.14	99.80	99.69	99.60	<b>97.42</b>	<b>99.82</b>	81.69	<b>72.55</b>	<b>91.55</b>
Sitting	<b>99.25</b>	99.72	99.68	99.50	<b>99.84</b>	99.64	<b>85.19</b>	90.15	<b>94.13</b>
Lying	99.92	99.96	99.98	99.95	<b>99.08</b>	<b>100</b>	<b>99.39</b>	<b>84.56</b>	99.30
Running	99.40	99.23	99.40	99.62	<b>98.63</b>	<b>99.67</b>	-	-	-
Jumping	99.73	<b>99.83</b>	<b>99.54</b>	99.77	99.60	99.80	93.90	<b>90.78</b>	<b>94.52</b>
Sit-up	99.42	99.53	<b>99.06</b>	99.18	<b>100</b>	99.36	-	-	-
Push-up	94.28	98.82	<b>99.40</b>	98.46	<b>52.36</b>	99.01	-	-	-
Dancing	98.52	99.18	98.71	99.45	<b>96.60</b>	<b>99.48</b>	-	-	-
Walking downstairs	-	-	-	-	-	-	88.98	<b>90.18</b>	<b>88.15</b>
Walking upstairs	-	-	-	-	-	-	<b>91.21</b>	<b>93.17</b>	91.23
Transition	-	-	-	-	-	-	<b>57.36</b>	<b>62.93</b>	57.65

\*Note: Values in bold and red color indicate the minimum F1 score of each activity in each dataset while values in bold and green color indicate the maximum values.

### C. Results and Discussion

Different combinations of sensor data are considered in both datasets: a) single IMU sensor (wrist, waist, ankle), multiple IMU sensors, multiple stretch sensors and combined IMU-stretch sensors in the iSPL IMU-stretch dataset; b) single IMU sensor (ankle), single stretch sensor and combined IMU-stretch sensors in w-HAR dataset. The average accuracy of three DL-based HAR models on these sensor data combinations are shown in Fig. 4.

In terms of deep learning models, the average accuracies are similar for the three models in the iSPL IMU-Stretch dataset. Overall, the hybrid CNN-LSTM model obtains the most stable performance in all the data combinations of the two datasets. Although the LSTM model achieves the highest accuracy in most of the IMU sensor data, especially in the iSPL IMU-Stretch dataset, it gets the worst performance in stretch sensor data with 3% of accuracy lower than the CNN-LSTM model in the w-HAR dataset. This instability of the LSTM model can be caused when the model is incapable of retaining long-term information in univariate time series (*i.e.*, single stretch sensor data in the w-HAR dataset). In contrast, in the hybrid CNN-LSTM model, since the univariate stretch sensor data is distilled and changed to multivariate time series by the CNN block, the LSTM layer is able to preserve the long-term dependencies.

In terms of sensor data, data collected from single IMUs have higher accuracy than the data of stretch sensor with a gap of 1.5% and 5% in the iSPL IMU-Stretch and w-HAR dataset, respectively when the LSTM-based HAR is used. This is because the stretch sensor provides only 1-dimensional data which is the stretching degree of the knees while the IMU sensor provides a wide range of information (*i.e.*, 3-axial acceleration and angular velocity). Combining data of IMU sensor and stretch sensor helps improve the system performance by approximately 0.5% and 2% of accuracy in the two datasets.

To have a detailed look at the sensor efficiency, F1 score of different sensor data combinations on different activities

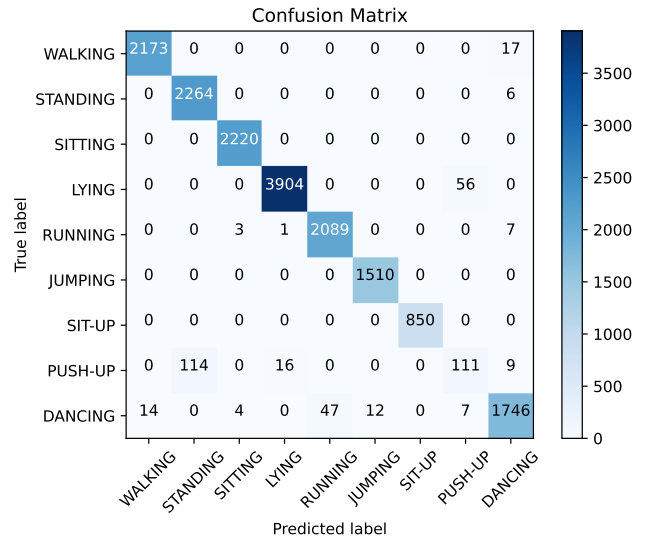


Fig. 5. Confusion matrix of the CNN-LSTM model on the stretch sensor of the iSPL IMU-Stretch dataset.

across all the datasets are computed and described in Table I. Some interesting findings have been found from this result. First, even stretch sensor has the lowest classification accuracy compared to IMU sensor in all three deep learning model according to the results in Fig. 4, it still achieves the highest F1 score values in sitting, sit-up (in the iSPL dataset) and walking, walking downstairs, walking upstairs and transition (in the w-HAR dataset). Thus, when being attached at the knee, stretch sensor has high potential in differentiating activities that have similar patterns from the lower limbs. Second, by only using a simple sensor fusion method, the combination of 3 IMUs in iSPL IMU-Stretch dataset does not gain any highest F1 score. Similarly, the data combination of all the sensors does not gain the highest F1 score at all activities in two datasets even though it has the most information compared to other data combinations. In particular, stretch sensor data in w-HAR dataset obtains the higher number of activity-wise F1-score

than the combination data. This finding suggests that by simply concatenating data of all the sensors might dull some important information that exists in the data of single sensors. Thus, this urges the need for developing a sensor fusion algorithm that can ignore noise data of a sensor but pay more attention to essential parts of other sensors. The synchronization among multiple sensors during data collection can also be a reason for this sensor fusion inefficiency.

Furthermore, problem of intra-class variance and inter-class similarity in human activity recognition is investigated. Firstly, the big gap in performance between the two datasets is the consequence of intra-class variance as the iSPL IMU-Stretch dataset is collected from one subject while the w-HAR is collected from 22 subjects. With the fact that each person has their own way to carry out an activity due to different factors such as age and height, data samples can be highly different from each other even though they are from the same activity. Secondly, a confusion matrix of the hybrid CNN-LSTM model on the stretch sensors of the iSPL IMU-Stretch dataset is demonstrated in Fig. 5 for inter-class similarity. It can be seen that push-up, standing, lying are easily misclassified to each other because the sensor can capture only the stretching and bending of the knees and the legs are stationary in a stretching state when these activities are carried out. Therefore, more research on tackling these intra-class variances and inter-class similarities in HAR is necessary.

#### IV. CONCLUSION

In this paper, several testing scenarios on different sensor data combinations and various deep learning-based models were performed to thoroughly study the effectiveness of IMU and stretch sensors in human activity recognition. With a wide range of sensor information, the multi-dimensional IMU sensor has outperformed the one-dimensional stretch sensor. However, the investigation results have also corroborated the potential of applying fabric, soft stretch sensors in capturing human motion. While this paper is a preliminary assessment of the practicality of soft, wearable sensors in human motion capture, we are expecting the future research in which multi-dimensional stretch sensors are developed and applied to promising applications of home healthcare such as diagnostic and therapeutics. Furthermore, on the basis of the promising findings presented in this paper, work on developing an optimal sensor fusion method is continuing and will be presented in future papers.

#### REFERENCES

[1] M. Bächlin, M. Plotnik, D. Roggen, I. Maidan, J. M. Hausdorff, N. Giladi, and G. Troster, "Wearable assistant for parkinson's disease patients with the freezing of gait symptom," *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 2, pp. 436–446, 2010.

[2] C. Chatzaki, M. Padiaditis, G. Vavoulas, and M. Tsiknakis, "Human daily activity and fall recognition using a smartphone's acceleration sensor," in *International Conference on Information and Communication Technologies for Ageing Well and e-Health*. Springer, 2016, pp. 100–118.

[3] J.-L. Reyes-Ortiz, L. Oneto, A. Samà, X. Parra, and D. Anguita, "Transition-aware human activity recognition using smartphones," *Neurocomputing*, vol. 171, pp. 754–767, 2016.

[4] Y. Ling, T. An, L. W. Yap, B. Zhu, S. Gong, and W. Cheng, "Disruptive, soft, wearable sensors," *Advanced Materials*, vol. 32, no. 18, p. 1904664, 2020.

[5] H. Chander, E. Stewart, D. Saucier, P. Nguyen, T. Luczak, J. E. Ball, A. C. Knight, B. K. Smith, R. Prabhu *et al.*, "Closing the wearable gap—part iii: use of stretch sensors in detecting ankle joint kinematics during unexpected and expected slip and trip perturbations," *Electronics*, vol. 8, no. 10, p. 1083, 2019.

[6] G. Bhat, N. Tran, H. Shill, and U. Y. Ogras, "w-HAR: An activity recognition dataset and framework using low-power wearable devices," *Sensors (Switzerland)*, vol. 20, no. 18, pp. 1–26, 2020.

[7] B. Klaassen, B. J. van Beijnum, M. Weusthof, D. Hofs, F. van Meulen, E. Droog, H. Luinge, L. Slot, A. Tognetti, F. Lorusi, R. Paradiso, J. Held, A. Luft, J. Reenalda, C. Nikamp, J. Buurke, H. Hermens, and P. Veltink, "A full body sensing system for monitoring stroke patients in a home environment," *Communications in Computer and Information Science*, vol. 511, pp. 378–393, 2015.

[8] R. A. Light, "Mosquito: server and client implementation of the mqtt protocol," *Journal of Open Source Software*, vol. 2, no. 13, p. 265, 2017.

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[10] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 37. PMLR, 2015, pp. 448–456.

[11] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ser. ICML'10. Madison, WI, USA: Omnipress, 2010, p. 807–814.

[12] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in Neural Information Processing Systems*, D. Touretzky, Ed., vol. 2. Morgan-Kaufmann, 1990, pp. 396–404.

[13] N. T. H. Thu and D. S. Han, "HiHAR: A hierarchical hybrid deep learning architecture for wearable sensor-based human activity recognition," *IEEE Access*, vol. 9, pp. 145 271–145 281, 2021.

# Comparative analysis of solar power generation prediction system using deep learning

So-yeong Kim  
Department of Information and Comm.  
Engineering  
Chosun University  
Gwangju, Republic of Korea  
kimsy9808@chosun.kr

Eun-ji Lee  
Department of Information and Comm.  
Engineering  
Chosun University  
Gwangju, Republic of Korea  
20175122@chosun.kr

Uttam Khatri  
Department of Information and Comm.  
Engineering  
Chosun University  
Gwangju, Republic of Korea  
uttamkhatri03@gmail.com

Seokjoo Shin  
Department of Computer Engineering  
Chosun University  
Gwangju, Republic of Korea  
sjshin@chosun.ac.kr

Ji-In Kim  
Department of Information and Comm.  
Engineering  
Chosun University  
Gwangju, Republic of Korea  
ji\_kim87@naver.com

Goo-rak kwon  
Department of Information and Comm.  
Engineering  
Chosun University  
Gwangju, Republic of Korea  
grkwon@chosun.ac.kr

**Abstract**— Due to the seriousness of environmental pollution, modern society is making efforts to switch from fossil fuel-centered energy to new and renewable energy worldwide. Among these new and renewable energies, solar energy is currently attracting attention due to its high growth potential. Among the renewable energy currently used, solar energy can generate a large amount of power without air pollutants and is widely used due to its high utilization. However, it is difficult to accurately predict the amount of power generation because solar energy, which is the source of energy from the sun, is greatly affected by seasons, weather, and installation environment. Therefore, this paper compares and analyzes the prediction and accuracy of solar power generation according to the environment through weather forecasts provided by the Meteorological Administration, using Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU) learning models.

**Keywords**—Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Solar energy, Power generation

## I. INTRODUCTION

Eco-friendly energy refers to energy that can be used so that the environment is no longer polluted by escaping from carbon energy in modern society. Since wind power, earth power, sunlight, and the like are used instead of carbon such as coal and petroleum, pollutants such as carbon dioxide are not emitted in the process of converting the natural phenomenon into thermal energy, and thus research has been steadily conducted. Fig. 1 is obtained from the data of 「Renewable energy generation (excluding non-renewable waste, from the 4th quarter of 2019)」 provided by the National Statistical Office [1]. The graph compares the amount of power generated by each energy source of renewable energy among the total power generation of 47,805,649 (MWh) of renewable energy in 2019. On the graph, the proportion of waste is the highest, but there is a disadvantage that air pollutants are generated when along with power generation using waste [2]. However, in the case of power generation using solar power, which has the next highest amount of power generation, except for the disadvantage that the initial cost is high, there are no air pollutants and a large amount of electricity can be obtained, so power generation using solar power is valuable as an alternative to fossil fuels. Therefore, in this paper, we intend

to compare and analyze the deep learning models that predict the amount of power generation according to the weather with accuracy so that solar power generation can be efficiently performed.

Subtotal power generation as of 2019  
(comprehensive business and private use)

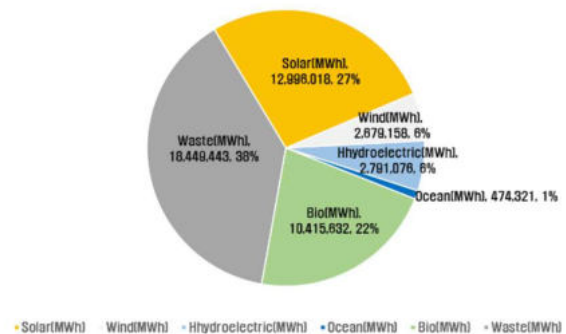


Fig. 1. Subtotal of power generation as of 2019 (Comprehensive for business and private use)

## II. DATASET

Considering the fact that the data values differ depending on the Korea Meteorological Administration, for accurate prediction and verification of power generation, Cheongju, Chungcheongbuk-do, Republic of Korea was used as an observation point, and the Korea Meteorological Administration's synoptic meteorological observation data and the Norwegian Meteorological Agency's data were used together.

### A. Training data

The training data used is the synoptic meteorological observation (ASOS) data provided by the Meteorological Data Open Portal of the Korea Meteorological Administration [3]. The observation point corresponds to Cheongju-si, Chungcheongbuk-do, and the observation period is from April 1, 2019 to September 1, 2021. The features used for training are 'Time', 'Temperature (°C)', 'Precipitation (mm)', 'Wind speed (m/s)', 'Humidity (%)', 'Dew point temperature (°C)', 'Amount of cloud cover (decile)' and 'Amount of power generation'.



### B. Validation data

For the validation data, forecast data provided by the Norwegian Meteorological Agency was used [4]. The observation point corresponds to Cheongju-si, Chungcheongbuk-do as the observation point of the training data, and the observation period is from October 22, 2021 to December 21, 2021. About 31 days of data recorded during the period were used, and the features used for validation are 'temperature', 'precipitation', 'wind speed', 'humidity', 'dew point', 'cloud cover', 'power generation' and 'time'.

### C. Output data

As the output data, the inverter data obtained through the monitoring system provided by MRT was used as the amount of power generation [5]. The location of the power plant corresponds to Cheongju-si, Chungcheongbuk-do, and the period is from April 1, 2019 to September 1, 2021. The total number of inverters is 11, and the total capacity of the inverters is 449.5kW. Features used for output data are 'accrue', 'start', 'day', and 'time'.

$$\text{The amount of time generated} = \text{accrue} - \text{start} - (\text{the sum of days before the relevant day}) \quad (1)$$

The formula for calculating the amount of time generated is equation (1), and the unit is W.

## III. DATA PROCESSING

### A. Training Data

In the ASOS weather observation data, partial Nan values existed for each column, and the data values before and after Nan values were compared and corrected. In the case of cloud cover, it was provided in the tenth quartile, so the training was conducted by multiplying it by 10. Tables I and II correspond to the data set observed in Cheongju-si, Chungcheongbuk-do.

### B. Validation Data.

In the forecast data of the Norwegian Meteorological Administration, wind direction and wind speed are provided in combination, so it was processed so that only the wind speed could be extracted separately. Tables III and IV correspond to the weather forecast data set corresponding to Cheongju-si, Chungcheongbuk-do.

TABLE I. WEATHER OBSERVATION DATA FOR ASOS IN CHEONGJU-SI, CHUNGCHEONGBUK-DO (1)

Date	Temp (°C)	Precipitation QC flag	Wind speed(m/s)
2019-04-01 0:00	2.7	9	0.4
2019-04-01 1:00	2.4		1
2019-04-01 2:00	2		1
2019-04-01 3:00	1.7		1.1
2019-04-01 4:00	1.3		0.9
2019-04-01 5:00	1.3		1.2
2019-04-01 6:00	1.1		0.8

TABLE II. WEATHER OBSERVATION DATA FOR ASOS IN CHEONGJU-SI, CHUNGCHEONGBUK-DO (2)

Humidity(%)	Dew point temp (°C)	Sunlight (hr)	Sunlight QC flag
-------------	---------------------	---------------	------------------

48	-7.2		9
51	-6.7		9
53	-6.5		9
56	-6.1		9
59	-5.8		9
62	-5.1		9
66	-4.5		9

TABLE III. WEATHER FORECAST DATA IN CHEONGJU-SI, CHUNGCHEONGBUK-DO (1)

Date	Time	Temp	Wind speed	Precipitation
2021-10-22 0:00	0	7		
2021-10-22 1:00	1	6		
2021-10-22 2:00	2	5		
2021-10-22 3:00	3	5		
2021-10-22 4:00	4	5		
2021-10-22 5:00	5	4		
2021-10-22 6:00	6	4		
2021-10-22 7:00	7	5		
2021-10-22 8:00	8	6		
2021-10-22 9:00	9	8		
2021-10-22 10:00	10	10		
2021-10-22 11:00	11	11		
2021-10-22 12:00	12	12		

TABLE IV. WEATHER FORECAST DATA IN CHEONGJU-SI, CHUNGCHEONGBUK-DO (2)

Wind direction	Humidity	Dew point	Cloud cover	Power generation
1 from north west1	67	1	0	0
1 from north1	64	0	0	0
1 from west1	68	0	0	0
1 from west1	70	0	0	0
1 from west1	69	-1	0	0
1 from west1	71	0	0	0
1 from west1	71	-1	0	0
1 from west1	68	-1	0	0
1 from north west1	68	0	0	10.6
1 from north west1	63	1	0	31.1
2 from north west2	59	2	0	119.8
3 from north west3	55	3	0	216.2
3 from north west3	53	3	0	291.2

### C. Output Data

As for the output data, the amount of data stored is flexible because the facility operates at the moment of power generation and has a large environmental influence.

The amount of power generation at the time = accumulated value of the time – accumulated value of the previous time (2)

The formula for calculating the amount of power generation at the time corresponds to equation (2), and when extracting data by time, if it is 0 o'clock, the accumulated value is initialized to 0W. The maximum power generation per hour of 1 inverter is 45,409W, and if the power generation exceeds 45,000W, it was estimated as an error value and performed by 11 inverters 1-11 respectively to delete the data on the relevant day. In addition, data on the day of partial omission due to the loss of communication at some time zones were deleted and processed for accurate prediction. Table V corresponds to a data set of 11 inverters installed in Cheongju-si, Chungcheongbuk-do.

TABLE V. INVERTER DATA IN CHEONGJU-SI, CHUNGCHAEONGBUK-DO

Name	Unit	Start	Day	Accrue	Time
Inverter1	60	4748406	267	4748673	2019-04-01 6:00
Inverter1	60	4748406	1853	4750259	2019-04-01 7:00
Inverter1	60	4748406	5770	4754176	2019-04-01 8:00
Inverter1	60	4748406	20239	4768645	2019-04-01 9:00
Inverter1	60	4748406	43732	4792138	2019-04-01 10:00
Inverter1	60	4748406	70598	4819004	2019-04-01 11:00
Inverter1	60	4748406	95883	4844289	2019-04-01 12:00
Inverter1	60	4748406	115707	4864113	2019-04-01 13:00
Inverter1	60	4748406	130818	4879224	2019-04-01 14:00
Inverter1	60	4748406	146957	4895363	2019-04-01 15:00
Inverter1	60	4748406	162571	4910977	2019-04-01 16:00

#### IV. METHODS

Long Short-Term Memory (LSTM), one of the deep neural network algorithms, has the specificity of Cell State, which remembers the previous state of the system being simulated. Before the input vector (processed information) is stored in the current state along with the old state according to the Timestamp, the current power generation of the solar power generation system can be predicted through the previous weather information [6]. On the other hand, the Gated Recurrent Unit (GRU) is a modified model of the LSTM in which the structure of the LSTM is more simply processed. The forgetting gate and input gate of LSTM are integrated into update data, and the cell state and hidden state are integrated into one to become a simpler structure than LSTM, resulting in faster learning due to reduced weight count, but almost the same performance as LSTM [7]. And for more accurate comparison, a keras density layer is added. After providing an input layer to the model, it is a model in which a dense layer is added by activating a function that is generated to solve Dying ReLU (the phenomenon in which neurons die) of ReLU called Laeky ReLU.

##### A. Long Short-Term Memory (LSTM)

Four LSTM layers followed by one dense layer were used as conditions for the long short-term memory (LSTM) learning model. As an activation function, leakyReLU, alpha value was 0.01, lose was mse, and as an optimizer, 'adam' was

used. Table VI corresponds to the LSTM learning model summary.

TABLE VI. LONG SHORT-TERM MEMORY LEARNING MODEL SUMMERY

Layer(type)	Output Shape	Param#
gru(GRU)	(None, 128)	52608
dense(Dense)	(None, 64)	8256
dense_1(Dense)	(None, 32)	2080
dense_2(Dense)	(None, 16)	528
dense_3(Dense)	(None, 1)	17

##### B. Gated Rucurrent Unit (GRU) Model

One GRU layer, three dense layers were used as conditions for the Gated Recurrent Unit (GRU) learning model. leakyReLU was used as the activation function, alpha value was 0.01, lose was mse, and 'adam' was used as the optimizer. Table VII corresponds to the GRU learning model summary.

TABLE VII. GATED RECURRENT UNIT LEARNING MODEL SUMMERY

Layer(type)	Output Shape	Param#
lstm(LSTM)	(None, 1, 32)	5120
lstm_1(LSTM)	(None, 1, 16)	3136
lstm_2(LSTM)	(None, 1, 8)	800
lstm_3(LSTM))	(None, 1, 4)	208
dense_4(Dense)	(None, 1, 1)	5

##### C. Dense Neural Network

Four Dense layers were used as conditions for the Dense Neural Network model. As an activation function, leakyReLU, alpha value was 0.01, lose was mse, and as an optimizer, 'adam' was used. Table VIII corresponds to the Dense Neural Network learning model summary.

TABLE VIII. DENSE NEURAL NETWORK LEARNING MODEL SUMMERY

Layer(type)	Output Shape	Param#
dense_5(Dense)	(1, 1, 64)	512
dense_6(Dense)	(1, 1, 32)	2080
dense_7(Dense)	(1, 1, 16)	528
dense_8(Dense)	(1, 1, 8)	136
dense_9(Dense)	(1, 1, 1)	9

#### V. RESULTS AND DISCUSSION

To obtain the error rate applied to the demonstration system, the actual generation was subtracted from the predicted generation and divided by the installed capacity. Table I corresponds to the results when each model was trained. When Timestamp is 1 and Epoch is 50, the training loss values are 953 for LSTM, 974 for GRU, and 1003 for Dense Neural Network. The validation loss values were 2244 for LSTM, 2389 for GRU, and 2692 for Dense Neural Network. In terms of the error rate applied to the demonstration system, LSTM was 13.01%, GRU was 13.67%, and Dense Neural Network was 14.23%. Among

them, LSTM showed the highest efficiency. When Timestamp is 24 and Epoch is 200, the training loss value is 31 for LSTM and 985 for Dense Neural Network. The validation loss values are 3769 for LSTM and 2174 for Dense Neural Network. In terms of the error rate applied to the demonstration system, the efficiency of the Dense Neural Network was higher, with LSTM being 17.03% and Dense Neural Network 13.03% as shown in TABLE IX.

TABLE IX. LONG SHORT-TERM MEMORY, GATED RECURRENT UNIT, DENSE NEURAL NETWORK LEARNING MODEL COMPARISON

Table Head	LSTM		GRU	Dense Neural Network	
	Timestamp : 1	Timestamp : 24	Timestamp : 1	Timestamp : 1	Timestamp : 24
	Epoch : 50	Epoch : 200	Epoch : 50	Epoch : 50	Epoch : 200
Training Loss Value	953	31	974	1003	985
Validation Loss Value	2244	3769	2389	2692	2174
Error rate when applying the demonstration system	13.01%	17.03%	13.67%	14.23%	13.03%

## VI. CONCLUSION

Since the establishment of a solar energy generation prediction system is a key role in efficiently managing and distributing energy, many studies are being conducted. Therefore, accurate prediction of power generation for various variables such as season, weather, and installation environment is essential for solar energy generation. In this paper, we used ASOS data of Cheongju-si, Chungcheongbuk-do as training data, forecast data of Cheongju-si, Chungcheongbuk-do as validation data, and generation data of Cheongju-si, Chungcheongbuk-do as output data. It was shown that the amount of power generation can be predicted using LSTM (Long Short-Term Memory), GRU (Gated Recurrent Unit), and Dense Neural Network as the learning model algorithm. As a result, it was confirmed that the error rates of LSTM, GRU, and Dense Neural Network vary according to the detailed conditions of Timestamp and Epoch. When Timestamp is 1 and Epoch is 50, the efficiency of LSTM is the highest, and when Timestamp is 24 and Epoch is 200, the Dense Neural Network shows high efficiency.

## ACKNOWLEDGEMENT

This work was supported by the ICT R&D program of MSIP/IITP. [No. 2021-0-01264, AI-based photovoltaic system with real-time prediction of failure and generation amount]

## REFERENCES

- [1] Korea Energy Agency, "New and renewable energy supply performance survey," [https://kosis.kr/statHtml/statHtml.do?orgId=337&tblId=DT\\_337N\\_A002](https://kosis.kr/statHtml/statHtml.do?orgId=337&tblId=DT_337N_A002).
- [2] S. K. Kim, S. B. Lee, J. M. Park, C. Yu, J. H. Hong, H. C. Kim, N. E. Jung, M. R. Jo, J. H. Kim, and S. H. Heo, "A Study of Air Pollutants Reduction Plan for Waste Solid Fuel Fired Facilities," *Journal of Korean Society for Atmospheric Environment*, vol. 28, Issue 6, pp. 656-668, 2012.
- [3] Meteorological Administration Weather Data Open Portal Jonggwan Weather Observation, <https://data.kma.go.kr/data/grnd/selectAsosRltmList.do?pgmNo=36>.
- [4] Norwegian Meteorological Administration forecast data, <https://www.yr.no/en/details/table/28691116/Republic%20of%20Korea/North%20Chungcheong/Miwon-myeon>.
- [5] MRT co. Ltd., <http://www.mrt.co.kr/>.
- [6] Dongkyun Kim and Seokgu Kang. "Data Collection Method for Construction of Precipitation-Daily Runoff Estimation LSTM Model" *Proceedings of the Korean Society for Water Resources* 54, no.10, pp. 795-805, 2021.
- [7] Tamal Datta Chaudhuri and Indranil Ghosh, "Artificial neural network and time series modeling based approach to forecasting the exchange rate in a multivariate framework," *Journal of Insurance and Financial Management*, vol. 1, no. 5, 2016.

# Multi-head CNN and LSTM with Attention for User Status Estimation from Biometric Information

Hyunseo Park

*School of Electrical Engineering*  
*Korea Advanced Institute of Science and Technology*  
Daejeon, Rep. of Korea, 34141  
tkf92001@kaist.ac.kr

Gyeong Ho Lee

*School of Electrical Engineering*  
*Korea Advanced Institute of Science and Technology*  
Daejeon, Rep. of Korea, 34141  
gyeongho@kaist.ac.kr

Hyeontaek Oh

*Institute for Information Technology Convergence*  
*Korea Advanced Institute of Science and Technology*  
Daejeon, Rep. of Korea, 34141  
hyeontaek@kaist.ac.kr

Nakyoung Kim

*Institute for Information Technology Convergence*  
*Korea Advanced Institute of Science and Technology*  
Daejeon, Rep. of Korea, 34141  
nkim71@kaist.ac.kr

Jaeseob Han

*School of Electrical Engineering*  
*Korea Advanced Institute of Science and Technology*  
Daejeon, Rep. of Korea, 34141  
j89449@kaist.ac.kr

Jun Kyun Choi

*School of Electrical Engineering*  
*Korea Advanced Institute of Science and Technology*  
Daejeon, Rep. of Korea, 34141  
jkchoi59@kaist.ac.kr

**Abstract**—With Internet of Things technologies, healthcare services for smart homes are emerging. In the meantime, the number of households of single-living elderly who are distant from using smart devices is increasing, and contactless radar-based sensors are recently introduced to monitor the users in single households. In this paper, contactless radar-based sensors were installed in over 100 households of single-living elderly to collect their biometric data under uncontrolled environments. In addition, a deep learning-based classification model is proposed that estimates the user status in predefined classes. In particular, the classification model is designed with a multi-head convolutional neural network with long-short-term memory and an attention mechanism. The proposed model aims to extract features in diverse resolutions from the biometric data while capturing the temporal causalities and relative importance of the features. The experimental results verify that the proposed classification model improves the status classification accuracy by 2.8% to 31.7% in terms of  $F_1$  score for the real-world dataset.

**Index Terms**—Radar-based status monitoring, status classification, multi-head feature extraction

## I. INTRODUCTION

With the recent advancements of Internet of Things (IoT) technologies, various smart home applications are developed. A smart home means a residence where a household can benefit from smart services with remote access, monitoring, and control capabilities [1]. Meanwhile, the number of single households is anticipated to rise worldwide, with a significant

This research was financially supported by the Ministry of Trade, Industry and Energy (MOTIE) and Korea Institute for Advancement of Technology (KIAT) through the International Cooperative R&D program. (Project No. 0011879)

increase in the ratio of the elderly in the next two decades [2]. This growing aging population could cause a weakening of the socio-economic structure of many countries in terms of healthcare and related costs [3]. Accordingly, many efforts to develop the solution for healthcare services for the elderly in single households (e.g., monitoring of underlying diseases, detecting emergencies, etc.) are put together.

On-body and contactless sensors have been considered as the two main approaches for healthcare services to collect and monitor users' biometric data [4]. The on-body method uses wearable devices for service provision, and the sensors need to be placed on the user's body. Hence, the biometric data collected from the on-body sensors are relatively accurate. On the other hand, the user requires to continuously wear the sensor device and frequently charge the device, which raise inconvenience especially for the high age group who are often distant from using smart devices [5]. Accordingly, when it comes to the status monitoring in the users' living space, the methods to monitor the users' status from the data collected with contactless sensors are widely investigated for service provision without affecting the users' daily life [6].

As the sensor technologies are getting mature than ever, contactless radar-based data acquisition methods are recently investigated that use the phase change of the radar reflected by the movement of the human body. As a consequence, an individual's biometric features (e.g., heart rate, respiration, etc.) can now be accurately obtained without contact, and the development of analysis on the collected biometric data is in progress [7]–[10]. In this paper, a method to estimate a user's

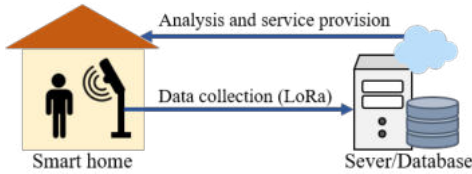


Fig. 1. High level radar-based biometric monitoring system model

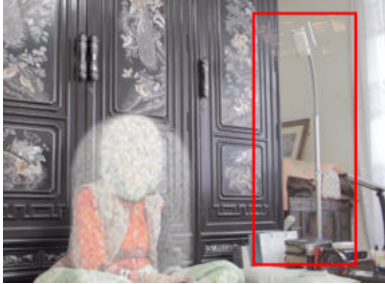


Fig. 2. Illustration of the installed radar sensor

TABLE I  
SPECIFICATIONS OF THE INSTALLED RADAR SENSOR

Category	Specification
Chip	Sharp DC6M4JN3000
Method	Microwaves (24.05 24.25GHz)
Resolution	60cm
Range (Max.)	1.5m (Heartbeats, Breathing) 7m (Body motion)
Directionality	Azimuth: 25°, Elevation: 20°
Error rate	±10% ( 3m)

status from the biometric information collected through a radar sensor is investigated. While radar sensors have a limitation that the accuracy of the measurement drops when the subject is intensely moving, this paper aims for the situations when the displacement and the motion of the subject are bound to some degree (e.g., sleeping in the bed, having a meal on the table, watching television, etc.) In the meantime, the time-series data on human activity collected from sensors can be decomposed into a set of features in multiple levels (i.e., low, middle, and high), and the means to capture the feature in diverse resolution, as well as their temporal causalities, are needed for the user status estimation from biometric data. Accordingly, a deep learning-based model is proposed in this paper that analyzes a fixed length of biometric data and classifies the user status in that period. The contactless radar-based sensors were installed in the living spaces of over 100 single households of elders, and real-world biometric information is collected. In particular, this paper utilizes the biometric information of 22 elders collected under uncontrolled environments and proposes a deep learning-based user status classification model.

The rest of this paper is organized as follows. Section II provides previous studies. Section III-B introduces the status classification method. The performance evaluation is presented in Section IV, followed by the conclusion in Section V.

TABLE II  
SAMPLE DISTRIBUTION OF THE HR DATASET

Class	Status	Number of data samples	Ratio
1	Not detected	32,790	33.6%
2	In sleep	45,148	46.3%
3	In active activity	5,762	5.9%
4	In stationary Activity	6,460	6.6%
5	In unidentified activity	7,405	7.6%
Total		97,565	100%

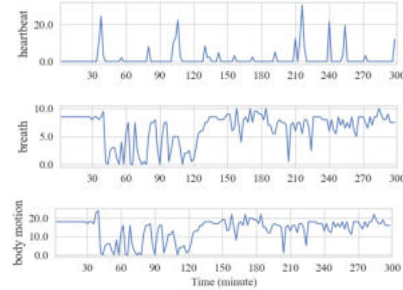


Fig. 3. Example of the biometric data of a subject in sleep

## II. RELATED WORKS

The research on biometric information monitoring technologies with contactless radar-based sensors are largely in two fields of study. One is on increasing the accuracy of the contactless monitoring sensors' measurements compared to on-body sensors. The other is on improving the performance of biometric data analysis to use them for applications and services. With the recent advancements in sensor technology, the second field of study that focuses on biometric data analysis is actively in progress. In the meantime, research on radar-based cardio-respiration monitoring is widely investigated, which is mainly conducted in an impulse-radio ultra wide band (IR-UWB) radar-based environment [7]. This research includes studies that propose a method to accurately measure of the heart rate, to reliably monitor the continuous cardio-respiration rate, to estimate the sleep stages, etc. [10]–[12]. In particular, many studies have applied deep learning technologies to enhance the accuracy of the analysis on biometric data. Long-short-term memory (LSTM) with an attention mechanism is used in [12] to analyze the heart and respiratory rates collected by an IR-UWB radar-based device. In [13] and [14], electrocardiogram signals are analyzed respectively based on a multi-head and residual convolutional neural network (CNN). In the meantime, the well-known deep learning technologies are applied to a set of publicly available biometric datasets, and their performances are compared and evaluated in [15]. However, to the best knowledge of the authors, there has not been a work that comprehensively utilizes multi-head CNN with LSTM and an attention mechanism to enhance the performance of the user's status classification. Accordingly, this study proposes a deep learning-based user status classification model designed with multi-head CNNs, LSTMs, and a multi-head self-attention mechanism to integrally capture the contexts in diverse feature levels as well as their temporal causalities and importance.

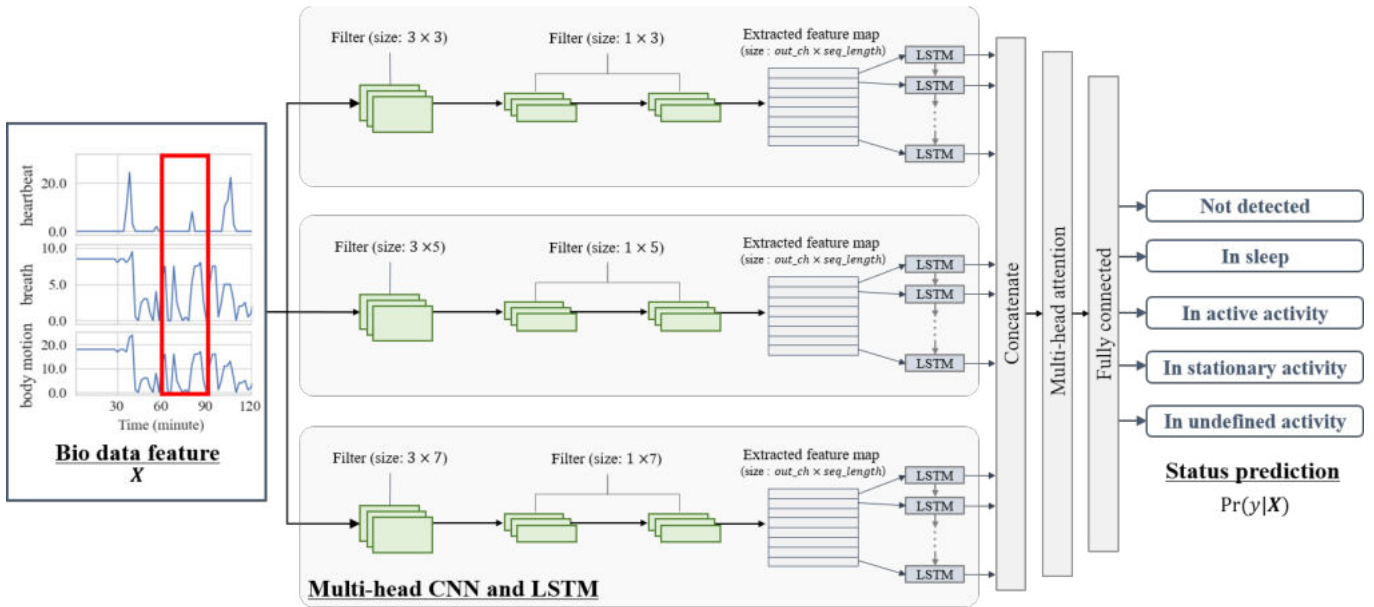


Fig. 4. Proposed deep learning-based user status estimation model with a multi-head CNN and attention mechanism

### III. STATUS CLASSIFICATION FROM BIOMETRIC FEATURE

In this section, a deep learning-based status classification model is proposed that extracts and analyzes the multi-level features from biometric data. The proposed model classifies the status of a subject in one of the predefined status classes: 1) not detected, 2) in sleep, 3) in an active activity, 4) in a stationary activity, and 5) in an undefined activity. In advance of introducing the proposed classification model, we provide details on the dataset.

#### A. Dataset Generation and Data Preprocessing

In each living space of over 100 individual single households, a radar-based sensor with the specifications given in Table I was installed as described in Fig. 1, and the sensor measures the heart rate, respiratory rate, and body motion every 2 minutes. The biometric data were collected under uncontrolled environments at different times for each individual from 2020 to 2021. A part of the collected dataset is illustrated in Fig. 3 as an example. Only the data that are reliably collected without a network failure for a reasonable period are used for this study. In addition, whereas this study utilizes only heart rate, respiratory rate, and body motion data, the method proposed in this paper can be easily replaced with other types of biometric features.

The biometric data are labeled by experts as one of the five predefined statuses of the subject, such that the subject is 1) not detected, 2) in sleep, 3) in an active activity, 4) in a stationary activity, and 5) in an undefined activity. Min-max normalization is applied to each dimension of the biometric features. The continuous time-series data are then segmented into fully non-overlapping subsequences in the length of 30 minutes. Overlapping subsequences can be used in a practical system, but the non-overlapping subsequences are used in this

paper to prevent the training, validation, and test dataset for assessment from containing partially overlapping information. The most frequently detected status in the period of the segmented subsequence is assigned as the final status of the segmented subsequence. The distribution of the generated dataset is described in Table II.

#### B. Status Classification with Multi-head CNN

The objective of the user status classification problem is defined as

$$\Pr(y | \mathbf{X}), \quad (1)$$

where  $y$  denotes the status class, and  $\mathbf{X}$  is the input sequence of biometric features. That is, the status classification model aims to estimate the status of a user from the observed biometric data. Considering the biometric data  $\mathbf{X}$  as a multivariate time series, it can be decomposed into low and high-level features like the other types of data, such as signals and images. The time-series data about human activities are often perceived as a combination of actions in low and high levels, and the extraction of features in diverse levels, therefore, can facilitate the status classification.

To extract the features from the time-series data, CNNs have been widely used in various types of research around biometric data analysis. However, vanilla CNNs have shown their limitations as they capture the features at a defined resolution. In this paper, we thereby utilize CNNs in a multi-head structure to identify the features in diverse levels from the biometric data. To additionally capture the temporal relations of the features, LSTMs are applied afterward to each of the feature maps extracted by the multi-head CNN. In addition, a multi-head self-attention mechanism is used as well to differently weigh the importance of the extracted features. The

TABLE III  
CONFUSION MATRIX FOR ERROR MEASUREMENTS

Actual \ Predicted	Positive	Negative
	Positive	True positive (TP)
Negative	False positive (FP)	True negative (TN)

self-attention mechanism computes attention weights for each hidden unit of the LSTM from all CNN heads to generate context vectors as a weighted sum of the hidden units.

The proposed classification model is illustrated in Fig. 4. A 30-minute long multidimensional biometric data consisting of heart rate, respiratory rate, and body motion information is fed into each head of the multi-head CNN designed with different filter sizes. Each head of the multi-head CNNs extracts the feature maps in different resolutions, and LSTMs additionally insert the information about temporal causalities into the captured features. The extracted features are differently scaled based on the attention weights computed by the multi-head attention mechanism, which is then mapped into the probability of each status class through a fully connected layer with softmax.

#### IV. PERFORMANCE EVALUATION

In this section, the performance of the proposed status classification model is evaluated in terms of its classification accuracy.

##### A. Evaluation Metric

The performance of the proposed model is evaluated in accuracy and  $F_1$  score of the classification results, which are computed based on the confusion matrix illustrated in Table III. The accuracy and  $F_1$  score are given by

$$\text{Accuracy} = \frac{\sum_{i=1}^N \text{TP}_i}{\sum_{i=1}^N \text{TP}_i + \sum_{i=1}^N \text{FP}_i}, \quad (2)$$

and

$$F_1 \text{ score} = \frac{1}{N} \sum_{i=1}^N \frac{\text{TP}_i}{\text{TP}_i + \frac{1}{2}(\text{FP}_i + \text{FN}_i)}, \quad (3)$$

where  $\text{TP}_i$ ,  $\text{TN}_i$ ,  $\text{FP}_i$ , and  $\text{FN}_i$  are the numbers of the true positive, true negative, false positive, and false negative instances for the  $i$ -th class from  $N$  classes. The ranges of both accuracy and  $F_1$  score are  $[0, 1]$ , where a value closer to 1 implies better classification performance. While accuracy is the most well-known metric for classification problems over various domains, the macro  $F_1$  score is widely used to evaluate the performance of a classifier on a dataset with imbalanced class samples like the generated dataset.

TABLE IV  
 $F_1$  SCORE OF THE STATUS CLASSES (%)

Model	Accuracy	$F_1$ score
Gaussian Naive Bayes	68.0	<b>55.8</b>
k Nearest Neighbor	78.8	51.6
Support Vector Machine	82.8	52.6
Random Forest	<b>83.3</b>	54.9
Multi-layered Perceptrons (16-8)	<b>85.6</b>	<b>65.8</b>
Multi-layered Perceptrons (16-16-8)	85.3	65.8
Multi-layered Perceptrons (16-16-16-8)	85.2	62.4
Single-head CNN and LSTM	85.7	69.1
Single-head CNN and LSTM with attention	<b>86.7</b>	<b>71.5</b>
Multi-head CNN and LSTM	86.4	71.5
Multi-head CNN and LSTM with attention	<b>87.1</b>	<b>73.5</b>

##### B. Experimental settings and results

For performance evaluation of the proposed classification model, experiments on the methods with conventional machine learning and deep learning techniques are conducted. For conventional machine learning techniques, Gaussian naïve Bayes (NB), k-nearest neighbor (kNN), support vector machine (SVM), and random forest (RF) are applied as the benchmarks of the performance comparison. The conventional machine learning methods are implemented with the scikit-learn library in python. For baselines, multi-layered perceptions (MLP) with different numbers of layers and nodes are investigated. In addition, the models composed of CNNs followed by LSTMs with and without an attention mechanism are used for more advanced baselines. A multi-head self-attention mechanism is used for advanced baselines, and the multi-head CNN with LSTM are respectively implemented with 3 layers of CNNs with the convolutional filter sizes of (3, 5, 7) and 2 layers of bidirectional LSTMs with the hidden size of 16. For single-head CNN models, the 4 layers of CNN is used with the convolutional filter size of 5 for all layers. A fully connected layer is added before the softmax layer for all CNN and LSTM-based models. The deep learning models are implemented with TensorFlow and Keras in python. All deep learning models are trained using Adam optimizer with a learning rate 0.001 until their losses are converged with the learning rate decay and early stopping. For training, validation, and test, 60%, 20%, and 20% of instances randomly selected from the generated dataset are used, respectively. For a fair assessment, 10 independent trials of experiments are conducted, and the classification accuracy of all trials are averaged.

The experimental results are provided in Table IV. The results show that the deep learning approaches greatly outperform the conventional machine learning approaches in terms of  $F_1$  score while there are not remarkable improvements in terms of accuracy. This implies that the deep learning approaches are more robust to the imbalanced dataset compared to the conventional machine learning approaches. In the meantime, the deep learning-based classification models tend to be easily overfitted when the fully connected layers are heavily stacked because the input feature of the biometric data is relatively simple in its contexts and small in its size compared to

datasets in the other domains. However, the deep CNN and LSTM models show improved performance compared to the MLP models as the CNN and LSTM facilitate more advanced feature extraction before the last fully connected layer. In the same manner, the proposed model composed of multi-head CNN and LSTM with an attention mechanism outperforms all the other models with 2.8% to 31.7% improvements in  $F_1$  score as it successfully extracts the contexts in diverse levels and their temporal causalities with proper importance weights. In summary, the experimental results on the empirically generated dataset show that the proposed status classification model achieves performance improvements compared to all benchmark and baseline models both in terms of accuracy and  $F_1$  score.

## V. CONCLUSION

In this paper, the biometric data (i.e., heart rate, respiratory rate, and motion volume) collected in the living spaces of elders under uncontrolled environments are investigated, and a classification model is proposed that estimates a user's status in one of the five predefined classes (i.e., not detected, in sleep, in an active activity, in a stationary activity, and in an undefined activity). In particular, the status classification model is designed to support the extraction of features in diverse resolutions and their temporal causalities with importance weights. The experimental results show that the proposed model enhances the status classification performance by up to 31.7% in  $F_1$  score and 4.6% in accuracy. As the results on the empirical data collected under uncontrolled environments verify that the proposed model achieves noticeable improvements in  $F_1$  score, the effectiveness and practicality of the proposed model are shown. In the meantime, to improve the overall classification accuracy for effective service provision, the issue of imbalanced class in the training dataset requires to be comprehensively managed in the future work of this study. In addition, to consider the scalability of the proposed status classification system, automatic labeling on the generated data also needs to be considered while handling the privacy concerns as well.

## ACKNOWLEDGMENT

The authors of this paper thank Seung Chul Kim from KULS, João Garcia from Ubiwhere, and Ricardo Gonçalves from PROEF for co-working to develop the dataset and prototype for the proposed system as partners of a research project (KIAT, Project No. 0011879).

## REFERENCES

- [1] N. Balta-Ozkan, R. Davidson, M. Bicket, and L. Whitmarsh, "Social barriers to the adoption of smart homes," *Energy Policy*, vol. 63, pp. 363–374, 2013.
- [2] Organisation for Economic Co-operation and Development (OECD). The Future of Families to 2030. Accessed: Feb., 2020. [Online]. Available: <https://doi.org/10.1787/9789264168367-en>
- [3] S. Majumder, E. Aghayi, M. Nofaresti, H. Memarzadeh-Tehran, T. Mondal, Z. Pang, and M. J. Deen, "Smart homes for elderly health-care—recent advances and research challenges," *Sensors*, vol. 17, no. 11, 2017.

- [4] H. Mshali, T. Lemlouma, M. Moloney, and D. Magoni, "A survey on health monitoring systems for health smart homes," *International Journal of Industrial Ergonomics*, vol. 66, pp. 26–56, 2018.
- [5] J. Kim, "How much do we know about the use of smartphones in the silver generation?: Determinants of the digital divide within the silver generation," *Information Society & Media*, vol. 21, no. 3, pp. 33–64, 2020.
- [6] M. Bahache, J. P. Lemayian, W. Wang, and J. Hamareh, "An inclusive survey of contactless wireless sensing: A technology used for remotely monitoring vital signs has the potential to combating covid-19," 2020.
- [7] J. Kranjec, S. Beguš, G. Geršak, and J. Drnovšek, "Non-contact heart rate and heart rate variability measurements: A review," *Biomedical Signal Processing and Control*, vol. 13, pp. 102–112, 2014.
- [8] A. Ni, A. Azarang, and N. Kehtarnavaz, "A Review of Deep Learning-Based Contactless Heart Rate Measurement Methods," *Sensors*, vol. 21, no. 11, 2021.
- [9] F. Zhang, C. Wu, B. Wang, M. Wu, D. Bugos, H. Zhang, and K. J. R. Liu, "Smars: Sleep monitoring via ambient radio signals," *IEEE Transactions on Mobile Computing*, vol. 20, no. 1, pp. 217–231, 2021.
- [10] Y. Lee, J. Y. Park, Y. W. Choi, H. K. Park, S. H. Cho, S. H. Cho, and Y. H. Lim, "A Novel Non-contact Heart Rate Monitor Using Impulse-Radio Ultra-Wideband (IR-UWB) Radar Technology," *Scientific Reports*, vol. 8, no. 13053, 2018.
- [11] C. H. Chang and W. W. Hu, "Design and implementation of an embedded cardiorespiratory monitoring system for wheelchair users," *IEEE Embedded Systems Letters*, vol. 13, no. 4, pp. 150–153, 2021.
- [12] H. B. Kwon, S. H. Choi, D. Lee, D. Son, H. Yoon, M. H. Lee, Y. J. Lee, and K. S. Park, "Attention-based lstm for non-contact sleep stage classification using ir-uwband radar," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 10, pp. 3844–3853, 2021.
- [13] M. Kachuee, S. Fazeli, and M. Sarrafzadeh, "Ecg heartbeat classification: A deep transferable representation," in *2018 IEEE International Conference on Healthcare Informatics (ICHI)*, 2018, pp. 443–444.
- [14] N. Ahmed, A. Singh, S. K. S., G. Kumar, G. Parchani, and V. Saran, "Classification Of Sleep-Wake State In A Ballistocardiogram System Based On Deep Learning," 2020.
- [15] S. Purushotham, C. Meng, Z. Che, and Y. Liu, "Benchmarking deep learning models on large healthcare datasets," *Journal of Biomedical Informatics*, vol. 83, pp. 112–134, 2018.



# An Explainable Computer Vision in Histopathology: Techniques for Interpreting Black Box Model

Subrata Bhattacharjee  
Dept. of Computer Engineering  
Inje University  
Gimhae, Republic of Korea  
subrata\_bhattacharjee@outlook.com

Yeong-Byn Hwang  
Dept. of Digital Anti-Aging Healthcare  
Inje University  
Gimhae, Republic of Korea  
hyb1345679@gmail.com

Kobiljon Ikromjanov  
Dept. of Digital Anti-Aging Healthcare  
Inje University  
Gimhae, Republic of Korea  
kobiljonikromjanov@gmail.com

Rashadul Islam Sumon  
Dept. of Digital Anti-Aging Healthcare  
Inje University  
Gimhae, Republic of Korea  
Sumon39.cst@gmail.com

Hee-Cheol Kim  
Dept. of Digital Anti-Aging Healthcare  
Inje University  
Gimhae, Republic of Korea  
heeki@inje.ac.kr

Heung-Kook Choi  
Department of Computer Engineering  
Inje University  
Gimhae, Republic of Korea  
cschk@inje.ac.kr

**Abstract**—Computer vision is a field of artificial intelligence (AI) that is being used increasingly in histopathology to identify pathologies in slide images with a high degree of accuracy. In this paper, we focus on the different interpreting techniques of explainable computer vision (XCV). Analysis of histopathology images is a challenging task, and specialized knowledge is mandatory to make AI decisions. To carry out this analysis, a deep learning model has been used to classify and differentiate the scoring (i.e., benign and malignant) of Prostate cancer (PCa). However, the AI models are complex and opaque, and it is important to understand model decision-making. Therefore, to address this problem, we present three techniques for accountability and transparency of the model, namely Activation Layer Visualization (ALV), Local Interpretable Model-Agnostic Explanation (LIME), SHapley Additive exPlanations (SHAP), and Gradient-weighted Class Activation Mapping (Grad-CAM). XCV is AI in which the results of the black-box model can be understood by humans. The robustness of our model has been confirmed by using an external test dataset including 100 histopathology images. The model performance has been evaluated using the receiver operating characteristic (ROC) curve.

**Keywords**— explainable computer vision, histopathology, artificial intelligence, black box, prostate cancer

## I. INTRODUCTION

The analysis of histopathology images is a gold standard for the detection of different types of cancer regions and performs diagnosis using AI algorithms [1, 2]. Histopathology study is carried out under the microscope for disease diagnosis. Hematoxylin and Eosin (H&E) staining is used routinely in histopathology laboratories to analyze different types of cells and tissue and provides important information about the pattern, shape, cell structure in a tissue sample [3, 4]. Also, H&E dyes make it easier for pathologists to see different parts of the cell under a microscope Hematoxylin has a deep blue-purple color which shows the ribosomes, chromatin within the nucleus. In contrast, Eosin has an orange-pink-red color which shows the cytoplasm, cell wall, collagen, connective tissue, and other structures that surround and support the cell. Image classification and segmentation are two basic tasks in digital histopathology. Image classification is carried out by categorizing and labeling groups of pixels within an image [5]. In this study, the image classification task was carried out using PCa tissue samples, and it is a type of cancer that has always been an important challenge for pathologists. For manual diagnosis of PCa, expert pathologists

need more attention to analyze the tissue pattern, structure of cells, and glands under a microscope, which is time-consuming. However, to make the work easier for pathologists, many researchers are developing different types of computer-aided diagnosis (CAD) systems that can make decisions automatically.

In recent years, AI algorithms have shown tremendous performance in different kinds of applications, especially in medical health. It has been used in many fields as exemplified by computer vision and is well-recognized for image classification [6]. Recently, the activation features of convolution neural networks (CNN) have achieved splendid triumphs in computer vision [7-9]. XCV is AI in which the results of the black-box model can be understood by humans. Nowadays, AI systems and machine learning (ML) algorithms are widespread in many areas. Data is used almost everywhere to solve problems and help humans, a large factor for this success is the progress in the DL area, but also generally the development of new and creative ways how we can use data.

As a consequence, the complexity of these systems becomes incomprehensible even for AI experts. Therefore, the models are usually also referred to as black boxes. The meaning of “black-box” is that it is generally difficult to clearly explain the decisions made by the models [10]. Explainability and interpretability are very important in medical areas because the CAD system needs to be transparent and understandable to gain the trust of doctors and patients. The procedures and methods that allow human users to understand and trust the results and output created by the models are called XCV [11]. Fig. 1 shows a schematic representation of XCV.

In this paper, we introduce a Light-Dense CNN (LDCNN) model for histopathology image classification. The model consists of multiple layers, including the input layer, convolutional layers, concatenation layers, dropout layers, and classification layer. This model has been modified from the light-weight CNN (LWCNN) architecture which was introduced in our previous study [12]. Also, we have explained the processes and outputs of the supervised model so that it is understandable for other readers. The ALV, LIME [13], SHAP [14, 15], and Grad-CAM [16] techniques were used to generate the activated feature maps and visualize the model’s decisions which produce a coarse localization map of the important region in the image, thus interpreting the decision of the neural network.

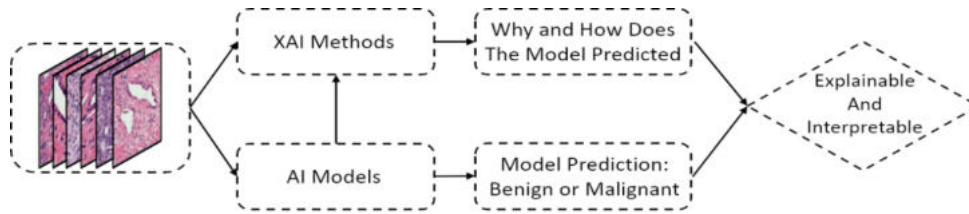


Fig. 1. A brief schematic representation of explainable computer vision

The contributions of this paper are summarized as follows:

- The binary classification was performed successfully using the LDCNN model.
- Experiments were conducted using PCa histopathology images with different magnifying factors (i.e., 20× and 40×).
- Two types of datasets were used to perform the experiments: a public dataset (i.e., PANDA Challenge) and a private dataset.
- The XCV methods are used to visualize the results of the black-box model, which include ALV, SHAP, and Grad-CAM.

The remainder of this article is structured as follows. In Section II, we described the related work and recent studies about images classification and XAI. Section III illustrates the complete methodology of this study, which includes data acquisition, model development, and XCV methods. Results of the AI models are presented in Section IV. Section V discusses the paper and lastly, the paper is concluded in Section VI.

## II. RELATED WORK

Research on computer vision for histopathology image analysis provided valuable findings regarding the problems of automatic detection and classifying PCa tissue images. In [17], they developed a patch-based classifier using CNN for the automated classification of histopathology images. Their proposed method achieved promising results for both binary and multiclass classification. In [18], they developed a dual-channel residual convolution neural network to classify the histopathology images of the lymph node section. They performed binary classification to discriminate between cancerous from noncancerous tumors. In [19], a novel method was proposed for histopathological image classification of colorectal cancer. They developed a novel bilinear convolution neural network (BCNN) model that consists of two CNNs, and the outputs of the CNN layers are multiplied with the outer product at each spatial domain. This proposed model of this paper performed better than the traditional CNN by classifying colorectal cancer images into eight different classes. In [20], they developed an Inception Recurrent Residual Convolution Neural Network (IRRCNN) model for the histopathology image classification of Breast Cancer. They developed their model based on three powerful DL architectures, namely Inception, Residual, and Recurrent Network. In [12], the author proposed a lightweight CNN model to classify the histopathology images of prostate cancer. The model achieved promising accuracy of 94.0% for binary classification. The comparative analysis was performed with other state-of-the-art pre-trained models. In [21], the author proposed a fully automatic method that detects prostatectomy WSIs with a high-grade Gleason score. The

model achieved an accuracy of 78% in a balanced set of 46 unseen test images.

In recent years, researchers are focusing on the XAI because the decision-making process of deep neural networks is largely unclear, and it is difficult to understand for humans. In [22], the author used different methods to generate the importance map from the black-box model indicating how salient each pixel importance using gradients or other internal network states. Also, they address the problem of XAI for deep neural networks that take images as input and output a class probability. In [13], the author proposed a novel explanation technique (i.e., LIME) to explain the predictions of nay classifier. Also, they demonstrated the flexibility of this method by explaining ML models (e.g. random forests) for text and DL models (e.g. neural networks) for image classification. In [14], the author present a unified framework for interpreting model prediction, SHAP. It assigns feature importance for a particular prediction. Also, they proposed new methods that show better consistency with human intuition than previous approaches. In [15], the author surveyed the current progress of XAI and in particular its advances in healthcare applications. They discussed different approaches (i.e., Grad-CAM, LIME, and SHAP) to unbox the black-box for medical explainable AI via multi-modal and multi-center data fusion. In [23], the author evaluated k-means clustering and random forest algorithms using two very popular xExplainable techniques (i.e., LIME and SHAP) to see and understand the output of the black-box model.

In previous research works, the authors developed different kind of CNN models for histopathology image classification and achieved promising results. Also, few explainable techniques were proposed for interpreting model prediction. However, in the present study we developed a LDCNN model which is a modified version of LWCNN [12], and it is not complicated like other CNN models discussed in this section. The proposed model performs better than LWCNN in terms of accuracy, overfitting issue, and computational cost.

## III. MATERIALS AND METHODS

### A. Dataset

We have used two different datasets from two different centers. Out of which, one is public and the other one is private.

**Public Dataset:** It was collected online from the Kaggle repository [24]. The whole slide images (WSIs) were prepared at Radboud University Medical Center, Netherland. The slides were scanned using 3Dhitech Panoramic Flash II 250 scanner at 20× magnification. Sample patch images used for model testing are shown in Fig. 2 The size of the patch images extracted from WSIs is 512 × 512 pixels.

**Private Dataset:** This dataset is not publicly available online. It was collected from the Severance Hospital of Yonsei

University, South Korea. The slides were scanned at  $40\times$  optical magnification with 0.3 NA objective using a digital camera (Olympus C-3000) attached to a microscope (Olympus BX-51). The sizes of patches extracted from WSIs are  $256 \times 256$  and  $512 \times 512$  pixels. Fig. 3 shows the sample images of PCa used for model training and validation.

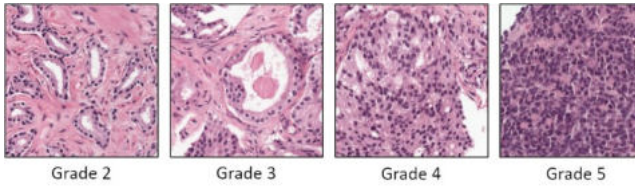


Fig. 2. The four types of PCa histopathology images from the PANDA challenge dataset. Grade 2 is considered a benign tumor and Grade 3, Grade 4, and Grade 5 is considered a malignant tumor

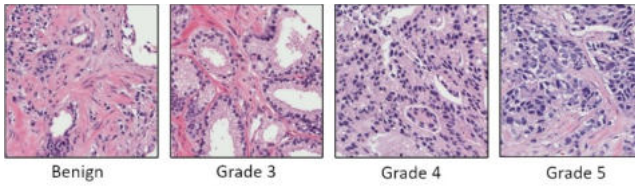


Fig. 3. The four types of histopathology images of PCa (benign, grade 3, grade 4, and grade 5) from a private dataset

Image resizing is a crucial step in computer vision. DL models train faster on smaller images and require the same input image dimensions (i.e., height  $\times$  width) for all the input samples. Also, the model produces an error if the image is too small or too big. So according to the rule of thumb method, we decided to use  $256 \times 256$  pixel images for the DL model.

Moreover, we performed data augmentation to increase the number of samples in the dataset because the huge data increases the likelihood that it contains useful information, which is advantageous for the DL model. Also, by adding more data in the training set the chances of overfitting decrease rather than increase. Therefore, we generated 2 samples from each input sample with rotation (i.e.,  $10^\circ$  and  $180^\circ$ ) augmentation technique. Table I shows the statistics for the PCa classification datasets.

TABLE I. STATISTICS FOR PRIVATE AND PUBLIC DATASET

Training and Validation	Private Dataset		
	Benign	Malignant	Total
Total Number of Samples	900	900	1800
Number of Training Samples Before Augmentation	810	810	1620
Number of Training Samples After Augmentation	1620	1620	3240
Number of Validation Samples	90	90	180
Testing	Public Dataset		
Number of Test Sample	50	50	100

### B. LDCNN Model for Prostate Cancer Recognition

To perform supervised learning, we introduce an LDCNN model for PCa classification. We have used two different datasets from two different centers. Out of which, one is

public and the other one is private. Although the model was not trained with a sufficient amount of data, the proposed model has shown state-of-the-art performance. The model provides better recognition performance using fewer network parameters.

To construct the model, we utilized a concatenation operation between the CNN layers to build the dense connections in the network. Here, the output feature maps of the layer are concatenated with the incoming feature maps. The model has several advantages: it strengthens feature propagation, encourages feature reuse, and substantially reduces the number of parameters. Nonetheless, the model may require high graphics processing unit (GPU) due to concatenation operation. The model included CNN layers, such as those for input, convolution, rectified linear unit (ReLU), concatenation, dropout, global average pooling (GAP), and classification. In this model, ‘Stride=2’ was utilized in the convolution layer instead of the ‘Maxpooling ( $2 \times 2$ )’ to down-sample an input representation (image, hidden-layer output matrix, etc.), reducing its dimensionality and allowing for assumptions to be made about features contained in the sub-regions binned. The entire model is shown in Fig. 4.

For classification, we set the input shape to  $256 \times 256 \times 1$  while building the model. The model contains 9 convolutional layers, 3 concatenation layers, a GAP layer, and a classification layer. Softmax activation function was utilized for the binary classification. An Adam [25] optimizer was used during training and the ReduceLROnPlateau function was used to control the learning rate (LR) of the model. To avoid model overfitting, we used the early stopping function which is a form of regularization. A total of 100 epochs were set for training the model and the learning stopped at 45 because there was no progress on the validation set for consecutive 10 epochs. All the experiments were conducted on a workstation with an NVIDIA GeForce RTX 3060 GPU, 32 GB of RAM using Tensorflow and Keras libraries.

### C. Activation Layer Visualization

ALV is the technique for visualizing the feature maps by digging into neural networks. In the CNN model, activation layers are a crucial part of the design, and each layer produces a different number of feature maps that are the result of applying the filters to an input image. Therefore, visualizing the activation layer of the black-box model is important because it shows the output (i.e., the activated feature maps) of specific activation layers and this is done by looking at each specific layer.

### D. Gradient Weighted Class Activation Map

Grad-CAM is another popular and effective technique for interpreting black-box models. It is a simple method compared to SHAP but we can see which regions in the image were relevant in a specific class. To visualize the attention regions in the image, a GAP layer was used instead of fully connected (FC) layers at the end before the final classification layer, and the network takes the convolutional feature maps as input through the GAP layer to produce the outputs for each class. Therefore, the class-discriminative localization map is obtained by computing the gradient of class  $C$  with respect to feature maps  $F$  of a convolutional layer. The global average pooled gradients flow back to obtain the importance weights  $\alpha_k^C$ . The computation for Grad-CAM can be expressed as:

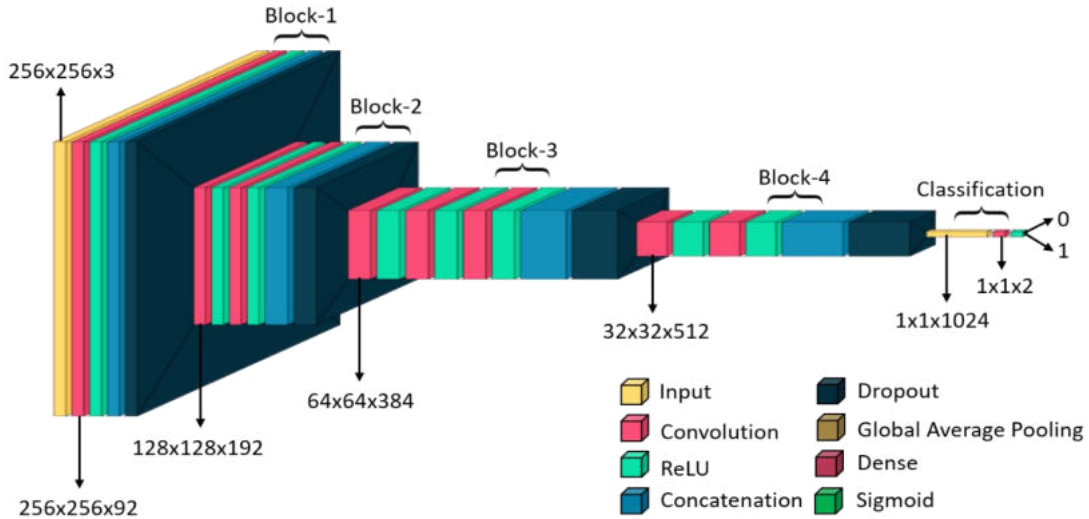


Fig. 4. Light Dense Convolutional Neural Network Architecture

$$\alpha_k^c = \frac{1}{uv} \sum_i^u \sum_j^v \frac{\partial y^c}{\partial F_{ij}^k} \quad (1)$$

$$GradCAM_c = ReLU \left( \sum_{k=1}^k \alpha_k^c F^k \right) \quad (2)$$

where  $\alpha_k^c$  is the average of all gradient score of class  $C$  for the  $k$ th feature map,  $u$  and  $v$  are the length and width of the image,  $\frac{1}{uv} \sum_i^u \sum_j^v$  is the global average pooling,  $\frac{\partial y^c}{\partial F_{ij}^k}$  are the gradients via backpropagation,  $ReLU$  is the activation function of a convolutional layer, and  $GradCAM_c$  is the final attention result for the predicted class.

#### E. Local Interpretable Model-Agnostic Explanation

LIME is one of the novel explanation methods that explains the model predictions from each data sample in a faithful way by approximating the local interpretable models [23]. The implementation strategies of LIME are different for different data format (i.e., tabular, text data, and image data) [26].

To explain the decision of the black-box model for image data, LIME technique was used to visualize the important regions that contributed to the prediction results. In LIME algorithm, first an image segmentation method (i.e., Quickshift) is utilized to separate the original data into multiple pixel blocks. Then the pixel block is used as the original data set and perturbs it to achieve model interpretation. The equation for LIME explainer can be expressed as:

$$\xi(x) = \frac{\text{argmin}}{g \in G} Loss(f, g, \pi_x) + \Omega(g) \quad (3)$$

where  $f$  is an original predictor (i.e., the CNN model),  $x$  is the original features,  $g$  is a local model,  $Loss(f, g)$  signifies the local approximation degree of the target model  $f$  and the proxy model  $g$  [26],  $\pi_x$  is the measure of proximity of an instance  $y$  from  $x$ ,  $\Omega(g)$  is measure of complexity of  $g \in G$  [23], and  $\xi(x)$  is an interpreter.

#### F. SHAP Gradient Explainer

SHAP is a very popular AI technique and game theory-based approach used for explaining the output of any black-box model (e.g., DL or ML). SHAP technique is used in this paper to measure feature importance and explain model decisions using expected gradients (i.e., an extension of integrated gradients). Generally, the feature attribution method is called integrated gradient which is used for deep neural networks. Therefore, the SHAP value indicates the importance of each feature in the model and how much it is contributed to the predictions for each given instance.

To explain and interpret the decisions of the black-box model, we used the SHAP algorithm to visualize the attention regions by plotting the SHAP values of every important feature for every predicted tissue sample. The equation for Shapely value estimation can be expressed as:

$$\phi_i(f, x) = \sum_{S \subseteq F} \frac{|S|! (F - |S| - 1)!}{F!} \times [f_x(S) - f_x(S \setminus i)] \quad (4)$$

where  $\phi_i$  Shapely value for feature  $i$ ,  $f$  is the black-box model,  $x$  is the input dataset,  $S \subseteq F$  is the feature subsets, and  $F$  is the set of all features. The SHAP value is computed based on the model prediction with the training dataset  $f_x(S)$  and testing dataset  $f_x(S \setminus i)$ .

#### IV. RESULTS AND DISCUSSION

We trained and tested the LDCNN model on high-resolution H&E stained image datasets collected from two different centers. A total of 900 images were used from private dataset and 100 from public dataset for training and testing, respectively. We performed data augmentation to avoid overfitting issues and improve the performance and outcomes of the CNN model by creating new and different samples to train the dataset. Further, to learn our CNN model, we divided private dataset into training and validation datasets according to an 9:1 ratio. Both LWCNN and LDCNN models were trained and tested on private and public datasets, respectively. From the comparative analysis (Table II), it can be observed that our proposed model achieved the best performance. In particular, LDCNN obtained a better performance by 6.0% on

accuracy compared to LWCNN. Moreover, at testing phase, the public dataset was further separated into five-split for determining the generalizability of the learned model (i.e., LDCNN). Therefore, the model showed promising results and achieved an accuracy of 100%, 100%, 95.0%, 90.0%, and 85.0%, and area under the curve (AUC) of 1.00, 1.00, 1.00, 0.90, and 0.99 at test split 1, 2, 3, 4, and 5, respectively. In addition, we also provided the ROC curves to evaluate and compare the CNN models which illustrates the diagnostic ability of a binary classifier system, shown in Fig. 5. From the figure, we can observe that both the models performed well at training phase and achieved an overall AUC of 98.0%. In contrast, at testing phase, LWCNN model did not achieve better results compared to LDCNN.

TABLE II. COMPARISON RESULTS OF LWCNN AND LDCNN

Model Performance at Testing Phase	Public Dataset	
	LWCNN	LDCNN
Accuracy (%)	87.0	93.0
Precision (%)	96.0	92.0
Recall (%)	81.4	93.8
F1-Score (%)	88.1	92.9
AUC (%)	98.0	98.0

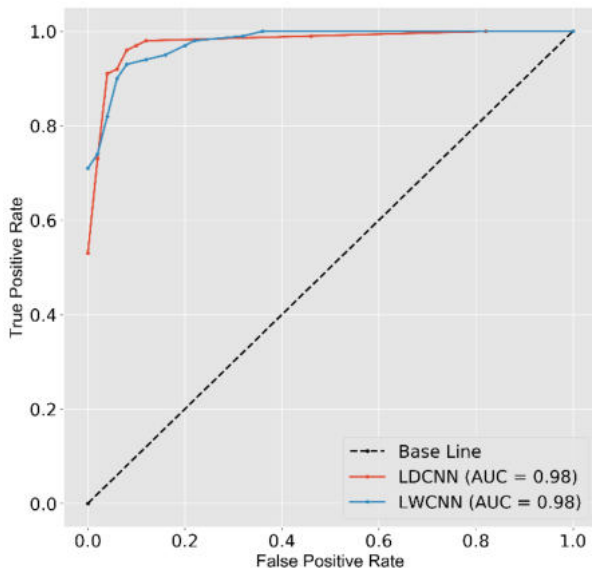


Fig. 5. ROC curve for analyzing the model performance on each test split generated by plotting the model's confidence scores

Unboxing the black-box model is very important for the medical image analysis. Many AI algorithms cannot provide any evident how and why a decision has been cast. Therefore, in this paper, we adopted few techniques for interpreting the black-box model. Fig. 6 shows the visualization results of four different activation layers extracted from our proposed CNN model. From the figure, we can observe low- and high-level feature maps obtained by CNN to identify cancer types (i.e., benign and malignant).

We also examined to interpret the decisions of the CNN black-box model using the SHAP technique (Fig. 7). This shows how each feature is significant in determining the final prediction of the model outputs. This technique could

recognize the cell nuclei surrounding circular regions and highly scattered in other sections in benign and malignant tissue samples. Fig. 8 shows another popular effective method (Grad-CAM) for interpreting the CNN model. This technique is utilized to see which regions in the image are relevant to the particular class. Fig. 9 shows the model explanation via LIME to visualize the super-pixels that contributed to the benign and malignant prediction results. To visualize the interpretable results, we used the predicted benign and malignant samples, shown in Fig. 7a and b, respectively.

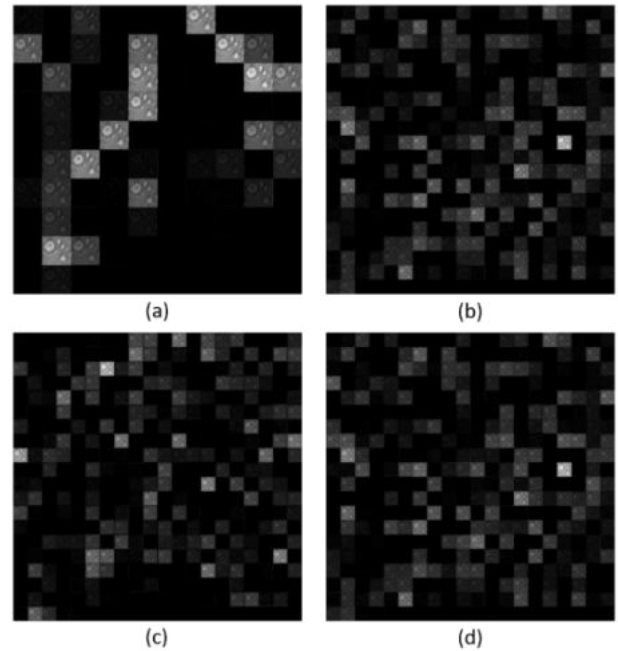


Fig. 6. Feature maps are generated from the activation layers of the CNN model. (a) Block 1 activation layer. (b) Block 2 activation layer. (c) Block 3 activation layer. (d) Block 4 activation layer. The bright pixels represents the activated and significant features

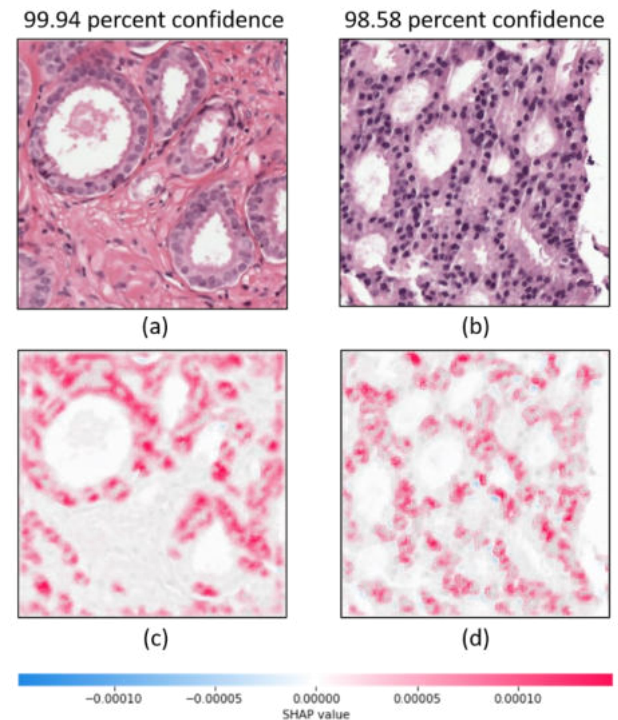


Fig. 7. Visualization of the attention regions that are positively contributed to the prediction via the SHAP method. (a) and (b) Predicted benign and

malignant tissue samples. (c) and (d) Interpretable results of (a) and (b), respectively. The color bar signifies the SHAP value. The red and blue color represents the positive and negative value that increases and decreases the model's output, respectively

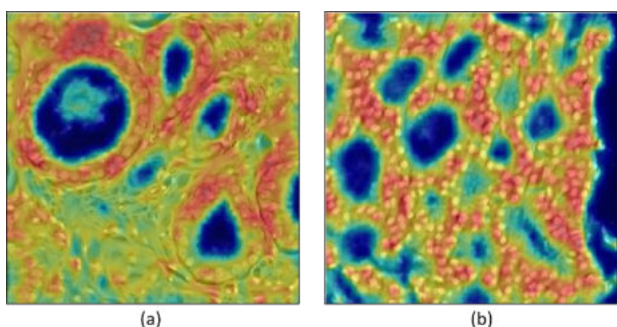


Fig. 8. Visualization of the attention regions using the Grad-CAM method. (a) Benign tissue sample. (b) Malignant tissue sample. The red color signifies the most class-specific discriminative parts of the image

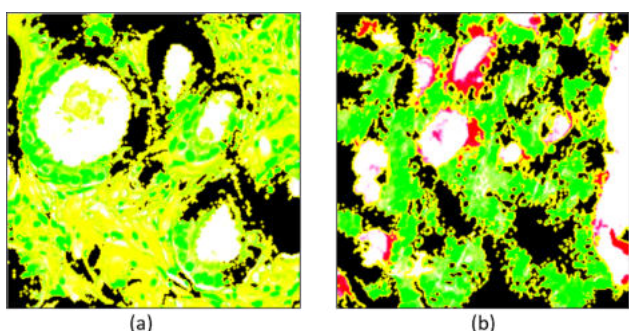


Fig. 9. Visualization of the attention regions and super-pixels that are positively contributed to the prediction via the LIME method. (a) Benign tissue sample. (b) Malignant tissue sample. The green color signifies the most class-specific discriminative parts of the image

In this study, we have first discussed the methods utilized for image classification and XCV. Binary classification (i.e., benign vs. malignant) was performed using LDCNN and LWCNN models, and comparative analysis was carried out to analyze their performance on a public dataset used for testing. Then, we adopted different explainable and interpretable techniques to visualize the attention regions and contribution of each pixels for the prediction. The main difference between the two models is that the LDCNN consists of the concatenation layers that create dense connections in the network and each convolutional block is constructed using the combinations of activation functions (i.e., Tanh and ReLu).

Diagnosis of PCa using histopathology images has been one of the key topics in recent oncology. In this study, we focused on interpretability and explainability in DL. Our proposed CNN model classified the H&E stained images of PCa into benign and malignant and achieved the best result at test split 1 and 2, obtaining an AUC of 1.00. The interpretable and explainable techniques (i.e., ALV, Grad-CAM, LIME, and SHAP) are very beneficial for individual diagnosis by analyzing each super-pixel in the predicted sample. Also, these methods explain how local explanations affect the final prediction. It is of note that PCa detection and classification is a widely investigated problem in medical data analysis. Therefore, meta-explanation is important to describe the behavior of the black-box model at a more human-understandable level [14]. Research in XCV should be more precise and meaningful because human users are the viewers of XCV results.

Nevertheless, the visualization results extracted from our CNN model are interpretable and explainable which shows the activated and significant feature maps for classification and attention regions in the predicted outputs. However, from the existing research works, it has been analyzed that there are no standardized metrics to evaluate the explainability techniques of CNN.

## V. CONCLUSION

In this paper, an LDCNN model was developed for the classification of H&E stained images of PCa (i.e., benign and malignant). This model was modified from the LWCNN model introduced in our previous study. The modified model plotted some astounding results by prompting an accuracy of 100% at test split 1 and 2. The approaches we used in this study are significantly superior for tissue image classification and detection of the cancer regions. Therefore, the quantitative and qualitative results are promising to rationalize further research of our approach for other domains. In the future, the deliberated approach will be applied to other cancers. However, the present work motivated and encouraged us by providing an excellent output that could be useful in real-life scenarios for the healthcare industry.

## ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1A2C2008576).

## REFERENCES

- [1] M. N. Gurcan, L. E. Boucheron, A. Can, A. Madabhushi, N. M. Rajpoot, and B. Yener, "Histopathological Image Analysis: A Review," *IEEE Rev. Biomed. Eng.*, vol. 2, pp. 147–171, 2009, doi: 10.1109/RBME.2009.2034865.
- [2] M. Veta, J. P. W. Pluim, P. J. van Diest, and M. A. Viergever, "Breast Cancer Histopathology Image Analysis: A Review," *IEEE Trans. Biomed. Eng.*, vol. 61, no. 5, pp. 1400–1411, May 2014, doi: 10.1109/TBME.2014.2303852.
- [3] J. P. Hinton *et al.*, "A Method to Reuse Archived H&E Stained Histology Slides for a Multiplex Protein Biomarker Analysis," *Methods Protoc.*, vol. 2, no. 4, p. 86, Nov. 2019, doi: 10.3390/mps2040086.
- [4] J. P. Hinton *et al.*, "A Method to Reuse Archived H&E Stained Histology Slides for a Multiplex Protein Biomarker Analysis," *Methods Protoc.*, vol. 2, no. 4, p. 86, Nov. 2019, doi: 10.3390/mps2040086.
- [5] Y. Xu *et al.*, "Large scale tissue histopathology image classification, segmentation, and visualization via deep convolutional activation features," *BMC Bioinformatics*, 2017, doi: 10.1186/s12859-017-1685-x.
- [6] A.-S. Metwalli, W. Shen, and C. Q. Wu, "Food Image Recognition Based on Densely Connected Convolutional Neural Networks," in *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIC)*, Feb. 2020, pp. 027–032, doi: 10.1109/ICAIC48513.2020.9065281.
- [7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2014, pp. 580–587, doi: 10.1109/CVPR.2014.81.
- [8] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-scale Orderless Pooling of Deep Convolutional Activation Features," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014, pp. 392–407.
- [9] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.
- [10] A. Singh, S. Sengupta, and V. Lakshminarayanan, "Explainable deep learning models in medical image analysis," *Journal of Imaging*, 2020, doi: 10.3390/JIMAGING6060052.

- [11] A. Barredo Arrieta *et al.*, “Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI,” *Inf. Fusion*, vol. 58, pp. 82–115, Jun. 2020, doi: 10.1016/j.inffus.2019.12.012.
- [12] S. Bhattacharjee, C.-H. Kim, D. Prakash, H.-G. Park, N.-H. Cho, and H.-K. Choi, “An Efficient Lightweight CNN and Ensemble Machine Learning Classification of Prostate Tissue Using Multilevel Feature Analysis,” *Appl. Sci.*, vol. 10, no. 22, pp. 71–93, Nov. 2020, doi: 10.3390/app10228013.
- [13] M. Ribeiro, S. Singh, and C. Guestrin, ““Why Should I Trust You?”: Explaining the Predictions of Any Classifier,” in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, Feb. 2016, pp. 97–101, doi: 10.18653/v1/N16-3020.
- [14] S. Lundberg and S.-I. Lee, “A Unified Approach to Interpreting Model Predictions,” *Adv. Neural Inf. Process. Syst.*, May 2017, [Online].
- [15] G. Yang, Q. Ye, and J. Xia, “Unbox the black-box for the medical explainable AI via multi-modal and multi-centre data fusion: A mini-review, two showcases and beyond,” *Inf. Fusion*, 2022, doi: 10.1016/j.inffus.2021.07.016.
- [16] Y. Zhang, D. Hong, D. McClement, O. Oladosu, G. Pridham, and G. Slaney, “Grad-CAM helps interpret the deep learning models trained to classify multiple sclerosis types using clinical brain magnetic resonance imaging,” *J. Neurosci. Methods*, vol. 353, p. 109098, Apr. 2021, doi: 10.1016/j.jneumeth.2021.109098.
- [17] K. Roy, D. Banik, D. Bhattacharjee, and M. Nasipuri, “Patch-based system for Classification of Breast Histology images using deep learning,” *Comput. Med. Imaging Graph.*, vol. 71, pp. 90–103, Jan. 2019, doi: 10.1016/j.compmedimag.2018.11.003.
- [18] S. Chakraborty, S. Aich, A. Kumar, S. Sarkar, J.-S. Sim, and H.-C. Kim, “Detection of cancerous tissue in histopathological images using Dual-Channel Residual Convolutional Neural Networks (DCRCNN),” in *2020 22nd International Conference on Advanced Communication Technology (ICACT)*, Feb. 2020, pp. 197–202, doi: 10.23919/ICACT48636.2020.9061289.
- [19] C. Wang, J. Shi, Q. Zhang, and S. Ying, “Histopathological image classification with bilinear convolutional neural networks,” in *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Jul. 2017, vol. 2017, pp. 4050–4053, doi: 10.1109/EMBC.2017.8037745.
- [20] M. Z. Alom, C. Yakopcic, M. S. Nasrin, T. M. Taha, and V. K. Asari, “Breast Cancer Classification from Histopathological Images with Inception Recurrent Residual Convolutional Neural Network,” *J. Digit. Imaging*, 2019, doi: 10.1007/s10278-019-00182-7.
- [21] Del Toro, O. J., Atzori, M., Otálora, S., Andersson, M., Eurén, K., Hedlund, M., ... & Müller, H. (2017, March). Convolutional neural networks for an automatic classification of prostate tissue slides with high-grade gleason score. In *Medical Imaging 2017: Digital Pathology*, vol. 10140, p. 101400. International Society for Optics and Photonics.
- [22] V. Petsiuk, A. Das, and K. Saenko, “RISE: Randomized Input Sampling for Explanation of Black-box Models,” *Br. Mach. Vis. Conf. 2018, BMVC 2018*, Jun. 2018.
- [23] A. Gramegna and P. Giudici, “SHAP and LIME: An Evaluation of Discriminative Power in Credit Risk,” *Front. Artif. Intell.*, vol. 4, pp. 1–6, Sep. 2021, doi: 10.3389/frai.2021.752558.
- [24] “Prostate cANcer graDe Assessment (PANDA) Challenge”, Accessed on: Oct. 10, 2021. [Online]. Available: <https://www.kaggle.com/c/prostate-cancer-grade-assessment/overview/description>
- [25] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” 2015.
- [26] Y. Liang, S. Li, C. Yan, M. Li, and C. Jiang, “Explaining the black-box model: A survey of local interpretation methods for deep neural networks,” *Neurocomputing*, vol. 419, pp. 168–182, Jan. 2021, doi: 10.1016/j.neucom.2020.08.011.

# Whole Slide Image Analysis and Detection of Prostate Cancer using Vision Transformers

Kobiljon Ikromjanov  
Dept. of Digital Anti-Aging Healthcare  
Inje University  
Gimhae, Republic of Korea  
kobiljonikromjanov@gmail.com

Subrata Bhattacharjee  
Dept. of Computer Engineering  
Inje University  
Gimhae, Republic of Korea  
subrata\_bhattacharjee@outlook.com

Yeong-Byn Hwang  
Dept. of Digital Anti-Aging Healthcare  
Inje University  
Gimhae, Republic of Korea  
hyb1345679@gmail.com

Rashadul Islam Sumon  
Dept. of Digital Anti-Aging Healthcare  
Inje University  
Gimhae, Republic of Korea  
sumon39.cst@gmail.com

Hee-Cheol Kim  
Dept. of Digital Anti-Aging Healthcare  
Inje University  
Gimhae, Republic of Korea  
heeki@inje.ac.kr

Heung-Kook Choi  
Dept. of Computer Engineering  
Inje University  
Gimhae, Republic of Korea  
cschk@inje.ac.kr

**Abstract**—Prostate cancer (PCa) is the most frequently diagnosed non-skin malignancy in men and the second leading cause of fatality from cancer. The most prognostic marker for PCa is the Gleason grading system on histopathology images. Pathologists examine the Gleason grade on stained tissue specimens of Hematoxylin and Eosin (H&E) based on tumor structural growth patterns from whole slide image (WSI). According to the Gleason grading system, prostate cancers are scaled into five grades based on glandular patterns of differentiation. It varies from grade 1 (normal tumor) to grade 5 (abnormal tumor). Cancer cells that look similar to healthy cells receive a low score. Recent developments in Computer-Aided Detection (CAD) using Artificial Intelligence (AI), mainly Deep learning (DL) have brought the immense scope of automatic detection and recognition at better accuracy in adenocarcinoma like other medical diagnoses. Automated DL systems have delivered promising results from histopathological images to accurate grading of prostatic adenocarcinoma. This study aims to classify multiple patterns of images extracted from the WSI of a prostate biopsy based on the Gleason grading system. First, extract patches from the detected region of interest (ROI), then applying Vision Transformers (ViT) model for classification. Finally, the classified patches are scored and graded. The proposed deep learning model in this research will be able to assist the pathologist and other researchers to identify and treat of prostate cancer.

**Keywords**— whole slide image, prostate cancer, vision transformers, artificial intelligence

## I. INTRODUCTION

Deep learning AI architectures are developed and applied to medical images, making high-precision diagnosis possible. For diagnosis, the medical images need to be labeled and standardized, before data pre-processing and training DL model. The final predicted diagnosis results can be obtained immediately and accurately. Tumor detection and classification in histopathology images are important for early diagnosis and treatment planning. Many techniques have been proposed for classifying medical image data through quantitative assessment [1-3]. However, some quantitative ways of evaluating medical images are inaccurate and require considerable computation time to analyze large amounts of data. Analytical strategies applied to AI algorithms can improve diagnostic accuracy and save time.

To identify different kinds of prostate tumors, pathologists use different screening methods. Male hormones such as testosterone cause prostate cancer to grow and survive. Like all cancers, prostatic adenocarcinoma begins once a mass of cells has grown out of control and invades other tissues. Cells

become cancerous due to the accumulation of defects, or mutations, in their DNA. Mutations in the abnormal cells' DNA cause the cells to grow and divide more rapidly than normal cells do. Histological examination of tissues and the detection of cancer by physicians remains the gold standard in cancer diagnosis. The diagnosis of PCa is heavily time-consuming. In addition, it is based on subjective grading. For example, the study by Ozkan et al. reported that two pathologists disagreed about the presence of cancer in 31 of 407 baseline biopsies and that the total concordance of the accessed Gleason score was only 51.7%, describing these challenges in diagnosing the PCa consistently [4]. Therefore, the development of computer-assisted decision support tools is essential for saving time, predicting disease outcomes, and improving precision medicine for pathologists.

Automated diagnosis can reduce workloads and pathologist variability. Researchers face difficulty in studying the Gleason scoring system [5, 6]. Accurate annotations and pathological accuracy are required to train the model correctly. At present, automated computerized techniques are in high demand for medical image analysis and processing. However, we propose a ViT [7-9] model to classify the grading of PCa. We detect ROI, then patches for classification and scoring [10-12]. After scoring all the patches, overall grading is performed for each WSI which is helpful for pathologists to save time.

## II. DATASET COLLECTION

The dataset was collected online, which is publicly available on the Kaggle PANDA challenge [13]. It has 10,616 WSI images and 10,516 corresponding WSI mask images. Radboud University Medical Center and Karolinska Institute have teamed up to organize this PANDA competition. Fig. 1 shows some examples of Radboud and Karolinska images and their annotations. However, in our experiment, we have used more than 5,000 images with their masks from Radboud University Medical Center dataset. After having a patching process on those images, we got approximately 304,000 acceptable patch images of size  $256 \times 256$  pixels, corresponding to 5 classes: stroma, benign, score 3, score 4, and score 5. We have split about 240,000 patch images for training, 60,000 patch images for validation, and around 4,000 patch images for testing.



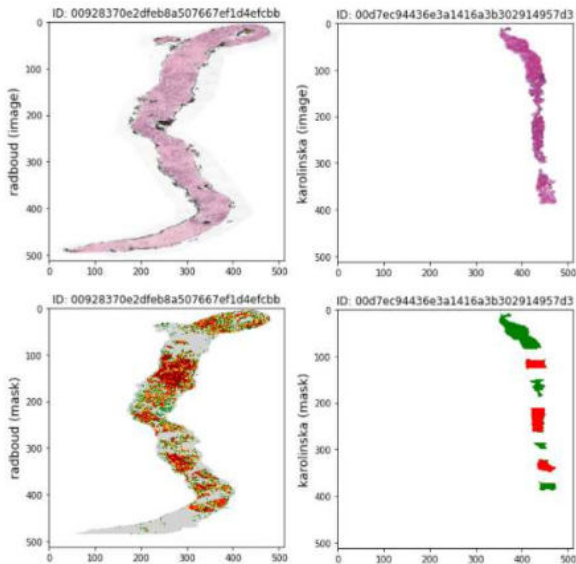


Fig. 1. Sample images and ground masks

### III. METHODOLOGY

To follow up the proposed model, we applied patching for making an acceptable dataset for the model. Then, we manipulated the ViT model and get the training output according to the Gleason Scoring System. The following Fig. 2 shows the general view of the applied method for the training of the ViT model and its architecture. In the following, we will clarify patching, a ViT model, and a Gleason Score System.

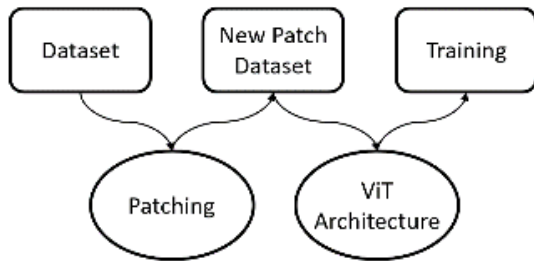


Fig. 2. The process of patching and training ViT model

#### A. Patching

The ability to compare image regions (patches) has been the basis of many approaches to core computer vision problems, including object, texture, and scene categorization. The WSI is also called a gigapixel image that is composed of more than 1 billion pixels [14], and it is computationally unfeasible to perform ROI-based image analysis in such a high dimensional space. Therefore, in many existing works, image analysis have been performed over small image patches. This has the advantage of making computational tasks such as learning, inference, and likelihood estimation much easier than working with gigapixel image directly. In this order, the entire WSI cannot be trained in GPU memory at once, so one solution is to select a subset of patches from the high dimensional image. In this study, we have patched all 10,516 WSIs and their ground truth mask images cickected from the Kaggle dataset. Fig. 3 illustrates the steps for patching the WSI. Fig. 3 (a) shows 395 patches of size  $256 \times 256$  pixels, while Fig. 3 (b) indicates the annotation done by pathologists. In Fig. 3 (c), the patches are extracted from ROI annotated in Fig. 3 (b).

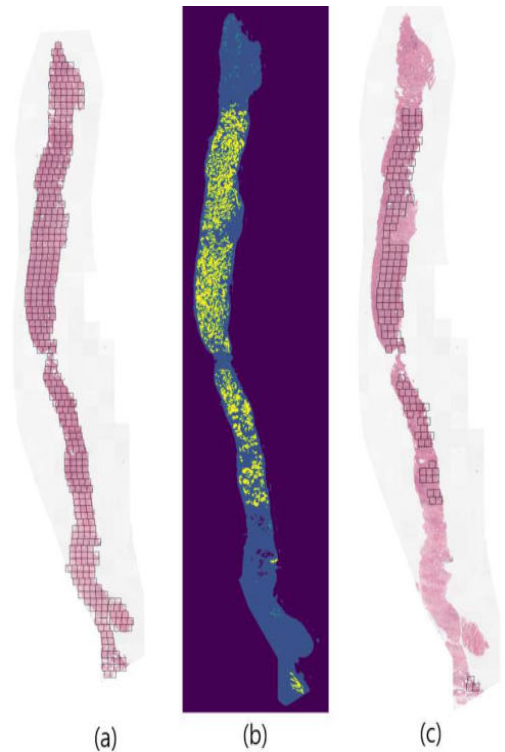


Fig. 3. The steps for patching a WSI

#### B. Vision Transformers

The Vision Transformer, or ViT, is a model for image classification that employs a Transformer-like architecture over image patches. An image is split into fixed-size patches, each of them is then linearly embedded, position embeddings are added, and the resulting sequence of vectors is fed to a standard Transformer encoder. The standard approach of adding an extra learnable “classification token” to the sequence is used to perform the classification.

Inspired by the ViT model for the classification of each patch, we experiment with applying the patched images from the WSI as input images. First, we split them into fixed-sized images, then flatten them. After creating lower-dimensional linear embeddings from flattened image patches, we include positional embeddings. Moreover, feeding the sequence as an input to a state-of-the-art transformer encoder, we can pre-train the ViT model with image labels, which are then fully supervised on a big dataset. Lastly, we can fine-tune the downstream dataset for image classification. Fig. 4 shows ViT architecture for classification.

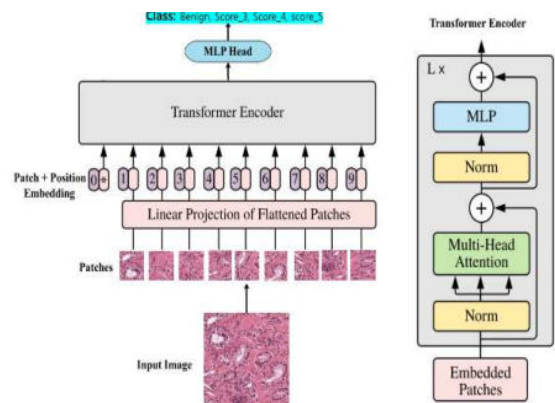


Fig. 4. ViT architecture for classification

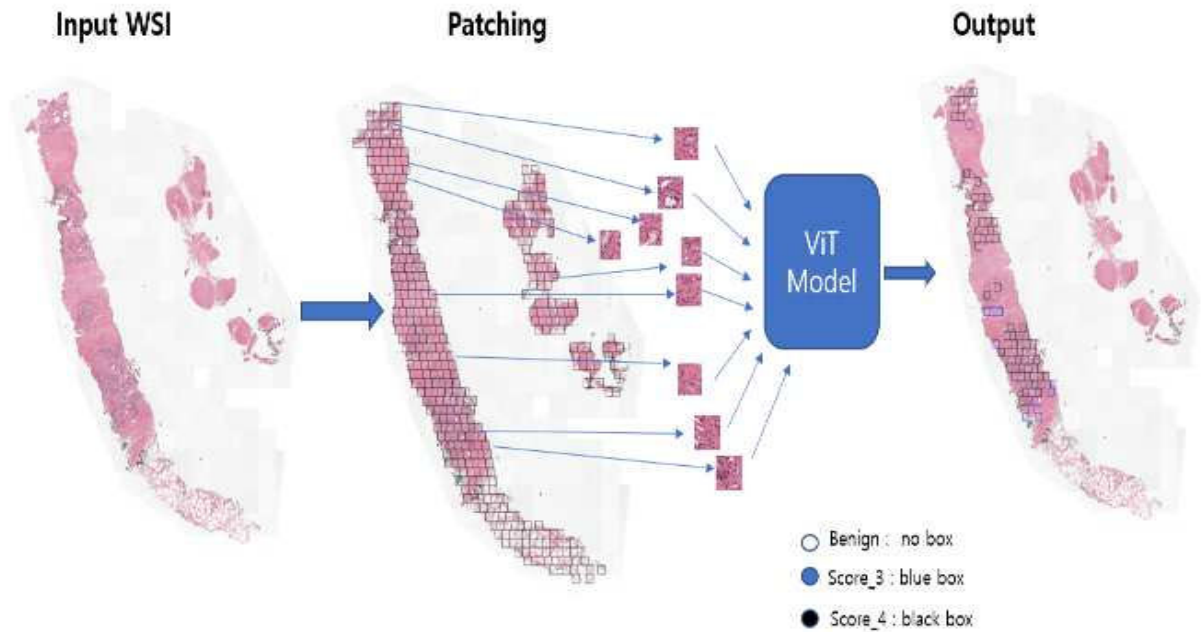


Fig. 5. Example of patching and prediction of ViT model

### C. Gleason Score System

The Gleason Score is the grading system used to determine the aggressiveness of prostate cancer. This grading system can be used to choose appropriate treatment options. The Gleason Score ranges from 1-5 and describes how much cancer from a biopsy looks like healthy tissue (lower score) or abnormal tissue (higher score). Most cancers score a grade of 3 or higher. Since prostate tumors are often made up of cancerous cells that have different grades, two grades are assigned for each patient. A primary grade is given to describe the cells that make up the largest area of the tumor and a second grade is given to describe the cells of the next largest area. For instance, if the Gleason Score is written as 3+4=7, it means most of the tumor is grade 3 and the next largest section of the tumor is grade 4, together they make up the total Gleason Score. If the cancer is almost entirely made up of cells with the same score, the grade for that area is counted twice to calculate the total Gleason Score. Typical Gleason Scores range from 6-10. The higher the Gleason Score, the more likely it is that cancer will grow and spread quickly. Our proposed method shows the potential of AI systems for Gleason grading, but more importantly, shows the benefits of pathologist-AI synergy.

## IV. RESULT

The ViT model evaluated the patches as stroma, benign, score 3, score 4, and score 5 according to the level of cancerous cells and differentiates score 3, 4, and 5 with three different colors which would be very helpful for the pathologists to identify the affected ROIs. In the following Fig. 5, we can see around 760 patches in the beginning, and the ViT model predicted the level of cancerous cells and differentiated a whole image as no box for the benign, blue box for score 3, and black box for score 4.

The performance measures used for model evaluated are precision, recall, and f1-score. Overall the model performed well and achieved an accuracy of 80.0%. The following Table I shows the performance measures for the ViT model. In the precision, the model achieved good scores on the stroma,

benign, and score 3. In the recall and f1-score, the model predicted stroma and benign cases very well compared to score 3, 4, and 5 cases. Fig. 6 demonstrates the overall architecture of the procedure for generating the final predicted WSI image. In the final image, the pathologists can see the affected areas and its level. The model efficiency can be evaluated through the confusion matrix, shown in Fig. 7.

TABLE I. ViT ARCHITECTURE FOR CLASSIFICATION

Classes	Precision (%)	Recall (%)	F1-score (%)
Stroma	99.0	90.0	94.0
Benign	84.0	93.0	88.0
Score 3	82.0	73.0	78.0
Score 4	63.0	72.0	67.0
Score 5	74.0	71.0	72.0
Average	80.4	79.8	79.8

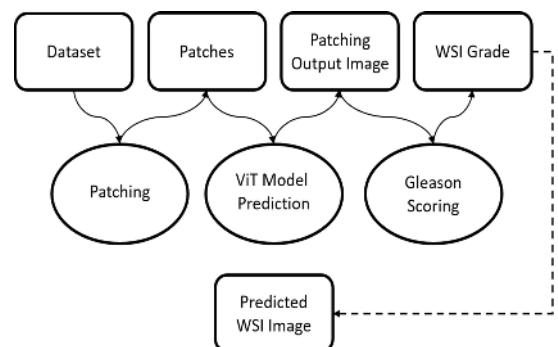


Fig. 6. The overall architecture for prediction and generating the final predicted image

True label \ Predicted label	Stoma	Benign	Score_3	Score_4	Score_5
Stoma	723	18	30	24	5
Benign	0	763	17	15	5
Score_3	8	93	575	100	24
Score_4	2	18	39	565	176
Score_5	0	29	25	147	599

Fig. 7. Confusion matrix of ViT model

## V. DISCUSSION

The proposed patching technique and ViT model were designed to help the pathologists classify different cancer images which consist of two active approaches for a vision processing task. The sliding window approach in image processing is used to get the local information by sub-dividing the images into many blocks (may be overlapping or non-overlapping). The kernel is a small matrix act as a transformation, it is used to map the original data into modified one. An overview of the model is depicted in Fig. 4. The first layer of the ViT linearly projects the flattened patches into a lower-dimensional space. The components resemble plausible basis functions for a low-dimensional representation of the fine structure within each patch. Fig. 5 illustrates the workflow for testing and prediction WSI samples. First WSI is divided into patches then each patch is fed to ViT model for scoring. If there are cancer patch images on prediction, it counts the number of predictions: score 3, score 4, and score 5 and makes bounding boxes on WSI that could be helpful for the doctors to make a better decision.

## VI. CONCLUSION

The proposed method is patching WSIs and selecting ROI patches using ground truth images to train the ViT model. As the ViT is an attention-based model, it may give better concentration on cancer tissue areas while training and testing. This paper proposes a deep learning-based classification of multiple patterns of images extracted from the WSI of a prostate biopsy based on the Gleason grading system. The results show possibilities to assist the pathologist and other researchers to identify and treat of prostate cancer. In the future, we will develop the Mask-RCNN architecture for more improvement, train the model on a greater number of datasets,

and explain the prediction of the model via different interpretable techniques.

## ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1A2C2008576).

## REFERENCES

- [1] K. Bera, K. A. Schalper, D. L. Rimm, V. Velcheti, and A. Madabhushi, "Artificial intelligence in digital pathology — new tools for diagnosis and precision oncology," *Nat. Rev. Clin. Oncol.*, 2019, doi: 10.1038/s41571-019-0252-y.
- [2] B. Aygüneş, S. Aksoy, G. Cinbiş, K. Kösemehmetoglu, S. Önder, and A. Üner, "Graph convolutional networks for region of interest classification in breast histopathology," 2020, doi: 10.1117/12.2550636.
- [3] S. Bhattacharjee, C. H. Kim, D. Prakash, H. G. Park, N. H. Cho, and H. K. Choi, "An efficient lightweight cnn and ensemble machine learning classification of prostate tissue using multilevel feature analysis," *Appl. Sci.*, 2020, doi: 10.3390/app10228013.
- [4] T. A. Ozkan, A. T. Eruyar, O. O. Cebeci, O. Memik, L. Ozcan, and I. Kuskonmaz, "Interobserver variability in Gleason histological grading of prostate cancer," *Scand. J. Urol.*, 2016, doi: 10.1080/21681805.2016.1206619.
- [5] W. Bulten *et al.*, "Automated deep-learning system for Gleason grading of prostate cancer using biopsies: a diagnostic study," *Lancet Oncol.*, 2020, doi: 10.1016/S1470-2045(19)30739-9.
- [6] N. Chen and Q. Zhou, "The evolving gleason grading system," *Chinese Journal of Cancer Research*. 2016, doi: 10.3978/j.issn.1000-9604.2016.02.04.
- [7] M. Is, R. For, and E. At, "An image is worth 16x16 words: visual image transformer," 2021.
- [8] A. Vaswani *et al.*, "Attention Is All You Need," *Adv. Neural Inf. Process. Syst.*, Jun. 2017, [Online]. Available: <http://arxiv.org/abs/1706.03762>.
- [9] Y. Xia *et al.*, "Effective Pancreatic Cancer Screening on Non-contrast CT Scans via Anatomy-Aware Transformers," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2021, pp. 259–269.
- [10] X. Zhu, J. Yao, and J. Huang, "Deep convolutional neural network for survival analysis with pathological images," 2017, doi: 10.1109/BIBM.2016.7822579.
- [11] J. Yao, X. Zhu, J. Jonnagaddala, N. Hawkins, and J. Huang, "Whole slide images based cancer survival prediction using attention guided deep multiple instance learning networks," *Med. Image Anal.*, vol. 65, p. 101789, Oct. 2020, doi: 10.1016/j.media.2020.101789.
- [12] M. Salvi, U. R. Acharya, F. Molinari, and K. M. Meiburger, "The impact of pre- and post-image processing techniques on deep learning frameworks: A comprehensive review for digital pathology image analysis," *Computers in Biology and Medicine*. 2021, doi: 10.1016/j.combiomed.2020.104129.
- [13] "Prostate cANcer graDe Assessment (PANDA) Challenge", Accessed on: Oct. 10, 2021. [Online]. Available: <https://www.kaggle.com/c/prostate-cancer-grade-assessment/overview/description>.
- [14] D. Tellez, G. Litjens, J. Van Der Laak, and F. Ciompi, "Neural Image Compression for Gigapixel Histopathology Image Analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021, doi: 10.1109/TPAMI.2019.2936841.

# A Generative Adversarial Network Approach to Metastatic Cancer Cell Images

Seohyun Lee<sup>1</sup>, Hyuno Kim<sup>2</sup>, Hideo Higuchi<sup>3</sup>, Masatoshi Ishikawa<sup>4</sup>, and Ryuichiro Natato<sup>1</sup>

<sup>1</sup>Laboratory of Computational Genomics, Institute for Quantitative Bioscience, The University of Tokyo  
1-1-1 Yayoi, Bunkyo-ku, Tokyo, Japan

<sup>2</sup>Department of Mechanical and Biofunctional Systems, Institute of Industrial Science, The University of Tokyo  
4-6-1 Komaba, Meguro-ku, Tokyo, Japan

<sup>3</sup>Department of Physics, Graduate School of Science, The University of Tokyo

<sup>4</sup>Data Science Research Division, Information Technology Center, The University of Tokyo

7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan

seohyun\_lee@iqb.u-tokyo.ac.jp

h-kim@iis.u-tokyo.ac.jp

higuchi@phys.s.u-tokyo.ac.jp

ishikawa@ishikawa-vision.org

rnatato@iqb.u-tokyo.ac.jp

**Abstract**—The shapes of metastatic cancer cells are considered to be relatively different from non-metastatic cancer cells, especially regarding the degree of development of lamellipodia or the pattern of internal organ arrangement. However, understanding the specific pattern of the metastatic cancer cell has just started to emerge. In this paper, based on the generative adversarial network approach, we attempted to generate metastatic cancer cell images using human breast cancer cells where the metastasis-promoting protein, PAR1, is expressed.

**Index Terms**—generative adversarial network, cell image classification, breast cancer cell, biomedical image analysis, deep learning.

## I. INTRODUCTION

Cancer indicates the phenomenon that the cells do not follow the apoptotic pathway but grow uncontrollably [1]. Although the possibility of staying as a benign tumor is still high if the cells remain at their original position, it is considered malignant when the cells spread from the primary location to the other parts of the body [2]. The process of this dissemination is known as tumor metastasis, which makes cancer the most deadly disease with high morbidity rates [3]–[5].

Therefore, there have been many studies to understand the internal signal pathway in the cancer cell and to develop medical treatments using the knowledge of the nanoscale movement of intracellular information carrier, because the uncontrolled cell signal cascade is considered to be one of the essential key factors for the cancer research [1], [2].

Regarding intracellular dynamics, a huge amount of studies have been conducted to elucidate the specific mechanism of information delivery. Particularly, the three-dimensional movement of vesicles in terms of the interaction with the cytoskeletons such as actin filament and microtubule was interpreted based on numerical analysis, [6], [7] with the report

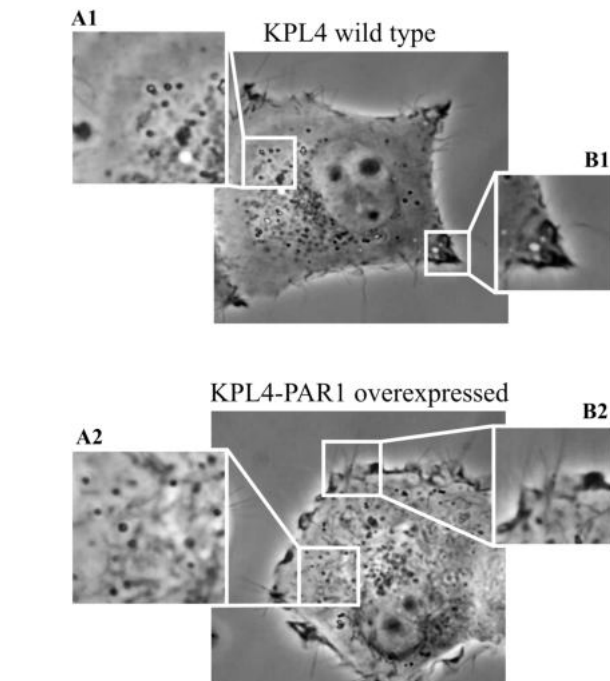
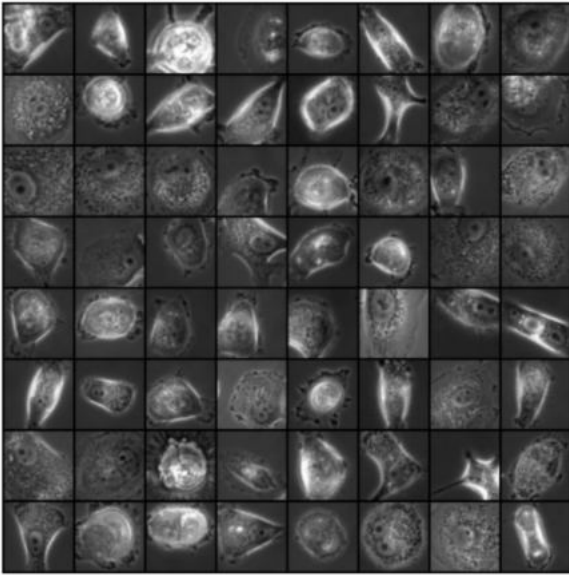


Fig. 1. Recognizable difference in cell morphology between representative KPL4 wild type which is relatively non-metastatic (upper) and protease-activated receptor 1 (PAR1) overexpressing KPL4 cancer cell which is considered highly metastatic (lower). A1 and A2 show the pattern of vesicle distribution while B1 and B2 indicate the difference in the development of filopodia and lamellipodia, and the contour profile of the cell.

about the characteristic rotational movement of cargoes on the cytoskeletal networks [8], [9]. Additionally, recent studies also attempted to introduce machine learning and image processing method based on computer vision into the cellular dynamics of the information carriers of interest [10]–[12].

A.

Training Images of KPL4-PAR1 overexpressed cell



B.

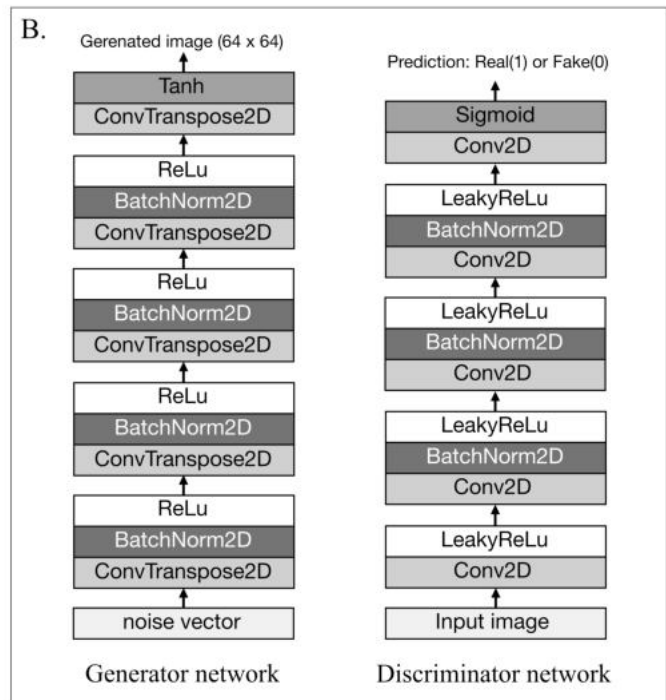


Fig. 2. (A) A representative training dataset of KPL4-PAR1 overexpressed cancer cell images. (B) The structures of the generator network and the discriminator network exploited in this study.

The studies introduced above shed light on our understanding of the practical movement of the signal carriers in the cancer cell, but the process of metastasis, which is in charge of leading cancer to lethal disease, has not yet been fully elucidated. Recently, because there have been studies to identify the mutation of a specific protein that promotes the mobility of metastatic cancer cells, many researchers have started to focus on the morphology of the cancer cell, as the irregular shape of the cancer cell is frequently accompanied by the enhanced mobility for the higher rate of cell migration [13].

For example, as shown in Fig. 1, the shape of the metastatic cancer cell can differ from relatively non-metastatic cancer cells, in the case of the KPL4 human breast cancer cell. Human observers can detect the different characteristics in morphology between two different cell types, such as the distribution pattern of internal organs (A1 and A2) and the development of filopodia and lamellipodia as well as the contour profile of the cell (B1 and B2). Therefore, recent studies have started to focus on big data of metastatic cancer cell images to develop deep-learning algorithms for the classification mainly using such differences in tissue-level morphology [14]–[16]. In our previous study, we suggested a deep-learning approach to classifying the type of KPL4 cancer cell, only using their phase-contrast microscopy images of colony morphology.

In this study, we attempted to introduce a generative adversarial network (GAN) approach [17] into the mimicry of the morphology of metastatic cancer cells, in order to augment the cell shape dataset for a more refined classification task [18].

## II. DATA PREPARATION

### A. Live Cell Imaging

The KPL4 human breast cancer cell line was kindly provided by Dr. Kurebayashi [19] (Kawasaki Medical School, Kurashiki, Japan). The cells exploited for the imaging were cultured in a complete growth medium (Dulbecco’s modified Eagle’s medium with high glucose, Nacalai Tesque, Inc., Japan) and then incubated at 37°C with 5% of CO<sub>2</sub>. During the imaging experiment, cells were stored in the heater (IN-ONI-F1, Tokai HIT, Shizuoka, Japan) to maintain the physiological conditions of living cells.

The images of the individual KPL4-PAR1 overexpressed cancer cells were taken by CMOS camera (Andor, DG-152X-C0E-F1, Belfast, Northern Ireland) with the microscope (IX70, Olympus, Tokyo, Japan) where 60× of objective lens is installed with a stage stabilizer [20]. In order to obtain clearly defined images of the intracellular area as well as cell edges, the phase-contrast imaging method, which enhances the contrast based on optical technique, was exploited in the imaging experiment. The size of raw images was 500×500.

### B. Data Preprocessing and Augmentation

Followings are the process to prepare the KPL4-PAR1 overexpressed cancer cell image dataset as shown in Fig. 2(A). First of all, the size of all the raw images was reduced to 200×200 to lower computational cost. Second, since the initial number of prepared images was only 360, the dataset was augmented by rotating and cropping the images. Because the size of the image after the transformation basically decreases

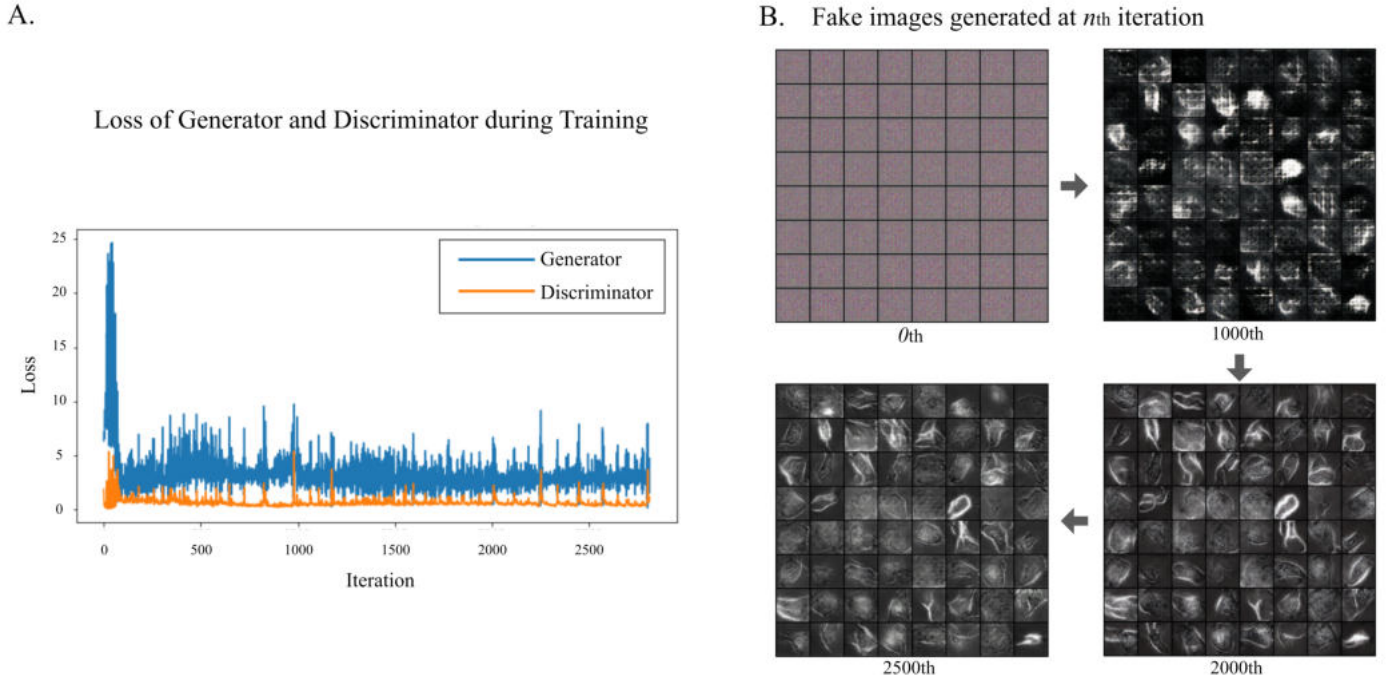


Fig. 3. (A) Loss of generator and discriminator during training. (B) A series of fake images created by generator network at 0th, 1000th, 2000th, and 2500th iteration.

as the rotation angle increases, we set the incremental angle as 0.1 degrees, and the initial angle as  $n\pi/2$  ( $n=1, 2, 3, 4$ ). The rotation was conducted for 5 degrees in maximum ( $0 < \theta < 5$ ). After the rotation, the images were cropped to generate the largest possible square using the image crop function in Python Image Library. The four arguments for the crop, which determine the left, upper, right, and lower positions, respectively, were calculated as below.

$$(\text{left, upper, right, lower}) = (\alpha, \alpha, s - \alpha, s - \alpha) \quad (1)$$

where  $s$  indicates the size of expanded image produced by the rotation of the original image and  $\alpha$  refers to the value computed by the following calculation in the case of the rotation by  $\theta$ .

$$\alpha = s \times (\sin \theta \times \cos \theta / (\sin \theta + \cos \theta)) \quad (2)$$

Utilizing the above data augmentation method, the volume of the total image dataset was enlarged to 72,000 and the images were fed to the generator as well as discriminator models with the batch size 128.

### III. MODEL CONSTRUCTION

#### A. Experimental Condition

In the experiment, the models were constructed by using the Pytorch framework (version 0.4.1) with Python version 3.7.9, in Linux (Ubuntu 16.04 LTS) operating system. The number of epoch for the training was set to 5, which produces

approximately 2,800 iterations with a batch size of 128 for 72,000 images in total. Additionally, the loss was computed using Binary Cross-Entropy loss, and Adam optimizer [22] was exploited for both generator and discriminator networks.

#### B. Deep Convolutional Generative Adversarial Network

The practical structure for the cancer cell image generator and discriminator networks were constructed based on the deep convolutional generative adversarial network (DCGAN), one of the extended GAN, which respectively utilizes convolutional and convolutional-transpose layers for the discriminator and the generator network [21]. Composed of convolutional-transpose layers, batch normalization layers, and ReLU activation, the generator network produces fake images from a noise vector in order to deceive the discriminator. On the other hand, the discriminator network comprises convolution layers, batch normalization layers, and LeakyReLU activation to discern the real images from fake images created by the generator network. The entire structures of the generator and discriminator are as shown in Fig. 2(B).

## IV. RESULT AND DISCUSSION

The loss over iteration for generator and discriminator network is as shown in Fig. 3(A) and the fake images created by generator network at 0th, 1000th, 2000th, and 2500th iterations are as shown in Fig. 3(B). The generator loss was computed by  $\log D(G(z))$  where  $z$  indicates the latent vector and  $G$  represents the generator network which outputs the image while  $D$  refers to the discriminator network which produces a

scalar probability that  $G(z)$  is real. Likewise, discriminator loss is calculated as  $\log D(x) + \log(1 - D(G(z)))$  where  $D(x)$  means the average output for all the real batches by the discriminator, which is the sum of the losses for the real and fake batches. The goal of GAN is achieved when the generator can produce perfect fake images so that the discriminator can guess with 50% of confidence. In this experiment, the fake images were stably created when the iteration reaches approximately 2000, as shown in Fig. 3(A), and the level of loss lingers until the final epoch. Accordingly, the quality of fake images produced by the generator network remains similar when iteration is over approximately 2000, as shown in Fig. 3(B).

Although the generator loss was reduced overall and the value of  $D(G(z))$  oscillates between 0.1 and 0.6, the goal of which is 0.5, the generated cell images are still recognizable as fake by human observers. First, the generator network is good at defining the edges of cells, but the location of the nucleus and the distribution of internals still seems obscure in the produced images. In addition, as the loss of the generator network did not apparently decrease after the 2000th iteration, the quality of generated cell images was not seemingly enhanced.

Because the main reason we attempted to produce fake cancer cell images is to augment the image dataset which can be subjected to the classification task for distinguishing metastatic and non-metastatic cancer cell lines only by their images, optimization of the number of epochs as well as the structural composition of both generator and discriminator are remained to be improved in the future work. Moreover, not only the KPL4-PAR1 overexpressed cell line but also the KPL4 wild type cell line which is known to be relatively less metastatic should be included in the next experiment, in order to prepare similar amounts of the image dataset. If we are able to compose and train the image dataset made of DCGAN-based augmentation besides real microscopy images for KPL4-PAR1 overexpressed cell and KPL4 wild type cell, it is expected to help our understanding of the morphological characteristics of metastatic cancer cell lines.

## V. CONCLUSION

In this paper, the microscopy images of the KPL4-PAR1 overexpressed cell line, which is known to be a metastatic human breast cancer cell, were generated by DCGAN. The raw cell images were obtained by using phase-contrast microscopy, and the volume of the dataset was enlarged based on the rotation and cropping of the images. After the 2000th iteration, the generator network was able to produce the images that discriminator can tell the genuineness of the images by approximately between 0.1 to 0.6, and a human observer can detect the similar shapes of cell edges compared to real cell images. With more improvement in layer composition and image data preparation, the result shown in this study is expected to be exploited in data augmentation for metastatic cancer cell classification task, which aims to understand the individual morphology of highly metastatic cancer cell lines.

## REFERENCES

- [1] R. J. Bold, P. M. Termuhlen, and D. J. McConkey, "Apoptosis, cancer and cancer therapy," *Surgical oncology*, 6(3), pp. 133–142. 1997.
- [2] H. Chen, W. Zhang, G. Zhu, J. Xie, and X. Chen, "Rethinking cancer nanotheranostics," *Nature Reviews Materials*, 2(7), pp. 1–18. 2017.
- [3] Petersen, "Oral cancer prevention and control—the approach of the World Health Organization," *Oral oncology*, 45(4-5), pp. 454–460. 2009.
- [4] F. Bray, J. Ferlay, I. Soerjomataram, R. L. Siegel, L. A. Torre, and A. Jemal, "Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries," *CA: a cancer journal for clinicians*, 68(6), pp. 394–424. 2018.
- [5] B. Weigelt, J. L. Peterse, and L. J. Van't Veer, "Breast cancer metastasis: markers and models," *Nature reviews cancer*, 5(8), pp. 591–602. 2005.
- [6] S. Lee, H. Kim, and H. Higuchi, "Numerical method for vesicle movement analysis in a complex cytoskeleton network," *Optics express*, 26(13), pp. 16236–16249. 2018.
- [7] S. Lee, H. Kim, and H. Higuchi, "Extended dual-focus microscopy for ratiometric-based 3D movement tracking," *Applied Sciences*, 10(18), 6243.
- [8] J. P. Bergman, M. J. Bovyn, F. F. Doval, A. Sharma, M. V. Gudheti, S. P. Gross, J. F. Allard, and M. D. Vershinin, "Cargo navigation across 3D microtubule intersections," *Proceedings of the National Academy of Sciences*, 115(3), pp. 537–542. 2018.
- [9] S. Lee and H. Higuchi, "3D rotational motion of an endocytic vesicle on a complex microtubule network in a living cell," *Biomedical Optics Express*, 10(12), pp. 6611–6624. 2019.
- [10] S. Lee, H. Kim, M. Ishikawa, and H. Higuchi, "3D Nanoscale tracking data analysis for intracellular organelle movement using machine learning approach," In *Proc. IEEE Int. Conf. Artificial Intelligence in Information and Communication*, pp. 181–184. 2019.
- [11] S. Lee, H. Kim, H. Higuchi, and M. Ishikawa, "Visualization and data analysis for intracellular transport using computer vision techniques," In *Proc. IEEE Sensors Applications Symposium*, pp. 1–6. 2020.
- [12] S. Lee, H. Kim, H. Higuchi, and M. Ishikawa, "Visualization method for the cell-level vesicle transport using optical flow and a diverging colormap," *Sensors*, 21(2), pp. 1–13. 2021.
- [13] T. Lv, X. Wu, L. Sun, Q. Hu, Y. Wan, L. Wang, Z. Zhao, Z. Tu, and Z. X. J. Xiao, "p53-R273H upregulates neuropilin-2 to promote cell mobility and tumor metastasis," *Cell death & disease*, 8(8), pp. e2995–e2995. 2017.
- [14] D. Wang, A. Khosla, R. Gargeya, H. Irshad, and A. H. Beck, "Deep learning for identifying metastatic breast cancer," *arXiv preprint arXiv:1606.05718*. 2016.
- [15] B. E. Bejnordi, M. Veta, P. J. Van Diest, B. Van Ginneken, N. Karssemeijer, G. Litjens, J. A. W. M. van der Laak, and O. Geessink, "Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer," *Jama*, 318(22), pp. 2199–2210. 2017.
- [16] W. Jiao, G. Atwal, P. Polak, R. Karlic, E. Cuppen, A. Danyi, J. de Ridder, C. van Herpen, M. P. Lolkema, N. Steeghs, G. Getz, Q. Morris and L. D. Stein, "A deep learning system accurately classifies primary and metastatic cancers using passenger mutation patterns," *Nature communications*, 11(1), pp. 1–12. 2020.
- [17] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, 27, 2014.
- [18] S. Lee, H. Kim, H. Higuchi, M. Ishikawa, "Classification of Metastatic Breast Cancer Cell using Deep Learning Approach," In *Proc. IEEE Int. Conf. Artificial Intelligence in Information and Communication*, pp. 425–428. 2021.
- [19] J. Kurebayashi, T. Otsuki, C. K. Tang, M. Kurosumi, S. Yamamoto, K. Tanaka, M. Mochizuki, H. Nakamura, and H. Soono, "Isolation and characterization of a new human breast cancer cell line, KPL-4, expressing the Erb B family receptors and interleukin-6," *British journal of cancer*, 79(5), pp. 707–717. 1999.
- [20] S. Lee, H. Kim, and H. Higuchi, "Focus stabilization by axial position feedback in biomedical imaging microscopy," In *Proc. IEEE Sensors Applications Symposium*, pp. 1–6. 2018.
- [21] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*. 2015.
- [22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

# UIRNet: Facial Landmarks Detection Model with Symmetric Encoder-Decoder

Savina Colaco, Young Jin Yoon and Dong Seog Han\*

*School of Electronic and Electrical Engineering*

*Kyungpook National University*

Daegu, Republic of Korea

savinacolaco@knu.ac.kr, skag2603@knu.ac.kr, dshan@knu.ac.kr\*

**Abstract**—One of the challenging problems for facial landmarks detection is learning important features from faces that contain different deformation of face shapes and pose. These important features include eye centres, jawline points, nose points, mouth corners etc that are helpful in various computer vision-related applications. The detection of facial landmarks is difficult when faces have a lot of variation in different conditions. These conditions could be various imaging conditions such as illumination, occlusion, or head poses. In this paper, we propose a deep learning-based facial landmarks detection model called Unet-Inception-ResNet (UIRNet) to predict distinct feature points. The model predicts 68-point landmarks from the detected faces from digital images or video.

**Index Terms**—facial keypoint detection, convolutional neural network, encoder-decoder

## I. INTRODUCTION

Due to various implications of face recognition, the performance gap between machines and the human visual system domain becomes a huge obstacle. Since recognising faces for humans can be done effortlessly but it is a challenging problem for the machines in the computer vision area over many years [1]. In particular, the identification methods for fingerprint or iris scans are more accurate than face recognition. Extensive research has been carried out for face recognition since it is an important method for the identification of the person. Face recognition is related to many domains such as computer and pattern recognition, security, biometrics, neuroscience, and multimedia processing. One of the difficult fields in face recognition is face alignment or facial landmark detection. The facial landmark detection goal is to detect the location of predefined facial landmarks, such as the corners of the eyes, eyebrows, the tip of the nose. It has been widely applied to a large variety of computer vision applications. For example, head pose estimation, facial re-enactment, 3D face reconstruction, etc. Recent advances in facial landmark detection focus on learning vital features from different deformation of face shapes and poses, different expressions, partial occlusions and so on. A simple framework is to construct features to depict the facial appearance and shape information by the convolutional neural networks (CNNs), and then learn a model, to map the features to the landmark locations. A CNN captures the complex semantic relationship between the features for a variety of applications. In this paper, we propose a symmetric

encoder-decoder network with an Inception-ResNet module to better capture the landmarks for the detected faces in real-time.

## II. EXPERIMENT

### A. Implementation Details

The facial landmarks model is trained with a combined dataset of 300W [2] and 300VW [3]–[5] with a total number of 112,111 images. The 300W dataset comprises AFW, HELEN, LFPW, XM2VTS, and IBUG datasets where images are annotated with 68 landmarks. The images in the dataset are resized to  $112 \times 112$  resolution in grayscale. The Keras framework is used for model implementation and trained with a batch size of 32 and epochs of 100. We also apply early stopping once the model performance stops improving on the validation data. The model is continuously optimized with the Adam optimization technique [6] with a learning rate of  $10^{-4}$ . The whole dataset is split with a ratio of 60 to 20% for training and testing subsets. The testing subsets are further split with 20% of validation subsets from testing subsets. For the model training, mean squared error (MSE), which is defined as the average of the square of all of the errors, is used between ground truth and predicted points.

### B. Models

The proposed model adopts symmetric architecture called Unet [7] as a baseline model as shown in Fig. 1. The original Unet architecture is also called a contraction-expansive path or encoder-decoder. The Unet model is effective where the output is of similar size as the input and the output needs that amount of spatial resolution. As the name, the architecture resembles a U shape, which has a downsampling or encoder path on the left part and upsampling or decoder path on the right part. The downsampling path consists of recurring layers of two  $3 \times 3$  convolutions followed by the Hswish activation function and batch normalization. The spatial dimensions are reduced with the application of a  $2 \times 2$  max-pooling operation. This downsampling or encoder path helps to capture the contextual information in the image. Moreover, the number of feature channels is doubled with spatial dimensions being halved. In the upsampling step, several transposed convolutions are used and the corresponding feature map from the encoder is concatenated with  $3 \times 3$  convolution. The final layer has a



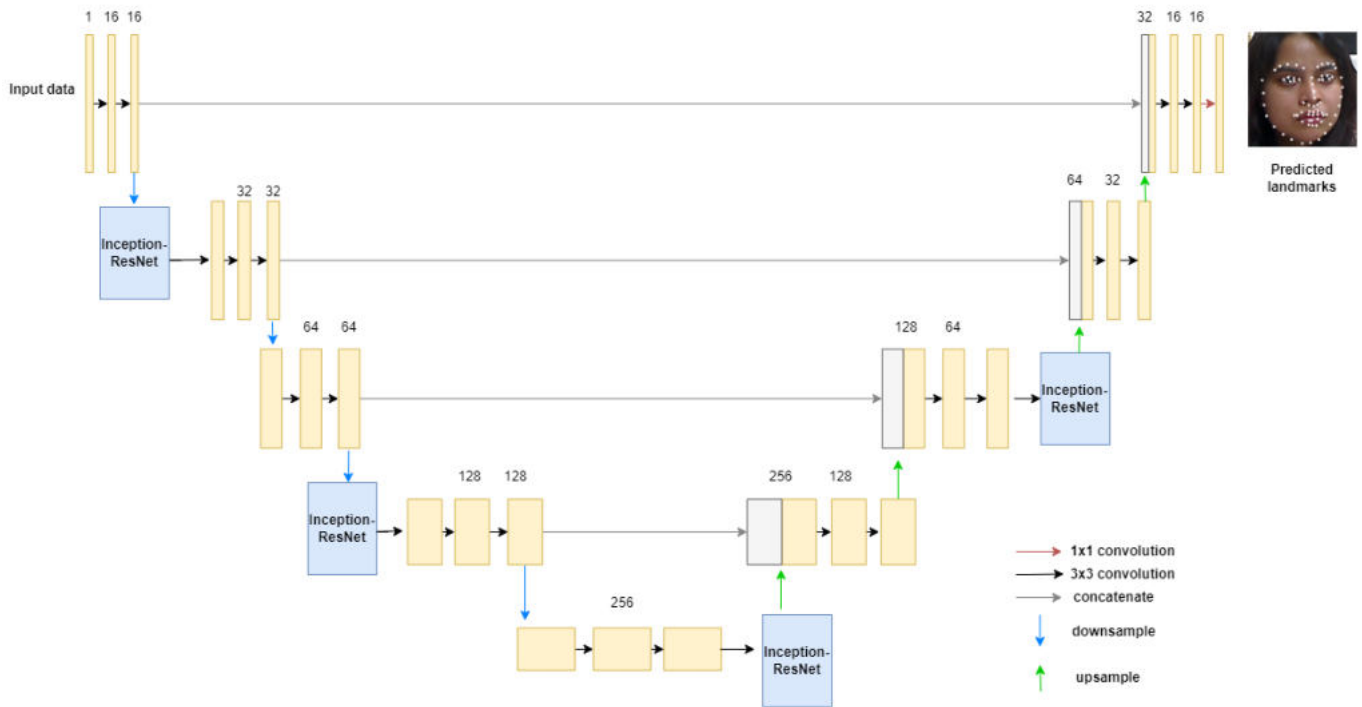


Fig. 1. Proposed symmetric encoder-decoder with Inception-ResNet module.

$1 \times 1$  convolution to map the channels to the desired number of classes. The upsampling step helps to get the precise localization which is important for facial landmarks detection. The number of landmarks to be predicted can depend on the different target tasks. The Unet architecture combines the high-level features which are semantically low from the encoder and reused with upsampled output in the decoder. The proposed model called Unet-Inception-ResNet (UIRNet) is further extended with an inception-resNet module to get a better level of abstraction.

The Inception-ResNet module as shown in Fig. 2 is a tunable structure that gives several possibilities to change the number of filters in the layers. For the model, a different number of filters such as  $1 \times 1$ ,  $3 \times 3$ , dilated filters are used to extract features. The different filters help to concentrate on the different parts of face images to detect facial landmarks. A skip connection performs an identity mapping by adding the original input features to the output of the stacked layers. The Inception-ResNet modules are placed after the 1<sup>st</sup>, 3<sup>rd</sup>, 5<sup>th</sup> and 7<sup>th</sup> stacked convolutions layers of UNet. Each layer in the module is followed by batch normalization and Hswish activation. The Hswish activation function replaces the expensive sigmoid with its piece-wise linear in swish which could be a disadvantage for mobile devices.

### III. DISCUSSION

The facial landmarks detection is being experimented with three models such as simple encoder-decoder, Unet and UIRNet. The simple encoder-decoder model used in the experiment has a similar structure with Unet without the con-

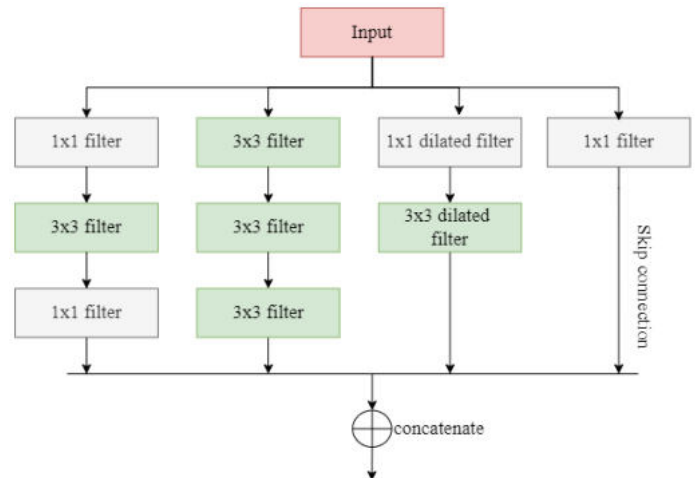


Fig. 2. Inception-ResNet module structure.

catenation of a higher-level feature map from the encoder to upsampled output from the decoder. All the models are evaluated with the MSE loss function to measure the average of the squares of the errors. The faces are detected with the ResNet- single-shot detector(ResNet-SSD) face detector from images or video. The SSD [8] is faster than Faster R-CNN since it does not need an initial object proposals generation step.

The simple encoder-decoder has a similar structure with Unet but without concatenation. As shown in Table 1, it

Table I: Comparison with different CNN models with proposed model

Model	Accuracy	Parameters (in Millions)
Simple encoder-decoder	64%	3.8M
UNet	39%	3.6M
UIRNet	73%	4.3M

achieves 64% of prediction accuracy with 3.8 M total parameters on the combined dataset. The encoder reduces the spatial dimensions in every layer and increases the channels. But decoder increases the spatial dimensions while the reduction in channels. Hence the spatial dimensions are restored to predict each pixel in the input image. Real-time detection of facial landmarks with simple encoder-decoder in Fig. 3, shows approximate localization of facial landmarks. It does not align well around the nose, mouth and jaw area.

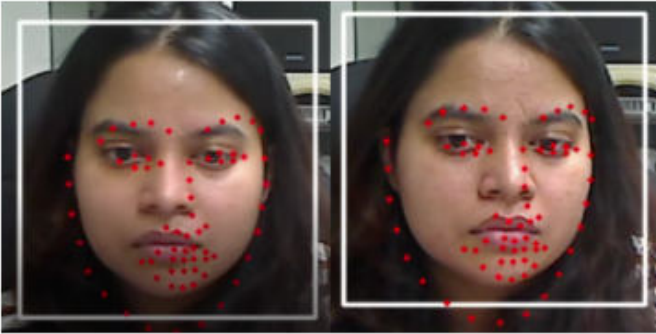


Fig. 3. Facial landmarks detection with simple encoder-decoder.

The Unet which is a symmetric encoder-decoder with the concatenation of high-level features with upsampled output gives 39% of prediction accuracy on the combined dataset with 3.6 M parameters. One of the reasons it shows lower accuracy on the combined dataset is less variation in data needed to tackle different conditions such as head pose, occlusion and illumination. The Unet is limited in extracting complex features from images. In Fig. 4, facial landmarks detected suffers completely with extreme variations in head poses.

The proposed model, UIRNet, is extended with the Inception-ResNet module and achieves 73% of prediction accuracy with 4.3 M parameters on the combined dataset. The Inception-ResNet allows changes to the number of filters in various layers without affecting the quality of the fully trained network. The different filters help to extract features at different scales especially in a different part of the face region. The skip connection added with the Inception makes the architecture deeper to prevent degradation problems. Fig. 5 shows the real-time detection of facial landmarks with the proposed model UIRNet. The UIRNet has better localization of facial landmarks compared to the other two models with most of the facial landmarks. It also shows approximate alignment with different head poses but suffers distortion with

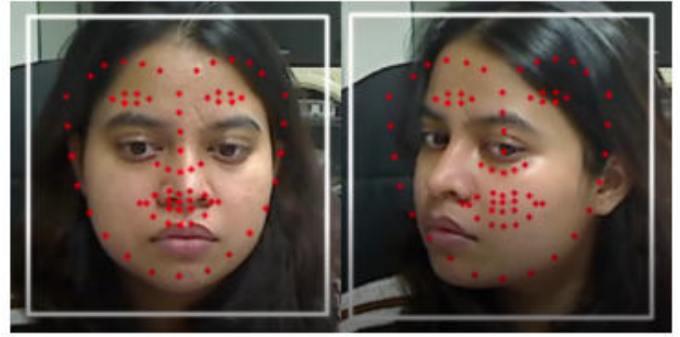


Fig. 4. Facial landmarks detection with Unet.

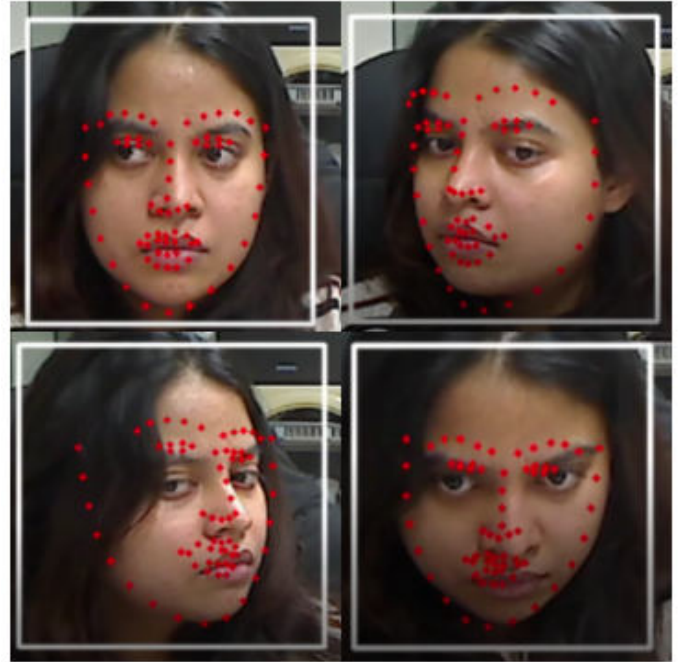


Fig. 5. Facial landmarks detection with UIRNet.

roll rotation around the z-axis.

#### IV. CONCLUSION

In this paper, we proposed a model called UIRNet for predicting facial landmarks in real-time. The UIRNet uses a Unet as a baseline network with the Inception-ResNet module to improve prediction accuracy. We trained and compared our proposed model with other CNN models such as simple encoder-decoder and Unet with the combined dataset of 300W and 300VW. The proposed model showed better prediction accuracy than the other two models. Though prediction accuracy is improved, the model still suffers from large localization errors for extreme variations. For future work, we aim to improve our model for different unconstrained conditions for robust detection.

#### ACKNOWLEDGMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF), funded by the Ministry of Education (2021R1A6A1A03043144).

#### REFERENCES

- [1] Shi, S., Facial Keypoints Detection. ArXiv 2017, abs/1710.05279.
- [2] Sagonas, C.; Tzimiropoulos, G.; Zafeiriou, S.; Pantic, M. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Sydney, Australia, December 2 – December 8 2013 2013; pp. 397-403.
- [3] Chrysos, G. G.; Antonakos, E.; Zafeiriou, S.; Snape, P. Offline deformable face tracking in arbitrary videos. In Proceedings of the IEEE international conference on computer vision workshops, Santiago, Chile, December 7 – December 13 2015; pp. 1-9.
- [4] Shen, J.; Zafeiriou, S.; Chrysos, G. G.; Kossaifi, J.; Tzimiropoulos, G.; Pantic, M. The first facial landmark tracking in-the-wild challenge: Benchmark and results. In Proceedings of the IEEE international conference on computer vision workshops, Santiago, Chile, December 7 – December 13 2015; pp. 50-58.
- [5] Tzimiropoulos, G. Project-out cascaded regression with an application to face alignment. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, June 7 – June 12 2015; pp. 3659-3667.
- [6] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.
- [7] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. MICCAI.
- [8] Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A. C. Ssd: Single shot multibox detector. In Proceedings of the European conference on computer vision, Amsterdam, The Netherlands, October 8 – October 16 2016; pp. 21-37.

# Design and Analysis of an Efficient Energy Sharing System among Electric Vehicles using Evolutionary Game Theory

MD Rizwanul Kabir<sup>1,\*</sup>, Muhammad Mutiul Muhaimin<sup>2</sup>, Md. Abrar Mahir<sup>3</sup> and K. Habibul Kabir<sup>4</sup>

Department of Electrical and Electronics Engineering, Islamic University of Technology (IUT)

Dhaka 1704, Bangladesh

Email: {rizwanulkabir<sup>1,\*</sup>, mutiulmuhaimin<sup>2</sup>, abramahir<sup>3</sup>, habib<sup>4</sup>}@iut-dhaka.edu

**Abstract**—Electric Vehicles (EV) are limited to a short driving range owing to battery constraints. The charging locations of EVs also tend to be quite distant from one another and often times they are not available in many places. The combination of these two problems lead to longer trip durations for an EV since the depletion of the battery requires travelling to distant locations or even taking detours to reach the charging stations. In the proposed network scheme, an EV that lacks energy to complete its trip can request for energy from EVs in its vicinity. The model proposed is based on a variant of evolutionary game theory. This model is premised on the selfishness of each individual EV and is expanded upon using replicator dynamics on graphs. The EV which is requesting charge, known as a *receiver*, offers an incentive to attract other EVs, known as *givers*, to share their energy. The model outlines the procedure by which the regulation of the incentive contributes to a change in the ratio of *givers* in the model. The results demonstrate that an equilibrium can be established where there is the consistent creation of *givers* in the system. This equilibrium is attained by varying the incentive offered by EVs with depleted energy levels. Thus, using a theoretical and numerical approach it is demonstrated that an effective energy sharing system is sustainable.

**Index Terms**—Electric vehicle, evolutionary game theory, replicator dynamics, bi-directional DC-DC conversion

## I. INTRODUCTION

Electric vehicles (EVs) and their widespread use are a major tool in humanity's arsenal to fend off the climate crisis. The transportation industry, in general, stacks up a significant share of the greenhouse gas emissions around the world [1]. Road transport alone contributes approximately 72% of the emissions from this industry [2]. As a result, EVs are likely to play an essential role in curbing greenhouse gas emissions. There is a major hindrance, however, to its adoption in everyday life and global coverage.

Firstly, the distance that can be traversed seamlessly by an EV is circumscribed by its current battery limitations. This drastically reduces the distances that can be travelled by an EV compared to traditional vehicles. A shorter driving range leads to a much more frequent need to visit Electric Vehicle Charging Stations (EVCS) to restore depleted energy levels of the battery. This leads to increased trip durations, consequently, disincentivising consumers from switching to EVs from traditional vehicles.

Secondly, infrastructure issues in the form of limited EVCSs add to the aversion of EVs among the general populace [2].

These stations tend to be few and far apart. The problem of limited driving range gets exacerbated with a paucity of these EVCSs in the vicinity of the drivers. This challenge gets compounded when such public infrastructure is out of service or malfunctioning. As a result, EVs are forced to travel to distant charging locations which leads to extended trip durations. Long queues at sparsely available charging sites can lead to longer charging times as well.

To boost the popularity of EVs- curbing emissions from vehicles in the process- an efficient energy sharing scheme is being propounded in this study. The EVs in this system are either autonomous or semi-autonomous. The bedrock of the scheme being proposed is a variant of an evolutionary game theoretic approach. In addition to the design of a stable and perpetual system, the aim is to establish automation in this system such that any EV with depleted charge levels can collect charge from surrounding EVs upon request. When the energy level drops below a designated threshold, the EV sends out requests using existing communication networks. The EV with depleted energy can be labeled as the *receiver*.

Fig. 1 below illustrates the proposed scenario where a *receiver* in its journey to its destination engages in an energy sharing scheme with other EVs. The EV which accepts the request and takes upon the role of sharing energy with this *receiver* is labeled as *giver*. The request sent out using communication networks reaches to all the EVs which are connected to the network. The distance that an EV can travel with its remaining charge is used to form a circular cluster around it with other EVs along the cluster's radius becoming participants in the game. This game-initiating EV ensures that minimum energy is expended in suit of this energy collection purpose by traveling the shortest path. Owing to the autonomous selection of *giver* EVs in the cluster by local interactions among them, the goal of the scheme is fulfilled.

Due to its logical, strategic decision-making features, the evolutionary game theoretic method is one of the most suited techniques for achieving such a system. It takes into account the process of natural selection which is advantageous in dealing with selfishness among the EVs. It uses a mathematical model called replicator dynamics to characterize it [3].

All participants are assumed to be homogenous agents as EVs with same charging port, equal battery capacity etc. All

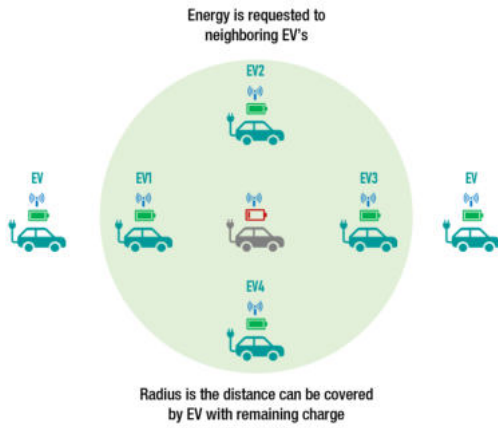


Fig. 1: Proposed scenario.

the EVs along form Internet of things (IoT) devices which communicate with each other on existing networks. Charge transfer occurs using bidirectional DC-DC converter technology using a wired setting. It is heavily employed already in the domain of stored energy systems. This converter technology is nowadays demonstrated to be effective at the transfer of charge between vehicles. Bidirectional DC-DC technology leads to an increase of 1% SOC in 1 minute 8 seconds [4]. The change in SOC is interpreted by the *receiver* and *giver* EVs in terms of the cost per unit they are expending or in terms of the incentive being provided to them. This is further elaborated upon in sections below. This also helps to validate the repetitive nature of the game as every time the *receiver's* charge is depleted, it requests for charge and the game is re-initiated. It is noted that all the EVs and their batteries are part of a vehicle-to-energy network supported via the Internet of Things (IoT)

This process continues till the receiver can reach its destination. Other technologies such as battery swapping and V2G (Vehicle to Grid) charging are also prevalent in the EV charging technology domain. In light of the proposed scheme which requires vehicles to exchange their energy and then continue in their respective journeys, each EV is fitted with this converter technology and its component devices [5], [6]. Afterwards the applicability and suitability of this energy-sharing scheme is discussed where section II ponders upon other studies conducted in relevant fields, section III describes the proposed scheme, section IV demonstrates the numerical results and section V concludes this study.

## II. RELATED WORKS

Vehicle-to-vehicle (V2V) energy sharing technology is becoming increasingly popular. There is research which focuses on the challenge of routing, scheduling, and matching vehicles in a V2V Wireless Power Transfer (WPT) medium on an energy-time extended network, and provides a dynamic programming solution approach to solve it [7]. Other researchers are positing charging sharing schemes among EVs (V2V) which employs inductive power transfer. Those works posited novel solutions into different ways inductive charging can be deployed which are not yet practically implementable [5],

[6]. Research is going on to develop a cost-effective energy-sharing framework for electric vehicles. In one such energy-sharing model, the study offers a comprehensive approach to energy management, integrating the applicability and existence of grids, thus taking a holistic view of energy management [8]. Other works have focused on V2V energy sharing using fog computing. Those works have delved into ensuring security in such transactions using blockchain methods to enable a robust network of EVs [9]. There are other research proposals which suggest the charging of EVs while driving on the road wirelessly [10]. However, such systems entail the overhaul of current infrastructure which can be extremely costly. The advantage of the scheme proposed in this paper is that it does not impose onerous restrictions on the type of infrastructure needed for implementation. The following section dives into the details of the proposed scheme.

## III. PROPOSED SCHEME

The discussion in this section encompasses the formulation of the game to be modeled, the various components of the game and its implementation in the system. The relationship between the EVs in this system is thoroughly outlined alongside the different roles they might play in the game. It is noted that all the EVs and their batteries aided by IoT is the vehicle to energy networking.

### A. Overview

This paper suggests an automated system which aims at increasing the range of EVs and reducing trip durations. The objective is to reduce trip durations by not having to divert from the route of the destination to reach an Electric Vehicle Charging Station (EVCS). In cases where there are no EVCSs in the vicinity, this approach can help the community even more. The design of the system is such that it does not necessitate changes to public infrastructure and bypasses all the bureaucracy and cost its change entails.

The cluster formed around the *receiver* EV dictates the environment of the game. If an EVCS is far away, even within this cluster, and takes the *receiver* farther from its desired route, trip duration can increase. The problem is exacerbated if no EVCSs are found within the cluster. This brings forth the need for energy sharing among EVs. However, the challenge is that all vehicles are potentially selfish and are unwilling to share their energy. As such, a system is proposed that *a)* selects a *giver* EV that rendezvous and shares its energy with the requesting (*receiver*) EV and *b)* takes the vehicle's selfishness into account. The system must not have any form of human intervention.

Here, every individual EV in the overall population formulates a strategy for itself. The strategy that gets chosen by an EV changes in every round of the game. This change can be visualised with the help of replicator dynamics on graphs [11], [12], [13], [14], [15]. Replicator dynamics on graphs is a derivative of the original replicator dynamics in the case of a finite population. According to replicator dynamics, the prevalence of a particular strategy being adopted by EVs in

the system is dependent upon the payoff that those EVs can attain from that strategy. The members of the population of EVs in this case represent the vertices of a regular graph. The utilisation of the scheme allows the selection of certain EVs in a cluster referred to as *giver* EVs which are willing to share their energy with the *receiver* EV. For this work, *giving* and *not giving* are the two allowed strategies in the population upon the *receiver* requesting energy.

To adopt this modeled game, there needs to be the recurrence of the following three stages:

1. *Giver Selecting Stage*: Every individual EV chooses to be a *giver* or a *non-giver* depending on mutual interactions with other EVs in the cluster.

2. *Receiver Responding Stage*: When the selection is complete among the EVs in the cluster, all the EVs communicate with the *receiver* and the *receiver* responds to rendezvous for energy sharing.

3. *Energy Sharing Stage*: Each *giver* meets and shares energy with the *receiver*. *Non-giver* EVs do not meet with any EV and do not take part in energy sharing.

A *round* of the game is defined as the repetition of these three stages which is undergone by the system. It is assumed that all the vehicles synchronise with each other and are cognizant of the duration of the round. The amount of energy intake by the *receiver* and the amount it further needs to reach its destination determines every individual round. The length of the round is communicated among the EVs and can be updated if necessary. This model is a variant of evolutionary game theory based on a self-organized data aggregation technique described in [16].

The fundamental theme propelling this scheme is that EVs are selfish about their energy. As a result, an *incentive* is needed to prod them to share their energy. The *giver* in every round is imparted with a type of benefit to attract them which serves as the *incentive*. This brings forth two challenges: 1) How can the selection of *givers* be done automatically under situations where all EVs are potentially selfish? 2) Is it possible to control the number of *givers*? Here, the application of this variant of evolutionary game theory can assist in resolving these since it takes into account the different influences caused by the mutual interactions among the EVs.

### B. Selection of the Givers

The vehicles in the cluster are under mutual dependency with other cluster members. It is assumed that each node communicates with its neighboring *receivers* and then determines to be a *giver* or a *non-giver* based on the potential benefits it can reap. In this system, let  $c$  be the equivalent amount of credit that is expended as a *giver* travels and consumes battery energy to meet, i.e. equivalent to the decrease in SOC, with a *receiver* for energy sharing purposes. Let, the amount of incentive offered by the *receiver* be  $b$ , i.e. equivalent to the increase in SOC it can attain using that gain in credit. This is the energy equivalent amount of credit that the *givers* are to be provided if they expend  $c$  amount of energy equivalent credit to service the request of *receivers*. The EVs can utilise this  $b$

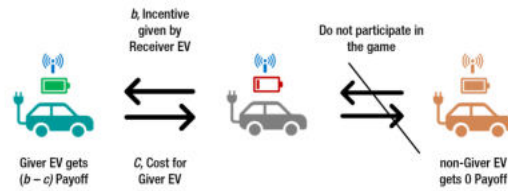


Fig. 2: Selection of *giver* EVs.

amount of credit automatically at any EVCS to get recharged by the energy level afforded by  $b$  without any additional payments. However, since the *non-giver* EVs are not traveling in the quest of sharing energy, they are also not spending energy for the cause. So, *non-givers* are neither losing nor consuming any energy in order to service the request of the *receiver*. Fig. 2 above illustrates the *giver* selection and the corresponding payoff received when the other EV chooses to be a *non-giver*.

Since the EVs are selfish in nature it must be ensured that their energy expenditures are offset by subsequent incentives. Hence, the incentive offered by the *receiver* needs to be a level of  $b$  such that  $b \geq c$ . Intuitively, it can be asserted that larger the  $b$  being offered, the more the number of *givers*. This condition also helps to subdue the number of *non-givers* in a system. This incentive  $b$  can only be offered by the *receiver* which allows it to regulate the number of *givers* created in the system.

Initially, the bargain among vehicles is modeled as a game between two neighboring EVs, i.e. players in evolutionary game theory. There are two roles (strategies) for each player: *giver* (shares energy with the *receiver*) and *non-giver* (declines to share energy with the *receiver*). Table I below illustrates the possible combinations among the two players in this game.

TABLE I: The payoff matrix in between two EVs.

		EV 2	
		Giver	Non-Giver
EV 1	Giver	$\frac{b}{2} - c, \frac{b}{2} - c$	$b - c, 0$
	Non-Giver	$0, b - c$	$0, 0$

The ensuing payoffs for all of the individual scenarios can be modeled by taking the credit offered by the *receiver* and the equivalent credit of the energy consumed by the EVs into account. An individual EV knows its own payoffs in light of the possible strategies of other players, without knowing the exact strategies. A *giver* strategy is likely to be preferred by an EV which results in the highest possible payoff. Each EV is cognizant of the payoff it can receive when the competing EV chooses the same or different strategy.

In a case where one of the EVs is a *giver* with the other being a *non-giver*, the largest gain is by the *giver* whereas the *non-giver* remains at a net neutral point with no loss or profit.

The *receiver* offers the full incentive to the giver ( $b-c \geq 0$ ) for this scenario since other EV chooses to become a *non-giver*.

In the case where none of the EVs choose to share their energy, i.e. both choose to be *non-givers*, they encounter no net loss but they also do not get the potential profit that sharing their energy could have reaped.

In the case where both the neighboring EVs decide to become *givers*, the *receiver* meets with both the vehicles for energy sharing. As a result, the incentive offered to each car is halved ( $\frac{b}{2} \geq c$ ). Afterwards, it is possible to abstract Table I into Table II.

TABLE II: The abstracted payoff matrix in between two EVs.

EV 1 \ EV 2		Giver	Non-Giver
		Giver	$P, P$
Non-Giver	$S, T$	$R, R$	

In the abstracted payoff matrix,  $T > R$  and  $P > S$ . In this proposed scenario every vehicle is incentivised to become a giver as  $T > R$ . The larger the  $b$ , the greater is the incentive. Resultantly, this proves that the *receiver* EV can change the value of  $b$  to control the creation of *givers*. In addition, if both EVs taken into consideration choose to be *givers*, the *receiver* can take energy from both the EVs offering half the incentive.

Furthermore, the condition  $T + S > R + P$  facilitates the selection of a role which is an evolutionarily stable strategy [17]. The most preferred role is the case where a player can get the highest reward for sharing their energy while the other player does not wish to share their energy. In conjunction with the payoff matrix and evolutionary game theory, when every EV undertakes appropriate strategies to maximise its own payoff, then the system reaches a fully stable situation where both *givers* and *non-givers* stably coexist [13]. A distinguishing characteristic of this model is that the selections are entirely dependent on the mutual interactions among the vehicles by taking into account the selfishness of each player. The time it takes for charging during each interaction is assumed to be constant. This is predicated upon the homogenous nature of the EVs considered and hence, the latency of the giver and receiver side is also assumed to be constant throughout the system.

#### IV. ANALYTICAL RESULTS

In this section, the relationship between the ratio of *givers* and the parameters of the payoff matrix is derived analytically with the aid of replicator dynamics on graphs [12], [15]. Each EV derives a payoff from the interactions with all of its neighbors. Afterwards, a comparison is performed in terms of the obtained payoff with a randomly chosen neighbour. The replicator equations on graphs are delineated upon along with a thorough elucidation of the analytical results obtained. Furthermore, the future scope of this research which can further validate this work is also expanded upon.

#### A. Replicator Equation on Graphs

In order to start the derivation of replicator equation on graphs [13], consider the ratio of *giver* EVs to the total number of EVs as  $x$ . On the other hand,  $1-x$  is the ratio of the *non-giver* EVs to the total number of EVs. The expected fitness  $f_1$  and  $f_2$  are given by

$$\begin{aligned} f_1 &= x\left(\frac{b}{2} - c\right) + (1-x)(b-c), \\ f_2 &= 0. \end{aligned} \quad (1)$$

Let  $k$  denote the number of neighbors of each EV, called degree of graph [14]. Despite the fact that the analysis given is based on the  $k$ -regular graph [12], the method may also be applied to non-regular graphs, like unit disk graphs [12], [14]. This allows us to visualise on graphs how the total number of EVs in a cluster can affect the ratio of *givers* at a certain incentive level. The sum of the original payoff matrix and a modifier matrix is the modified payoff matrix for evolutionary game theory on graphs [13]. The modifier matrix is given as Table III.

TABLE III: Modifier matrix between two EVs.

EV 1 \ EV 2		Giver	Non-Giver
		Giver	$0, 0$
Non-Giver	$-m, m$	$0, 0$	

$$m = \frac{(k+3)\left(\frac{b}{2} - c\right) + 3(b-c)}{(k+3)(k-2)}, \quad \forall k > 2. \quad (2)$$

The local competition among the strategies is denoted by  $m$  (as shown in Table III). Each tactic's gain is another's loss, and local competition between the identical strategies yields zero. The expected payoff for the local competition  $g_1$  and  $g_2$  of *giver* and *non-giver* are

$$\begin{aligned} g_1 &= (1-x)m, \\ g_2 &= -xm. \end{aligned} \quad (3)$$

Consequently, the average payoff consisting of the two strategies becomes

$$\phi = x(f_1 + g_1) + (1-x)(f_2 + g_2). \quad (4)$$

Putting the values from Eqns. (1) and (3) into Eqn. (4),

$$\phi = x\left[x\left(\frac{b}{2} - c\right) + (1-x)(b-c)\right]. \quad (5)$$

From Eqns. (1), (3) and (5), the replicator equation on graphs [13] is found for  $k > 2$  to be

$$\dot{x} = x(f_1 + g_1 - \phi).$$

This can be manipulated to

$$\dot{x} = x(1-x)\left[-\frac{xb}{2} + (b-c) + \frac{k\left(\frac{b}{2} - c\right) + 4.5b - 6c}{(k+3)(k-2)}\right], \quad \forall k > 2.$$

$\dot{x}$  represents the derivative of  $x$ . Therefore, in order to obtain the maxima of  $x$ ,  $\dot{x} = 0$  is taken. Therefore,  $x^* = 0, 1$  and

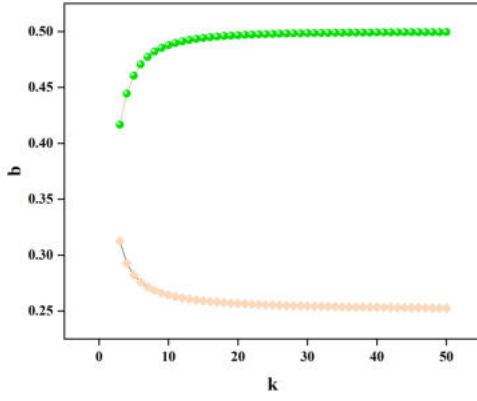


Fig. 3: The supremum and infimum of  $b$  varying with  $k$ .

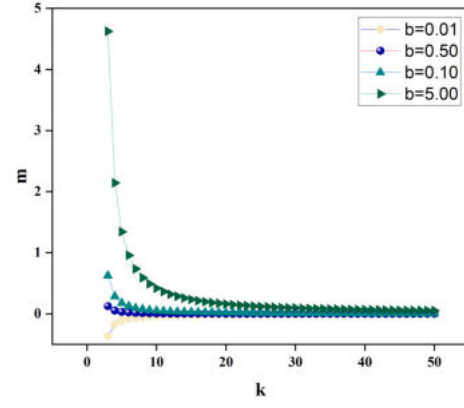


Fig. 4: Modifier  $m$  controlled by changing the value of  $b$ .

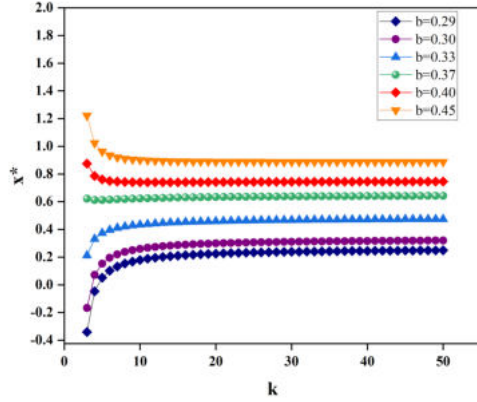


Fig. 5: The value of  $x^*$  controlled with the value of  $b$ .

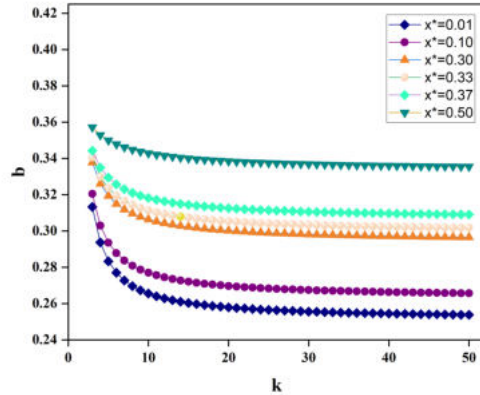


Fig. 6: The value of  $b$  controlled with the value of  $x^*$ .

$$x^* = \frac{2}{b} \left[ (b - c) + \frac{k(b - c) + 4.5b - 6c}{(k + 3)(k - 2)} \right], \quad \forall k > 2. \quad (6)$$

Now, this is feasible for  $0 < x^* < 1$ . So, from Eqn. 6,

$$\frac{c(k^2 + 2k)}{k^2 + \frac{3k+9}{2} - 6} < b < \frac{2c(k^2 + 2k)}{k^2 + 2k + 3}, \quad \forall k > 2 \quad (7)$$

It is found

$$0 < \frac{c(k^2 + 2k)}{(k^2 + \frac{3k+9}{2} - 6)} < \frac{2c(k^2 + 2k)}{(k^2 + 2k + 3)}, \quad \forall k > 2$$

such that  $b > c$ . The numerical results with the aid of figures are expanded in the following section. The equilibrium achieved in Eqn. (6) plays a vital role in controlling this model. The impacts of the different variables and their inter-relationship is discussed in the end.

### B. Numerical Results

The three variables that the model depends on are  $b$ ,  $c$  and  $k$ . The values of these variables affect the value of  $x$ . In this scheme,  $c$  is a very prominent variable in the case of all the graphs obtained from replicator dynamics. For the purpose of this study, we have considered  $c$  to be equivalent to 0.25 units but  $c$  can be chosen as any value depending on how far a *giver* needs to travel for upholding our model. We have

kept the value sufficiently small for the ease of calculation. All the following graphs have been extrapolated keeping this assumption in mind, without loss of generality. The units of  $b$  and  $c$  must be the same. Also, the value of  $c$  would change according to the pricing scheme. In short, the value of  $c$  in our work is just an assumption. It is not universal. And this value and its unit impacts the entire system comprising of  $b$  and  $x^*$ .

Fig. 3 demonstrates the supremum and infimum of  $b$  as satisfied by Eqn. 7. It is observed that even in the worst case, the value of  $b$  cannot be less than  $c$ , which satisfies a cornerstone of this work stating that  $b \geq c$ . Furthermore, it is observed that the maximum value of  $b$  results in  $2c$ . This substantiates the model because it proves that the *givers* will not lose. This also clarifies that the *givers* cannot demand an unreasonable amount of credit, since this is controlled by the saturation of  $b$  for higher numbers of EV in the radius.

Afterwards, Fig. 4 represents the local competition among the EVs. The modifier equation is represented from Eqn. 2. The modifier graph converges to 0 for different values of  $b$  with large  $k$  values which implies that the likelihood of EVs all choosing similar strategies is minimal.

Further, the variation of the value of  $x^*$  for the controlling of the value of  $b$  is also evident from Fig. 5. Here,  $x^*$  can be any value in between 0 and 1, depending on  $b$  for a fixed



number of EVs. It can be deduced that for a specific  $b$ ,  $x^*$  does not change when  $k$  becomes large owing to the fact that for limiting value of  $k$ ,  $m$  tends to zero as illustrated in Fig. 4. On the other hand, for relatively smaller values of  $k$ , for instance, less than 15,  $x^*$  shows different characteristics, depending on  $b$ . With decreasing values of  $b$ , the value of  $x^*$  decreases and vice versa. Therefore, our model is valid since the changing of incentive  $b$  affects the ratio of *givers*  $x^*$  in the system. As  $x^*$  can be controlled by  $b$ , similarly,  $b$  can also be controlled by  $x^*$ . This is evident from Fig. 6. In essence, this means that, the more the number of *givers* in the vicinity, the more the *giver* will have to pay, unless the excessive amount of EVs has already saturated the value of  $b$ .

### C. Future Scope

Future works can be directed at further validating this model with the aid of agent-based dynamics to conduct micro level experiments. These experiments are to be worked on to ensure these mathematical models are elaborated upon. It can allow the model to be viewed from a micro level perspective in contrast to the macro level perspective present here. Micro level considerations can be included in these studies ranging from time, distance and the relative velocity of the EVs. Since in this analysis time is assumed to be constant, future work can include the factor of time and thus form a dynamic game. A dynamic game can further be executed with the aid of Reinforcement learning (RL). It can then elucidate the latency of the *giver* and *receiver*. Varying latencies of *giver* and *receiver* can be investigated. These considerations can reinforce the robustness of this proposed model.

## V. CONCLUSION

Widespread acceptance of EVs as viable alternatives is stunted by many impediments. The main reason for this is range anxiety. The objective of this study is reduce trip durations for EVs by providing charging sources where there are no charging stations nearby, whittling down trip durations in the process. This is achieved by the design of an efficient model to share energy among EVs which is conducted autonomously with the aid of a variant evolutionary Game Theory. It paves the way for the creation of a system where an EV can collect energy from other EVs by touting an incentive. The level of incentive offered by *receivers* is used to control and induce the creation of *givers*, thereby, formulating an automated system of EVs connected in a network to share energy amongst themselves. The elucidation of the model using replicator dynamics on graphs in conjunction with the numerical results allows the comprehensive analysis of this proposed framework. The theoretical and numerical approach employed here substantiates the equilibrium of *givers* and the corresponding incentives that regulate this equilibrium. In this way an efficient energy sharing scheme among electric vehicles can be established.

## REFERENCES

- [1] Kawamoto, R., Mochizuki, H., Moriguchi, Y., Nakano, T., Motohashi, M., Sakai, Y., Inaba, A.: Estimation of CO2 Emissions of internal combustion engine vehicle and battery electric vehicle using LCA. *Sustainability (Switzerland)* **11**(9) (2019). <https://doi.org/10.3390/su11092690>
- [2] Bobanac, V., Pandzic, H., Capuder, T.: Survey on electric vehicles and battery swapping stations: Expectations of existing and future EV owners. In: *IEEE International Energy Conference* pp. 1–6 (2018). <https://doi.org/10.1109/ENERGYCON.2018.8398793>
- [3] Weibull, J.W.: *Evolutionary Game Theory*, vol. 148. MIT Press (1997)
- [4] Vempalli, Sai K., K, Deepa, G, Prabhakar: A Novel V2V Charging Method Addressing the Last Mile Connectivity. *IEEE*, 1–6 (12 2018). <https://doi.org/10.1109/PEDES.2018.8707602>
- [5] Dutta, P.: Coordinating rendezvous points for inductive power transfer between electric vehicles to increase effective driving distance. In: *International Conference on Connected Vehicles and Expo, ICCVE - Proceedings* pp. 649–653 (2013). <https://doi.org/10.1109/ICCVE.2013.6799872>
- [6] Dutta, P.: Use of inductive power transfer sharing to increase the driving range of electric vehicles. In: *IEEE Power and Energy Society General Meeting* (2013). <https://doi.org/10.1109/PESMG.2013.6672635>
- [7] Abdolmaleki, M., Masoud, N., Yin, Y.: Vehicle-to-vehicle wireless power transfer: Paving the way toward an electrified transportation system. *Transportation Research Part C: Emerging Technologies* **103**, 261–280 (2019). <https://doi.org/https://doi.org/10.1016/j.trc.2019.04.008>
- [8] Shurrab, M., Singh, S., Otrok, H., Mizouni, R., Khadkikar, V., Zeineldin, H.: An efficient vehicle to vehicle (v2v) energy sharing framework. *IEEE Internet of Things Journal* (2021). <https://doi.org/10.1109/JIOT.2021.3109010>
- [9] Sun, G., Dai, M., Zhang, F., Yu, H., Du, X., Guizani, M.: Blockchain-enhanced high-confidence energy sharing in internet of electric vehicles. *IEEE Internet of Things Journal* **7**(9), 7868–7882 (2020). <https://doi.org/10.1109/JIOT.2020.2992994>
- [10] Lee, S., Huh, J., Park, C., Choi, N.S., Cho, G.H., Rim, C.T.: On-Line Electric Vehicle using inductive power transfer system. In: *IEEE Energy Conversion Congress and Exposition, ECCE - Proceedings* pp. 1598–1601 (2010). <https://doi.org/10.1109/ECCE.2010.5618092>
- [11] Ohtsuki, H., Nowak, M.A.: Evolutionary stability on graphs. *Journal of Theoretical Biology* **251**(4), 698–707 (2008). <https://doi.org/10.1016/j.jtbi.2008.01.005>
- [12] Ohtsuki, H., Nowak, M.A.: The replicator equation on graphs. *Journal of Theoretical Biology* **243**(1), 86–97 (2006). <https://doi.org/10.1016/j.jtbi.2006.06.004>
- [13] Nowak, M.: *Evolutionary dynamics: exploring the equations of life*, vol. 51. Belknap Press/Harvard University Press (2008)
- [14] Ohtsuki, H., Hauert, C., Lieberman, E., Nowak, M.A.: A simple rule for the evolution of cooperation on graphs and social networks. *Nature* **441**(7092), 502–505 (2006). <https://doi.org/10.1038/nature04605>
- [15] Szabó, G., Fáth, G.: Evolutionary games on graphs. *Physics Reports* **446**(4-6), 97–216 (2007). <https://doi.org/10.1016/j.physrep.2007.04.004>
- [16] Kabir, K.H., Sasabe, M., Takine, T.: Evolutionary game theoretic approach to self-organized data aggregation in delay tolerant networks. In: *IEICE Transactions on Communications* **E93-B**(3), 490–500 (2010). <https://doi.org/10.1587/transcom.E93.B.490>
- [17] Osborne, M.J., Rubinstein, A.: *A course in game theory*, vol. 5. MIT press (1994)

# GAN-based Data Augmentation for UWB NLOS Identification Using Machine Learning

Duc Hoang Tran  
Department of Electronics Engineering  
Kookmin University  
Seoul 136-702, Korea  
duchoangbkdn.1995@gmail.com

ByungDeok Chung  
ENS. Co. Ltd  
Ansan 15655, Korea  
bdchung@ens-km.co.kr

Yeong Min Jang  
Department of Electronics Engineering  
Kookmin University  
Seoul 136-702, Korea  
yjjang@kookmin.ac.kr

**Abstract**— Indoor position system based on ultra-wideband technology was recognized recently as its great potential to guarantee accurate localization. Non-line-of-sight identification attracts lots of attention. Extracted from the different characters of channel impulse response using Machine Learning is proposed to reduce the localization error, caused by non-line-of-sight condition. In this paper, we proposed an efficient method using Generative Adversarial Network for data augmentation cooperating with autoencoder for enhancing the training model. The results show our framework obtained state-of-art identification performance.

**Keywords**— ultra-wideband, generative adversarial network, non-light-of-sight, machine learning

## I. INTRODUCTION

Ultra-wideband (UWB) communication recently is one of the most outstanding technologies because of its low-power consumption, high accuracy, robust operation, and low complexity for building accurate Indoor Positioning System (IPS). However, obstacles and other interferers can lead to non-line-of-sight (NLOS) situations between the emitter and the receiver as well as decreasing the ultimate location in indoor environment [1]. Therefore, NLOS detection is important to improve the overall performance of UWB-based systems by excluding or correcting NLOS contaminated distance before employing range estimation.

There are a lot of methods are discussed for NLOS identification. These methods can be classified into four groups. In the first method, they try to find the differences of the estimated distance information under Light-of-Sight (LOS) or NLOS conditions based on a detection threshold determined by mean values of Gaussian distribution for each situation respectively [2]. The second method use the energy of the first path is dramatically greater than the energy of the delayed paths in channel impulse response (CIR) [3]. However, many factors influence the signal propagation path loss model, and manually picked features may not be sufficient for LOS/NLOS classification. The third method is built based on the context awareness of the mobile user or environment data (e.g., geometries and attenuation factors) [4]. However, this method requires large computation and also a prior position is necessary. Inspired by the superior performance of machine learning method for data classification and tackle the imbalance samples problem in NLOS and LOS dataset, we proposed a novel UWB NLOS detection and classification method based on Generative Adversarial Network (GAN) and improved Machine Learning (ML) using Autoencoder network which demonstrate the enhancement of detection accuracy. The remainder of the paper is organized as follows. Section II presents the NLOS

problem and CIR model; Then, we describe imbalance data issues and data augmentation method to tackle this issue in Section III; Section IV provides detail of our frameworks; we also detail the experimental result and comparison with others method, some discussion and future works are given.

## II. LOS/NLOS PROBLEM STATEMENT

Distance information from various channels is used to determine location results in UWB-based IPS. The Time of Arrival (TOA) method is used in UWB-based IPS for distance calculation because it provides more accurate. Meanwhile, the CIR measurements are used to obtain distance information by using the TOA technique. Figure 1 shows the CIRs of single preamble pulse generated by the LOS and NLOS signal. As can be observed, the magnitude of the LOS is substantially larger than that of the NLOS, and the curves are different. The CIR can be regarded as time series while the NLOS and LOS CIRs differ owing to the differing transmission pathways. Indeed, the NLOS reception affects the CIR curves heavily. By using machine learning methods, we can employ to deal with NLOS/LOS classification directly utilizing the CIR as the input vector.

## III. DATA AUGMENTATION METHOD

Data imbalance is one of the main challenges of LOS/NLOS classification in UWB systems. Suffering from the data imbalance problem, various ML-based methods cannot correctly identify NLOS from minority classes. Thus, reducing the interference of the imbalanced dataset to improve detection accuracy in UWB-based IPS is also a fundamental issue. In practice, the works of data augmentation in the time-series region are very limited and mostly focus on the traditional data transformation methods such as jittering,

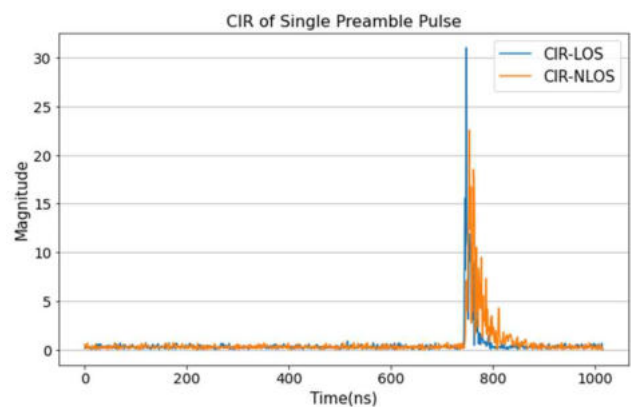


Figure 1: CIRs under LOS and NLOS condition

scaling, window slicing, and flipping and do not significantly improve the accuracy of the model [5]. The most popular generative method in data augmentation is the generative adversarial network (GAN) [6]. The GAN algorithm is mostly applied in image processing and image generation and recently it also provides ability to increase performance in time series data. In this paper, we generate the NLOS signal to solve imbalance data issues. Using different approaches in both the experiment and test, we evaluated the generated data comprehensively and avoided misjudging during the data generation process for obtaining the final results.

#### IV. PROPOSED FRAMEWORK

In general, data augmentation is mostly used in image processing because it is easy to evaluate whether the generated data is similar to the original data based on the judgement of human. Moreover, much recent research proposed simple model ML with low accuracy in training or the use Deep Learning with high complex but still not archive high performance. In our framework, we proposed training and testing process using both generated data by GAN and real data shown in Figure 2. We introduce an adaptive update strategy for Machine learning model with the Autoencoder network, so that the encoder as a data preparation step when training a machine learning improve detection model.

In practical, we introduce data augmentation using GAN to generate NLOS data similar to the original data. With different approaches and AI models, we can guarantee the evaluation process with high accuracy and similarity with the original data, which can help to improve the predictive model.

##### A. Machine learning model and Deep Autoencoder

The ML models analyzed in this study are Logistic Regression (LR) [7], Random Forests [8], Support Vector Machines (SVM) [9] and XGBoost Classifier [10]. These models have proved to be robust for classification applications. Moreover, these methods are very flexible when deal with different data types and structures. However, these model unoptimized in most of recently research on LOS and NLOS classification. Thus, we propose the framework using the encoder in Autoencoder network to perform feature extraction on raw CIR data that can be used to prepare input data before train and evaluate with different machine learning models.

Firstly, we provide an autoencoder network to learn compressed representation of raw data. In this scheme, the encoder compresses the input, and the decoder attempts to regenerate the input from the compressed data provided by encoder. Then, we only use encoder for feature extraction on raw data and it can be as a feature vector in a supervised learning model, for visualization, or more generally for dimensionality reduction. Figure 3 shows the loss function of the autoencoder framework while Table I summarizes the model training

##### B. GAN

GAN is a technique designed by Ian Goodfellow [6] to generate new data from a fixed training data set. In this technique, the discriminative and the generative neural networks compete in a zero-sum game to improve themselves. Using a limited training set, the GAN techniques learn by themselves to generate data using the specific structure [11]. The most well-known GAN applications are those in

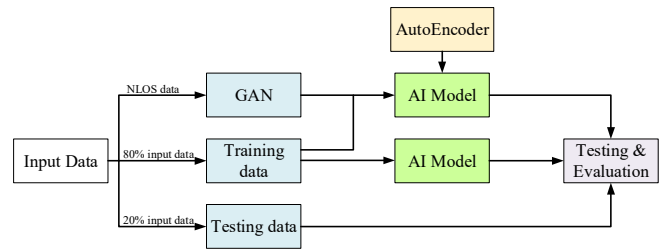


Figure 2. Training process using both real data and generated data by GAN

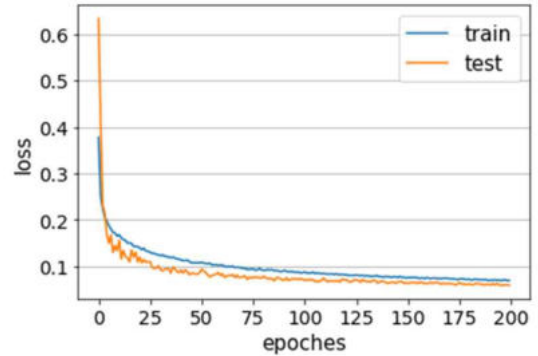


Figure 3. The Autoencoder loss

computer vision, in which a photograph set is trained to generate new output with realistic characteristics for human observers.

The adversarial procedure is illustrated in Figure 3. Most existing GANs perform a similar adversarial procedure in different adversarial objective functions. In this paper, the GAN algorithm is used to generate the NLOS data signal; therefore, only NLOS data is fed into the generator. The generator generates the NLOS data using random noise, which ranges from 0 to 1 with normal distribution to guarantee the difference in the output data. Meanwhile, the discriminator distinguishes the generated samples and the data samples. Given adequate capacity and training time, the generative neural network and the discriminator network will converge and achieve a point where the generator produces samples so real that make the discriminator cannot distinguish them from the real data.

TABLE I  
MODEL ARCHITECTURE FOR FEATURE EXTRACTION

	Layer (type)	Output Shape
Encoder	Input Layer	[(None, 1016)]
	Dense	(None, 2032)
	Batch Normalization	(None, 2032)
	Leaky ReLU	(None, 2032)
	Dense	(None, 1016)
	Batch Normalization	(None, 1016)
	Leaky ReLU	(None, 1016)
	Dense	(None, 1016)
Decoder	Dense	(None, 1016)
	Batch Normalization	(None, 1016)
	Leaky ReLU	(None, 1016)
	Dense	(None, 2032)
	Batch Normalization	(None, 2032)
	Leaky ReLU	(None, 2032)
	Dense	(None, 1016)

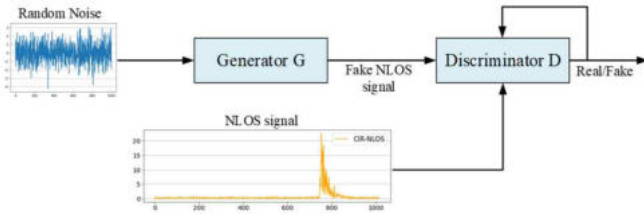


Figure 5. GAN model for NLOS data augmentation

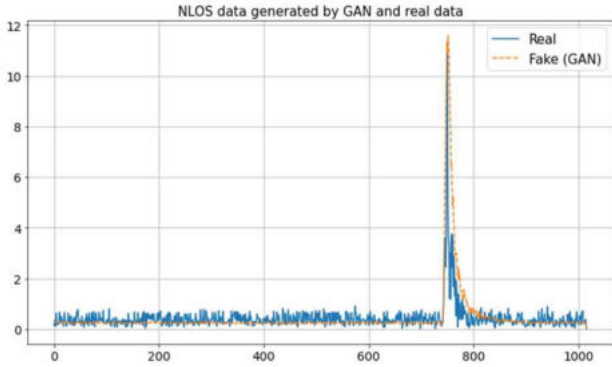


Figure 4. Data Generation using GAN

### C. Data Generating

The preprocessing procedure for the generated signal is the same as that for the original signal, and, based on that, we can evaluate its quality using previous LOS/NLOS detection methods. Note that we generate the signal only for the NLOS signal because this signal is assumed to be less than the signal obtained for the original data.

In our proposed framework, we trained 500 sample NLOS using GAN with noise dimension was 32 and 1016 features CIR respectively with real input data. After completing the generation, we provided these synthetic sample for training dataset and execute ML model for classification. The results archived from our proposed framework was presented in subsection IV.D.

### D. Classification Accuracy Comparison

After selecting approximate Autoencoder model, this subsection aimed to evaluate these ML model with our proposed framework. Autoencoder and ML consisted of LR, RF, SVM and XGBoost were deployed to classify the NLOS/LOS in UWB-based IPS. Besides, we generated NLOS data for solve imbalance issues using GAN which described as subsection IV.B. For comparing these methods fairly, they were incorporated with the parameter. Dataset for training and testing were the same as that described in Figure 2.

For evaluation of our algorithm in various conditions, we utilized a dataset from different locations described in [12]. Specifically, the dataset was created by SNP-N-UWB board with DWM1000 UWB radio module in seven different indoor locations. After applying the original model with two scenarios: (1) comparison among different locations and (2) comparison among different ranging distances, we perform k-fold cross validation with  $k=10$ . The result according to Figure 6 show that XGBoost has a higher value of average detection accuracy than other traditional methods for all location. The high performance of XGBoost has been achieved thanks to the optimization of this algorithm. Besides, these results imply that different contexts have an impact on UWB signal quality,

causing signal classification to be inconsistent under LOS and NLOS conditions. ML algorithms still perform well at NLOS/LOS classification as indicated by the fact that the accuracy is still relatively good, averaging 80% and higher. To enhance the traditional ML, we also use the encoder layer from Autoencoder model which was defined in Table I. For instance, the accuracy of SVM support with encoder layer increased more than the original one from 2-3% due to extracting important feature of signal before putting data into SVM model. Figure 6 also shows this advanced SVM has the most efficient detection accuracy compared to the other algorithms. However, the result still not be improved with lower ranging because of imbalance issue.

We deployed our model with the ranging distance from 0 to 2m which we applied GAN for data augmentation before evaluating the NLOS identification models. The classification accuracy comparison results were listed in the Table II. In without GAN scenario, we archived 79.92% to 83.58% based on LR, RF, SVM and XGBoost method. Specifically, XGBoost show the highest accuracy when identifying NLOS signal. After applying encoder layer, we got accuracy up to 82.25% with LR and 83.67% with SVM, that imply the improvement of classification accuracy when we deploy Autoencoder for transform input data before training model. Moreover, with data augmentation by using GAN, ML model perform more accurate than that without GAN. These results show that our proposed GAN can solve data imbalance and it can reach the accuracy of ML up to 86.98%. Specifically, SVM performed the best result for NLOS identification compare with another ML which supported by encoder processing and GAN.

Data augmentation is useful in the training process when the number of NLOS samples is so small that the model cannot be trained effectively. This characteristic is very suitable in NLOS detection in UWB-based IPS because of the lack of NLOS signal at the start of the implementation phase. With the improvement of GAN, we can generate NLOS data for applying the classification with high similarity to the original

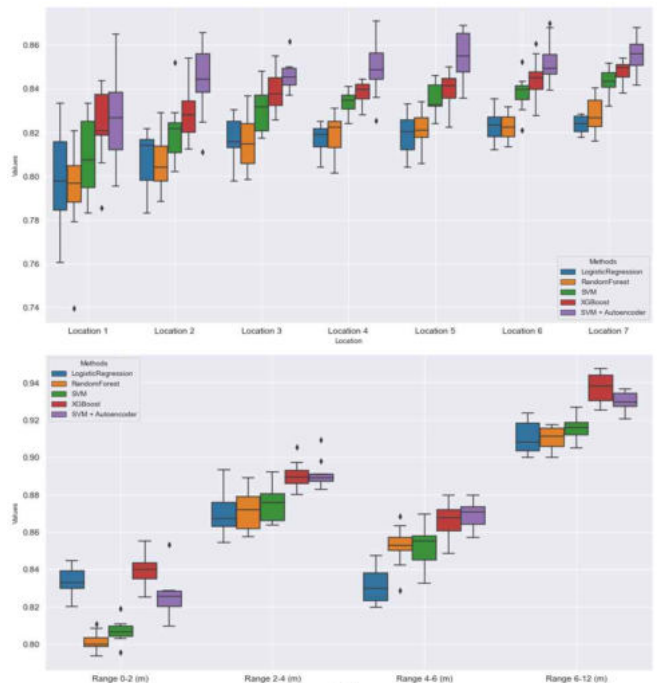


Figure 6. Accuracy of NLOS detection based on ML

TABLE II  
MODEL EVALUATION WITH GAN USING DIFFERENT APPROACHES

	Method	Accuracy (%)
No-GAN	Logistic Regression	79.92
	Random Forests	80.42
	Support Vector Machines	82.00
	XGBoost Classifiers	83.58
	RF + Autoencoder	82.25
	SVM + Autoencoder	83.67
GAN	Logistic Regression	82.58
	Random Forests	81.45
	Support Vector Machines	85.12
	XGBoost Classifiers	86.67
	RF + Autoencoder	83.33
	SVM + Autoencoder	86.98

data. Using various experiments and evaluations, we can conclude that the generated data has a high similarity with the original data in both the time domain and frequency domain. The generation data significantly improved the application of training performance with a large CIR samples in UWB-based IPS. Although we could generate high-quality input data, the original NLOS data are also necessary for testing and partial training.

## V. CONCLUSIONS

This study proposed a novel method to generate the NLOS signal data, thus enhancing NLOS identification accuracy in the case of a limited dataset for training. After testing, we conclude that the generated data has high similarity to the original data and significantly improves the accuracy of the model with limited real NLOS data in the training dataset.

However, the data augmentation method using GAN still has a limitation, since the high variety can reduce the output signal and unstable during the training process. Therefore, the architectures of both the generator and discriminator should be considered carefully, and the output of GAN has to be carefully evaluated. With these remain challenges, we

consider providing other generative AI models for the data augmentation and compare with the current scheme.

## ACKNOWLEDGMENT

This work was supported by the Technology development Program (S3098815) funded by the Ministry of SMEs and Startups (MSS, Korea).

## REFERENCES

- [1] X. Yang, "NLOS mitigation for UWB localization based on sparse pseudo-input Gaussian process", *IEEE Sensors J.*, vol. 18, no. 10, pp. 4311-4316, May 2018.
- [2] J. Khodjaev et al., "Survey of NLOS identification and error mitigation problems in UWB-based positioning algorithms for dense environments", *Ann. Telecommun. Ann. Commun.*, vol. 65, no. 5, pp. 301-311, Jun. 2010.
- [3] K. Bregar et al., "NLOS channel detection with multilayer perceptron in low-rate personal area networks for indoor localization accuracy improvement", *Proc. 8th Jožef Stefan Int. Postgraduate School Students Conf.*, vol. 31, pp. 1-8, May 2016.
- [4] S. S. Wu et al., "NLOS error mitigation for UWB ranging in dense multipath environments", *Proc. IEEE Wireless Commun. Netw. Conf.*, pp. 1565-1570, Oct. 2007.
- [5] Iwana, B.K.; Uchida, S. Time Series Data Augmentation for Neural Networks by TimeWarping with a Discriminative Teacher. arXiv 2020, arXiv:2004.08780.
- [6] Goodfellow, I.J. NIPS 2016 Tutorial: Generative Adversarial Networks. arXiv 2017, arXiv:1701.00160.
- [7] Tolles, Juliana; Meurer, William J (2016). "Logistic Regression Relating Patient Characteristics to Outcomes". *JAMA*. 316 (5): 533-4. doi:10.1001/jama.2016.7653. ISSN 0098-7484. OCLC 6823603312. PMID 27483067.
- [8] Breiman, L. Random Forests. *Mach. Learn.* 2001, 45, 5-32
- [9] Vapnik V.N. *The Nature of Statistical Learning Theory*. Springer; Berlin, Germany: 1995.
- [10] Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. In 22nd SIGKDD Conference on Knowledge Discovery and Data Mining, 2016.
- [11] Bui, V.; Pham, T.L.; Nguyen, H.; Jang, Y.M. Data Augmentation Using Generative Adversarial Network for Automatic Machine Fault Detection Based on Vibration Signals. *Appl. Sci.* 2021, 11, 2166.
- [12] Bregar, Klemen & Hrovat, Andrej & Mohorcic, Mihael. (2016). NLOS Channel Detection with Multilayer Perceptron in Low-Rate Personal Area Networks for Indoor Localization Accuracy Improvement.

# BER Minimization by User Pairing in Downlink NOMA Using Laser Chaos-Based MAB Algorithm

Masaki Sugiyama

Department of Electrical Engineering  
Tokyo University of Science  
Tokyo, Japan  
4318038@ed.tus.ac.jp

Aohan Li

Department of Electrical Engineering  
Tokyo University of Science  
Tokyo, Japan  
aohanli@ee.kagu.tus.ac.jp

Zengchao Duan

Department of Electrical Engineering  
Tokyo University of Science  
Tokyo, Japan  
b-danzocho@haselab.ee.kagu.tus.ac.jp

Makoto Naruse

Department of Information Physics and  
Computing  
The University of Tokyo  
Tokyo, Japan  
makoto\_naruse@ipc.i.u-tokyo.ac.jp

Mikio Hasegawa

Department of Electrical Engineering  
Tokyo University of Science  
Tokyo, Japan  
hasegawa@ee.kagu.tus.ac.jp

**Abstract**—Non-Orthogonal Multiple Access (NOMA) is the technology that allows multiple users' downlink communications to be transmitted in the same resource block by proper user pairing. In realizing real-time operations, an ultrafast pairing decision scheme is required. Previous studies have shown that decision making using laser chaos is ultra-fast and effective as a Multi-Armed Bandit (MAB) algorithm. In this paper, we demonstrate user pairing using laser chaos-based MAB algorithm on the basis of the bit error rate of the physical layer. That is, we define the conditions for successful or failed communication by bit error, leading to benefits from the efficient decision-making ability of the laser chaos-based MAB strategy. The numerical results show that the proposed method provides better performances than conventional ones, C-NOMA and UCGD-NOMA.

**Index Terms**—Non-Orthogonal Multiple Access (NOMA), Laser Chaos, Multi-Armed-Bandit Problem, MAB algorithm, Bit error, User Pairing

## I. INTRODUCTION

With the advent of 5G, the upcoming years will see an explosive growth of mobile data traffic and a dramatic increase in the number of mobile devices, calling for the introduction of revolutionary wireless technologies to sustain the ever-increasing demand for bandwidth and services [1].

Non-orthogonal multiple access (NOMA) has been recognized as an essential technology for improving connectivity, spectral efficiency, cell-edge throughput, and user fairness in the fifth generation and beyond wireless networks. NOMA allows multiple users to use the same frequency band and time by assigning different power [2]. On the other hand, at the receiver side, multiuser-detection (MUD) algorithms such as successive-interference-cancellation (SIC) are implemented to identify specific signals [2]. User pairing is a way to select two users who are multiplexed in the same resource block. There is a growing interest in this user pairing [3] [4]. Pairing is important in NOMA for a variety of reasons. Reasons include

the fact that the larger the channel gain difference between users, the smaller the impact of SIC execution errors on the system performance, and the different throughput that can be obtained from pairing [3] [4] [5]. It has been found that the pairing schemes proposed in [4] do not provide optimal pairing [5].

Reinforcement learning methods, e.g., deep reinforcement learning methods and (Multi-Armed Bandit) MAB methods, are often used to solve problems in wireless communication [6] [7]. In [6],[7], Q-Learning and deep Q-Learning are applied to NOMA respectively. However, in order to apply these reinforcement learning methods, state information such as the user's location is required. Therefore, it takes time to obtain the state information, which results in delays. The MAB algorithm, on the other hand, can make decisions without state information, so it can be applied to real-time operation. User pairing problems and some other problems in wireless communication are sometimes treated as MAB problems [5] [8] [9]. The MAB problem is the problem of finding the machine with the highest reward probability from multiple slot machines with unknown reward probabilities [9].

In [10], Naruse et al. solves the MAB problem by using chaotic oscillatory waves generated by a semiconductor laser. Decision making is very fast using the MAB algorithm based on laser chaos [10]. In MAB problems, it is important to explore alternatives in order to find the best choice [11]. In [12], it is possible to generate a good quality random bit sequence at a very high bit rate. It has been shown that the laser chaotic time series can be used to solve the MAB problem very fast [10], [13]. It has been shown that scalable decision making up to 64-arm bandit problems is possible [13]. The MAB algorithm is based on laser chaos, and it selects one slot machine by successive large and small comparisons between the set threshold and the sampled data of the laser chaos waveform. Then, depending on the results of playing

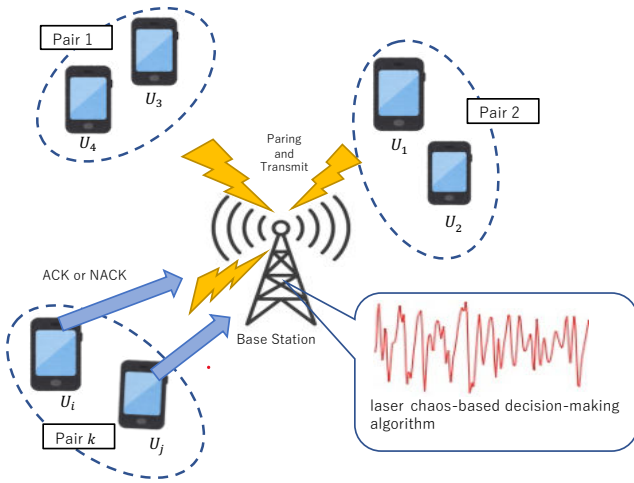


Fig. 1. System model.

the selected slot machine, the threshold is adjusted to make a better choice. If the option to be finally selected is mapped to a bit string, one slot machine is finally selected by setting the result of the comparison between the threshold and the sampled data of the laser chaotic waveform as “0” or “1”.

Duan et al. propose a pairing method using MAB algorithm based on laser chaos for NOMA systems [5]. The slot machines are associated with user pairing options in MAB problem, and the selected slot machine by the laser chaos-based decision maker means the selected pairing. In [5], the rewards of the MAB problem were defined by communication throughputs. This way, however, requires a not-so-small time duration to realize the rewards; therefore, the fast decision ability of the laser chaos-based MAB algorithm is not unfortunately fully utilized. Furthermore, the system model in [5] is too simple to evaluate the exact performance in NOMA systems.

In this paper, to overcome such limitations, the pairing problem is considered by examining bit error rate (BER) in adapting the MAB algorithm based on laser chaos. We compare the BER performance of the proposed method with conventional pairing strategies, such as C-NOMA [4] and UCDG-NMA [4].

The rest of the paper is organized as follows. In Section II, we provide the system model and the problem formulation. In Section III, we introduced the operating principle of the laser chaos-based MAB algorithm. In Section IV, the pairing method in the NOMA system based on laser chaos-based MAB algorithm is proposed on the basis of channel quality or BER. Section V shows numerical demonstrations of the proposed method. Section VI concludes the paper.

## II. SYSTEM MODEL

In this paper, we consider a downlink single cell system where one base station (BS) provides services to multiple users. Fig. 1 shows an overview of the system model of the NOMA system under study. All transmitters and receivers

have one antenna each. Let  $U=\{U_1, U_2, \dots, U_n, \dots, U_N\}$  be defined as the set of  $N$  users in the circular cell and  $K=\{1, 2, \dots, k, \dots, N/2\}$  be defined as the index of each pair.  $N$  is the total number of users, which is an even number. We assume that the distance ordered as  $d_1 \leq d_2 \leq \dots \leq d_N$  where  $d_n$  is the distance from the base station (BS) to the  $n^{\text{th}}$  user. The multiplexed signal  $x_k$  to the  $k^{\text{th}}$  pair is defined as follows [14]:

$$x_k = \sqrt{a_k P_k} x_k^i + \sqrt{(1-a_k) P_k} x_k^j \quad (1)$$

where  $x_k^i$  and  $x_k^j$  are the signals for the  $i^{\text{th}}$  and  $j^{\text{th}}$  users that from the pair  $k$  ( $d_i \leq d_j$ ).  $a_k$  is the power allocation factor for the  $k^{\text{th}}$  pair.  $P_k$  is the power value assigned to each pair for the  $k^{\text{th}}$  pairs.

At the receiver of the  $k^{\text{th}}$  pair, the received signal  $y_k$  is defined as follows [14]:

$$y_k = d_n^{-\lambda} \tilde{h}_n x_k + w_n \quad (2)$$

where  $d_n^{-\lambda}$  is the pathloss between BS and  $n^{\text{th}}$  user and  $\lambda$  is the pathloss exponent, is the Rayleigh fading of  $n^{\text{th}}$  user and  $w_n$  is the additive white gaussian noise (AWGN).

The base station performs user pairing using the MAB algorithm based on laser chaos and sends data to the user according to the pairing. After receiving and decoding the data, the user detects the bit errors and returns a response (ACK or NACK) to the base station depending on the number of bit errors. Let *bit error* $_n$  be defined as the number of bit errors detected by the  $n^{\text{th}}$  user and  $r_n = \{0, 1\}$  be defined as the  $n^{\text{th}}$  user's response to the base station.  $r_n = 1$  means that the  $n^{\text{th}}$  user sent ACK to the base station.  $r_n = 0$  means that the  $n^{\text{th}}$  user sent NACK to the base station. At the user side, the specific signal of each pair is restored via the SIC.

Our goal is to maximize the communication success rate by optimizing user pairing, which can be expressed as follows.

$$\max_b \sum_{n=1}^N r_n. \quad (3)$$

where  $b$  is the pairing option, which refer to which user to pair with. The larger of the value  $\sum_{n=1}^N r_n$  in Eq. (3) is, the higher communication success rate that the NOMA system can obtain while the lower of the users' BER will be. The maximum value of  $\sum_{n=1}^N r_n$  in Eq. (3) is  $N$ . At that time, all users communicate successfully while the BER is minimized. The obtained pairing option is the optimal user pairing solution of our formulated problem when  $\sum_{n=1}^N r_n = N$ . In order to achieve this goal, optimal user pairing must be performed. In this paper, the optimal user pairing is conducted using the MAB algorithm based on laser chaos.

## III. DECISION MAKING AS A MAB ALGORITHM BASED ON LASER CHAOS

Laser chaos is the chaotic output produced by a semiconductor laser. The following methods are used to generate laser chaos. The oscillation of lasers becomes unstable and leads to chaos when a portion of the output light is fed back to the

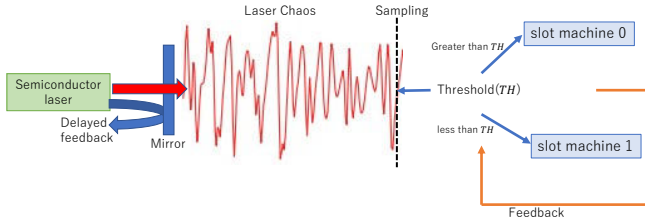


Fig. 2. MAB algorithm based on laser chaos.

laser cavity after a certain delay via an externally arranged mirror [9].

Fig. 2 schematically outlines the decision making based on MAB algorithm using laser chaos time-series generated by a semiconductor laser [9]. The principle of decision making based the laser chaos can be summarized as follows.

First, the initial value of the threshold is set. Next, a decision is made by comparing the sampled laser time series data with the threshold value. If the sampled time series data is greater than the threshold value, slot machine 0 is selected. Otherwise, slot machine 1 is selected. The threshold is adjusted according to whether the selected slot machine can be executed and rewarded, so that slot machines with higher reward probability will be selected in order to increase the reward obtained in the future.

The threshold  $TH(t)$ , which is compared with the sampling of laser chaos in step  $t$ , is given by

$$TH(t) = k \times [TA(t)] \quad (4)$$

where  $TH(t)$  is the threshold adjuster value at step  $t$ ,  $[TA(t)]$  is the closest integer to  $TA(t)$  rounded to zero, and  $k$  is a constant to control the range of  $TH(t)$ .  $[TA(t)]$  is  $-Z, \dots, -1, 0, 1, \dots, Z$ , where  $Z$  is a natural number. Thus, the number of thresholds is  $2Z + 1$ .  $[TA(t)]$  is limited to the range of  $-kZ$  to  $kZ$ .

The threshold adjuster TA is updated according to the following:

$$TA(t+1) = \begin{cases} \pm\Delta + \alpha TA(t), & \text{if rewarded. (a)} \\ \mp\Omega + \alpha TA(t), & \text{otherwise. (b)} \end{cases} \quad (5)$$

where  $\alpha$  ( $0 \leq \alpha \leq 1$ ) is a forgetting rate to control the impacts of past experiences,  $\Delta$  is a reward and  $\Omega$  is a penalty. In the case of a reward, i.e., if a benefit is obtained by playing the selected slot machine, the threshold adjustment value  $TA(t)$  is updated according to Eq. (5a). If it is not a reward, i.e., you did not get a benefit by playing the selected slot machine, the threshold adjustment value  $TA(t)$  is updated according to Eq. (5b).

$\Omega$  is a value based on past choices and benefits. Let,  $S_i$  be the number of times slot machine  $i$  was selected in step  $t$ . Let  $L_i$  be the number of times you played the selected slot machine  $i$  in step  $t$  and obtained a benefit. At this time, the estimated reward probability of  $i^{\text{th}}$  slot machine  $P_i$  is given by:

$$P_i = \frac{L_i}{S_i} \quad (6)$$

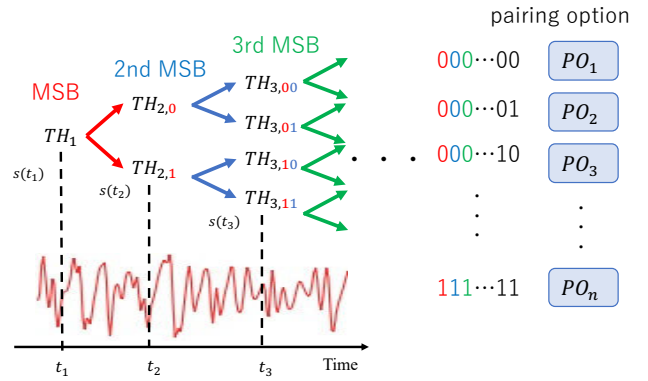


Fig. 3. User pairing using MAB algorithm based on laser chaos.

In the two-armed MAB problem, we use the estimated reward probability in Eq. (6) to define  $\Omega$  as follows:

$$\Omega = \frac{P_0 + P_1}{2 - (P_0 + P_1)} \quad (7)$$

If it is not a reward, i.e., you did not get the benefit by playing the selected slot machine, the threshold adjustment value  $TA(t)$  is updated according to Eq. (5b), using Eq. (7).

#### IV. USER PAIRING BY LASER CHAOS-BASED MAB ALGORITHM BASED ON BIT ERROR

In this section, we describe MAB algorithm based on a laser chaos for user pairing in NOMA. The proposed method is a reinforcement learning algorithm that reaches the optimal pairing by repeating the following process: determine the pairing options by the MAB algorithm based on laser chaos, communicates with the pairings, and adjusts the threshold based on the communication results. Fig. 3 shows the structure of MAB algorithm based on laser chaos for the NOMA user pairing problem, where the slot machines in the MAB problem are treated as user pairing options. The user pairing options are selected by comparing the sampled data from the laser chaos time series with a threshold value. The threshold is adjusted based on the reward of communicating with the selected user pairing option. In Fig. 3,  $PO_n$  is the  $n^{\text{th}}$  option selected. The next step is to describe the rewards and penalties needed to adjust the threshold.

Next, we explain that the detailed time series and threshold comparison method. In the laser chaos-based MAB algorithm, the identity of the pairing option to be selected can be determined bit by bit in a pipelined fashion from the most significant bit (MSB) to the least significant bit (LSB). For each bit, the decision is based on a comparison between the measured chaotic signal level and a specified threshold value. First, we determine the most significant bit (MSB). At  $t = t_1$ , compare the level of the chaotic time series with the threshold value  $TH_1$ . If the time series is greater than or equal to the threshold value, the chosen option is 0, which we denote as  $D_1$  (MSB) = 0. Otherwise, the result will be 1 ( $D_1 = 1$ ).

Then, we decide the second most significant bit. In the case the MSB is determined by  $D_1 = 0$  and compare the level



of the time series with the threshold value  $TH_{2,0}$ . The first number 2 in the  $TH_{2,0}$ 's subscript indicates that the threshold value is related to the second most significant bit, and the second number 0 in the subscript indicates that the previous decision was 0 ( $D_1 = 1$ ). If the time series is greater than or equal to the threshold  $TH_{2,0}$ , then the second most significant bit is 0 ( $D_2 = 0$ ), otherwise it is 1 ( $D_2 = 1$ ).

Finally, we have to determine the least significant bit. According to the above rules, threshold value comparison finishes when all  $L$  bit information of the specified option is determined. If  $L = 4$ , the result of the 4<sup>th</sup> comparison is the least significant bit of the combination to be selected. The update formula of the threshold adjuster value ( $TA$ ) is expressed as follows:

$$TA_{L,M_1,M_2,\dots,M_{(L-1)}}(t+1) = \begin{cases} \pm\Delta + \alpha TA_{(L,M_1,M_2,\dots,M_{(L-1)})}(t), & \text{if rewarded. (a)} \\ \mp\Omega + \alpha TA_{(L,M_1,M_2,\dots,M_{(L-1)})}(t), & \text{otherwise. (b)} \end{cases} \quad (8)$$

In Eq. (8a), the reward is given when  $M_L = 0$  or 1,  $D_1 = M_1, \dots, D_{L-1} = M_{L-1}$  are determined. In Eq. (8b), the reward is not given when  $M_L = 0$  or 1,  $D_1 = M_1, \dots, D_{L-1} = M_{L-1}$  are determined. In this time,  $D_1 D_2 \dots D_L$  is the selected option associated with the bit sequence.  $\alpha$  ( $0 \leq \alpha \leq 1$ ) is a forgetting rate to control the impacts of past experiences.

It is mentioned that each user responds to the base station according to the success or failure of the communication in Section II and the conditions of success or failure are explained below. After receiving and restoring the data, each user detects the bit error. The  $n^{\text{th}}$  user compares this *bit error* <sub>$n$</sub>  with the  $n^{\text{th}}$  user's bit error reference value *err* <sub>$n$</sub>  to determine the success or failure of communication. If *bit error* <sub>$n$</sub>   $\leq$  *err* <sub>$n$</sub> , the communication is regarded as successful; ACK is sent to the base station. In the case of *bit error* <sub>$n$</sub>   $>$  *err* <sub>$n$</sub> , the communication is regarded as failed; NACK is sent to the base station. The equation is as follows:

$$r_n = 1 \quad \text{if } \textit{bit error}_n \leq \textit{err}_n \quad (9)$$

$$r_n = 0 \quad \text{if } \textit{bit error}_n > \textit{err}_n \quad (10)$$

The base station rewards and penalties the threshold according to the number of ACKs received. If the number of ACKs received at the base station is greater than or equal to  $X$ , give a reward, otherwise give a penalty.

## V. SIMULATION RESULTS

In this section, we present the numerical results to evaluate the performance based on BER. Besides, we compared our proposed method Laser Chaos Decision Making (LCDM)-NOMA to C-NOMA [4] and UCGD-NOMA [4]. In C-NOMA and UCGD-NOMA, users are first divided into two categories according to their distance from the base station: "near area" and "far area". C-NOMA's pairing scheme is that pairs the closest user to a base station in the short-range region with the farthest user to a base station in the long-range region, the second closest user in the short-range region with the second

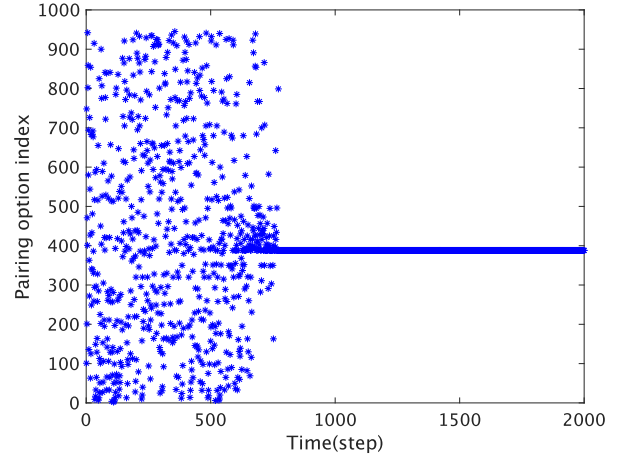


Fig. 4. Pairing option selected by MAB algorithm based on laser chaos at each step.

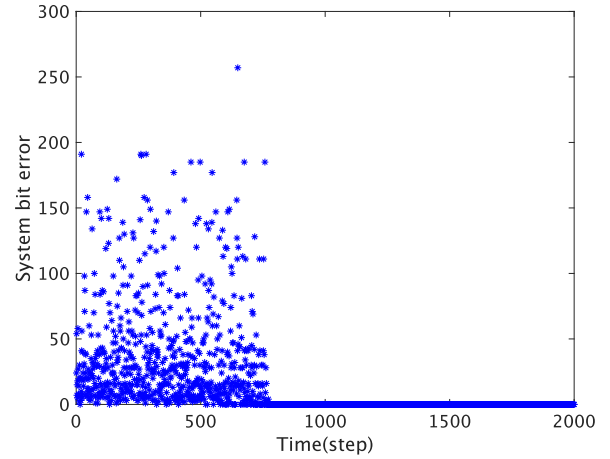


Fig. 5. System bit error for the pairing option selected at each step.

farthest user in the long-range region, and so on, and pairs other users as well. UCGD-NOMA's pairing scheme is that pairs the user who is closest to a base station in the short-range area with the user who is closest to a base station in the long-range area, the second closest user in the short-range area with the second closest user in the long-range area, and so on, and pairs other users as well. In our simulation, we consider 10 users ( $N = 10$ ) and a cell with a radius of 1000 m where users are arranged randomly. Path loss exponent is set as 2.7. The power allocation is fixed, what it means that all pairs are allocated 30 dBm regardless of pairing option and the power allocation factor is  $a = 0.1$ . We assume that the standard is  $k = 128$  and  $Z = 1$  in MAB algorithm based on laser chaos. In addition, the forgetting rate  $\alpha$  is 1.0 and  $\Delta$  is 1.0 in that one. data bit per user is 256 bit.  $x_k^i$  and  $x_k^j$  is a signal of "data bit per user" modulated by QPSK and made into OFDM symbols by IFFT. The number of bit errors to be tolerated, *err* <sub>$n$</sub> , can be fixed or adaptive. In the fixed case,

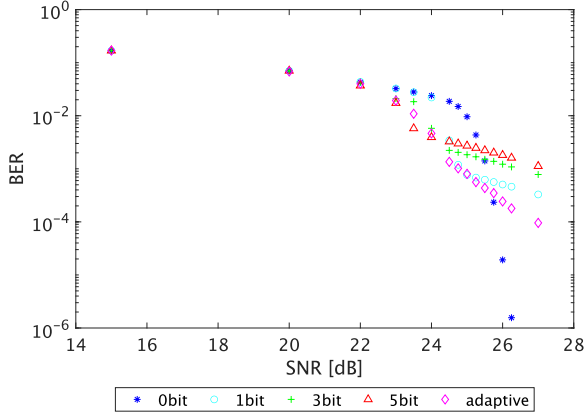


Fig. 6. BER under different  $err$  settings when the cell radius and the path loss exponent are 1000 m and 2.7, respectively.

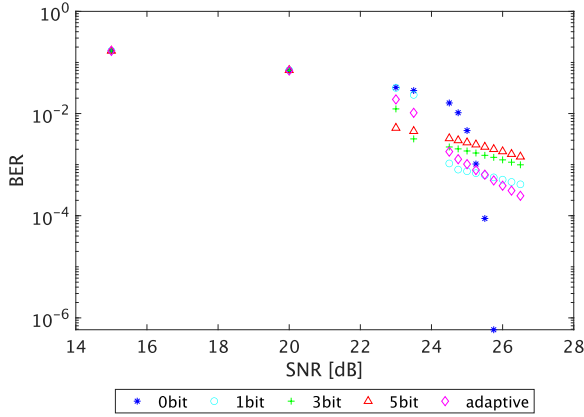


Fig. 7. BER under different  $err$  settings when the cell radius and the path loss exponent are 500 m and 3.5, respectively.

$err_n$  is the same for all steps. In the case of adaptive,  $err_n$  is the average of the last five bit errors of its own.

In this simulation, 2000 steps are used as the threshold learning step, and this 2000<sup>th</sup> step is executed 10000 times. 2000<sup>th</sup> step pairing option is considered to be a better one on the basis of the MAB algorithm in Section IV. We show the effectiveness of the proposed method by evaluating the BER of this 2000<sup>th</sup> step. The BER is calculated as follows:

$$BER = \frac{\sum_{r=1}^{10000} \sum_{n=1}^N bit\ error_{n,2000^{th}}}{data\ bit\ per\ user \times N \times 10000} \quad (13)$$

where  $biterror_{n,2000^{th}}(r)$  is the bit error at the 2000<sup>th</sup> step of the  $n$ <sup>th</sup> user in the  $r$ <sup>th</sup> execution.

Fig. 4 shows that the results of the pairing option selected by MAB algorithm based MAB laser chaos in one run. The horizontal axis of Fig. 4 is the number of steps, and the vertical axis is the pairing option index. Fig. 4 illustrates the pairing selected according to MAB algorithm based on laser chaos is

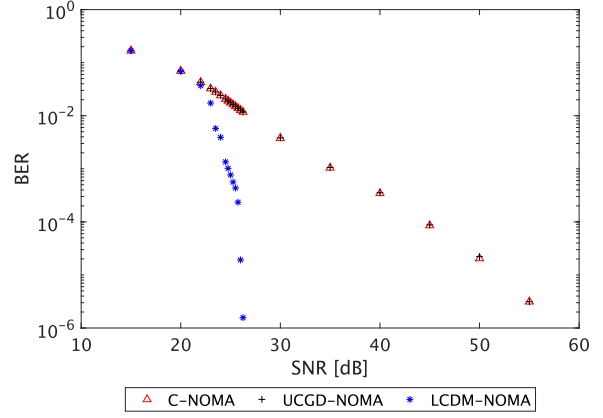


Fig. 8. BER comparison with LCDM-NOMA and C-NOMA and UCGD-NOMA.

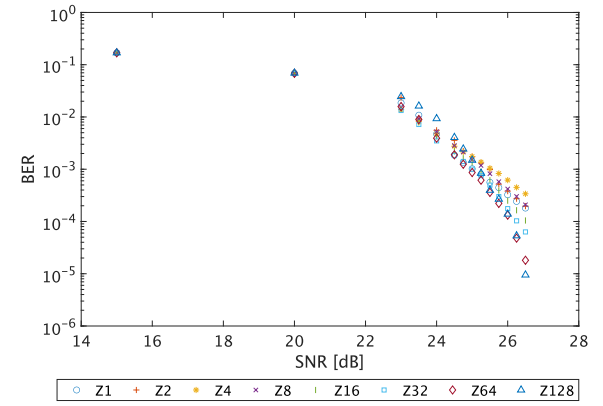


Fig. 9. BER with different number of thresholds.

changing before about 750 steps. In this run, after about 750 steps, the proposed scheme can converge to the 388<sup>th</sup> pairing option and no longer change.

Fig. 5 shows that the variation in the number of  $system\ bit\ error$  at each step in one run.  $system\ bit\ error$  is the sum of  $bit\ error_n$  in each step, expressed in a formula,  $system\ bit\ error = \sum_{n=1}^{10} bit\ error_n$ . From Fig. 4, we can see that the pairing options converge to *one*, so the system bit error also converges and its value is zero.

Fig. 6 shows that the BER after 2000 steps to each SNR. In Figure 6, “0 bit, 1 bit, 3 bit, 5 bit” is the result when  $err_n$  is fixed at 0, 1, 3, or 5 bits for all users in all steps and all runs, respectively. “adaptive” is the result when  $err_n$  is set to the average of the last five bit errors of the user. It can be seen that SNR is more higher, BER becomes more smaller. Furthermore, it can be seen that the optimal  $err_n$  differs depending on the SNR. That is why we have to consider setting  $err_n$ .

Fig. 7 shows the BER when the cell radius and path loss exponent are set to 500 m and 3.5, respectively. Other

parameter settings are the same as in Fig. 6. From Fig. 7, we can get the same conclusion as that from Fig. 6, which shows that the variation trend of BER under different  $err_n$  settings do not change with the cell radius and path loss exponent.

Fig. 8 shows that a BER comparison to C-NOMA and UCGD-NOMA. We observe from Fig. 8 that the proposed LCDM-NOMA achieves a smaller BER than the C-NOMA and UCGD-NOMA. Therefore, we conclude LCDM-NOMA is better than C-NOMA and UCGD-NOMA when BER is concerned.

Fig. 9 summarizes the effect of the number of thresholds on the BER in LCDM-NOMA and the resultant bit error rate. The number of thresholds is given by  $2Z + 1$  where  $Z$  is a natural number. The fewer the number of thresholds, the quicker for the threshold to reach the upper or lower limit. Hence, the convergence of the selection becomes generally fast, but the selection accuracy becomes worse. Conversely, the higher the number of thresholds, the more likely that the threshold reaches its upper or lower limit through sufficient exploration. Therefore, the convergence needs longer time duration, but the selection accuracy becomes better. In the vicinity of 23 dB in SNR, the BER becomes smaller when the number of threshold steps is reduced; when the number of thresholds is larger than 23 dB in SNR, the BER becomes smaller when the number of thresholds increased. Therefore, it is necessary to set the appropriate number of thresholds depending on the given SNR.

## VI. CONCLUSION

In this paper, we demonstrate an optimization method for user pairing in NOMA systems by using MAB algorithm based on laser chaos on the basis of the bit error rate of the physical layer. The performance of the proposed method is verified by applying MAB algorithm based on laser chaos in the NOMA system. Simulation results show that the proposed algorithm accomplishes a smaller BER than conventional NOMA algorithms. In future work, we will evaluate the performance in more realistic setting problems and implement error correcting codes.

## REFERENCES

- [1] A. Abrardo, M. Moretti, and F. Saggese, "Power and Subcarrier Allocation in 5G NOMA-FD Systems," *IEEE Trans. Commun.*, Vol. 19, No. 12, December 2020.
- [2] Mohammed Abd-Elnaby, Germien G. Sedhom and Mohamed Elwekeil "Subcarrier-User Assignment in Downlink NOMA for Improving Spectral Efficiency and Fairness," *IEEE Access* January 11, 2021, Digital Object Identifier 10.1109/ACCESS.2020.3047985.
- [3] L. Zhu, J. Zhang, X. Cao and D. O. Wu, "Optimal User Pairing for Downlink Non-Orthogonal Multiple Access (NOMA)," *IEEE Wireless Commun. Lett.*, Vol. 8, pp. 328–331, April 2019.
- [4] M. B. Chahab, M. Irfan, M. F. Kader and S. Y. Shin, "User pairing schemes for capacity maximization in non-orthogonal multiple access systems," *Wirel. Commun. Mob. Comput.*, Vol. 16, pp. 2884–2894, December 2016.
- [5] Z. Duan et al., "High-speed Optimization of User Pairing in NOMA System Using Laser Chaos Based MAB Algorithm," 2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC) 13–16, April 2021.
- [6] M. Valente, R. Demo Souza, H. Alves and T. Abrao, "A NOMA-Based Q-Learning Random Access Method for Machine Type Communications," *IEEE Wireless Commun. Lett.*, Vol. 9, NO. 10, October 2020.
- [7] S. Wang, T. Lv, W. Ni, N. C. Beaulieu and Y. J. Guo, "Joint Resource Management for MC-NOMA: A Deep Reinforcement Learning Approach," *IEEE Trans. Wireless Commun.*, Vol. 20, NO. 9, September 2021.
- [8] H. Kanemasa, A. Li, M. Naruse, N. Chauvet and M. Hasegawa, "Dynamic Channel Bonding Using Laser Chaos Decision Maker in WLANs," 2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC) 78–82 April 2021.
- [9] L. Lai, H. Jiang, and H. V. Poor, "Cognitive medium access: Exploration, exploitation, and competition," *IEEE Trans. Mob. Comput.*, Vol. 10, No.2, pp. 239-253, Feb. 2011.
- [10] M. Naruse, Y. Terashima, A. Uchida and S.-J. Kim, "Ultrafast photonic reinforcement learning based on laser chaos," *Scientific Reports*, Vol. 7, pp. 8772, 2017.
- [11] M. Naruse, N. Chauvet, A. Uchida, A. Drezet, G. Bachelier, S. Huant, H. Hori, "Decision Making Photonics: Solving Bandit Problems Using Photons," *IEEE J. Sel. Topics Quantum Electron.*, Vol. 26, July. 2019.
- [12] A. Uchida, K. Amano, M. Inoue, K. Hirano, S. Naito, H. Someya et al., "Fast physical random bit generation with chaotic semiconductor lasers," *Nat. Photon.*, Vol.2, pp. 728–732, Nov. 2008.
- [13] M. Naruse et al., "Scalable photonic reinforcement learning by time-devision multiplexing of laser chaos," *Scientific Reports*, Vol. 8, pp. 10890, 2018.
- [14] H. Yahya, E. Alsusa and A. Al-Dweik, "Exact BER Analysis of NOMA with Arbitrary Number of Users and Modulation Orders," *IEEE Trans. Commun.*, Vol. 69, No. 9, September 2021.

# Hybrid Energy Management Systems based on Edge Processing for Electric Transportation Applications

Henar Mike O. Canilang

Department of Electronics Engineering  
Kumoh National Institute of Technology  
Gumi, South Korea  
hmocanilang@kumoh.ac.kr

Danielle Jaye S. Agron

Department of Electronics Engineering  
Kumoh National Institute of Technology  
Gumi, South Korea  
danielleagron@kumoh.ac.kr

Wansu Lim

Department of Electronics Engineering  
Kumoh National Institute of Technology  
Gumi, South Korea  
wansu.lim@kumoh.ac.kr

**Abstract**—In this paper, a hybrid energy management system (HEMS) based on an edge processing scheme is presented. This HEMS is considered for deployment in the electric transportation industry which is a major sector for energy management applications. This approach paves the way for the emerging convergence of energy management systems (EMS) and intelligent applications. In this proposed scheme, the HEMS is integrated with edge processing to realize its intelligent and sustainable deployment. The HEMS is tested on a designed simulation platform and is re-deployed on an edge device for model verification. The implemented HEMS utilizes a battery and ultra-capacitor pack as the source. The battery pack and the ultra-capacitor pack have a rated maximum voltage of 50.4 V each.

**Index Terms**—Battery, edge device, edge processing, hybrid energy management system, ultra-capacitor.

## I. INTRODUCTION

Since the industrial revolution and up to the present day, the world is still heavily relying on energy produced by fossil fuels which have major implications to global climate change, air pollution that is pivotal to health issues which tally of at least 5 million premature deaths each year and natural energy resources depletion. The energy market is gearing towards renewable energy sources and storage. The energy production of the modern energy sector contributes to at most 75% of the total carbon dioxide emission (CO<sub>2</sub>) emissions worldwide which is a major factor in global climate change and health pollution [1], [2]. In order to provide a cleaner energy supply for the demand worldwide, the energy industry is gearing towards an alternative approach of greener and sustainable energy. Rapid development and deployment of renewable energy sources are scaling towards industrial applications. This trend aims to promote the sustainable and stable deployment of renewable energy resources-based applications [3]. As of 2018, a total of 179 countries have started an initiative to invest in the application of renewable energy in their countries. As of 2020, the renewable energy sector provides 15% to 20% of the total global energy demand [4]. Renewable energy systems are composed of energy harvesters and storage which vary in terms of application and deployment. One of the most promising sectors utilizing renewable energy and storage is the transportation sector whereas electric vehicles (EVs) are mostly utilized. From 2020 to 2021, the global EV sale ramped up to an outstanding sales growth of 98%. Based on statistics

presented in [5], a total of over 4-million EVs and 2.4 plug-in hybrid vehicles were sold in the first half of 2020 to 2021. By the year-end of 2021, the total EVs globally are at least 16 million whereas two-thirds are pure EVs.

Early electric vehicles are powered by a battery. The manufacturers use a nickel-metal hydride battery or lithium-ion-based battery [6] however, due to the high-power demand of EV parts, the driving range is limited and the capacity of the batteries depletes over time since it is known to have high energy but low power density. These characteristics are affected by the peak power variations that cause rapid battery life degradation. In terms of the transportation sector, this is in line with the speed variations in motor traction demands such as for speed and braking variations. To improve the battery-based applications, the researchers integrate an ultra-capacitor to enhance the EV car energy source. The combination of battery and capacitors is gaining research interest for actual deployment since it realizes sustainable systems for a plethora of applications such as for energy storage and source. On the other hand, capacitors have low energy density and high-power density. Going further, capacitors have rapid charging and discharging capability with a known high output power density and low-power-to-weight ratio. The combination of battery and capacitors can realize an efficient system capable of high energy density for the driving range and a high-power density for acceleration. The capacitor is capable of a strong charge based on energy regeneration which is also a pivotal factor. This demands a hybrid management system that can handle the dynamics difference between batteries and capacitors.

Hybrid energy management system (HEMS) applications and deployment are rapidly increasing owing to its benefits compared with other energy systems such as battery management systems (BMSs) and capacitor management systems (CMSs). By combining batteries and capacitors, the high power and energy density enhances the overall performance of the deployed system and also maximize the life of the battery and UCs. Though conventional HEMS is in demand nowadays, the deployment of HEMS requires innovation in this intelligent systems era. The deployment of HEMS nowadays requires the state-of-the-art capability to realize intelligent applications.

In the introduction of the hybrid era for transportation applications such as EVs, the researchers are still developing

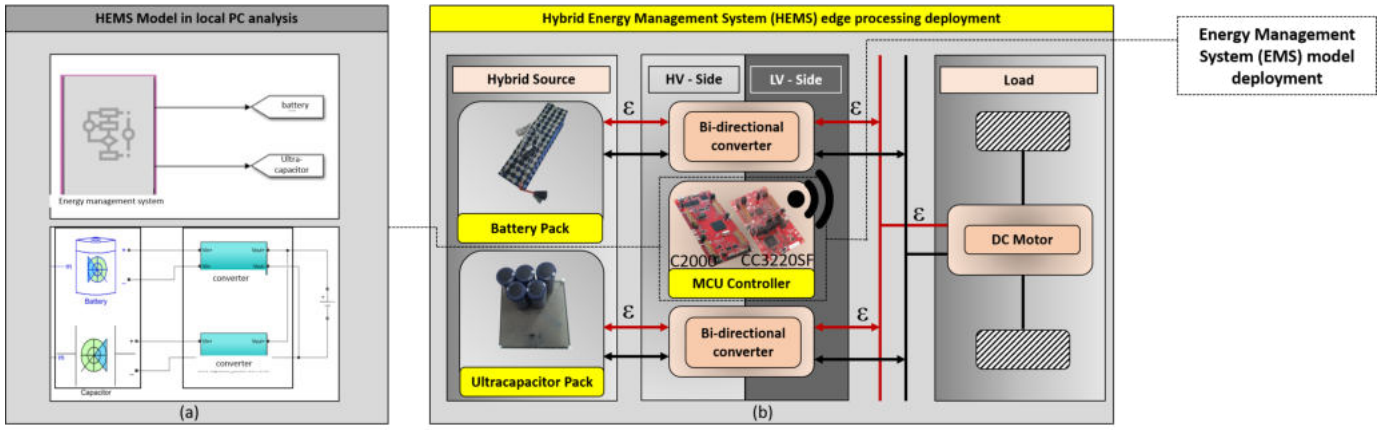


Fig. 1. Overview of the proposed HEMS based on edge processing for electric transportation application (a) HEMS modelling in local PC (b) edge based processing HEMS with the deployed HEMS model.

an optimal way to allocate the power that is coming from batteries and capacitors. This management system aims at monitoring, balancing, controlling, protecting, and enhancing the efficiency of these battery cells and capacitor cells on their deployment. This paper focuses on the analysis of HEMS for intelligent applications, particularly for electric transportation applications. We comprehensively analyze and implement a system for HEMS that enables internet-of-things-based computing paradigms such as edge processing. Edge processing is used to address the demand for intelligent applications of HEMS and the constraints of the current intelligent approach such as data processing efficiency. Edge-based application realizes the convergence of energy management and intelligent applications.

## II. RELATED STUDIES

### A. Energy management system

Energy management system (EMS) is a research hotspot in terms of sustainable application and deployment. EMS research aims to improve the overall safety, efficiency, reliability, stability, and deployment capability of both energy storage and sources. In summary, opportune EMS is being deployed to ensure that energy storage and sources adhere to the deployment parameter and standards [7]–[9].

### B. Battery management system

One of the most commonly used EMS is for battery-based applications. This is in line with the ongoing demand for battery-based applications such as in the electric transportation industry such as EVs. Researchers focused on the improvement of BMS in terms of design and cost trade-off, fault adaptivity [10], and intelligent-based applications [11]. This is in line with the demand for intelligent energy management applications for the transportation industry such as the internet-of-vehicles (IoV) and edge processing.

### C. Capacitor management system

Capacitors are often compared with the battery for their deployment characteristics. The energy management for

capacitor-based applications is known as the capacitor management system (CMS). Research and studies prove that capacitors such as ultra-capacitors and super-capacitors last longer than batteries. This is due to the fact that capacitors can handle the peak variation of a system demand such as voltage and current during the charging and discharging phase. This strengthens the physical toll tolerance of capacitors compared with batteries [12]. With the emerging demand for intelligent applications as mentioned in the BMS section, CMS and capacitor-based applications are also gearing toward intelligent applications [13].

### D. Hybrid energy management system

Hybrid energy systems combine an energy source with another. This is to address the dynamics and peak variations which are the constraints of the deployed energy sources. For this case, a hybrid energy management system (HEMS) is deployed to ensure the effectiveness and co-deployment of the combined energy source. HEMS is the combination of the BMS and CMS. HEMS aims to maximize the interrelation between batteries and capacitors considering their distinct dynamic behavior [14], [15].

TABLE I  
COMPARATIVE ADVANTAGE OF EXISTING HEMS APPROACH AND THE PROPOSED EDGE HEMS DEPLOYMENT

Reference	Criteria	Conventional HEMS	Proposed Model
[16]	Intelligent applications consideration	No	Yes
	Edge processing application consideration	No	Yes
[17]	Intelligent applications consideration	Yes	Yes
	Edge processing application consideration	No	Yes
[18]	Intelligent applications consideration	Yes	Yes
	Edge processing application consideration	No	Yes

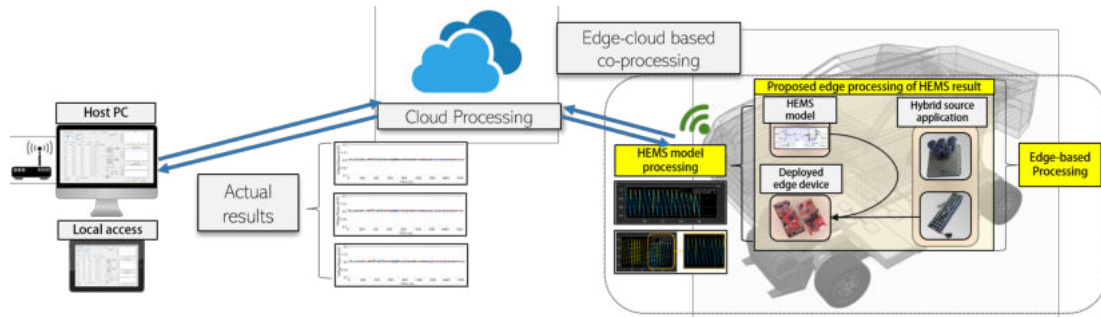


Fig. 2. The proposed HEMS actual result for edge processing.

### E. HEMS convergence to the edge computing paradigm

The convergence of EMS and this intelligent era is pivotal and rapidly increasing nowadays. The emerging edge AI devices innovate the conventional deployment of EMS towards intelligent applications [19], [20]. In terms of HEMS convergence to intelligent applications, its core focus is to adapt with the overall demand and parameters of the system to ensure the maximum efficiency of both sources (battery, capacitors, etc.) when deployed using EMS models.

Table I summarizes the proposed HEMS deployment approach of this paper, which realizes its convergence to the edge-computing paradigm. Most papers and research focus on the development of EMS models and on optimizing existing HEMS. The proposed approach of this paper is the convergence of the HEMS model for edge-based processing. Edge-based processing considers intelligent applications such as the deployment of state-of-the-art HEMS models.

### III. METHODOLOGY

This paper aims to presents a HEMS processing approach using edge processing to realize the convergence of EMS to intelligent applications as shown in Fig. 1. The sources used for this proposed HEMS are 1) batteries and 2) ultra-capacitors. With the edge-based processing, the load and peak requirement of the HEMS is learned and predicted by the system. This makes the HEMS adaptive to a multitude of deployment parameters and applications. Two specific conditions are managed by the HEMS through the learned parameters which are 1) utilize ultra-capacitors at peak power demand and 2) utilize batteries at stable power demand. The analysis of HEMS is performed at the edge device which for this application is the Texas Instrument TMS320F28035 C2000. The model utilizes an electric transportation platform for simulation particularly with the motor as the load. The battery pack and ultra-capacitor pack has a rated voltage of 50.4 V respectively for the HEMS implementation of this paper. The model is first deployed in a local PC for verification and analysis prior to the edge device deployment. The deployed model realizes soft and hard real-time applications whereas, in the actual deployment, a wireless communication module is integrated into the Texas Instrument TMS320F28035 C2000.

In the Fig. 1 shows the overview of the proposed HEMS based on edge processing for electric transportation applica-

tions. Fig. 1(a) shows the HEMS model which is simulated and designed in a local PC. Fig. 1(b) shows the HEMS deployed with the EMS model. The figure highlights which represents the energy of the two sources and the demand energy of the load which is a motor based on electric transportation. The C2000 is the central MCU or the edge device for this application where an EMS model is deployed. The CC3220SF is the proposed wireless communication module used to realize soft and hard real-time applications. The edge-based application depends on the model deployed whereas, for this application, the total current demand and power demand of the system are processed.

### A. HEMS design considerations

The design consideration of the HEMS is the peak current demand of the system with respect to the load. The demand current of the system should equalize the total current of the HEMS. An energy management model is deployed to process the parameters of the HEMS such as the average load demand and learn this parameter. The HEMS adheres to the total load demand and equalizes the total demand and supply by managing both sources.

### IV. RESULTS AND DISCUSSION

In the Fig. 2 shows the actual HEMS result for edge processing together with its processing capability. The HEMS model is deployed for electric transportation applications whereas this papers' methodology addresses the demand of HEMS application of this era, which is for intelligent transportation and smart vehicles. The deployed edge device with

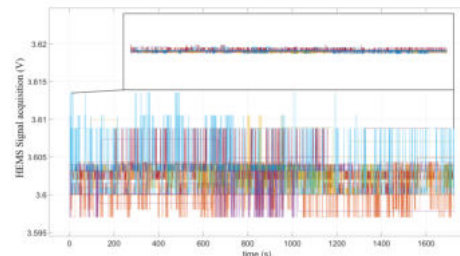


Fig. 3. The acquired HEMS parameters from local PC.

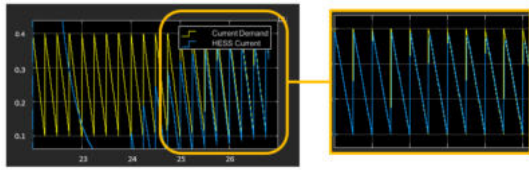


Fig. 4. The model deployment on local PC simulation.

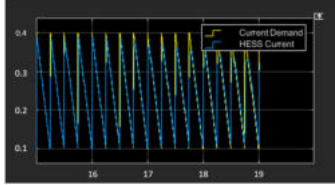


Fig. 5. The model deployment on edge device.

the HEMS model can independently process and control the HEMS. The integrated wireless communication allows the HEMS to communicate and transmit the actual result to the cloud or to local servers paving way for co-processing capability. In the Fig. 3 shows the actual monitoring result of the HEMS whereas these acquired HEMS parameter is accurate and is stable. The stability of the signal acquired proves that the deployed HEMS model on edge is effective.

In the Fig. 4 shows the actual initial results of the energy management model deployed on a local PC. It can be seen that the overall current demand and the hybrid energy storage system composed of two energy sources are equalizing. Though this is an initial result, this model proves to be promising in terms of deployment to the edge devices whereas when this model was uploaded to the C2000, the same results is yielded.

In the Fig. 5 shows the actual deployment of the HEMS model on the edge device. The current demand for the hybrid system is equalized with the current supplied by the batteries and ultra-capacitors. With this result, the peak variations in terms of the current demand of a hybrid system are analyzed. The peak variations represent the driving cycle of electric transportation. An energy management model for the hybrid of battery and ultra-capacitor is simulated and modeled in a local PC followed by the deployment on an edge device. This realizes HEMS edge processing. With the help of a wireless communication module integrated into the C2000, this HEMS model can realize soft and hard real-time applications, which is pivotal for its deployment in this intelligent system era.

## V. CONCLUSION

The proposed hybrid energy management's (HEMS) main goal is to maximize the correlation of two sources ensuring that both perform at maximum efficiency. In terms of this application, two different sources were used and tested for edge-based processing namely, batteries and ultra-capacitors. The batteries provide the average current whilst the ultra-capacitors provide the transient current. The energy management system deployed equalized the HEMS current demand and the supply from batteries and ultra-capacitor.

## VI. ACKNOWLEDGMENT

This work was supported by the Ministry of SMEs and Start-ups, S. Korea (S2829065, S3010704), and by the National Research Foundation of Korea (2020R1A4A10177511, 2021R111A3056900).

## REFERENCES

- [1] A. Bindra. Global climate change: A norwegian perspective. *IEEE Power Electronics Magazine*, 6:34–35, 12 2019.
- [2] R.M. Elavarasan, G. Shafiullah, P. Sanjeevikumar, Nallapaneni Manoj K., A. Annam, A. Vetrichevan, Lucian Mihet, P., and J. Holm-Nielsen. A comprehensive review on renewable energy development, challenges, and policies of leading indian states with an international perspective. *IEEE Access*, PP, 04 2020.
- [3] T. Kurbatova and T. Perederii. Global trends in renewable energy development. pages 260–263, 10 2020.
- [4] F. Ayadi, I. Colak, I. Garip, I. Halil, and H. Bulbul. Targets of countries in renewable energy. 10 2020.
- [5] S. Lee, N. Ahmad, S. Son, and A. Khattak. How many electric vehicle owners will repurchase a similar vehicle? 01 2022.
- [6] V.K Raja, I. Raja, and R. Kavvampally. Advancements in battery technologies of electric vehicle. *Journal of Physics: Conference Series*, 2129:012011, 12 2021.
- [7] S. Aznavi, P. Fajri, A. Asrari, and R. Sabzehgar. Energy management of multi-energy storage systems using energy path decomposition. In *2019 IEEE Energy Conversion Congress and Exposition (ECCE)*, pages 5747–5752, 2019.
- [8] D. Menniti, A. Pinnarelli, N. Sorrentino, P. Vizza, A. Burgio, G. Brusco, and M. Motta. A real-life application of an efficient energy management method for a local energy system in presence of energy storage systems. In *2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I CPS Europe)*, pages 1–6, 2018.
- [9] N. Yan, S. Li, T. Yan, and S. H. Ma. Study on the whole life cycle energy management method of energy storage system with risk correction control. In *2020 IEEE 4th Conference on Energy Internet and Energy System Integration (EI2)*, pages 2450–2454, 2020.
- [10] H. Canilang, A. Caliwag, and W. Lim. Design of modular bms and real-time practical implementation for electric motorcycle application. *IEEE Transactions on Circuits and Systems II: Express Briefs*, pages 1–1, 2021.
- [11] J. Tharun, V. Jegadeesan, S. Muthumanickam, and C. Krishnan. Intelligent battery management system. pages 1–5, 07 2021.
- [12] A. Abdelhakim and F. Dijkhuizen. Integration of batteries and super-capacitors in a vehicular power supply system, 06 2021.
- [13] L. Shuguang, Z. Wenpu, S. Tianle, S. Wenquan, and G. Yi. Design of intelligent power capacitor with synchronous switching. In *2021 40th Chinese Control Conference (CCC)*, pages 6992–6997, 2021.
- [14] M. Gaber, S.H. El-Banna, M. Eldabah, and M.S. Hamad. Design energy management system for generic hybrid power based on intelligent fuzzy logic technique. In *2021 International Telecommunications Conference (ITC-Egypt)*, pages 1–4, 2021.
- [15] P.N. Bhat and M.B. Veena. Design of efficient power management system using ultra capacitors. In *2019 Global Conference for Advancement in Technology (GCAT)*, pages 1–5, 2019.
- [16] X. Li, R. Ma, L. Wang, S. Wang, and D. Hui. Energy management strategy for hybrid energy storage systems with echelon-use power battery. pages 1–2, 10 2020.
- [17] H. Chen, C. Lin, R. Xiong, and W. Shen. Model predictive control based real-time energy management for hybrid energy storage system, 07 2021.
- [18] S.C. Choi, M.H. Sin, D.R. Kim, C.Y. Won, and Y.C. Jung. Versatile power transfer strategies of pv-battery hybrid system for residential use with energy management system. pages 409–414, 05 2014.
- [19] B. Bachir, M. Boukhniher, and L. Degaa. Energy management strategies for a fuel-cell/battery hybrid power system. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, page 095965182095851, 10 2020.
- [20] C. Deutsch, A. Chiche, S. Bhat, C. Lagergren, G. Lindbergh, and J. Kutenkeuler. Energy management strategies for fuel cell-battery hybrid aavs. pages 1–6, 09 2020.

# Studies on Intelligent Curation for the Korean Traditional Cultural Heritage

Jae-Ho Lee, Hee-Kwon Kim, Chan-Woo Park  
Creative Content Research Division  
Electronics and Telecommunications Research Institute  
Daejeon, Korea  
jhlee3@etri.re.kr, hkkim79@etri.re.kr, gamer@etri.re.kr

**Abstract**—In this paper, we introduce the necessary technologies to use the Korean traditional cultural heritage in immersive content by applying artificial intelligence technology. In fact, the data stored in museums has already been expanded to a huge amount. Recently, there are increasing efforts to convert such vast amounts of traditional cultural heritage image or text data into usable content through the analysis and connection of information using them. The main purpose of this study is to support the response to various content demands such as meta-verse and virtual reality for traditional cultural heritage of Korea. Representative technologies used in Korean traditional cultural heritage introduced in this study can be classified into artificial intelligence-based object detection and high-resolution conversion, and text analysis suitable for the characteristics of Korean traditional cultural heritage.

**Index Terms**—traditional heritage, digital heritage, Korean heritage, super-resolution, Named Entity Recognition

## I. INTRODUCTION

Recently, a lot of technologies for applying cultural heritage to immersive content are introduced and in progress under the influence of the spread of meta-verse, virtual reality, mixed reality, etc. Access to these new markets is an important factor in expanding the new role of museums and galleries in historical culture and art. However, to effectively compose cultural heritage-based immersive content, high-quality digitization of cultural heritage must be preceded. Additionally, it is necessary to develop AI visual search and search-based curation support technology and platform to manage vast digital heritage data.

With the development of artificial intelligence technology, researches to use the extensive cultural heritage data that have been continuously built until now as actual content and to activate it as a multifaceted approaches are gradually increasing, mainly in developed countries [1][2][3][4]. Meaning-based image search technologies that identify and compare major meanings between visual data, rather than simple keyword searches for vast cultural heritage data, are expanding to related application fields [5][6]. Artificial intelligence-based high-quality data conversion technologies are continuously being introduced [7][8].

Despite the continuous development of technology, the practical application of traditional cultural heritage in museums and exhibition halls has not yet been properly implemented.

The reason for this inadequate is that most of the staff working at the museum are mainly composed of studies based on archaeology, and the main task of the museum until now has actually been the preservation and management of these relics. Due to the increase in user-experienced exhibitions around the world, a change of times that requires new contents fused with technology as a new task of museums is rapidly being pursued. However, this new approach must be promoted based on digital transformation to ensure its effectiveness and continuity.

Digital transformation of traditional cultural heritage includes simply digitizing data, and it is necessary to consider new data generation methods and standards according to technology and equipment. In addition, it is necessary to define methods for changing and improving existing data according to the use of content. These digitized data can be easily retrieved and, if necessary, the relationship between each relic's must be established to be reborn as information necessary for practical use. When the data analysis is completed in this way, the traditional cultural heritage management platform that provides an intuitive interface that can be easily accessed and used by curators working in the museum and continuously updates the development of technologies must be completed.

For this digital transformation, we are currently conducting research to develop a platform, including interfaces for creation, enhancement, analysis, search, relationship definition, and practical use of traditional cultural heritage data. In this paper, we present applied artificial intelligence technologies in our development of an effective Korean traditional cultural heritage management platform. Among the research in progress in our work, object detection technology such as animals, plants, and people in traditional Korean painting, high-resolution conversion technology of the previously photographed low-resolution Korean traditional cultural heritage image data, and recognition of related information in texts related to the Korean traditional cultural heritage. We also introduce the application contents related to this research and development in the following sections.

## II. OBJECT DETECTION IN KOREAN TRADITIONAL PAINTINGS

The development of object detection technology in images is currently showing excellent results based on natural images.



However, to apply this object detection to Korean traditional painting, other works are needed. In particular, in the case of Korean painting, there is not much development of object detection technologies in painting using artificial intelligence. Therefore, in this study, from the preparation of the dataset to the ground truth dataset, the research was conducted in parallel with the preparations to complete the experimental environment.

### A. Preparation of Object Detection

For the composing of the experiment, in this study, a GT data generation tool for learning about cultural heritage image data was first developed. This GT data generation is a basic function to verify the effectiveness of research results, and it was conducted in parallel with the study because there is no defined data set for the information of individual objects in Korean traditional painting.

Due to the characteristics of Korean traditional painting, experts with an understanding of traditional painting were employed as image annotators. In the tagging tool, various functions were added to make it easier for workers based on the traditional culture major to increase the ease of work. By supporting the progress bar that allows you to check the current progress and before and after images of the image to be worked on, the user can make tagging easier. Additionally, a shortcut function was supported to facilitate annotation work on image extraction properties. It supports the property status window that shows the image information (file name, image size, etc.) currently being worked on to the worker to check the work progress. Simple image processing and image editing functions (image resize, image filtering, etc.) are also provided for accurate bounding box selection. Figure 1 shows examples of traditional Korean paintings that were tagged using these tools.

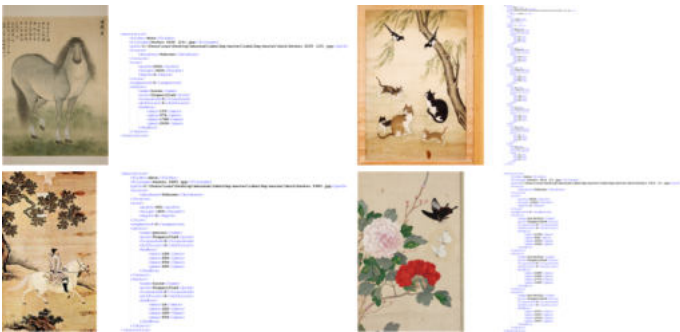


Fig. 1. Examples of annotated Korean traditional paintings.

### B. Examples of Object Detection

First, in this study, object detection was applied to images of Korean cultural heritage based on the previously learned deep learning model and the results were reviewed. The deep learning framework used was Tensor-Flow, and the COCO data-set was used as the training data, and the applied network was Faster-rcnn-inception-v2. Figure 2 shows the object detection results in Korean traditional painting using the existing model.



Fig. 2. Object detection based on the conventional model.

After attempting object detection using the existing learned model, Korean painting cultural properties are additionally learned, and additionally, transfer learning technology is applied to the object detection model to make better results. The deep learning framework used in this study is Tensor-Flow 1 and Keras, the training data are additionally learning the painting cultural heritage data-set to the existing COCO data-set, and the applied network is using RetinaNet.

RetinaNet is a structure that combines Focal Loss and Feature Pyramid Network and has an advantage in fast detection time and improves object detection performance in degradation problem of one stage detector. Using RetinaNet, we are confirming results that are advantageously applied to the detection of small objects included in painting cultural properties. Figure 3 is an overview of the deep learning model used in this study, and Figure 4 is the research result in the current situation, and it can be confirmed that the detection performance is improving little by little compared to the previous one.

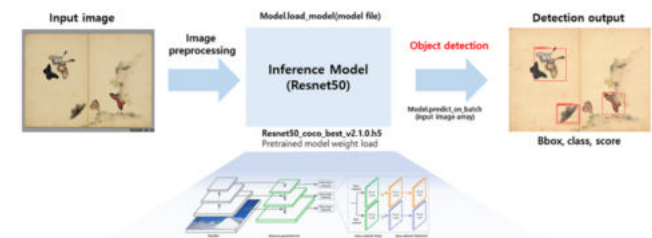


Fig. 3. Overview of deep learning model.

## III. SUPER-RESOLUTION OF THE OLD KOREAN CULTURAL HERITAGE IMAGE

Super-resolution is an image processing technology that converts a low-resolution image into a high-resolution image. It can be used in various fields by allowing images that have



Fig. 4. Current Object detection Results.

been lost or stored in a low resolution due to transmission or storage to be viewed as clear, high-definition images. This image high-resolution technique started with simple interpolation methods, such as bilinear interpolation and bicubic interpolation and has developed into a form using machine learning. Before the advent of super-resolution technology using deep learning, non-deep learning machine learning methods such as SelfExSR [9] and SI [10] were typical. SI is a method for learning linear mapping from low-resolution to high-resolution images with a small amount of data. However, in most super-resolution research fields, since a large amount of data can be easily obtained, the performance is inevitably inferior to that of the deep learning method that uses a large amount of data. While deep learning is a convolutional neural network that can learn various features through non-linear mapping, the non-deep learning method has limitations because it learns linear mapping. Additionally, there is a clear limitation in that performance changes significantly even if hyper-parameters such as patch size change even slightly.

The super-resolution method using deep learning started with SRCNN ((Super-Resolution Convolutional Neural Network) [11], and many technologies such as VDSR (Accurate Image Super-Resolution Using Very Deep Convolutional Networks) [12], ESPCN (Efficient Sub-Pixel Convolutional Neural Network) [13], SRResnet [14], and EDSR (Enhanced Deep Residual Networks for Single Image Super-Resolution) [15] have been introduced. Models suitable for super-resolution have been rapidly developed and have shown steady performance improvement. In particular, RCAN (Residual Channel Attention Networks) [16] has an uncomplicated structure and consistently shows high performance in benchmark tests, so it is an excellent model for natural images. In anticipation that it will show excellent results in cultural property images, this study was conducted by modifying the model structure of the previous studies.

Existing super-resolution research has been conducted in

various ways, but mostly it has been done only on unspecified general natural images, and has not been verified in cultural heritage image data so far. In this respect, there is a distinct difference from natural images, so a specialized method is needed. This paper proposes a method for super-resolution 4x and 8x images of cultural assets using deep learning. The model structure inspired by RCAN We propose a patch extraction method that uses the characteristics of cultural assets images and a deep learning method that uses cultural heritage image data-sets in various ways. Therefore, it is expected that it will be helpful in research related to cultural heritage in the future.

#### A. Learning Model of super-resolution of the Korean cultural heritage images

In this paper, the study was conducted by referring to the structure of the Residual in Residual Network. The network is composed of two ResGroups in a reduced form than in the paper. Each group has 10 ResBlocks, and each ResBlock is the type referenced in [16]. After adding the input image to the result after passing through the ResGroup, the 2x magnification module consisting of convolution, ReLU activation function, and pixel shuffle [13] is used to increase the image size to the desired size. At this time, in the case of a 4x magnification network, the above module was repeatedly configured 2 times, and in the case of an 8x magnification network, 3 times.

For the learning of cultural data properties, it was conducted in parallel with learning using general images. For general image learning, the DIV2K [17] dataset given in NTIRE2017 was used, and 2K quality images consist of 800 images for training and 100 images for testing. It is composed of general natural images not specific to either side, so it is easy to learn the characteristics of general images necessary for super-resolution.

We used the weight learned above to further learn by using cultural assets images for transfer learning. In this way, with the model learning the characteristics of natural images, additional learning was conducted using the cultural assets image dataset of Korean traditional cultural heritage. The advantage of this method is that it allows additional learning from cultural heritage image data while having a filter that can extract significant features for super-resolution from natural images.

#### B. Results of super-resolution

In all experiments, L1 loss was used, optimization was performed using the Adam optimizer, and the initial learning rate was set to  $1e-4$ . The batch size was 8, and patches cut to  $32 \times 32$  were used. In the case of training with only one dataset, the test was conducted with the weight with the lowest valid loss while training up to 65 K iterations. In the case of transfer learning, the test was conducted with the weight with the lowest effective loss by additionally learning as much as 10 K repetitions from the cultural heritage image data with the best weight obtained from learning with DIV2K.

All the result images of the deep learning model show better results both numerically and visually than the results of bicubic interpolation. In particular, the average PSNR value for the resulting image is at least 1.1 dB at 4x super-resolution and at least 1.09 dB at 8x super-resolution, showing a significant performance improvement. Among the deep learning methods, numerically, the result image of the learning model through transfer learning is the best, but only subtle differences can be seen with the naked eye. Figure 5 shows the results of the proposed method, the linear interpolation method, and the ground thumb image.

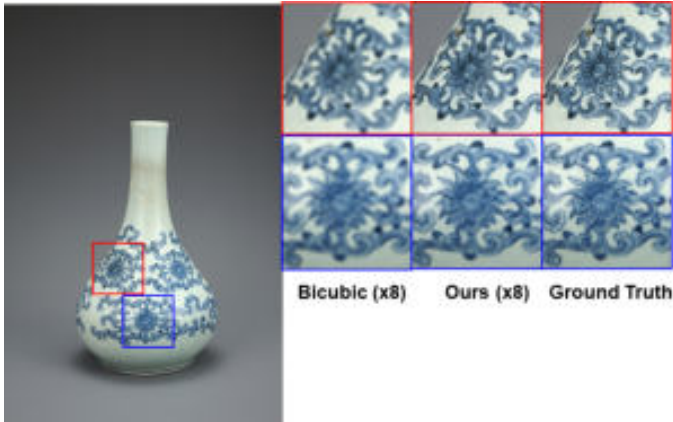


Fig. 5. Comparison of super-resolution result.

Most cultural heritage image data have a monotonous background and an object is located in the middle of the image. To this end, we newly construct a cultural property dataset, apply a method for extracting patches only from the central part, and propose a learning method using DIV2K, a natural image dataset, and cultural property data appropriately. As a result, compared to bicubic interpolation, about 1.25 dB at 4 times super-resolution and about 1.26 dB at 8 times super-resolution increased. Compared to the simple DIV2K learning method, performance increased by 0.06 dB at 4x magnification and 0.17 dB at 8x magnification. Figure 6 shows one of the super-resolution images of the proposed method.

#### IV. TEXT ANALYSIS IN KOREAN HERITAGE DESCRIPTION DATA

To extract formal and meaningful information from atypical traditional cultural heritage text data, this study applied a deep learning-based language model to learn semantic information and structural information of Korean sentences to develop an entity name recognition model and relationship extraction model.

##### A. Named Entity Recognition

Named Entity Recognition (NER) technology is one of the basic technologies, and it is also an important research area in terms of application. The process of learning with a language model optimized for traditional cultural heritage through post-training and fine-tuning based on the Korean



Fig. 6. Super-resolution result of the Korean cultural heritage image.

language model can further improve the model's performance. To recognize proper nouns in traditional culture, the above method is required, and to better understand traditional culture individual information in sentences, research and development is needed accordingly.

To understand the stylistic characteristics of domain-specific sentences and the complex understanding of the language, it is necessary to construct learning data, and fine-tuning of learning is required for more accurate proper noun entity recognition. NER is a technology for extracting individual names, such as person names (PS), place names (LC), and organization names (OG), which have unique meanings from a document and recognizing the extracted entity names.

Recently, NER in the field of natural language processing proceeds by pre-learning the LM through the encoder of the Bi-directional Transformer that can consider the context from a large corpus, and then apply it to the NER task. For the Pre-trained Language Model (PLM) to more effectively handle semantic and structural information of language from the text, model structures such as BERT, RoBERTa, and ELECTRA should be used [18][19][20].

This approach is because these models consider self-supervised learning objectives of language models such as Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). For this purpose, the NSP technique suggested by BERT was excluded. In this study, the RoBERTa model, which adopts the masking pattern of the MLM technique as a dynamic method, and the Korean language model of ELECTRA, which introduces the replaced token detection technique that learns each sample data more effectively than the MLM technique, were applied.

Compared to the multilingual language model, the Korean version of the language model, pre-learned with a large-capacity Korean Wiki corpus, can better capture the features of the Korean language and is built as a lexicon that expresses the Korean language better in terms of vocabulary. For this study, we tried to enable the language model to capture the semantic and structural meaning of tokens by using various

Korean-based language models considering Korean, which has agglutinative language characteristics. In this study, not only the existing traditional vocabulary-based tokenization method, but also a study was conducted to extract objects according to the characteristics of the language model by using the Korean language model that can consider the Korean semantic form.

The proposed model represents the structure of entering Korean language models such as KoBERT, RoBERTa, and KoELECTRA through the tokenization process based on the segmentation method by receiving sentences as input. Classifies input tokens through a language model that extends the transformer encoder structure. In the fine-tuning process, we tried to improve the model performance by adding MLP Layers to consider in more detail the context that not be considered only with the language model. The used MLP Layers structure can be divided into Bi-LSTM, Bi-LSTM-CRF, and CRF. In addition to KoBERT, RoBERTa, and KoELECTRA, the language models used in this study include KorBERT, KorBERT-morph, and HanBERT.

### B. Results of NER in the Korean cultural heritage description

Using the published NER dataset, preprocessing for performance evaluation of each language model and performance evaluation in several models were performed. The corpus used in this study is the below [21].

- NIKL – The NER corpus distributed by NIKL with data of 3 million words (2 million written, 1 million spoken) includes 15 analysis markers to recognize the entity name boundary.
- AIR and NAVER NER Challenge – The data designed based on the CoNLL-2003 data format are for competition data, and the test set is not disclosed. Therefore, the verification data set will be used for testing in the evaluation of actual learning.
- KMOU-NER Corpus – The data constructed by Korea Maritime University is constructed by dividing approximately 24 K of ignition data into 10 classes.
- KLUE – data built to perform 8 Korean Natural Language Understanding (NLU) tasks, including 6 classes for Korean NER

The actual experimental result is a simulation in progress based on the recognition data of all corpus objects, and the entire dataset is divided into train, dev, and test 8:1:1, respectively, and early stopping is applied to prevent overfitting problems. The experimental model was KoBERT, KoELECTRA base v3 model of the Korean model, and in the case of the multilingual model, the experiment was conducted in xlm-roberta-based. So far, although the multilingual model has the largest model size, the KoBERT model pre-trained in Korean has recorded the best performance.

In this study, research is in progress to properly prove the effectiveness of constructing NER data in the traditional culture domain through the developed Korean model. The goal is to visualize the knowledge relationship graph through the NER model and relationship extraction results developed by

understanding the linguistic characteristics of the Korean language from traditional cultural heritage text data. By using this knowledge, it is expected to provide a service to researchers and learners who want to use traditional cultural content to understand the subject matter they want by looking at the graph connected according to the conditions such as genre and era.

### V. CONCLUSIONS AND FURTHER WORKS

Recently, as the performance of computers is improved and the capacity of memory is increased, digitalization around the world has become a realistically feasible state. This phenomenon shows the possibility of digitization even for vast relics in museums and exhibition halls. Whereas digitization in the past was to create simple digital data, current digitization is applying it to the digital world or reality, such as virtual reality or digital twin. It is being changed for be used in the world.

In line with this trend, the digitization of relics in museums and exhibition halls around the world is changing for use for other applications such as virtual exhibition halls beyond the purpose of preserving the information on existing relics. To use it for other purposes, information must be put into the data stored. Such information is made through data analysis, but there is a clear limitation in the fact that a person performs analysis on a large amount of information and informatics it.

For this reason, many technologies for data processing using artificial intelligence technology are being studied. Technologies for analyzing information from numerous images or text and processing it into the desired form have been developed to an astonishing level. Unfortunately, however, these technologies are concentrated on universal images and natural language, so it is difficult to use them directly for analysis of old museums or ancient documents. For this reason, the actual development of artificial intelligence-based analysis and transformation of Korean traditional cultural heritage is still insufficient.

In this situation, this study studied the analysis and transformation technology of artificial intelligence-based cultural heritage data that can be used more by using the data of the actual museum and presenting its direction. In this ongoing study, the main purpose is to create an optimal usage model for the data of Korean traditional cultural heritage in the actual museum.

### ACKNOWLEDGMENT

This research is supported by the Ministry of Culture, Sports and Tourism and Korea Creative Content Agency (Project Number: R2020040045)

### REFERENCES

- [1] [www.wikiart.org](http://www.wikiart.org).
- [2] [www.europeana.eu](http://www.europeana.eu).
- [3] Gjorgji Srezoski, et al., "Omniart: Multi-task Deep Learning for Artistic Data Analysis," arXiv preprint arXiv:1708.00684, CoRR, Aug. 2017.
- [4] [www.museumnext.com](http://www.museumnext.com).

- [5] Nam Vo, et al., "Composing Text and Image for Image Retrieval – An Empirical Odyssey," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6439-6448, California, America, June 2019.
- [6] Marvin Teichmann et al., "Detect-to-Retrieve: Efficient Regional Aggregation for Image Search," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5109-5118, California, America, June 2019
- [7] Tao Dai, et al., "Second-order Attention Network for Single Image Super-Resolution," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11065-11074, California, America, June 2019.
- [8] J Bee Lim, et al., "Enhanced Deep Residual Networks for Single Image Super-Resolution," NTIRE 2017 Workshop. pp.136-144, Hawaii, America, July 2017.
- [9] J. B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 5197–5206, 2015.
- [10] J. Choi and M. Kim, "Super-Interpolation With Edge-Oriented-Based Mapping Kernels for Low Complex  $2\times$  Upscaling," IEEE Transactions on Image Processing, vol. 25, no. 1, pp. 469-483, January 2016, doi: 10.1109/TIP.2015.2507402.
- [11] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," European Conference on Computer Vision (ECCV), pages 184–199. Springer, 2014.
- [12] J. Kim, J. K. Lee and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," IEEE Conference on Computer Vision and Pattern Recognition, pp. 1646-1654, June 2016.
- [13] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1874–1883, 2016.
- [14] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al., "Photo-realistic single image super-resolution using a generative adversarial network," arXiv preprint arXiv:1609.04802, 2016.
- [15] B. Lim, S. Son, H. Kim, S. Nah, and K.M.Lee, "Enhanced Deep Residual Networks for Single Image Super-Resolution," In CVPRW, 2017.
- [16] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image Super-Resolution Using Very Deep Residual Channel Attention Networks," In ECCV, 2018.
- [17] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," IEEE Conference on Computer Vision and Pattern Recognition Workshop, pp. 1122-1131, July 2017.
- [18] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C., "Neural architectures for named entity recognition," arXiv preprint arXiv:1603.01360, 2016.
- [19] Nie, B., Ding, R., Xie, P., Huang, F., Qian, C., and Si, L., "Knowledge-aware Named Entity Recognition with Alleviating Heterogeneity," AAAI Conference on Artificial Intelligence, Vol. 35, No. 15, pp. 13595-13603, May 2021.
- [20] Tran, Q., MacKinlay, A., and Yepes, A. J., "Named entity recognition with stack residual lstm and trainable bias decoding," arXiv preprint arXiv:1706.07598, 2017.
- [21] Park, S., Moon, J., Kim, S., Cho, W. I., Han, J., Park, J., et al. "KLUE: Korean Language Understanding Evaluation," arXiv preprint arXiv:2105.09680, 2021.

# Community Detection with Graph Neural Network using Markov Stability

Shunjie Yuan

School of Cyber Engineering  
Xidian University  
Xi'an, China  
shunjiey@foxmail.com

Chao Wang

School of Cyber Engineering  
Xidian University  
Xi'an, China  
wangchao@xidian.com

Qi Jiang

School of Cyber Engineering  
Xidian University  
Xi'an, China  
jiangqixdu@xidian.edu.cn

Jianfeng Ma

School of Cyber Engineering  
Xidian University  
Xi'an, China  
jfma@mail.xidian.edu.cn

**Abstract**—Community detection is a fundamental task in network analysis. With the recent development of deep learning, some community detection methods related to deep learning have been proposed. However, these methods still face limitations with respect to accuracy and runtime. In this paper, we propose a graph neural network (GNN) based overlapping community detection method CDMG from the perspective of optimizing Markov Stability, which is a statistical property of the Markov process quantifying the quality of a community partition. Specifically, we train a graph neural network to generate the node embedding defined as the community affiliation weight matrix that denotes the strength of nodes' membership in communities while maximizing the Markov Stability. Then the community affiliation weight matrix is converted to a community affiliation matrix representing the community partition. Experiments on several real-world networks demonstrate the superiority of CDMG compared to other representative community detection algorithms. Additionally, since Markov Stability relies on a time parameter Markov Time, we observe that there exists a Markov Time threshold for a network. When using the Markov Time near the threshold, CDMG can produce a better community partition with much higher accuracy.

**Index Terms**—Complex Network, Community Detection, Graph Neural Network, Markov Stability

## I. INTRODUCTION

Many real-world systems can be represented as complex networks, such as the Internet, neural system, and transportation system. Community detection is fundamental research in network analysis because many further studies rely on community structure. For example, researchers elucidate the relationship between the structure of neuronal networks and the functional dynamics that they implement in the network of *Caenorhabditis elegans* connectome [1]. Because of the importance of community detection, it has attracted a great deal of attention from researchers and numerous algorithms have been proposed.

In recent years, Graph Neural Networks (GNNs) have become a new research hotspot, which is a powerful tool to deal with graph-structured data with deep learning algorithms. Research on GNNs generally can be divided into two categories, spectral-based and spatial-based approaches. Spectral-based methods define graph convolution operations as filters on the frequency domain of a graph. Because GNNs can reveal the higher-order structural information based on the non-linear

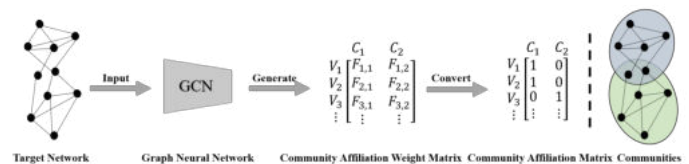


Fig. 1. The overview of the proposed algorithm CDMG. A graph neural network is trained to generate the community affiliation weight matrix that represents the strength of nodes affiliating to communities. Then the community affiliation weight matrix is converted to a community affiliation matrix that denotes the community partition.

feature aggregation and the information propagation across the network, which has been widely applied in several network analysis tasks, such as link prediction, node classification, and community detection [2]–[6]. And our approach is motivated by the success of these works that will be discussed in detail next.

The structure of a network can affect the dynamical behavior that takes place on the network in terms of its high-connectivity within communities. On the other hand, the dynamics can reveal features of the network structure. Based on this idea, Markov Stability [7]–[9], a statistical property of the Markov dynamic, is deployed to measure the quality of community structure in this work. A large Markov Stability always corresponds to a robust community structure, which indicates that a random walker is difficult to escape the communities [7]. For instance, we discuss the influence of community structure on Markov Stability in the well-studied network Karate that has two communities [10]. As we destroy the community structure of 10 percent of nodes each time in Fig. 2 (a), the Markov Stability decreases gradually. And as shown in Fig. 2 (b), the ground-truth partition corresponds to the largest Markov Stability. Not only Markov Stability can indicate the quality of community structure but also the community detection methods based on Markov Stability can provide multiple results compared with other methods as Markov Stability relies on Markov Time.

In this paper, we propose a new community detection algorithm named CDMG (Community Detection based on Markov Stability and Graph Neural Network) to detect communities with GNNs from the perspective of optimizing Markov

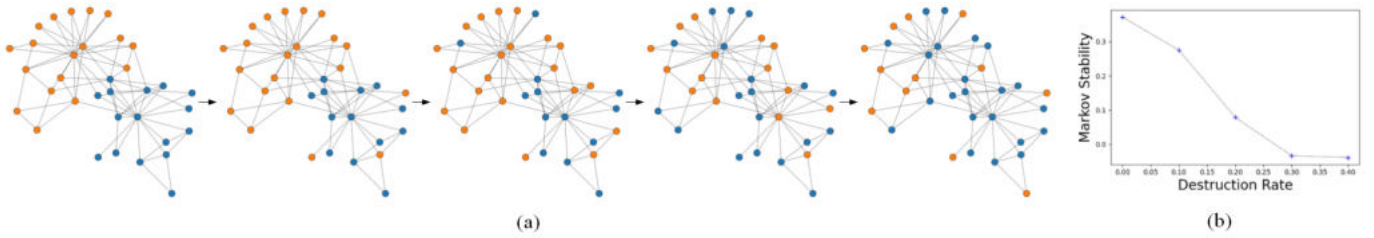


Fig. 2. The influence of community structure on Markov Stability in Karate. As we destroy the community structure step by step, the Markov Stability decreases gradually. And the ground-truth partition corresponds to the highest Markov Stability. Detecting communities with our method is a reverse process of the above.

Stability. Specifically, we use Markov Stability as the loss function which measures the quality of a community partition, and we train a graph neural network to generate the node embedding defined as the community affiliation weight matrix that represents the strength of nodes affiliating to communities while minimizing the loss function. The community affiliation weight matrix then is converted to a community affiliation matrix, a binary matrix representing the community partition. The overview of CDMG is shown in Fig. 1.

The main contributions are listed as follows:

- We propose a GNN based method CDMG for overlapping community detection from the perspective of optimizing Markov Stability.
- We conduct experiments on four real-world networks and the results demonstrate that CDMG outperforms other established methods in most cases.
- We discuss the influence of the time parameter Markov Time on the performance of CDMG and find out that there is a Markov Time threshold for a network. When Markov Time  $t$  is around the threshold, CDMG can result in better community partition with high accuracy.

The rest of the paper is organized as follows. In section II, we introduce some related work in community detection. Section III contains the explicit details about Markov Stability, Graph Neural Network, and the proposed method CDMG based on them. Section IV provides a thorough evaluation of our method and shows its superior performance compared with other representative algorithms on several real-world networks. Section V concludes our work.

## II. RELATED WORK

Many community detection methods have been proposed from different perspectives. One direction to uncover community structure is to optimize the measures that quantify the quality of community structure like modularity [5], [11]. Another direction to detect communities is to infer the relationship between vertices and communities based on nonnegative matrix factorization [12]–[15]. In this section, we mainly focus on some deep learning based methods. These methods can be broadly divided into methods based on certain neural networks like Generative Adversarial Networks (GANs) [16], GNNs [5], [6], and Attention Model [17] and methods based on Graph Representation with Cluster algorithms [18]–[22].

In [16], a generative adversarial network for community detection, CommunityGAN, is designed, which includes a generator that tries to generate a vertex subset with a high probability to be a clique, and a discriminator that tries to discriminate the clique from the generator. The output of CommunityGAN is the node embedding which is also the community affiliation weight matrix representing the community partition. Recently, several GNN based community detection methods have been proposed. For example, Tsitsulin et al. train a single-layer graph neural network to generate the community affiliation weight matrix while minimizing the loss function, the reformulated modularity composed of the community affiliation weight matrix [5]. Shchur et al. present the NOCD model that generates the community affiliation weight matrix with a graph neural network, where the balanced negative log-likelihood of the Bernoulli–Poisson (BP) model composed of the community affiliation weight matrix is used as the loss function [6]. The core idea of BP model is that the more communities two nodes are in common, the more likely they are to be connected by an edge. As we can see, the key point of community detection with GNNs is to build an efficient model, where community structure information such as community affiliation weight matrix can be integrated into the loss function. Moreover, the Attention Model has also been introduced into the community detection field. Lobov et al. propose a new model which is based on the Transformer model [23]. Specifically, they utilize the encoder part of the Transformer to transform the Bethe Hessian embeddings and produce the probability of each cluster for each node while optimizing the soft modularity loss function [17].

On the other hand, several graph representational learning algorithms have been designed. Given any graph, the graph representational algorithms can learn a low-dimensional vector for each vertex, which can then be used for various network analysis tasks, such as community detection. DeepWalk [19] utilizes random walk to generate node sequences and adopt Skip-Gram to learn node embedding, which preserves second-order proximity. Node2Vec [20] is an extended version of DeepWalk where it deploys a biased random walk to generate node sequences. LINE [21] preserves both the first-order and the second-order proximity while learning node embedding. MNMF [22] is an NMF-based representation learning model, which preserves both the microscopic structure (first and

second-order proximities) and mesoscopic community structure. ComE [24] is a framework that jointly solves community detection, community embedding, and node embedding together, and it adopts multivariate Gaussian distributions to represent communities based on the output node embedding.

### III. METHODOLOGY

Given an undirected network  $G = (V, E)$ , where  $V$  is a set of nodes and  $|V| = N$ ,  $E$  is a set of edges among nodes. The goal of community detection is to assign nodes into  $K$  communities. Such assignment can be represented as a community affiliation weight matrix  $F \in \{x \mid 0 \leq x \leq 1\}^{N \times K}$ . From this perspective detecting communities boils down to inferring the community affiliation weight matrix  $F$  when given the target network  $G$ .

#### A. Markov Stability

Markov Stability [7]–[9] is a statistical property of the Markov dynamic, which evaluates the quality of a community partition in terms of the persistence of the Markov dynamics within the communities during the time scale  $t$ , which means the larger Markov Stability is, the more unlikely a random walker is to escape the communities within time  $t$ . One main advantage of the community detection methods based on Markov Stability is that they can reveal community structure with different Markov Time  $t$  because Markov Stability is a time-parametrized function. Markov Time acts as a resolution parameter for community detection [1]. For an undirected and unweighted network  $G$ , its topology is encoded in the adjacency matrix  $A \in R^{N \times N}$ . We define the  $n$ -dimensional vector  $d$  with components  $d_i = \sum_{j=1}^n A_{ij}$ , the diagonal matrix  $D = \text{diag}(d)$  and the total weight of the degrees of the networks is  $m = \sum_{i,j} A_{ij}/2$ . Then we define a discrete-time Markov process governed by the following dynamics:

$$P_{t+1} = P_t D^{-1} A \equiv P_t M \quad (1)$$

where  $P_t$  is the probability vector and  $M$  is the transition matrix. Given a partition  $H$  at time  $t$ , the Markov Stability is defined as the trace of the clustered autocovariance of the diffusion process:

$$ms(t, H) = \text{trace} \left( H^T \left[ \prod P(t) - \pi^T \pi \right] H \right) \quad (2)$$

where  $\pi = d^T/2m$  is a unique stationary distribution of the process,  $P(t) = M^t$  and  $\Pi = \text{diag}(\pi)$ . The optimal community partition  $H$  corresponds to the maximal Markov Stability. But maximizing (2) is an NP-hard problem with no guarantees of global optimality, the existing methods utilize Louvain or other heuristic methods to optimize (2) [7], [25], but here we adopt a graph neural network.

#### B. Graph Neural Network

Graph Neural Networks are a class of models that can perform non-linear feature aggregation and information propagation with respect to network structure. For the purpose of this work, we use Graph Convolutional Network (GCN) [26] to

output node embedding for vertices. Given the node attributes  $X$ , a single-layer GCN can be defined as:

$$F := GCN(A, X) = \delta(\hat{A}XW) \quad (3)$$

where  $\delta$  is the non-linear activation function, such as ReLU,  $\hat{A} = \tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2}$  is the normalized adjacency matrix,  $\tilde{A} = A + I_N$  is the adjacency matrix with self-loops, and  $\tilde{D}$  is the diagonal degree matrix of  $\tilde{A}$ . If node attributes  $X$  are not available, we can use  $A$  as node features.

#### C. Method

Markov Stability can evaluate the quality of the network structure. The output of GCN can be considered as an embedding of nodes with the aim of preserving the network structure [6]. Therefore, we propose to optimize the output of GCN with Markov Stability. The core idea of our method is to generate a community partition with the maximal Markov Stability by using a graph convolutional network. Specifically, we utilize a 2-layer graph convolutional network to generate the node embedding defined as the community affiliation weight matrix  $F$ :

$$F := GCN(A, X) = \text{ReLU} \left( \hat{A} \text{ReLU} \left( \hat{A} X W^1 \right) W^2 \right) \quad (4)$$

The main difference between our GCN model and the standard GCN is that we introduce normalization after the second graph convolution layer, which leads to noticeable improvements in performance. The Markov Stability is used as the loss function which is defined as:

$$\mathcal{L}(F) = -\text{trace} \left( F^T \left[ \prod P(t) - \pi^T \pi \right] F \right) \quad (5)$$

Different from the Markov Stability mentioned before, here we use the community affiliation weight matrix  $F$  to approximate the community affiliation matrix  $H \in \{0, 1\}^{N \times K}$  where  $H_{uv}$  denotes that node  $u$  belonging to community  $v$  or not. By minimizing (5), we can find the optimal community affiliation weight matrix  $F$ , then we convert it to the community affiliation matrix  $H$ , assigning the nodes to the communities with a threshold  $p$ . If  $F_{uc}$  is bigger than the threshold  $p$ , we believe that node  $u$  belongs to community  $c$  and set  $H_{uc}$  to 1 else 0. The threshold  $p$  and the Markov Time  $t$  are two hyperparameters which will be discussed in next section. To sum up, using GCN and Markov Stability for community detection has several advantages. First, GCN can generate similar community affiliation weight vectors for neighboring nodes, which improves the quality of community detection. Second, the node attributes can be incorporated into the model. Finally, because Markov Stability relies on Markov Time  $t$ , our method can provide multiple results when using different Markov Time  $t$ . And we design experiments to analyze the influence of Markov Time  $t$  on the performance of CDMG in next section.

### IV. EVALUATION

In this section, we perform a thorough evaluation of CDMG and show its superior performance compared to other competing methods for overlapping community detection. First, we



describe the experimental networks, the accuracy metrics, and the comparative methods as well as the parameter setting of CDMG. Then we analyze the results and runtime of these community detection methods and explore the influence of Markov Time on the performance of CDMG especially. The experiments were performed on a computer running Windows Server 2016 with Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz CPUs, 96GB of RAM.

### A. Datasets

We conduct experiments on a variety of real-world networks including LiveJournal, Amazon, YouTube and DBLP. The statistics of these large-scale networks are summarized in Table I. Considering both the training time of these learning methods and the performance of the machine we used, we only sample four subgraphs with 100 ground-truth communities from these large-scale networks. LiveJournal is a free on-line blogging community where user can find friendship and form a group which other people can join. The vertices represent users and edges represent friendship. This subgraph has 1118 vertices and 2047 edges. The network of Amazon is collected by crawling its website, which is based on purchase information of the Amazon website. The vertices represent products, and edges represent those products that are frequently co-purchased. This subgraph has 3225 vertices and 10262 edges. YouTube is a popular American online video-sharing platform that includes a social network where the vertices represent users, and edges represent friendship among users. In the YouTube social network, users form friendships with each other and users can create groups that other users can join. This subgraph has 4890 vertices and 20787 edges. The DBLP computer science bibliography provides open bibliographic information on major computer science journals and proceedings. In the DBLP network, vertices represent authors, and edges represent that the authors have published at least one paper together. This subgraph has 10824 vertices and 38732 edges.

TABLE I  
STATISTICS OF SEVERAL LARGE-SCALE NETWORKS.

Network	$ V $	$ E $	$ C $
Amazon	0.34M	0.93M	49K
YouTube	1.10M	3.00M	30K
DBLP	0.43M	1.30M	2.5K
LiveJournal	4.00M	34.9M	310K

### B. Metrics

To quantitatively evaluate the performance of these community detection methods, we choose two metrics: overlapping Normalized Mutual Information (NMI) [27] and Omega Index ( $\Omega$ -Index) [28]. NMI is widely used to measure the performance of a community detection algorithm, which adopts the criterion used in information theory to compare the detected communities and the ground-truth communities. Omega Index is the overlapping version of Adjusted Rand Index, which is based on pairs of nodes in agreement in two partitions.

### C. Comparative Methods

Clique Percolation Method (CPM) [29] is a typical overlapping community detection algorithm which assumes that communities consist of overlapping complete subgraph. Sym-NMF [12] is a general framework for graph clustering, which inherits the advantages of NMF by enforcing nonnegativity on the clustering assignment matrix. NSED [13] is a nonnegative symmetric encoder-decoder approach proposed for community detection. Node2Vec [20] is a node embedding algorithm, which adopts biased random walk and Skip-Gram model to embed vertices. LINE [21] is a node embedding algorithm, which preserves the first-order and the second-order proximity among embeddings. MNMF [22] is an NMF-based node embedding method which considers the microscopic structure (the first-order and second-order proximities) and mesoscopic community structure. Because LINE, Node2Vec and MNMF are representational learning algorithms, we use the K-means algorithm to detect communities based on their output node embeddings. NOCD [6] is a community detection algorithm with a graph neural network, which is defined by Bernoulli-Poisson model. ComE [24] is a framework for community detection, community embedding, and node embeddings, which jointly detects communities and learns the embeddings.

### D. Parameter Setting

After obtaining the community affiliation weight matrix  $F$ , we need to convert it to the community affiliation matrix  $H$ . If  $F_{uc}$  is bigger than the threshold  $p$ , we believe that node  $u$  belongs to community  $c$ . The threshold  $p$  is defined as  $p = \sqrt{-\log(1 - \epsilon)}$  where  $\epsilon$  is the background edge probability  $\epsilon = 2|E|/|V|(|V| - 1)$ . The basic intuition about assigning nodes to communities is that if two nodes belong to the same community, then the probability of having an edge between them through the community should be bigger than the background edge probability [16]. Markov Time  $t$  is another hyperparameter that determines the time of the Markov process. In general, we set Markov Time  $t$  to 1. We also analyze the influence of Markov Time  $t$  on community detection with a certain community number.

### E. Results of Community Detection

In Table II, we summarize the community detection results of the proposed CDMG and other competing methods on the real-world networks in terms of NMI and Omega Index. Compared with other algorithms, CDMG can achieve the highest NMI score and Omega Index in most cases, which means CDMG can provide a more robust community structure with higher accuracy. Specifically, in Amazon, ComE outperforms other algorithms, achieving the highest NMI and Omega Index. In LiveJournal, YouTube and DBLP, our method performs the best, especially, the Omega Index of CDMG is much larger than other algorithms' in these networks. One possible explanation for the superior performance of CDMG in these experimental networks is that because both GCN and Markov Stability deeply depend on network structure, CDMG

TABLE II  
NMI SCORE AND OMEGA INDEX OF DIFFERENT METHODS ON SEVERAL EXPERIMENTAL NETWORKS

Network	LiveJournal		Amazon		YouTube		DBLP	
Metric	NMI	$\Omega$ -Index	NMI	$\Omega$ -Index	NMI	$\Omega$ -Index	NMI	$\Omega$ -Index
CPM	0.2196	0.0607	0.0995	0.0937	0.0000	0.0152	0.0439	0.0144
NSED	0.0293	0.0408	0.1104	0.0496	0.052	0.0543	0.0004	0.0000
SymNMF	0.0442	0.0556	0.1699	0.1621	0.1084	0.0963	0.0201	0.0458
MNMF	0.0219	0.0571	0.0000	0.0768	0.0872	0.1475	0.0030	0.0178
LINE	0.1324	0.0250	0.1815	0.0465	0.0473	0.0198	0.0171	0.0041
Node2Vec	0.2102	0.0636	0.2236	0.1793	0.0659	0.0487	0.0242	0.0223
NOCD	0.2007	0.1157	0.2323	0.1945	0.1773	0.2145	0.0631	0.0742
ComE	0.2542	0.0674	<b>0.2643</b>	<b>0.2447</b>	0.0973	0.0551	0.0419	0.0429
CDMG	<b>0.3015</b>	<b>0.5503</b>	0.1656	0.1909	<b>0.2453</b>	<b>0.4846</b>	<b>0.1141</b>	<b>0.2099</b>

TABLE III  
NMI SCORE AND OMEGA INDEX OF CDMG ON SEVERAL EXPERIMENTAL NETWORKS WHILE MARKOV TIME T INCREASING.

Network	LiveJournal		Amazon		YouTube		DBLP	
Metric	NMI	$\Omega$ -Index	NMI	$\Omega$ -Index	NMI	$\Omega$ -Index	NMI	$\Omega$ -Index
1	0.3015	0.5503	0.1656	0.1909	0.2453	0.4846	0.1141	0.2099
2	0.2625	0.4236	0.1822	0.2041	0.2439	<b>0.5201</b>	0.1040	0.2665
3	0.2915	0.4956	0.2212	0.2135	0.2469	0.5127	0.1205	0.2779
4	0.3423	0.5656	0.2061	0.2041	<b>0.2604</b>	0.5080	0.1259	0.3661
5	0.3540	0.6445	0.2338	0.2415	0.2467	0.4813	0.1474	0.4353
10	0.4014	0.7429	0.2329	0.2591	0.2273	0.4883	0.1982	0.5273
20	<b>0.4670</b>	<b>0.8295</b>	0.2937	0.2803	0.1852	0.3683	0.2779	0.6060
50	0.4531	0.7720	0.3276	0.3242	0.1575	0.3109	0.3664	0.6320
100	0.4430	0.7898	0.3679	0.4081	0.0761	0.1028	<b>0.3991</b>	<b>0.6585</b>
200	0.4308	0.7851	0.4012	0.3971	0.0242	0.0091	0.3943	0.2692
500	0.4257	0.8060	<b>0.4212</b>	0.3983	0.0226	0.0132	0.1328	0.2692
1000	0.4177	0.8187	0.3894	0.3983	0.0214	0.0155	0.0872	0.0994
2000	0.2800	0.5485	0.4044	<b>0.4185</b>	0.0233	0.0140	0.0748	0.0788

is sensitive to the modification of network structure causing that CDMG can capture community structure well.

### F. Influence of Markov Time on Community Detection

Table III and Fig. 3 demonstrate the influence of Markov Time  $t$  on the performance of CDMG. We can observe that for YouTube, at first the NMI score and Omega Index fluctuate within a certain range while Markov Time  $t$  increasing, but when  $t$  is above a threshold, the NMI score and Omega Index decay gradually. For LiveJournal, Amazon and DBLP, their NMI score and Omega Index increase gradually as  $t$  rises, and their highest NMI score and Omega Index are much larger than other algorithms' respectively. For DBLP when Markov Time  $t$  is above 100, both NMI score and Omega Index decrease gradually. For LiveJournal and YouTube their Markov Time thresholds are probably 20 and 4 respectively. To sum up, for CDMG there probably is a Markov Time threshold for a network. When Markov Time  $t$  is below the threshold, NMI score and Omega Index of community partition detected by CDMG either keeps relatively steady with a little fluctuation or increases gradually. Oppositely when Markov Time  $t$  is above the threshold, NMI score and Omega Index could decay gradually. According to this empirical observation, our method CDMG can provide a much better result when using the Markov Time  $t$  around the threshold.

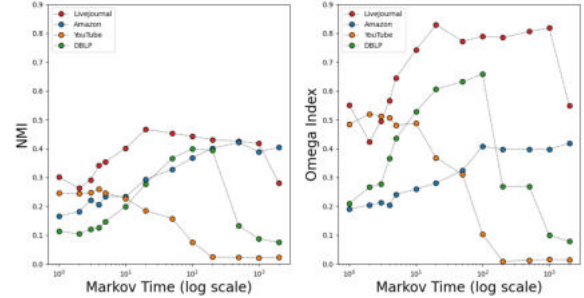


Fig. 3. NMI score and Omega Index of CDMG on the real-world networks including LiveJournal, Amazon, YouTube and DBLP with Markov Time  $t$  increasing.

### G. Runtime Comparison

We compare the runtime of these community detection algorithms on the experimental networks with different scale including LiveJournal, Amazon, YouTube and DBLP. Fig. 4 illustrates that CPM, NSED, SymNMF and MNMF are faster than other algorithms in these real-world networks. The GNNs-based methods such as CDMG and NOCD have shorter runtime in comparison to Node2Vec and LINE, and CDMG is relatively faster than NOCD when tackling networks with more vertices. As to ComE, its great consumption of time is reasonable for it jointly addresses community detection,

community embedding and node embedding. Even though CDMG is not the fastest method, the runtime of CDMG is still acceptable for its performance significantly outperforms other methods.

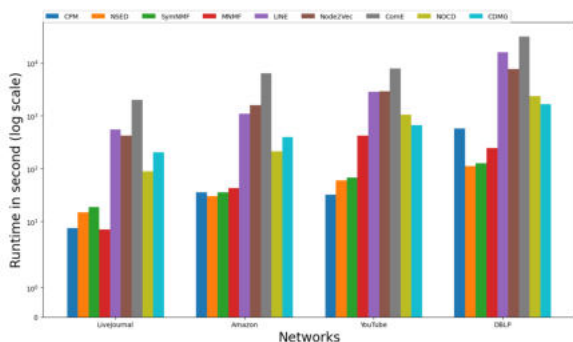


Fig. 4. Runtime of different community detection algorithms on the experimental networks including LiveJournal, Amazon, YouTube and DBLP.

## V. CONCLUSION

In this paper, we propose a graph neural network model CDMG for overlapping community detection from the perspective of optimizing Markov Stability. Specifically, we train a graph neural network to generate the community affiliation weight matrix indicating community structure while maximizing Markov Stability. The experiments confirm that on several real-world networks our method outperforms other baseline methods considering both the accuracy and runtime. And we also explore the influence of Markov Time  $t$  on the performance of CDMG and find out that there is a Markov Time threshold for a network. When Markov Time  $t$  is around the threshold, CDMG can provide a better result with higher accuracy. The results of CDMG also demonstrate how powerful Graph Network Networks are.

## ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grants 62072352, U1708262.

## REFERENCES

- [1] Bacik, Karol A., et al. "Flow-based network analysis of the Caenorhabditis elegans connectome." *PLoS computational biology* 12.8 (2016): e1005055.
- [2] Zhang, Muhan, and Yixin Chen. "Link prediction based on graph neural networks." *Advances in Neural Information Processing Systems* 31 (2018): 5165-5175.
- [3] Atwood, James, and Don Towsley. "Line graph neural networks for link prediction." *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).
- [4] Cai, Lei, et al. "Line graph neural networks for link prediction." *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).
- [5] Wu, Jun, Jingrui He, and Jiejun Xu. "Net: Degree-specific graph neural networks for node and graph classification." *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2019.
- [6] Tsitsulin, Anton, et al. "Graph clustering with graph neural networks." *arXiv preprint arXiv:2006.16904* (2020).
- [7] Shchur, Oleksandr, and Stephan Günnemann. "Overlapping community detection with graph neural networks." *arXiv preprint arXiv:1909.12201* (2019).
- [8] Lambiotte, Renaud, Jean-Charles Delvenne, and Mauricio Barahona. "Random walks, Markov processes and the multiscale modular organization of complex networks." *IEEE Transactions on Network Science and Engineering* 1.2 (2014): 76-90.
- [9] Delvenne, J.-C., Sophia N. Yaliraki, and Mauricio Barahona. "Stability of graph communities across time scales." *Proceedings of the national academy of sciences* 107.29 (2010): 12755-12760.
- [10] Delvenne, Jean-Charles, et al. "The stability of a graph partition: A dynamics-based framework for community detection." *Dynamics On and Of Complex Networks, Volume 2*. Birkhäuser, New York, NY, 2013. 221-242.
- [11] Zachary, Wayne W. "An information flow model for conflict and fission in small groups." *Journal of anthropological research* 33.4 (1977): 452-473.
- [12] Newman, Mark EJ. "Modularity and community structure in networks." *Proceedings of the national academy of sciences* 103.23 (2006): 8577-8582.
- [13] Kuang, Da, Chris Ding, and Haesun Park. "Symmetric nonnegative matrix factorization for graph clustering." *Proceedings of the 2012 SIAM international conference on data mining*. Society for Industrial and Applied Mathematics, 2012.
- [14] Sun, Bing-Jie, et al. "A non-negative symmetric encoder-decoder approach for community detection." *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 2017.
- [15] Li, Ye, et al. "Community detection in attributed graphs: An embedding approach." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32, No. 1. 2018.
- [16] Wang, Fei, et al. "Community discovery using nonnegative matrix factorization." *Data Mining and Knowledge Discovery* 22.3 (2011): 493-521.
- [17] Jia, Yuting, et al. "CommunityGAN: Community detection with generative adversarial nets." *The World Wide Web Conference*. 2019.
- [18] Lobov, Ivan, and Sergey Ivanov. "Unsupervised community detection with modularity-based attention model." *arXiv preprint arXiv:1905.10350* (2019).
- [19] Hu, Fang, et al. "Community detection in complex networks using Node2vec with spectral clustering." *Physica A: Statistical Mechanics and its Applications* 545 (2020): 123633.
- [20] Perozzi, Bryan, Rami Al-Rfou, and Steven Skiena. "Deepwalk: Online learning of social representations." *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2014.
- [21] Grover, Aditya, and Jure Leskovec. "node2vec: Scalable feature learning for networks." *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 2016.
- [22] Tang, Jian, et al. "Line: Large-scale information network embedding." *The World Wide Web Conference*. 2015.
- [23] Wang, Xiao, et al. "Community preserving network embedding." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 31, No. 1. 2017.
- [24] Vaswani, Ashish, et al. "Attention is all you need." *arXiv preprint arXiv:1706.03762* (2017).
- [25] Cavallari, Sandro, et al. "Learning community embedding with community detection and node embedding on graphs." *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 2017.
- [26] Beguerisse-Díaz, Mariano, Borislav Vangelov, and Mauricio Barahona. "Finding role communities in directed networks using role-based similarity, markov stability and the relaxed minimum spanning tree." *2013 IEEE Global Conference on Signal and Information Processing*. IEEE, 2013.
- [27] Kipf, Thomas N., and Max Welling. "Semi-supervised classification with graph convolutional networks." *arXiv preprint arXiv:1609.02907* (2016).
- [28] Aaron F McDaid, Derek Greene, and Neil Hurley. 2011. Normalized mutual information to evaluate overlapping community finding algorithms. *arXiv:1110.2515* (2011).
- [29] Gregory, Steve. "Fuzzy overlapping communities in networks." *Journal of Statistical Mechanics: Theory and Experiment* 2011.02 (2011): P02017.
- [30] Palla, Gergely, et al. "Uncovering the overlapping community structure of complex networks in nature and society." *nature* 435.7043 (2005): 814-818.

# Exploiting Heterogeneous Monitoring Data for Spatiotemporal Algal Bloom Prediction

Taewhi Lee, Miyoung Jang, Jang-Ho Choi, Jongho Won, and Jiyong Kim  
 Smart Data Research Section, AI Research Lab.  
 ETRI (Electronics and Telecommunications Research Institute)  
 Daejeon, Republic of Korea  
 {taewhi, myjang, janghochoi, jhwon, kjy}@etri.re.kr

**Abstract**—Harmful algal blooms need to be mitigated because they can cause significant negative effects to humans and other organisms. If such algal blooms can be predicted in advance by monitoring water quality, they can be suppressed at an early stage by making decisions to take actions. We describe our ongoing work on integrating the heterogeneous water quality monitoring data and on recovering the missing data using tensor completion techniques. We also discuss the challenges in carrying out this study.

**Index Terms**—algal bloom, data integration, tensor completion

## I. INTRODUCTION

Algal blooms are natural phenomena where the population of photosynthetic organisms rapidly increases in aquatic ecosystems [1]. Harmful algal blooms need to be mitigated because they can cause significant negative effects to humans and other organisms. When they occur in water sources, those effects may be exacerbated. If such algal blooms can be predicted in advance by monitoring water quality, they can be suppressed at an early stage by making decisions to take actions, such as dispatching algae harvesting ship,

water surface aerator, or ultrasonic algae controller, opening floodgates, and spraying yellow soil.

The accuracy of algal bloom prediction depends on the quality of the monitoring data. In order to collect water quality data more densely in near real-time, attempts are being made to collect data through various types of device as shown in Figure 1.

- **Fixed sensor data.** Water quality data collected from fixed sensors that are installed on the pontoons at specific points.
- **Moving sensor data.** Water quality data collected from an unmanned surface vehicle (USV) equipped with water quality sensors. The USV collects the water quality data as it travels the target area along the predefined route.
- **Hyperspectral image data.** Water quality data collected from an aerial drone equipped with a hyperspectral sensor camera. The concentration of chlorophyll-a (Chl-a) and phycocyanin (PC), which are indicators for algal biomass, can be extracted from the hyperspectral images.

However, these heterogeneous data cannot be directly used to algal bloom prediction. In order to predict algal blooms

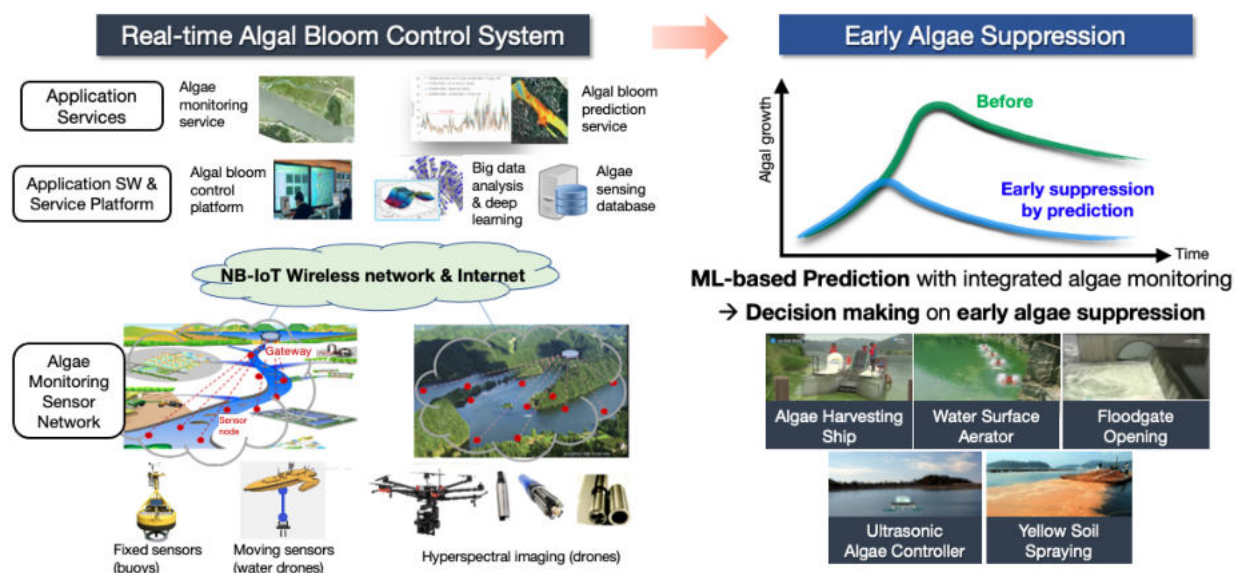


Fig. 1. algae monitoring data collection for algal bloom prediction

based on machine learning, it is necessary to preprocess these heterogeneous data to improve the quality of the data as follows.

- **Data integration.** Those monitoring data can be fed to machine learning by integrating data measured by different devices at different times and different locations.
- **Missing data recovery.** Data omissions occur frequently due to various reasons, such as weather conditions, hardware failure, and budget limitations.

In this paper, we describe our ongoing work on integrating the heterogeneous water quality monitoring data and on recovering the missing data using tensor completion techniques. Then, we conclude by discussing the challenges in carrying out this study.

## II. HETEROGENEOUS MONITORING DATA INTEGRATION

### A. Data Collection

Water quality monitoring data are being collected for our study area through the various devices mentioned in Section I. Our study area is the So-ok-cheon stream near to the Chu-so-ri region, a branch of the Geum river which is one of the six main rivers in South Korea, as shown in Figure 2. It is the area where green algae often occur due to topographic reasons. The study area is split into about 2000 grid cells, which are unit areas for algal bloom prediction.

The format of monitoring data is different depending on the device being measured. Fixed sensor data collected from the pontoon and moving sensor data collected from the USV include various kinds of features, including water temperature, pH, dissolved oxygen, electrical conductivity, total organic carbon, total nitrogen, total phosphorous, and chlorophyll-a. Such features cannot be extracted from hyperspectral images taken by aerial drones, except chlorophyll-a and phycocyanin.

The cycle of data collection is also different. While fixed sensor data are reliably collected on a hourly basis, moving

sensor data and hyperspectral image data are collected on a daily basis. Also, there are many omissions in moving sensor data and hyperspectral image data because USVs and aerial drones cannot operate in bad weather conditions, or they may also have to operate within budget.

### B. Data Integration

We integrate heterogeneous monitoring data by mapping into grid cells. The data integration process can be summarized as follows.

- 1) **Grid cell construction.** For each cell, polygon objects are created using the coordinates of its boundary points.
- 2) **Data-to-cell mapping.** Each monitoring data record is mapped to the corresponding cell polygon object, which contains the location coordinate of the data record.
  - **Fixed sensor data.** A pontoon is installed in a specific location, so fixed sensor data can be directly mapped into a specific cell.
  - **Moving sensor data.** Moving sensor data are collected every certain distance along the moving path of USVs. Multiple values may be mapped to in the same cell for the same datetime.
  - **Hyperspectral image data mapping.** Hyperspectral image data are spatially continuous because the data values are extracted from the image pixels. Therefore, it is required to sample the data by a certain distance.
- 3) **Representative value selection.** Since two or more values may be mapped to one cell for the same datetime, representative values have to be selected. It may be used to apply some statistics function like max(), avg(), and median() or to pick a certain record by policy.
- 4) **Missing data recovery.** Data omissions occur frequently due to various reasons, so it is necessary to recover missing data for more accurate prediction. It will be described in Section III.

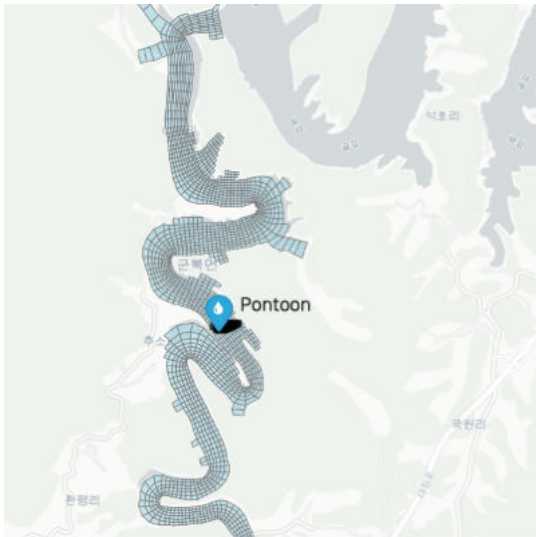


Fig. 2. Study area

## III. MISSING DATA RECOVERY

There are a large number of missing data in the collected data. In order to train machine learning models for algal bloom prediction, the missing data have to be handled in some ways. Naive methods, such as deleting data records with missing values or filling them with the average of surrounding values in the time or space dimension, would give inaccurate prediction results. Deep learning-based imputation techniques like DataWig [2] have been developed to impute missing values in tabular data. Many studies have been conducted to fill in the missing data [3], [4], but the larger the fraction of missing data, the more difficult it is to recover the data.

We have been trying to apply tensor data completion techniques using auxiliary information, Auxiliary Information Regularized CP model (AirCP) [5], which can be applied even in cases where the fraction of missing data is large. The authors applied the AirCP method to recover the spatiotemporal dynamics of hashtags. We apply this method to recover our

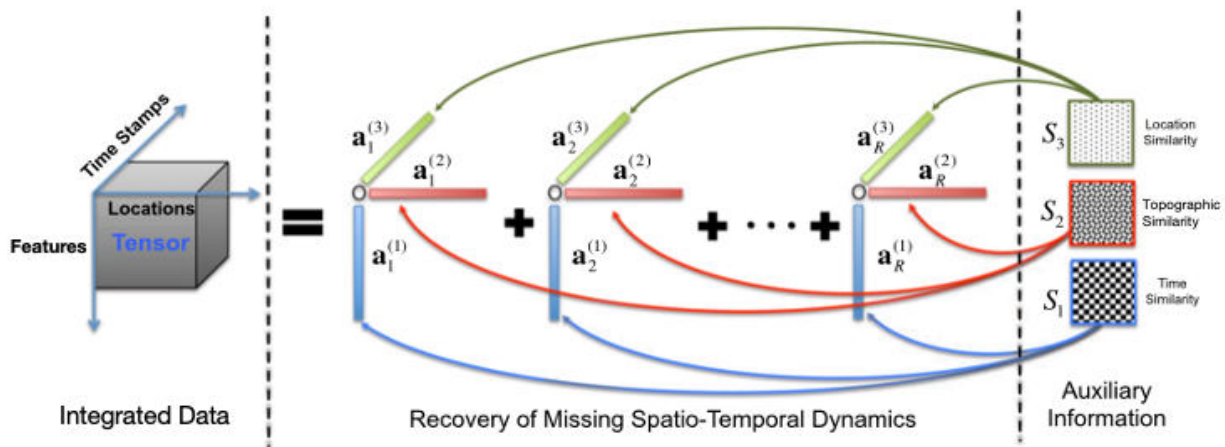


Fig. 3. Tensor data completion using Auxiliary Information Regularized CP model (adapted from [5])

integrated data by injecting topographic similarity, as well as location similarity and time similarity, as shown in Figure 3.

- **Location similarity.** A similarity matrix encoding spatial relationships. The location similarity matrix is derived under the assumption that the closer the grid cell, the more similar the concentrations of algae.
- **Topographic similarity.** A similarity matrix encoding topographic characteristics. For example, this matrix can indicate whether the grid cell is the edge or the middle of river by the distance to the nearest land. The topographic matrix is derived under the assumption that the more similar the topographical characteristics, the more similar the concentrations of algae.
- **Time similarity.** A similarity matrix encoding temporal relationships. The time similarity matrix is derived under the assumption that the closer the time of data collection, the more similar the concentrations of algae.

We continue to collect water quality data on our study area with a novel direct-readable water quality complex sensor, which is newly developed in our project. Unfortunately, enough data has not been accumulated yet. We are currently testing various topographic similarity matrices to enhance the accuracy of algal bloom prediction and are analyzing collected data for each cell.

#### IV. DISCUSSION AND CHALLENGES

As we are challenging to predict the concentration of algae for areas where water quality data has not been collected, we are facing a lot of difficulties. We describe the challenging issues that must be addressed.

To build machine learning models for algal bloom prediction, we need enough data to construct training and test datasets. It is better for the prediction to collect water quality data daily or more frequently, but this is impossible when the weather condition is bad or the hardware fails. This puts us in a situation where there is no exact real data to compare with predicted values.

A limited budget is another factor that lowers the frequency of data collection because of the costs involved in operating

USVs and aerial drones. The features that can be extracted are also different depends on the devices. It can be another research topic to determine the frequency or cycle of data collection under these constraints.

#### V. CONCLUSION

We presented a method to integrate the heterogeneous water quality monitoring data and to recover the missing data by applying tensor data completion techniques using auxiliary information. We used three types of auxiliary information to recover the missing data, i.e., location similarity, topographic similarity, and time similarity. We plan to compare the performance of algal bloom prediction using various auxiliary similarity matrices, or various missing data recovery methods. Also, we will address the challenging issues discussed in this paper.

#### ACKNOWLEDGMENT

This work was supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. 2018-0-00219, Space-time complex artificial intelligence blue-green algae prediction technology based on direct-readable water quality complex sensor and hyperspectral image).

#### REFERENCES

- [1] T. Lee, J.-H. Choi, M. Jang, J. Won, and J. Kim, "Enhancing prediction of chlorophyll-a concentration with feature extraction using higher-order partial least squares," in *Proceedings of 2020 International Conference on Information and Communication Technology Convergence*, 2020, pp. 1666–1668.
- [2] F. Biessmann, T. Rukat, P. Schmidt, P. Naidu, S. Schelter, A. Taptunov, D. Lange, and D. Salinas, "DataWig: Missing value imputation for tables," *Journal of Machine Learning Research*, vol. 20, no. 175, pp. 1–6, 2019.
- [3] T. Emmanuel, T. Maupong, D. Mpoeleng, T. Semong, B. Mphago, and O. Tabona, "A survey on missing data in machine learning," *Journal of Big Data*, vol. 8, pp. 140:1–37, 2021.
- [4] S. Jäger, A. Allhorn, and F. Biessmann, "A benchmark for data imputation methods," *Frontiers in Big Data*, vol. 4, p. 48, 2021.
- [5] H. Ge, J. Caverlee, N. Zhang, and A. Squicciarini, "Uncovering the spatio-temporal dynamics of memes in the presence of incomplete information," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, ser. CIKM '16, 2016, pp. 1493–1502.

# Three-dimensional Data Outlier Detected by Angle Analysis

Zhongyang Shen

China Mobile

Beijing, China

ORCID: 0000-0002-4826-0966

**Abstract**— We present a method to distinguish outliers from spherical cluster distributed three-dimensional data. The angle measurement method 3DOD transforms three-dimensional data to two-dimensional data, and then outliers can be detected by conventional two-dimensional data outlier algorithm.

**Keywords**— outlier, three-dimensional, angle transformation, dimension reduction

## I. INTRODUCTION

In some data application cases, we will meet the requirements related to the outliers of three-dimensional data. To be distinguished from most data, outliers show their special characters in some cases. Currently, there are a variety of methods for abnormal recognition of two-dimensional data, such as LOF, Isolation Forest, OGAD[1], ABOD[3], DBSCAN[4] and other algorithms. For three-dimensional or high-dimensional data, as the number of dimensions increases, the amount of calculations will increase exponentially, which brings challenges to find the correlation of high-dimensional features in outlier detection. There are current algorithms such as ABOD and other algorithms supporting three-dimensional data outlier detection, but it has disadvantages with calculation complexity, training data, and proportion setting. Other algorithms such as LOF with PCA algorithm are processed through dimension reduction, bringing the loss of multidimensional data information to get some obviously abnormal results.

In this paper, an unsupervised algorithm method for outlier detect of three-dimensional data (3DOD) is proposed to solve the problems of computational complexity caused by the increase in dimension. Through the angle transformation method, three-dimensional data is transformed into two-dimensional data, with the characteristics of three-dimensional data are remained, and then the conventional two-dimensional outlier analysis method is used to detect outliers. Experiments show that this method can effectively detect the outliers of spherical cluster distributed three-dimensional data.

This paper verifies that the three-dimensional data can be dimensional reduced to two-dimensional data by angle conversion in outlier recognition, with the data characteristics of the three-dimensional data remained. Also, this paper verifies that the two-dimensional outlier algorithm such as OGAD, LOF still has the ability to identify outliers for three-

dimensional data after reducing the dimension to two-dimensional data through angle conversion.

## II. ALGORITHM

### A. Principle

Outliers are some data that deviate from the normal data area. For the three-dimensional data of spherical cluster distribution, we assume there are a spherical cluster virtual boundary between normal points and outliers. To distinguish the outlier from normal points, we need to identify the virtual boundary of the spherical cluster. This method proposes an angle measurement method to determine the corresponding angle position of each three-dimensional data point by means of angle scanning from a spherical peripheral observation point, then convert the angle data into two-dimensional data to filter out the outliers.

### B. Proof

First, we will try to prove that angle measurement method is able to separate the normal data and abnormal data from the spherical cluster-like three-dimensional data when we intersect tangents from observation points to the spherical cluster. In the case, it will form a cone shape with possible normal values inside the cone and outliers outside the cone (Figure 1). After angle scanning from several observation points in the periphery, the possible normal values are in the overlapping areas of all scanning areas, and the outlier values are in the other areas, which separated from normal points.

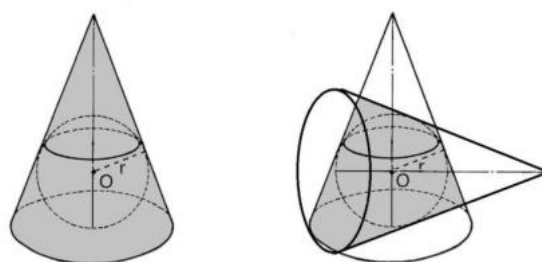


Figure 1. Example of angle scanning from observation points to detect abnormal values: shaded part is possible area of normal points

As an example, we take 14 external observation points with uniform distribution to compare the overlapping volume with the volume of the cluster. After angle scanning from all observation points, the difference between the two volumes is

compared to prove the effect of algorithm 3DOD on the identification and separation of outliers.

To simplify calculation, we assume that the observation point is a point at infinity, thus from the observation point, the scanning volume that intersects with the spherical cluster is to be close to a cylindrical volume. When we select fourteen evenly distributed observation points, these scanning operations will be converted into seven cylindrical volumes, and then the remaining volume will be formed by these seven cylinders intersected and overlapped. Now we can compare the remaining volume with standard spherical cluster size to analyze the difference.

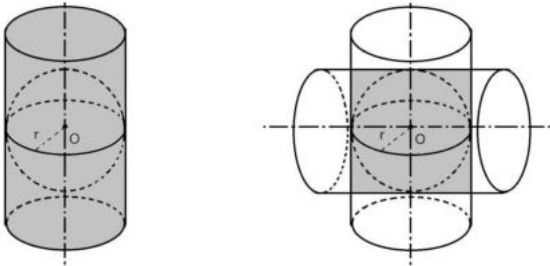


Figure 2. Example of angle scanning from observation point at an infinite distance: shadow portion is possible area of normal points

To calculate and compare the remaining volume, we use the following calculating process.

- Step 1: 14 uniformly distributed observation directions are selected, which the connecting lines from observation direction to the center of sphere were separated by equal angle.
- Step 2: Choose an observation direction.
- Step 3: The cylinder and the sphere are intersected on the surface of the sphere, which radius of the cylinder is equal to the radius of the sphere, and the height is the diameter of the sphere. Then the volume of cylinder is calculated.
- Step 4: The remaining volume will be the overlapped part with the above calculated cylinder volume and the previously calculated volume.
- Step 5: Go to Step 2 until calculation of all observation points are finished.
- Step 6: Calculate the remaining volume.

As calculation result, the ratio between remaining volume and spherical volume is 1.021:1, that is, the remaining volume is 2.1% larger than the spherical volume. Since the data in practical applications is not evenly distributed, the difference value can be reduced by increasing observation points or adjusting virtual boundary of the cluster, the fluctuation of 2.1% can be considered as an acceptable range of differences in the approximate calculation. When observation points are increased from 14 to 26 points or more, the difference value will be tended to be smaller.

### C. Thought of Algorithm

Based on above result, we can see that outliers can be detected by calculating each three-dimensional data from the peripheral observation point.

In this paper, we present a new angle conversion calculation method by converting three-dimensional data into two-dimensional. Two-dimensional data is formed by a certain three-dimensional observation point, the method is, from the observation point, we can form an angle in the direction of XY plane and another angle in the direction of Y axis, and then these two angle values are composed as two-dimensional data. When all two-dimensional data are formed, we can use conventional two-dimensional outlier algorithm to detect the outliers.

Algorithm steps are showed as following:

- Step 1: Select 14 or more external observation points that are evenly distributed relative to the three-dimensional data cluster, which the axis between each observation point and the center of the spherical cluster is spaced equal angle apart from each other.
- Step 2: Select an observation point and calculate the two-dimensional data angles of each measured point. We mark the line from observation point to measured point as LINE0, and mark LINE0 projection line on x, y plane as LINE1.

Angle A: Angle between X axis and LINE1.

Angle B: Angle between LINE0 and LINE1.

The Angles is rounded to simplify algorithm calculation.

Two-dimensional data are formed by angle A and angle B, which is used for outlier identification.

- Step 3: Use conventional two-dimensional outlier identification methods such as OGAD or LOF to calculate outliers of the two-dimensional data formed by angle A and angle B.
- Step 4: Go to Step 1, move to next point until all observation points are calculated.
- Step 5: Collect all the calculated values to sort the result of outliers.

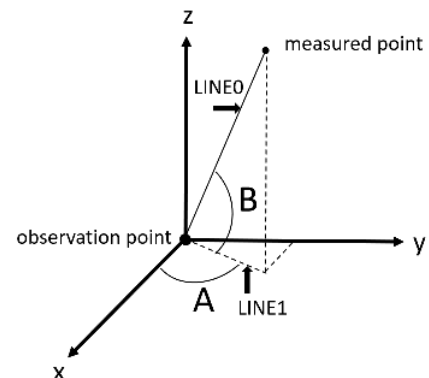


Figure 3. Example of converting angles into two-dimensional data



### III. PSEUDO-CODE

The following pseudo code is based on the thought of algorithm described above.

#### Algorithm Program

```

1: //Get the barycentre position and radius length;
2:  $m \leftarrow \text{count}(\text{Measured points})$ ;
3:  $\text{Barycentre}(x_0, y_0, z_0) = \frac{1}{m} \sum_{i=0}^m \text{point}(x, y, z)$ ;
4: for each point  $(x, y, z) \in \text{Measured points}$  do
5:    $\text{radius} = \max(\text{distance}(\text{Barycentre}(x_0, y_0, z_0), \text{point}(x, y, z)))$ ;
6: end for;
7: //Get observation points which evenly distributed around measured points;
8:  $p \leftarrow \text{quantity of observation points}$ ;
9: for  $i=0$  to  $p$  do
10:   $\text{obser}(x_i, y_i, z_i) = \text{position}$ (
11:     $\text{base: Barycentre}$ ,
12:     $\text{length: } (1 + \text{ratio}) * \text{radius}$ ,
13:     $\text{direction: evenly distributed around measured points}$ 
14:  );
15: //Get Density for every angle from observation point;
16: for each point  $(x, y, z) \in \text{Measured points}$  do
17:   $\text{angleA} = \text{integer}$  (
18:     $\text{vertex: obser}(x_i, y_i, z_i)$ ,
19:     $\text{sideline: } x\text{-axis}$ ,
20:     $\text{obser}(x_i, y_i, z_i)$  to point  $(x, y, z)$  projection line on  $x, y$  plane
21:  );
22:   $\text{angleB} = \text{integer}$  (
23:     $\text{vertex: obser}(x_i, y_i, z_i)$ ,
24:     $\text{sideline: obser}(x_i, y_i, z_i)$  to point  $(x, y, z)$ ,

```

```

25:     $\text{obser}(x_i, y_i, z_i)$  to point  $(x, y, z)$  projective line on  $x, y$  plane
26:  );
27:   $\text{pointAngle}(a_i, \beta_i) \leftarrow (\text{angleA}, \text{angleB})$ ;
28: end for;
29:  $\text{LOF}(\text{pointAngle}(a_i, \beta_i))$  or  $\text{OGAD}(\text{pointAngle}(a_i, \beta_i)) \rightarrow$ 
30:   $\text{RankingAnomaly2D}(\text{pointAngle}(a_i, \beta_i))$ ;
31:   $\text{RankingAnomaly2D}(\text{pointAngle}(a_i, \beta_i)) \rightarrow$ 
32:   $\text{RankingAnomaly3D}(\text{point}(x, y, z))$ ;
33: end for;

```

### IV. EXPERIMENTAL DATA AND ANALYSIS

The following experiments show experimental results and comparison results. As different algorithms, ABOD and LOF with PCA dimensional reduction are used in the experiments as comparisons.

First, we design a set of simulation three-dimensional data, define specifically normal values and abnormal values, and then use algorithm 3DOD, LOF with PCA dimensional reduction and ABOD to identify respectively. In the experiments two-dimensional outlier algorithm OGAD is used in 3DOD as comparison to LOF with PCA dimensional reduction.

Table I and Table II show the experiment results with radius of standard sample set as 200 distance unit, and each experiment tests 200 times.

Table I. Experiments designed: number of standard sample set to 600, number of outliers set to 10

Outlier radius range (distance unit)	number of observation points(3DOD)	Times that match 10 outliers exactly			times match 9 outliers			times match 8 outliers		
		3DOD	LOF with PCA	ABOD	3DOD	LOF with PCA	ABOD	3DOD	LOF with PCA	ABOD
200-400	14	131	55	153	191	137	195	200	182	200
	26	139			195			200		
250-400	14	190	71	196	199	145	200	200	190	200
	26	193			200			200		
300-400	14	197	88	200	200	170	200	200	196	200
	26	198			200			200		

Table II. Experiments designed: number of standard sample set to 605, number of outliers set to 5

Outlier radius range (distance unit)	number of observation points(3DOD)	Times that match 5 outliers exactly			times match 4 outliers			times match 3 outliers		
		3DOD	LOF with PCA	ABOD	3DOD	LOF with PCA	ABOD	3DOD	LOF with PCA	ABOD
200-400	14	162	109	181	200	181	198	200	195	200
	26	172			200			200		
250-400	14	198	125	200	200	189	200	200	199	200
	26	198			200			200		
300-400	14	200	146	199	200	192	200	200	200	200
	26	200			200			200		

To verify the results, we analyze and compare the difference results of the above experiment data. In these cases, we select the case which outlier distribution is 200-400 distance units, number of normal values is 605, number of

outliers designed is 5, and observation points of algorithm 3DOD is 14 points while the results are the same at 26 points.

The following Figure 4(a) and Figure 4(b) show difference between algorithm 3DOD and LOF with PCA dimensional

reduction, and Figure 4(c) and Figure 4(d) show difference between algorithm 3DOD and ABOD.

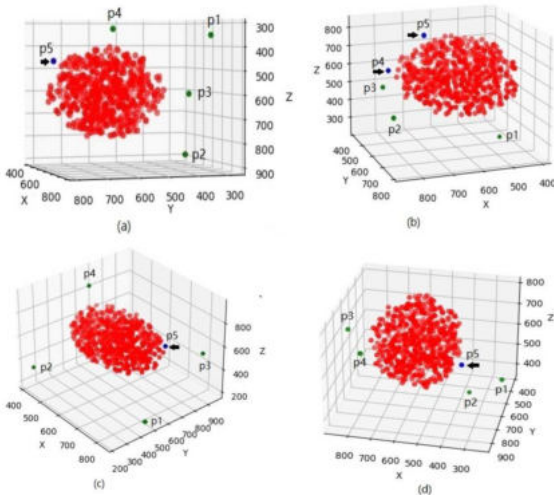


Figure 4. Outlier Cases

- Case 1: In Figure 4(a), algorithm 3DOD accurately identifies the defined 5 outliers (P1-P5), among which the blue point (P5) is ranked 5th in the outlier ranking, while in LOF with PCA dimensional reduction, 4 outliers are identified, and the blue point (P5) is ranked 216th in the outlier ranking, shown as in the normal value range. Result shows in this case algorithm 3DOD is more reasonable than LOF with PCA dimensional reduction.
- Case 2: In Figure 4(b), algorithm 3DOD accurately identified 5 defined outliers (P1-P5), among which 2 blue points (P4, P5) ranked 4th and 5th in outliers ranking, while 3 outliers are identified in LOF with PCA dimensional reduction, and 2 blue points (P4, P5) are ranked 14th and 507th respectively, which both showed in normal range. Result shows in this case algorithm 3DOD is more reasonable than LOF with PCA dimensional reduction.
- Case 3: In Figure 4(c), 4 outliers (P1-P4) are identified both by ABOD and algorithm 3DOD. The blue point (P5) is ranked 176th in ABOD and ranked 26th in algorithm 3DOD, which are in normal value range. In this case algorithm 3DOD and ABOD are similar in outliers detect.
- Case 4: In Figure 4(d), algorithm ABOD identifies 4 outliers (P1-P4), and algorithm 3DOD identifies 5 outliers (P1-P5). The blue point (P5) is ranked 11th in ABOD, while is within normal value range, and blue point (P5) is ranked 5th in algorithm 3GOD. Results show that in this case algorithm 3DOD is better than ABOD in outliers detected.

Through data analysis and comparison, we can find the following analysis results:

- As a conventional dimension reduction outlier algorithm, LOF with PCA dimensional reduction brings

loss of effective information also will bring to the deterioration of outlier data. Results show LOF with PCA dimensional reduction has less accuracy in outlier detection than 3DOD and ABOD.

- 3DOD and ABOD have similar result in accuracy.
- For 3DOD, accuracy of outlier detection will be improved with increasing quantity of observation points.

Experimental data indicates that algorithm 3DOD is proposed as a dimension reduction algorithm for outlier recognition of three-dimensional data cluster. It can be seen from the experimental data that algorithm 3DOD is effective in achieving outlier recognition of clustered three-dimensional data, and it is significantly better than algorithm LOF with PCA dimensional reduction in terms of stability and accuracy. Comparing with ABOD, which training data and abnormality ratio needed to be set in advance, unsupervised learning algorithm 3DOD has advantages in computational complexity.

When analyzing the algorithm implementation process, we can see that algorithm 3DOD is based on the mechanism of data accumulation, that is, when data increasing, the new data is added into previous data set instead of full-scale calculations, which bring more effective.

Based on angle analysis, algorithm 3DOD can bring about the study of three-dimensional data and three-dimensional above data cluster identification by calculating the strongest density direction.

## V. CONCLUSION

This paper presents a new algorithm 3DOD for three-dimensional data, the core idea and direction is three-dimensional data reduction by angle conversion from several external observation points. Experimental tests verify that the algorithm is reliable and accurate. It can be seen from experiment data that the method presented in this paper can play an effective role in the identification of outliers from three-dimensional data.

## REFERENCES

- [1] Zhongyang Shen, "Outlier Geometric Angle Detection Algorithm," 2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC), 2019, pp. 316-321, doi: 10.1109/ICAIIIC.2019.8669090.
- [2] Zhongyang Shen, "Cluster Quantity Distinguished by Geometric Angle Measurement," 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC), 2020, pp. 514-519, doi: 10.1109/ICAIIIC48513.2020.9065253.
- [3] Hans-Peter Kriegel, Matthias Schubert, Arthur Zimek, "Angle-Based Outlier Detection in High-dimensional Data," The 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2008.
- [4] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD), AAAI Press., 1996.
- [5] David Arthur, Sergei Vassilvitskii, "k-means++: the advantages of careful seeding," Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, 2007, pp 1027 - 1035.

- [6] Xie Huajuan, "Unsupervised Learning Methods and Applications," Publishing House of Electronics Industry, China, 2016.
- [7] Jiasi Shen and Martin Rinard, "Robust programs with filtered iterators," Proceedings of the 10th ACM SIGPLAN International Conference on Software Language Engineering (SLE), ACM, 2017, Pages 244 - 255, DOI:<https://doi.org/10.1145/3136014.3136030>
- [8] Masashi Sugiyama, "An Illustrated Guide to Machine Learning," Kodansha Ltd., Japan, 2013.
- [9] Toby Segaran, "Programming Collective Intelligence," O' Reilly Media, Inc., 2007.
- [10] Pankaj K. Agarwal and Nabil H. Mustafa, "k-means projective clustering," In PODS'04: Proceedings of the twenty-third ACM SIGMODSIGACT-SIGART symposium on Principles of database systems, pages 155 - 165, ACM, 2004.
- [11] Koki Saitoh, "Deep Learning from Scratch," O' Reilly Japan, Inc., 2016
- [12] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu, "A local search approximation algorithm for k-means clustering," Proceedings of the eighteenth annual symposium on Computational geometry, ACM, 2002, DOI:<https://doi.org/10.1145/513400.513402>.
- [13] Bardia Yousefi and Chu Kiong Loo, "Comparative study on interaction of form and motion processing streams by applying two different classifiers in mechanism for recognition of biological movement," The Scientific World Journal, 2014.
- [14] Andreas C. Muller, Sarah Guido, "Introduction to machine Learning with Python," O' Reilly Media, Inc., 2016.
- [15] Zhou Zhihua, "Machine Learning," Tsinghua University Press, 2016.

# Identification and Analysis of COVID-19-related Misinformation Tweets via Kullback-Leibler Divergence for Informativeness and Phraseness and Biterm Topic Modeling

Thomas Daniel S. Clamor<sup>1</sup>, Geoffrey A. Solano<sup>1</sup>, Nathaniel Oco<sup>2</sup>,  
Jasper Kyle Catapang<sup>3</sup>, Jerome Cleofas<sup>2</sup> and Iris Thiele Isip-Tan<sup>1</sup>

<sup>1</sup>University of the Philippines Manila

Manila City, Philippines

<sup>2</sup>De La Salle University

Manila City, Philippines

<sup>3</sup>University of Birmingham

Birmingham, United Kingdom

{tsclamor, gasolano}@up.edu.ph, nathanoco@yahoo.com, jxc1354@student.bham.ac.uk,  
jerome.cleofas@dlsu.edu.ph, icisiptan@up.edu.ph

**Abstract**—The interaction of Filipinos transitioned to a virtual setting making social media, like Twitter, their source of information since the pandemic started. The infodemic it caused has opened up avenues to understand the characteristics of misinformation tweets regarding COVID-19. In this paper, we present the classification and analysis of misinformation tweets related to COVID-19 towards identifying themes. We used pointwise KL divergence in scoring “informativeness” and “phraseness” to extract misinformation tweets and BTM for topic modeling. With a testbed of 7,711 tweets, the classifier model identified 3,533 misinformation tweets with an accuracy of 74.25%. The results of the topic modeling were analyzed and clustered to expose possible narratives in the data set. The three narratives showed that most Filipinos use Twitter to share jokes, spread information and awareness about the virus, express opinions about the government’s response, and share tips to prevent the disease. A wider date coverage could be included in future works.

**Index Terms**—COVID-19, misinformation, tweets, KLIP, BTM

## I. INTRODUCTION

The COVID-19 pandemic started after the first casualty outside China, in the Philippines, and inflicted a lasting impact on the lives of many people. COVID-19 or Coronavirus disease is caused by the newfound coronavirus, SARS-CoV-2 virus, that is transmissible through saliva droplets or nose discharge. With the pandemic, an “infodemic” or information outbreak has started and it contains both true and false information [1].

To lower the risk of acquiring the disease, people used social media to stay connected with the world. Social media, like Twitter, gives people access to news and information about the disease. It is also through it that people spread their knowledge, views, and opinion about the situation.

Twitter has become widely used by the public and organizations for gathering and spreading information during

emergency situations. Its design and features particularly the short burst style of posting, publicly available posts, attaching hyperlinks, easy way of re-posting or sharing, and multimedia capacities make Twitter an accommodating platform for people in sharing their personal experiences, express opinions and concerns, ask questions, and seek and share information. During emergencies and crisis situations, the platform acts as a broadcasting medium and a venue for sourcing news because of its capability for rapid information dissemination that reaches a vast audience; and also a place for people who seek help and who wants to give help during the disaster. Communicating through the platform is driven by the need to contribute to the situation and to connect with others and work together in helping the victims of the disaster [2].

In the Philippines, the presence of false information is not new. Disinformation campaigns are very common in the Philippine politics, and are systematic and strategic [3]. The presence of misinformation on social media platforms such as Twitter may have an impact on the spread of COVID-19. Understanding and detecting misinformation is important to prevent them from further spreading and creating more damage.

Extracting the common themes or topics can show the characteristics of a data and make sense out of it. An investigation on selected Telegram and WhatsApp groups in Iran were investigated to find the themes of misinformation related to COVID-19 [4]. Topic modeling on Twitter data during calamities, disasters, and other events can show how people behave and how they use social media during these times [5][6].

What are the themes of COVID-19-related misinformation? This is the main question that this study needs to address. But before determining the common topics of misinformation

tweets about COVID-19, we first need to identify them. Misinformation tweets from the data set were identified using the key terms or distinctive terms that were discovered using pointwise Kullback-Leibler divergence. Themes were extracted from COVID-19-related misinformation tweets using topic modeling and manual qualitative analysis.

## II. RELATED WORK

### A. Misinformation

Misinformation has been claimed to have a contribution to the spread of COVID-19. As misinformation regarding COVID-19 spreads rapidly on social media, several studies have been conducted to investigate its magnitude on Twitter. Understanding and detecting misinformation is important to prevent them from further spreading and creating more damage.

Basic characteristics of texts of fake and real news articles can be compared against each other. Kapusta et al. [7] did this in their study to show if there are statistically significant difference between the two and can be vital in choosing the suitable characteristics to use for the later classifier models of fake news. Their study showed that articles containing fake information have a more negative sentiment. The results also show that fake news articles tend to be more complicated and more descriptive than the real ones to mislead the reader and convince them that what they are reading is accurate [7].

Analysing the differences and similarities between misinformation regarding COVID-19 and general COVID-19 content can be crucial in understanding what kind of misinformation spreads on social media. In a study by Ranera et al. [8], they used Doc2Vec to retrieve judicial decisions of Philippine Supreme Court cases semantically similar. They used cosine similarity to quantify the similarity between two document vectors. The results prove that finding similar case decisions can be possible through Doc2Vec [8].

What are the key ideas in COVID-19-related misinformation tweets that are not in the general or not misinformation COVID-19-related tweets? This is one of the questions in the exploratory study of Shahi et al. [9] that they answered. Using pointwise Kullback-Leibler divergence for scoring both informativeness and phraseness (KLIP), which is combined into a single score to rank the phrases, they investigated the distinctive terms in their misinformation data [9].

In the approach presented by Tomokiyo and Hurst [10] in extracting distinctive terms or keyphrases, KL divergence or relative entropy was used to measure the difference between two language models. A *keyphrase* has two features namely *informativeness* and *phraseness*. Phraseness is a concept in which it determines how a cohesion of consecutive words be called a phrase. They defined informativeness as the ability of a phrase to represent the key ideas in the data, and the “new information” that we can get from a specific set of documents with respect to a background or general data set. They created language models for a *foreground corpus*, which is the target document from which phrases are to be extracted, and for a *background corpus*, to which the target document is compared.

The unigram model for the foreground corpus was denoted as  $LM_{fg}^1$  and higher order or N-gram model, where  $N$  is greater than 1, as  $LM_{fg}^N$ . The unigram model for the background corpus was denoted as  $LM_{bg}^1$  and N-gram model as  $LM_{bg}^N$  [10].

The amount of loss between  $LM_{fg}^1$  and  $LM_{fg}^N$  is related to phraseness and the amount of loss between  $LM_{fg}^N$  and  $LM_{bg}^N$  denotes informativeness. Fig. 1 illustrates this relationship.

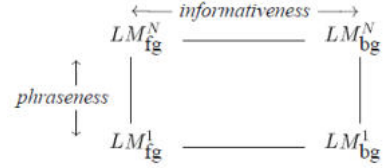


Fig. 1. Phraseness and informativeness as loss between language models

To compute the amount of loss between the language models, pointwise KL divergence  $\delta_w(p||q)$ , where  $w$  denotes phrase, was used [10].

$$\delta_w(p||q) = p(w) \log \frac{p(w)}{q(w)} \quad (1)$$

The phraseness of  $w$  is calculated by getting the amount of loss from assuming the independence of each word by comparing the N-gram from the unigram [10].

$$\delta_w(LM_{fg}^N || LM_{fg}^1) \quad (2)$$

The informativeness of  $w$  is calculated by getting the amount of loss from assuming that the phrase is extracted from the background instead of the foreground [10].

$$\delta_w(LM_{fg}^N || LM_{bg}^N) \quad (3)$$

### B. Topic modeling

Themes show the characteristics of the data, and theme extraction is the way to make sense out of it. Using discourse analysis, themes like “disease statistics”, “treatments”, “vaccines and medicines”, “prevention and protection methods”, “dietary recommendations” and “disease transmission” were discovered from COVID-19-related data from specific Telegram and Whats App groups in Iran [4].

One of the common topic modeling techniques is the latent dirichlet allocation (LDA). It is very effective in capturing topics in a document-level, but may not be very effective with short texts like tweets [5]. To capture the topics from typhoon-related tweets, Ligutom et al. [5] used biterm topic modeling (BTM) and used open coding to assess the topic models to show how Filipino people act during typhoons. The results show that certain behaviors like determination, unity and resiliency, and antagonism can be noted from the topics. The results also show that Filipinos express their opinions about the typhoon. They also stated that some topic models can be misleading as some tweets containing “typhoon” or

“bagyo” are not connected to typhoons but are included in the topic model [5].

Gorro et al. [11] used BTM and word2vec to answer the research question about the common topics on participants’ suggestions on disaster risk reduction (DRR) in their locality. The study showed that BTM and Word2vec can be used for qualitative analysis of data, like DRR, aside from using traditional manual approaches [11].

### III. METHODOLOGY

#### A. Data Collection

A Python library in getting historical tweets called GetOldTweets3 was used to collect tweets from January 1, 2020 to March 22, 2020. The keywords “covid”, “ncov”, and “coronavirus” were used to filter out COVID-19-related tweets. A total of 12,631 publicly available Twitter data from Metro Manila and nearby areas were gathered. From the total data gathered, 4,695 tweets prepared by the domain experts will be used for training, 508 tweets were labeled as “misinformation” and 4,187 tweets were labeled as “not misinformation”.

#### B. Data Preprocessing

Various preprocessing techniques are done in Natural language processing to prepare text data before proceeding to the actual experiment. An important step in the study is the cleaning of data wherein irrelevant and noisy data are removed to prevent misleading results. In this study, the following were performed:

- 1) Removal of URLs
- 2) Removal of words preceded by “#”
- 3) Removal of words preceded by “@”
- 4) Removal of punctuations
- 5) Tokenization
- 6) Lowercasing
- 7) Removal of English and Filipino stop words
- 8) Removal of words containing digits only
- 9) Removal of words with less than 2 characters
- 10) Removal of tweets with less than 2 words

A total of 4,634 tweets for the training data and 7,711 tweets for testing data were left after cleaning.

#### C. Analysis and Classification

1) *KLIP*: KL divergence or relative entropy is used to calculate the difference of two probability distributions. In extracting keyphrases, Tomokiyo and Hurst used pointwise KL divergence to score informativeness and phraseness. To do this, a foreground corpus and a background corpus is needed [10].

KLIP was used in this study to get the keyphrases from the data consisting of tweets tagged as misinformation and use these keyphrases to identify more tweets that contains misinformation. The tweets in the training data set was manually classified as “misinformation” or “not misinformation” by the domain experts. The tweets that are tagged as ‘misinformation’ with 506 tweets and as “not misinformation” with 4,128 tweets was set as the foreground corpus and the background corpus, respectively.

2) *Biterm Topic Modeling*: BTM is used to extract topics from short texts, like tweets, based on the collection of biterms from a whole corpus to address the problem with sparsity of data at document level [12]. This technique of topic modeling discovers the topic distribution over data set or corpus level instead of tweet or document level by using co-occurrence patterns in the whole corpus.

3) *Open Coding*: Open coding is a manual qualitative analysis used in labeling the various topic models generated by the BTM. Different topic models contain keywords which, as a whole, represents a topic. The tweets containing at least one of the keywords in a topic model are analyzed manually to formulate a label for that topic model.

### IV. EXPERIMENTAL RESULTS

To classify which among the tweets contain misinformation, we used the keyphrases obtained from using KLIP. A unigram and bigram model was created for the “misinformation” corpus, foreground, with 6,687 words; and a bigram model with Kneser-Ney smoothing to handle zero occurrences for the “not misinformation” corpus, background, with 54,444 words. For each bigram (x,y) in the foreground corpus, we calculated the phraseness from  $p(x, y)_{fg}$  and  $p(x)_{fg}p(y)_{fg}$ , and the informativeness from  $p(x, y)_{fg}$  and  $p(x, y)_{bg}$ . Table I shows the top 15 keyphrases or informative terms in COVID-19-related misinformation tweets compared to the not misinformation tweets.

TABLE I  
TOP KEYPHRASES IN COVID-19-RELATED MISINFORMATION TWEETS

Phrase	KLIP score
sec panelo	29.13652
winning labas	28.16712336
crowds never	28.16712336
large crowds	28.16712336
next election	28.02828692
becomes full	27.76165825
wisely next	27.76165825
checked regularly	27.47397618
chupa valentines	27.47397618
bio weapon	27.35619314

From the list of keyphrases, additional misinformation tweets were discovered by checking which of the tweets contains at least one of the keyphrases. From the 7,711 unlabeled data set, A total of 3,533 tweets were tagged as misinformation. A random sampling of two thousand tweets, one thousand each from the misinformation-tagged data and from the tweets not tagged as misinformation, was done to evaluate the model using confusion matrix, shown in Table II (“1” for “Misinformation” and “0” for “Not Misinformation”). Table III shows the metrics for every threshold for KLIP score.

We used the KLIP score threshold of exactly 20.2669 with a precision of 92.31% to get the misinformation tweets that can be combined with the training data containing misinformation. The combined data set of 671 tweets was prepared for topic

TABLE II  
KLIP CLASSIFIER MODEL CONFUSION MATRIX

		Predicted	
		1	0
Actual	1	593	108
	0	407	892

TABLE III  
KLIP CLASSIFIER MODEL METRICS

Threshold	Accuracy	Precision	Recall	F-score
0	74.25%	59.30%	84.59%	70%
5	70.20%	67.80%	28.53%	40%
10	66.85%	61.31%	14.69%	24%
15	65.80%	61.64%	6.42%	12%
20	65.45%	85.71%	1.71%	3%
25	65.25%	87.50%	1.00%	2%

modeling. We removed the top 15 words from the data based from the data preprocessing methods of [5] before performing BTM. We extracted 10 topics from the data and used open coding to label each topic model as shown in Table IV with a corresponding tweet related to the topic.

After extracting and labeling the topic models, a thorough analysis of the themes to cluster them into narratives was conducted. It is a way of “building a story” and have a deeper understanding of the themes from the data set [13].

## V. DISCUSSION

In Table I, it shows the top keyphrases in the misinformation data set compared to the background data set about COVID-19. The phrase “sec panelo” got the highest KLIP score which indicates that this is the most distinct or informative phrase in the misinformation training data set. Here is an example tweet from the misinformation training data that contains the phrase “sec panelo”: “Para maiwasan ang NCoV, Palakasin ang immune system sabi nga ni Sec. Panelo ‘no need to ban Chinese friends, you just need to boost your immune system’, so tayo na po ang mag adjust. -Baka hindi pa nakainom ng gamot si Tatang. Paka obob. Before and After” [in order to avoid NCoV, strengthen your immune system as what Sec. Panelo said ‘no need to ban Chinese friends, you just need to boost your immune system’, so let us adjust]. The tweet implies that the Chief Presidential Legal Counsel of the Philippines, Salvador Panelo, said that you only needed to strengthen your immune system to prevent yourself from contracting the COVID-19. Although it was the most informative phrase for our misinformation data, it did not appear in the unlabeled data set. The highest scoring phrase that appeared in the unlabeled data set was “next election” which got a score of 28.028. A sample tweet is “May this covid-19 pandemic remind us to #VoteWisely next election,” which implies that the COVID-19 pandemic reached the Philippines because of the government’s mishandling of the situation and let it be a reminder to vote worthy leaders next time. The phrase “ncov virus” got the

TABLE IV  
BTM RESULTS

Topic	Label	Topic models
Topic 1	Frustrations	kasi, bansa, baka, tapos, natin, ayan, tuloy, positive, sakit, kumain, pati, muna, safe, sitwasyon, please, like, hospital, mama, lockdown, buong
Topic 2	Government’s pandemic response	confirmed, cases, hospital, government, case, pilipinas, malapit, lalo, sobrang, natin, naka, philippines, mamatay, kamay, kasi, nakakatakot, inyo, number, deadly, building
Topic 3	Social responsibilities	always, matataong, keep, ligtas, home, lumayo, time, masks, jakol, everyone, maligo, safe, makaiwas, inside, sanitize, vitamins, clean, hygiene, lugar, message
Topic 4	Remedies or cure	people, natural, infected, kasi, please, prevent, safe, masyadong, everyone, testing, vitamin, china, epidemic, global, buying, vaccine, case, boost, meron, prevention
Topic 5	Undermining the virus	happy, needs, health, gave, like, pilipinas, outbreak, sars, masks, philippines, infected, year, period, really, people, going, prevent, china, proper, possible
Topic 6	Local governments’ pandemic responses	outbreak, niyo, city, officials, social, could, health, would, distancing, cases, spread, buti, sure, instead, know, like, corona, country, started, hirap
Topic 7	Boosting the immune system to fight COVID-19	help, washing, epidemic, satin, government, contact, vitamins, wash, experiencing, point, protect, wearing, proper, stupid, philippines, safety, natin, sakit
Topic 8	Precautionary measures	wash, precautions, need, sure, masks, distancing, quarantine, even, crowded, social, still, healthy, much, sanitizer, wear, vitamins, make, home
Topic 9	Fighting COVID-19	nasa, doctors, tapos, like, even, boost, araw, prevent, work, bwisit, symptoms, panlaban, advice, food, lysol, puro, vitamin, know
Topic 10	COVID-19 vs other illnesses	death, even, china, missing, first, without, public, year, something, must, outside, alerted, sars, competent, less, lethal, contagious, male, infectious, suddenly

lowest score with 0.009 which implies that it can not represent the key idea of the misinformation training data set.

A total of 3,533 tweets were tagged as misinformation with 74% accuracy, 59% precision, and 85% recall, as shown in Table III. The results show that almost all the tweets from the 4,178 testing data are correctly tagged by the model as “not misinformation”. With a precision of 92.31%, we used the 20.2669 KLIP score threshold to filter the misinformation tweets that can be combined with the training misinformation data for topic modeling.

After addressing the identification of misinformation tweets, topic modeling is performed to understand COVID-19 misinformation in Twitter. We extracted 10 themes from the misinformation data set as shown in Table IV. From conducting a final stage of manual analysis, we propose three narratives, among other possible narratives, clustered from the ten themes. The first narrative is about the tweets that contain humor and underestimating the COVID-19. The second narrative is

grouped from the tweets expressing frustrations during the start of COVID-19 outbreak in the Philippines. The third narrative is about the true nature of the virus according to the authors of the tweets.

#### A. *“It is just COVID”*

The first narrative emerges from the tweets that implies the inferiority of the COVID-19 among other illnesses. These tweets usually contain the words “less” and “more”. The tweet “I just don’t understand why people are more afraid of the nCOV than measles which has a higher mortality rate” is one of the tweets that can be interpreted that we should not be afraid of the new strain of coronavirus since it is less dangerous. Similar tweets compare COVID-19 and other specific viral infections like HIV and common flu.

Using the term “lang” [just/simply/only] is also common in tweets included in this narrative. The term is usually found in phrases like “ncov ka lang” [you are just nCoV] or “covid lang yan” [it is just COVID]. People tend to justify a future plan or an action done amidst the pandemic (“HAHAHHA ARAT COVID LANG YAN GALA TAYO” [it is just COVID let us hang out]).

Tweets with satirical content are also prevalent with 76 occurrences in the data set. Satirical tweets can be hard to detect without knowing their context. Tweets like “mas marami pa ang infected ng tiktok virus kesa covid jusko mama” [more are infected by the tiktok virus than COVID oh my God mama] could be interpreted in different ways such as: it could imply that the new strain of coronavirus should not be a concern since there are more people using the TikTok app and “infected” by it than the number of people in the Philippines infected by COVID-19, or it could suggest that a new virus named “Tiktok” does exist. Some tweets may suggest, as a joke, a cure for COVID-19; like “Puno na nang alcohol katawan ko malamang neto covid free nako” [my body is full of alcohol I’m probably COVID free]. There are also tweets that contain puns and word plays. One example is a portmanteau of Kobe Bryant and COVID-19, COVID Bryant. Other tweets show a different meaning to the terms “nCoV” and “COVID”. For instance, “NCoV- NEED CASH ON VALENTINES”.

The tweets covering this narrative show that Filipinos find hope in a disaster by making big problems seem small. Downplaying the COVID-19 by comparing it against other fatal virus may be from the fact that COVID-19 was new in the country at the time of posting such tweets that is why people had the impression on COVID-19 being a weaker virus. Adding humor to tweets about COVID-19 is another way of downplaying the virus and its effects that may demonstrate a less cautious way to live through the pandemic.

#### B. *“What now?”*

A second narrative emerges from tweets that express negative sentiments towards the situation. The tweets’ subjects in this cluster of themes revolve around government’s response (and lack of response). The phrase “vote wisely”, with 9

occurrences, can also be seen in the tweets, sometimes paired with “next election”, as a reminder for the people. There are tweets that express disappointment towards the mentioned name of official, either from a local or a national government post. A total of 9 tweets were seen mentioning ‘Mayor Joy’, mayor of Quezon City, Philippines. The phrase was seen together with words “anuna” or “ano na” [what now]. People also express their disappointment by not voting for the politician next election.

There are also tweets, a total of 32, that mention the name and nickname of President Rodrigo “Digong” Duterte. Some tweets contain a mockery of his own words, way of speaking, and strange solutions; for example, “Akala ko iihian na naman niya eh” [i thought he will pee on it again] wherein a solution Pres. Duterte proposed, last 13<sup>th</sup> of January 2020, to pee on the Taal volcano to stop the eruption was also suggested by a Twitter user to kill the coronavirus; and the term “veerus” which how he pronounced the word “virus” in his press conference last 3<sup>rd</sup> of February 2020. The tweet “State of calamity ba dahil ba sa coronavirus o sa Duterte Veerus!?! I really wanna know...” [state of calamity because of the coronavirus or the Duterte Veerus!?! I really wanna know...] implies that there is a possibility that Duterte was the one causing the big problem of the country. Some people are expressing their disapproval of not imposing a travel ban immediately – “4th is the very best, if this digong ordered at once the blocking of all chinese nationals from entering our ports and airports, there is no way for ncov to enter our country.”

In the data set, it is common to see tweets that contain government’s official announcements, protocols, and guidelines that comes with personal comments and opinions. Data shows that people are blaming the government for the COVID-19 outbreak in the Philippines.

#### C. *The virus*

The third narrative is clustered from tweets that tell stories or share knowledge about the COVID-19. In 7 tweets, it is implied that the COVID-19 was created as a “biochemical weapon”. For example, “Lemme make my own assumptions regarding CoVID reaching S.Korea. Not sure if everyone knows that this started as a Biochemical Weapon being made by the Chinese people & was spread for reasons we are not sure how. Also its meant to be used to attack somewhere.. And how this spread.”

There are also tweets that explains the weaknesses of the virus and how we can avoid contracting it. Some would say that heat can kill the virus (“Good amount of people wearing masks in Manila. Don’t they know heat kills coronavirus?). Drinking liquor, according to them, could kill the virus (“uminom ka lang..mabisang panlaban yan sa covid” [just drink..effective against COVID]). Since rubbing alcohol is effective in surface sanitation and killing the virus, we could also drink it (“Tara inom ng alcohol iwas sa covid haha” [let us drink alcohol to avoid COVID haha]).



In addition to drinking liquor or rubbing alcohol, tweets about boosting and strengthening the immune system is recurring 33 times. It is said that adequate sleep, healthy diet, proper hygiene, and drinking vitamins are enough to survive from the virus or even prevent yourself from acquiring it (“I was told a ketogenic diet improves the immune system, thus a good incentive in fighting nCoV.”). And people who have weaker immune system is more vulnerable to the disease (“Yari ako Kay CoVid-19 ..., Mahina Immune system ko Hays” [i am doomed because of COVID-19., my immune system is weak”]).

In the data pool, conspiracy theories are common. False knowledge about the nature of the virus could have an effect on how people will approach the situation. Some tweets covered by this narrative can also be taken as a joke especially the ones about drinking rubbing alcohol to kill the virus inside our body. It could also be observed that people share tips and reminders in fighting COVID-19 out of concern for others.

## VI. CONCLUSION

The study aims to have a deep understanding of the misinformation about COVID-19 in the Philippines through the analysis of Twitter data. This analysis can be done after finding the misinformation tweets, and that is also addressed in this study.

One limitation of this study is the date coverage and the small data size for misinformation training data. Future work could use a wider date coverage to achieve bigger data size which could give us more misinformation data for training that could yield better results for automated classification of tweets. In conclusion, key phrases extracted from the misinformation training data set using KLIP can be used to identify additional misinformation tweets.

After identifying which among the data gathered are misinformation, we used the best precision to filter the misinformation tweets before performing topic modeling and final stage of analysis. Results show different opinions, experiences, thoughts, and knowledge shared online. Ten topics were extracted, then clustered into three narratives. The suggested narratives showed that the shared misinformation on Twitter in the Philippines are from Filipinos who used Twitter during the start of the COVID-19 pandemic to share jokes, express frustrations, spread information and awareness, and share tips against the virus out of concern for others. Future work could also attach a demographic data like age, income, education, and career profession which could add insights on the misinformation data and how they spread, and deeper understanding on the authors or sharers of misinformation.

## REFERENCES

- [1] W. H. Organization, “Novel coronavirus (2019-ncov): Situation report, 13,” Technical documents, 2020, 7 p.
- [2] C. C. David, J. C. Ong, and E. F. T. Legara, “Tweeting supertyphoon haiyan: Evolving functions of twitter during and after a disaster event,” *PloS one*, vol. 11, no. 3, e0150190, 2016.
- [3] J. C. Ong and J. V. A. Cabañes, “Architects of networked disinformation: Behind the scenes of troll accounts and fake news production in the philippines,” *Architects of networked disinformation: Behind the scenes of troll accounts and fake news production in the Philippines*, 2018.
- [4] P. Bastani and M. A. Bahrami, “Covid-19 related misinformation on social media: A qualitative study from iran.,” *Journal of medical Internet research*, 2020.
- [5] C. Ligutom, J. V. Orio, D. A. M. Ramacho, C. Montenegro, R. E. Roxas, and N. Oco, “Using topic modelling to make sense of typhoon-related tweets,” in *2016 International Conference on Asian Language Processing (IALP)*, IEEE, 2016, pp. 362–365.
- [6] C. R. Soriano, M. D. G. Roldan, C. Cheng, and N. Oco, “Social media and civic engagement during calamities: The case of twitter use during typhoon yolanda,” *Philippine Political Science Journal*, vol. 37, no. 1, pp. 6–25, 2016.
- [7] J. Kapusta, L. Benko, and M. Munk, “Fake news identification based on sentiment and frequency analysis,” in *International Conference Europe Middle East & North Africa Information Systems and Technologies to Support Learning*, Springer, 2019, pp. 400–409.
- [8] L. T. B. Ranera, G. A. Solano, and N. Oco, “Retrieval of semantically similar philippine supreme court case decisions using doc2vec,” in *2019 International Symposium on Multimedia and Communication Technology (ISMAT)*, IEEE, 2019, pp. 1–6.
- [9] G. K. Shahi, A. Dirkson, and T. A. Majchrzak, “An exploratory study of covid-19 misinformation on twitter,” *Online social networks and media*, vol. 22, p. 100 104, 2021.
- [10] T. Tomokiyo and M. Hurst, “A language model approach to keyphrase extraction,” in *Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment*, 2003, pp. 33–40.
- [11] K. Gorro, J. R. Ancheta, K. Capao, N. Oco, R. E. Roxas, M. J. Sabellano, B. Nonnecke, S. Mohanty, C. Crittenden, and K. Goldberg, “Qualitative data analysis of disaster risk reduction suggestions assisted by topic modeling and word2vec,” in *2017 International Conference on Asian Language Processing (IALP)*, IEEE, 2017, pp. 293–297.
- [12] X. Yan, J. Guo, Y. Lan, and X. Cheng, “A bitern topic model for short texts,” in *Proceedings of the 22nd international conference on World Wide Web*, 2013, pp. 1445–1456.
- [13] Z. C. Pablo, N. Oco, M. D. G. Roldan, C. Cheng, and R. E. Roxas, “Toward an enriched understanding of factors influencing filipino behavior during elections through the analysis of twitter data,” *Philippine Political Science Journal*, vol. 35, no. 2, pp. 203–224, 2014.

# Blockchain based Secure Data Exchange between Cloud Networks and Smart Hand-held Devices for use in Smart Cities

Muneer Ahmad Dar<sup>†</sup>, Aadil Askar<sup>‡</sup> and Sameer Ahmad Bhat<sup>§,✉</sup>

<sup>†</sup>National Institute of Electronics and Information Technology (NIELIT), Jammu & Kashmir, India.

<sup>‡</sup>Dept. of Self Development Skills, King Saud University, Riyadh, Saudi Arabia.

<sup>§</sup>Gulf University for Science and Technology (GUST), Meref, Kuwait.

<sup>✉</sup>Gdansk University of Technology, Pomerania Gdansk, Republic of Poland.

Email: muneer@nielit.gov.in, aadil@ksu.edu.sa, bhat.s@gust.edu.kw

**Abstract**—In relation to smart city planning and management, processing huge amounts of generated data and execution of non-lightweight cryptographic algorithms on resource constraint devices at disposal, is the primary focus of researchers today. To enable secure exchange of data between cloud networks and mobile devices, in particular smart hand held devices, this paper presents Blockchain based approach that disperses a public/private key to save it on a block within a Blockchain. The proposed system generates public-private key pair to encrypt data digitally to allow data communication. This empowers communication devices to encipher data using keys stored in the Blockchain i.e. the public key. Generated cipher text can be decrypted/deciphered only with the respective private keys, meaning that only the communicating devices can obtain their own plain text in a data exchange process. Smart mobile employed in smart city can then encipher the data using the keys and store them on the cloud. The proposed system is able to decrease the number of overheads that relate to key generation, key delivery and key storage whilst providing solutions for data processing, information exchange and data over-collection, respectively. Thus, the study proposes a robust and secure solution to exchange keys and secure data communication based on Blockchain technology.

**Index Terms**—Blockchain, Cryptographic Algorithms, Digital Signature, Digital Signatures, Smart City.

## I. INTRODUCTION

With the inflation in score of population moving towards cities, the usage of technological solutions is of paramount priority of various governments across the world. The smart cities are coming up, which provides every kind of facility to its citizens. The technological solutions, be it digital transactions, smart healthcare, smart education and lot more, the data over-collection and securing the private/confidential data of citizens is of paramount importance. The EU group which looks for the information security of smart cities has come up with various protection measures that must be implemented in a smart city [1]. They recommend encryption of data that is transmitted, incorporation of intrusion detection system, Installation of VPNs, installation of alarms and surveillance and many other measures. Thus the user in a smart city is always on radar of intruders who always try to steal their critical data. With the introduction of Blockchain technology, many issues pertaining to the smart city can be resolved. The paramount advantage

of Blockchain is its scattered way of authentication and use of encryption that makes use of both public and private keys. With no centralized power, the Blockchain provides a vital security for the financial transactions. The distributed ledger concept of Blockchain has raised bitcoin in the form of digital crypto currency which is procured by millions. Thus the advantage of Blockchain can be exploited for the proficient, flexible and above all secure implementation of smart cities. Various countries have already used the Blockchain technology for the betterment of their citizens. Table I provides the list of initiatives [2].

TABLE I  
BLOCKCHAIN INITIATIVES BY VARIOUS COUNTRIES

Country Name	Blockchain Applied On
Sweden	The transactions related to real estate are maintained using Blockchain
Estonia	Patients Medical record is maintained using Blockchain
Ghana	The property documents are maintained using Blockchain
Russia	Shareholders transactions and secure transactions are maintained using Blockchain
Korea	Banking
Singapore	Blockchain based trading
Dubai	City Logistics, Paperless government System

## II. REVIEW OF BLOCKCHAIN TECHNOLOGY

The technique used by bitcoin introduced by Nakamoto is the most widely used append only distributed database for crypto-currency [4]. As demonstrated in the Fig 1, the Blockchain comprises of blocks. Each block has the following data members with the description as in Tabel II under:

The order of the Blockchain is controlled by the preceding blocks' hash value. The main advantages of Blockchain are distinguishability, directness and understanding, decentralized, verifiable, and many more. Transactions made between nodes i.e. users inside the Blockchain network will be obtained by certain nodes with the consensus protocol Proof-of-Work (PoW) as an illustration as in miners. Then the nodes compete for possibilities for generating the new block by listing the

TABLE II  
BLOCKCHAIN BLOCK STRUCTURE.

Block Data	Description
Version Number	For keeping track of modifications and updates that have occurred during the protocol's lifetime.
Previous Block Hash	Hash of preceding block
Merkle Root	It provides a unified hash value that allows you to validate anything that is contained within that block.
Time Stamp	A sequence of signs or encoded information that can be accurate to a fraction of a sec, if a specific event occurred, normally with a date and time of the day.
Difficulty Number	The difficulty of mining a block, or the difficulty of finding a hash below a specified goal, is a measure of how tough it is to mine a Bitcoin block.
Random Number	The generation of random numbers enables us to generate private keys— which are a component of your key pair.
Transaction Data	As soon as a transaction is entered in the Blockchain, its data, including as the price, the product, and control, are recorded and validated across all nodes within seconds, allowing it to be settled across the whole network.

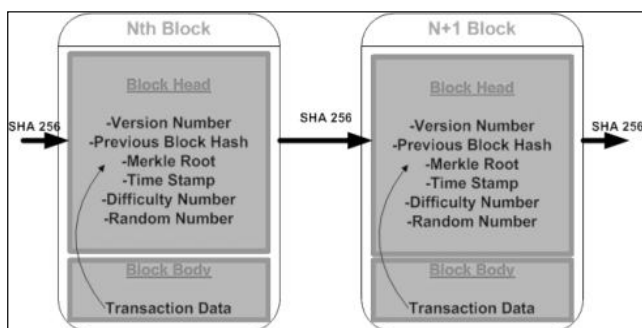


Fig. 1. Blockchain Structure.

previous block head hash values by increasing their parent block head random numbers. by growing the number of parent block Heads.

The miner will only construct a totally new block including transaction completed since the previous block is formed, depending on the difficulty number, if the hash value fulfills the criteria of the challenge number (ie, a system parameter for regulating block generation rate for Blockchain). The block is then sent to all network participants in the Blockchain system via the new block. Once the hash value confirms the supplied challenging number, all nodes will include the new block and link the new Blockchain to the local Blockchain sync to the global Blockchain. In addition, a Digital Signature Algorithm (DSA) is employed for communications security, and a Merkle Hash Tree for transaction information protection purposes is utilized. The Blockchain is therefore ready to offer us with a distributed, reliable and trustworthy consensus environment in the longer term.

A digital signature is a mathematical technique that confirms the validity of digital messages or documents throughout the communications process [5]. For a legitimate digital signal,

the recipient is certain that a recognized and verified sender has generated the message. In the meanwhile, neither the sender nor the receiver can deem the transmission of the communication, nor the transmission of the message. In other words, the receiver can immediately determine whether the message has been altered or deleted during transmission. The proposed technique uses a secure communications mechanism based on ECDSA's cryptographic algorithm in order to ensure data security throughout communication. ECDSA, but at the other hand, is a kind of DSA combining the DSA with the Cryptography Elliptical Curves, which Neal Koblitz presented [6] and Victor Miller offered [7], and it is both hybrid.

A Merkle Hash Tree [8] is a kind of binary tree constructed as building pieces with hash values. As shown in Fig 2: The information included within a leaf node is the hash value of a business, but it is the hash of the mixture of the child nodes of a leaf node in which it is recorded. This technique employs the safe SHA256 hash algorithm [9], an irregular computational process, and a cryptographic scheme with pseudo-randomness. It is used to secure transactions on the Blockchain of the scheme. As immediately as the load varies little, its output will thus be very changeable. This allows users to check if the processes or the data in the block body remain valid or not on the Blockchain network, i.e. nodes, due to this feature. Consider an example to better grasp what this means: Any change in the block contents of a block may be detected by MHT by considering each operation's hash value as new point in the MHT.

### III. EXISTING RESEARCH

In order to investigate and apply the technology Blockchain, Bitcoin's success led intellectuals to explore several fields, such as Smart Contracts [11], Finance [9], Management of HR [14], Supply Chain [15] and Internet of Things [11], [17]. In [11], for example, the authors stated that the Blockchain technology should be lightly installed for a smart IoT dwelling. Multiple IoT equipment is connected to one single network in every residence (e.g. smartphones, PCs and sensors). Under Bitcoin's success scholars have been driven to research and utilize Blockchain technology in many fields, including Smart Agreement [11] and Finance [4], the supply chain [15], and human resources management [14], and [11], [17]. In [11], for example, the authors suggested that the Blockchain technology be light installed in the intelligent IoT dwelling. Several IoT users are interconnected to the very same network in each home, such as smart devices like phones, personal PCs and sensors to collect data. The suggested model provides for only approved users to access and manage home devices and safeguarded and unable to modify the messages received by authorized users by the malicious users. The creators of [12] have established a new spread- out information managing policy that enables users to monitor their records so that third-party infringements can be avoided. The platform mixes cryptocurrency and off-Blockchain stores to build a framework for private data administration.

Since, the Blockchain acknowledges the users as their proprietors; therefore users are not needed to have confidence

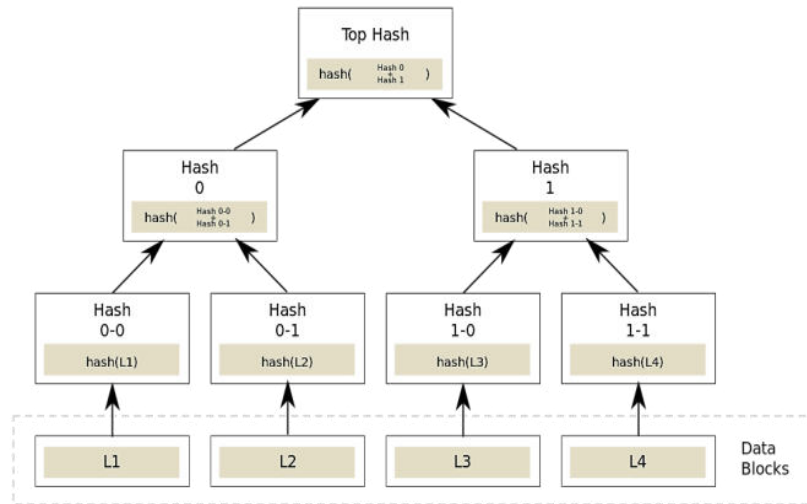


Fig. 2. Merkle Hash Tree [10]

in any third party. The authors have expanded [12] with the addition of a new approach, dubbed a Proof of Credibility Score (PCS), to improve the crypton algorithm for mining procedures. The suggested PCS technique uses the interconnection between nodes to determine the credibility score differently from [12] where the trust value of the node is gathered for how many beneficial acts the Node has taken. The numerical findings have then shown that the security measures can be upgraded with the proposed scheme Blockchain of credibility. The scalability of the Blockchain is a major difficulty as the application grows. The authors in [18] offered to tackle this problem with a BigchainDB system in NoSQL database format. The Blockchain pipeline is used to scale the system to the distributed database by adding Blockchain features. In MC, transactions at mobile nodes should be taken into account to enable the direct interchange and sharing of peer-to-peer data. This is particularly critical if connecting devices have really no internet connection. Presently some Android applications, such as Easy-Miner [12] and Scrypt-Miner PRO and LTC, are developing for mining on Mobile equipments for the Bitcoin relevance. They are still demo versions, though, and have not yet finished. Especially because Bitcoin applications are employed for crypto-currency applications alone, there are still a lack of the platform for broader Blockchain activities.

#### IV. PROPOSED ARCHITECTURE

A digital signature is an authentication method. It consists of a public key pair and digital certificate, to authenticate or verify either recipient's or sender's identity. Elliptic curve cryptography generates smaller keys compared to digital signing algorithms that generate average length keys. Elliptic curve cryptography implements the algebraic structure of elliptic curves over finite fields, and it is a public key cryptography. Elliptic curve cryptography helps to generate definite or random sequences, such as pseudo-random numbers, digital signatures, and more.

#### A. Elliptic Curve Digital Signature Algorithm (ECDSA)

Elliptic Curve Digital Signature Algorithm (ECDSA) is based on public key cryptography (PKC). It is a Digital Signature Algorithm (DSA), uses key derived from Elliptic Curve Cryptography (ECC). ECDSA signed certificates are used to encrypt connection requests of an HTTPS website that informs about the applied encryption by an image of a physical padlock displayed by the browser. ECDSA could be also found implemented in security systems, such as secure messaging apps, including Bitcoin security. While serving at Transport Layer Security (TLS), ECDSA encrypts connection requests between web browsers and a web application.

Compared to another popular algorithm- RSA, ECDSA offers high level security with short key lengths, is the primary feature of ECDSA. Apparently, ECDSA executes at low computational power requirements compared to RSAM, which is a less secure competing equation. Elliptic Curve Digital Signature Algorithm (ECDSA) is an elliptic curve-based encryption and digital signature scheme. ECDSA can be used to apply digital signature, however more efficiently. ECDSA bases on an elliptic curve, and the curve is analyzed for a point. After analyzing, a point is chosen on the curve, and then next step is to multiply the selected point by another number. This just creates a new point on the same curve. As a result of multiplication, the key lies in the fact that finding the the new point on the curve is really a complex and hard task. Even if the original point is known, the new point cannot be found. This complexity of ECDSA highlights its robustness against methods used to decrypt the data exchanged during the communication processes.

The proposed methodology allows a member in the suggested technique to join, relocate and consider leaving any subset. The proposed scheme, like Bitcoin, Ethereum applications sets up key encrypted duos without centralized authority for the Elliptic Curve Digital Signature Algorithm (ECDSA) - based data exchange. Unlike the Bitcoin Blockchain, which takes about an hour to make a transaction in Bitcoin (to actu-

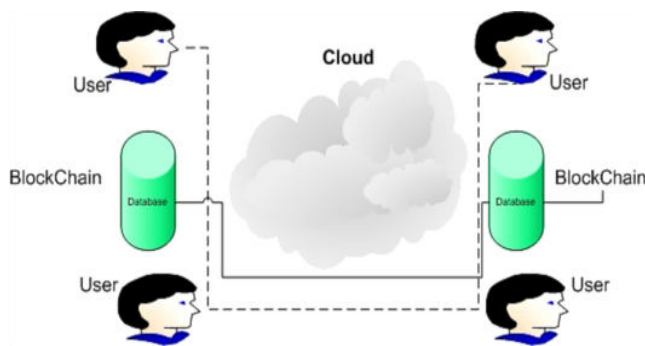


Fig. 3. Proposed Architecture.

ally create 6 blocks), Blockchain inside the scheme proposed adds blocks directly after they are received. And the proposed solution may prove to be significantly effective for the mobile devices used in the proposed scheme.

Fig. 3 shows a straightforward operating process of our proposed model. For the Blockchain, the open or public keys of all members eligible to establish a secure communication will be stored on a Cloud network. Any member may use an open/public key stored by other members on the Blockchain of the associated sub-network so as to connect with the other group members. A mobile device closer to a cloud may find easy access to a number of the available resources on Cloud, and conversely devices faraway from Cloud service may have limited access. As more and more mobile devices connect to nearest available cloud service, the availability of services needs to be ensured.. However, the services need to qualify and fulfill the increased demand for computational resources, communication channels bandwidth, and large storage resources.

## V. CONCLUSION

Currently available mobile devices are often categorized under resources constraint devices, as it is hard to execute computationally complex algorithms on such hardware platforms. As millions of citizens show trends to maintain social connectedness in a smart city, there is a need to adopt lightweight security architectures. Not only would such architectures protect the privacy of users, but also could run or execute at a low computational cost. Therefore, this study provides advantages of using technological innovations in Blockchain and proposes model for secure data exchange between the mobile devices used in smart cities. The model proposed shows a conceptual model and gives an insight of how implementing Blockchain can decrease the computational complexity of the algorithms to allow them run on hardware constraint devices. In our future studies, we plan to test the proposed system on the available different resource restricted platforms so as to investigate in details the hardware and software complexities posed to processes that implement Blockchain technology.

## REFERENCES

[1] Smart Cities Cyber Security Management, Consultancy Report. Available at <https://securingsmartcities.org/wp-content/uploads/2017/09/SSC-SCCCM.pdf>

[2] PWC – PWC’s Global Blockchain Survey, 2018. Available at <https://www.pwccn.com/en/research-and-insights/publications/global-Blockchain-survey-2018/global-Blockchain-survey-2018-report.pdf>

[3] Gupta, S.: ‘Using Blockchains in smart cities’, Meetings of The Mind, 2018. Available at <https://meetingoftheminds.org/using-Blockchain-in-smartcities-29319>

[4] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system”, <https://bitcoin.org/bitcoin.pdf> , 2008.

[5] Wikipedia contributors, “Digital signature — Wikipedia, the free encyclopedia,” [https://en.wikipedia.org/w/index.php?title=Digital\\_signature&oldid=876680165](https://en.wikipedia.org/w/index.php?title=Digital_signature&oldid=876680165), 2019, [Online; accessed 08-September-2021].

[6] N. Koblitz, “Elliptic curve cryptosystems,” *Mathematics of computation*, vol. 48, no. 177, pp. 203–209, 1987.

[7] V. S. Miller, “Use of elliptic curves in cryptography,” in *Conference on the theory and application of cryptographic techniques*. Springer, 1985, pp. 417–426.

[8] R. C. Merkle, “A digital signature based on a conventional encryption function,” in *Conference on the theory and application of cryptographic techniques*. Springer, 1987, pp. 369–378.

[9] N. T. Courtois, M. Grajek, and R. Naik, “Optimizing sha256 in bitcoin mining,” in *International Conference on Cryptography and Security Systems*. Springer, 2014, pp. 131–144.

[10] Merkle Tree, Wikipedia, the free encyclopedia [https://en.wikipedia.org/wiki/Merkle\\_tree](https://en.wikipedia.org/wiki/Merkle_tree) [Online; accessed 08-September-2021].

[11] A. Dorri, et al., “Blockchain for IoT security and privacy: The case study of a smart home,” in *IEEE International Conference on Pervasive Computing and Communications Workshops*, pp.618-623, Hawaii, USA, Mar. 2017.

[12] G. Zyskind, O. Nathan, and A.S. Pentland, “Decentralizing privacy: Using Blockchain to protect personal data,” in *IEEE Security and Privacy Workshops*, pp. 180-184, San Jose, USA, May 2015.

[13] D. Fu and L. Fang, “Blockchain-based trusted computing in social network,” in *IEEE International Conference on Computer and Communications*, pp. 19-22, Chengdu, China, Oct. 2016.

[14] X. Wang, L. Feng, H. Zhang, C. Lyu, L. Wang, and Y.You, “Human resource information management model based on Blockchain technology,” in *IEEE Symposium on Service-Oriented System Engineering*, pp. 168-173, San Francisco, USA, Apr. 2017.

[15] H. M. Kim and M.Laskowski, “Towards an ontology-driven Blockchain design for supply chain provenance,” *Open-Access Online Library*, Aug. 2016.

[16] Global M-commerce Market 2016-2020. Technavio’s Report.

[17] A. Dorri, S. S. Kanhere, and R. Jurdak, “Towards an optimized blockchain for IoT,” in *IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation*, pp. 173-178, Pittsburgh, USA, Apr. 2017.

[18] T. McConaghy, R. Marques, A. Miller, D. De Jonghe, T. McConaghy, T. G. McMullen, and A. Granzotto, “BigchainDB: a scalable Blockchain database,” *White Paper, BigChainDB*, 2016.

[19] M. A. Dar, S. Nisar Bukhari and U. I. Khan, “Evaluation of Security and Privacy of Smartphone Users,” 2018 Fourth International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), Chennai, 2018, pp. 1-4, doi: 10.1109/AEEICB.2018.8480914.

[20] Dar, Muneer & Khan, Ummer & Bukhari, Syed. (2019). Lightweight Session Key Establishment for Android Platform Using ECC. 10.1007/978-981-13-3122-0\_33.

[21] M. A. Dar and J. Parvez, “Security Enhancement in Android using Elliptic Curve Cryptography,” *Int. J. Secur. its Appl.*, vol. 11, no. 6, pp. 27–34, 2017.

The 4<sup>th</sup> International Conference on **Artificial Intelligence** in  
**Information and Communication**

**ICAIIIC 2022**



<http://icaiiic.org>