



IW-FCV 2023

The 29th International Workshop on Frontiers of Computer Vision

February 20-22, 2023, Yeosu, Utop Marina Hotel, South Korea

<https://iwfcv2023.github.io/>

Online Workshop Proceeding

| Organized by |

- IW-FCV 2023 Organizing Committee

| Co-organized by |

- Culture Technology Institute, Chonnam National University
- Korean Institute Smart Media
- CNU National Program of Excellence in Software

| Sponsored by |

- JeollaNamdo Tourism Organization, Yeosu City, Korean Computer Vision Society

Oral Presentations

Oral Session 1

When & Where: February 20th, 09:00 ~ 10:40, 1F Grand Ballroom

Chair: Prof. Kanghyun Jo, Prof. Go Irie

O1-1. Hierarchical Image Classification with Conceptual Hierarchies Generated via Lexical Databases.....	1
Tomoaki Yamazaki ¹ , Seiya Ito ¹ and Kouzou Ohara ¹	
¹ Aoyama Gakuin University, Japan	
O1-2. Action Recognition for Each Person with Feature Extraction by Large-scale Object Detector.....	14
Akira Mitsuoka ¹ and Kunihito Kato ¹	
¹ Gifu University, Japan	
O1-3. Structural Point Cloud Data Recovery to Learning 3D Feature Representation.....	29
Ryosuke Yamada ¹ , Ryu Tadokoro ² , Yue Qiu ² , Hirokatsu Kataoka ²	
and Yutaka Satoh ²	
¹ University of Tsukuba, Japan	
² AIST, Japan	
O1-4. Point Cloud Based Deep Molecular Pose Estimation for Structure-Based Virtual Screening.....	37
Ken Kariya ¹ , Go Irie ¹ , Ryosuke Furuta ² , Yota Yamamoto ¹ , Shin Aoki ¹	
and Yukinobu Taniguchi ¹	
¹ Tokyo University of Science	
² The University of Tokyo, Japan	
O1-5. Efficient Multi-Receptive Pooling for Object Detection on Drone.....	52
Jinsu An ¹ , Dwisnanto Muhamad Putro ² , Adri Priadana ¹ and Kanghyun Jo ¹	
¹ University of Ulsan, South Korea	
² Universitas Sam Ratulangi, Indonesia	

Oral Session 2

When & Where: February 20th, 14:30 ~ 16:10, 1F Grand Ballroom

Chair: Prof. Hae-Gon Jeon, Prof. Yuji Pyamada

O2-1. Robust Scene Text Detection under Occlusion via Multi-Scale Adaptive Deep Network..... 64

My-Tham Dinh¹, Minh-Trieu Tran¹, Quang-Vinh Dang¹ and Guee-Sang Lee¹

¹Department of Artificial Intelligence Convergence,

Chonnam National University, Gwangju, South Korea

O2-2. Detection and Tracking of Flying Small Bats under Complex Backgrounds..... 77

Kakeru Sugimoto¹, Kazusa Usio², Ryota Sugimori², Emyo Fujioka³,

Hiroaki Kawashima⁴, Shizuko Hiryu⁵ and Hitoshi Habe^{6,7}

¹Graduate School of Science and Engineering, Kindai University, Japan

²Graduate School of Life and Medical Sciences, Doshisha University, Japan

³Organization for Research Initiatives and Development,

Doshisha University, Japan

⁴School of Social Information Science, University of Hyogo, Japan

⁵Faculty of Life and Medical Sciences, Doshisha University, Japan

⁶Department of Informatics, Faculty of Informatics, Kindai University, Japan

⁷Cyber Informatics Research Institute, Kindai University, Japan

O2-3. Facial Depth and Normal Estimation using Single Dual-Pixel Camera..... 85

Minjun Kang¹, Jaesung Choe¹, Hyowon Ha⁴, Hae-Gon Jeon², Sunghoon Im³,

In So Kweon¹ and Kuk-Jin Yoon¹

¹KAIST

²GIST

³DGIST

⁴Meta RealityLabs

O2-4. Generative Bias for Robust Visual Question Answering..... 97

Jae Won Cho¹, Dong-Jin Kim², Hyeonggon Ryu¹, and In So Kweon¹

¹KAIST, Daejeon, South Korea

²Hanyang University, Seoul, South Korea

O2-5. DDConv Dilated Depthwise Convolution with YOLOv5 for Drone Imagery..... 105

Jehwan Choi¹, Minseung Kim², Donggue Kim², and Kanghyun Jo¹

¹Department of Electrical, Electronic and Computer Engineering,
University of Ulsan, Ulsan, South Korea

²School of Electrical Engineering, University of Ulsan, Ulsan, South Korea

Oral Session 3

When & Where: February 20th, 16:30 ~ 18:10, 1F Grand Ballroom

Chair: Prof. Dong-Geol Choi, Prof. Hiroaki aizawa

O3-1. DASO: Distribution-Aware Semantics-Oriented Pseudo-label for Imbalanced Semi-Supervised Learning	118
Youngtaek Oh ¹ , Dong-Jin Kim ² , and In So Kweon ¹	
¹ Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea	
² Hanyang University, Seoul, Republic of Korea	
O3-2. Improvement of Robustness to Noise for Medical Image Segmentation by using Self-Supervised Learning Approach	126
Yuta Konishi ¹ and Takio Kurita ¹	
¹ Hiroshima University, Kagamiyama, Higashi Hiroshima, Japan	
O3-3. Bidirectional Domain Mixup for Domain Adaptive Semantic Segmentation	139
Minseok Seo ² , Yuhyun Kim ¹ and Dong-Geol Choi ¹	
¹ Hanbat National University, Daejeon, South Korea	
² SI Analytics, Daejeon, South Korea	
O3-4. LabOR: Labeling Only if Required for Domain Adaptive Semantic Segmentation	149
Inkyu Shin ¹ , Dong-Jin Kim ¹ , Jae Won Cho ¹ , Sanghyun Woo ¹ , Kwanyong Park ¹ and In So Kweon ¹	
¹ KAIST, Daejeon, South Korea and Hanyang University, Seoul, SouthKorea	
O3-5. Attribute Auxiliary Clustering for Person Re-identification	159
Ge Cao ¹ and Kanghyun Jo ¹	
¹ Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Ulsan, Korea	

Oral Session 4

When & Where: February 21st, 09:00 ~ 11:00, 1F Grand Ballroom

Chair: Prof. Inseop Na, Prof. Hitoshi Habe

O4-1. UDA-COPE: Unsupervised Domain Adaptation for Category-level Object Pose Estimation	171
Taeyeop Lee ¹ , Byeong-Uk Lee ¹ , Inkyu Shin ¹ , Jaesung Choe ¹ , Ukcheol Shin ¹ , In So Kweon ¹ , and Kuk-Jin Yoon ¹	
¹ KAIST	
O4-2. Dynamic Circular Convolution for Image	182
Xuan-Thuy Vo ¹ , Duy-Linh Nguyen ¹ , Adri Priadana ¹ and Kang-Hyun Jo ¹	
¹ Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Ulsan, SouthKorea	
O4-3. Task-specific Scene Structure Representations	195
Seunghyun Shin ¹ , Jisu Shin ¹ and Hae-Gon Jeon ¹	
¹ AI Graduate School, GIST, SouthKorea	
O4-4. Learning Depth from Focus in the Wild	203
Changyeon Won ¹ and Hae-Gon Jeon ¹	
¹ Gwangju Institute of Science and Technology	
O4-5. Human Face Detector with Gender Identification by Split-based Inception Block and Regulated Attention Module	211
Adri Priadana ¹ , Muhamad Dwisnanto Putro ² , Duy-LinhNguyen ¹ , Xuan-Thuy Vo ¹ and Kang-Hyun Jo ¹	
¹ Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Ulsan, South Korea	
² Departmen tof Electrical Engineering, Universitas Sam Ratulangi, Manado, Indonesia	
O4-6. Novel Surveillance System for Suspicious Activities Analysis using Deep Learning	226
Aditya Kakde ¹ , Bhavana Kaushik ¹ , Deepika Koundal ¹ , Durgansh Sharma ² , and Neelu Jyothi Ahuja ¹	
¹ University of Petroleum and Energy Studies, Dehradun	
² Christ (Deemed to be University), Ghaziabad	

Oral Session 5

When & Where: February 21st, 14:30 ~ 16:30, 1F Grand Ballroom

Chair: Prof. Soon Ki Jung, Prof. Bhavana Kaushik

O5-1. Three-dimensional structure extraction and evaluation of microvessels in cardiac tissue imaged via confocal microscopy..... 243

Shotaro Kaneko¹, Yuichiro Arima², Masahiro Migita³ and Masashi Toda³

¹Graduate School of Science and Technology, Kumamoto University,
Kumamoto, Japan

²Dept. of Cardiovascular Medicine, Kumamoto University,
Kumamoto, Japan

³Center for Management of Information Technologies,
Kumamoto University, Kumamoto, Japan

O5-2. Multi-Attributed Face Synthesis for One-Shot Deep Face Recognition..... 258

Muhammad Shaheryar¹, Lamyanba Laishram¹, Jong Taek Lee¹,
and Soon Ki Jung¹

¹School of Computer Science and Engineering,
Kyungpook National University, Daegu, Republic of Korea

O5-3. Parallax-based Imitation Learning with Human Intervention for Uncertain Insertion Tasks..... 271

Yasuhiro Niwa¹, Kunihito Kato¹, Hiroaki Aizawa², Yoshiyuki Hatta¹
and Kazuaki Ito¹

¹Gifu University, Yanagido, Gifu City, Gifu, Japan

²Hiroshima University, Kagamiyama, Higashi-Hiroshima City,
Hiroshima, Japan

O5-4. A Style-Based Caricature Generator..... 286

Lamyanba Laishram¹, Muhammad Shaheryar¹, Jong Taek Lee¹
and Soon Ki Jung¹

¹School of Computer Science and Engineering,
Kyungpook National University, Republic of Korea

O5-5. Detecting Mounting Behaviors of Dairy Cows by Pre-Training with Pseudo Images..... 299

Yuta Okuda¹, Yota Yamamoto¹, Kazuaki Nakamura¹,
and Yukinobu Taniguchi¹

¹Tokyo University of Science, Tokyo, Japan

O5-6. Classification of Lung and Colon Cancer Using Deep Learning Method..... 313

Md. Al-Mamun Provath¹, Kaushik Deb¹, and Kang Hyun Jo²

¹Department of Computer Science and Engineering,
Chittagong University of Engineering & Technology (CUET),
Chattogram, Bangladesh

²Department of Electrical, Electronic and Computer Engineering,
University of Ulsan

Oral Session 6

When & Where: February 21st, 16:45 ~ 18:05, 1F Grand Ballroom

Chair: Prof. Jongil Park, Prof. Kazuhiko Sumi

O6-1. Reproduction of Artwork on Display using Hyperspectral Imaging and Monitor Calibration..... 325

Kyudong Sim¹ and Jong-Il Park¹

¹Hanyang University, Seoul, South Korea

O6-2. Game Engine Compatible 3D Clothes Modeling from a Single Image..... 330

Soyoung Yoon¹, Sojin Yun¹ and In Kyu Park¹

¹Department of Information and Communication Engineering,
Inha University, Incheon, Korea

O6-3. Event-Based Reflectance Separation..... 337

Ryota Kunimasu¹, Ryo Kawahara¹ and Takahiro Okabe¹

¹Department of Artificial Intelligence, Kyushu Institute of Technology,
Kawazu, Iizuka, Fukuoka, Japan

O6-4. A Set of Control Points Conditioned Pedestrian Trajectory Prediction..... 341

Inhwan Bae¹ and Hae-Gon Jeon¹

¹AI Graduate School, GIST, Gwangju, South Korea

Poster Session 1

When & Where: February 20th, 12:20 ~ 14:30, 2F Greenwich Hall

Chair: Prof. Choonsung Shin, Prof. Yota Yamamoto

P1-1. Format-Compatible 3D Metahuman Modeling from a Single Image..... 349

Sojin Yun¹, Soyoung Yoon¹, and InKyu Park¹

¹Department of Information and Communication Engineering,
Inha University

P1-2. YOLO5PKlot: A Parking Lot Detection Network Based on Improved YOLOv5 for Smart Parking Management System..... 356

Duy-Linh Nguyen¹, Xuan-ThuyVo¹, Adri Priadana¹ and Kang-Hyun Jo¹

¹Department of Electrical, Electronic and Computer Engineering,
University of Ulsan, Ulsan, South Korea

P1-3. Texture Synthesis Based on Aesthetic Texture Perception Using CNN Style and Content Features..... 337

Yukine Sugiyama¹, Natsuki Sunda¹, Kensuke Tobitani^{1,2} and Noriko Nagata¹

¹Kwansei Gakuin University, Sanda, Hyogo, Japan

²University of Nagasaki Nishi-Sonogi, Nagasaki, Japan

P1-4. Emotion Recognition by using optimised deep features_Irfan Haider..... 385

Irfan Haider¹, Guee-Sang Lee¹, Hyung-Jeong Yang¹ and Soo-Hyung Kim¹

¹Department of Artificial Intelligence Convergence,
Chonnam National University, Gwangju, South Korea

P1-5. Monitoring students' classroom attention on digital platform..... 397

Hirotoshi IBE¹ and Hiromasa NAKATANI¹

¹International Professional University of Technology in Nagoya, Japan

P1-6. Patent Image Retrieval Using Cross-entropy-based Metric Learning..... 401

Kotaro Higuchi¹, Yuma Honbu¹ and Keiji Yanai¹

¹Department of Informatics, The University of Electro-Communications,
Chofugaoka, Chofu-shi, Tokyo, Japan

P1-7. Pre-training of Pneumonia Classifier for Chest CT images using Fractal Database	409
Yuken Yoshioka ¹ , Daichi Ikefuji ² , Tomokazu Funatsu ¹ , Takashi Nagaoka ³ , Takenori Kozuka ⁴ , Mitsutaka Nemoto ⁵ , Takahiro Yamada ⁶ , Yuichi Kimura ^{7,8} , Kazunari Ishii ^{4,6} and Hitoshi Habe ^{7,8}	
¹ Graduate School of Science and Engineering, Kindai University, Japan	
² Department of Informatics, Faculty of Science and Engineering, Kindai University, Japan	
³ Department of Computational Systems Biology, Faculty of Biology-Oriented Science and Technology, Kindai University, Japan	
⁴ Department of Radiology, Faculty of Medicine, Kindai University, Japan	
⁵ Department of Biomedical Engineering, Faculty of Biology-Oriented Science and Technology, Kindai University, Japan	
⁶ Institute of Advanced Clinical Medicine, Kindai University Hospital, Japan	
⁷ Department of Informatics, Faculty of Informatics, Kindai University, Japan	
⁸ Cyber Informatics Research Institute, Kindai University, Japan	
P1-8. Advanced Video Inpainting method using Residual Query Connection	417
Youngjun La ¹ and Jong-Il Park ¹	
¹ Hanyang University, Department of Computer Science, Republic of Korea	
P1-9. Utilization of Temporal Detection Consistency for Improving the Multi-Object Tracking	425
Abhyudaya Singh Tak ¹ and SoonKi Jung ¹	
¹ Kyungpook National University, Daegu, South Korea	
P1-10. A Study on Tracking Moving Objects: Pig counting with YOLOv5 and StrongSORT	443
Seunggwan Lee ¹ , Wonhaeng Lee ¹ and Junghoon Park ¹	
¹ College of Computing and Informatics, Applied Artificial Intelligence, Ajou University, Republic of Korea	
P1-11. BRDF Measurement with TDCRA	451
Atsushi Kimura ¹ , Ryo Kawahara ¹ and Takahiro Okabe ¹	
¹ Department of Artificial Intelligence, Kyushu Institute of Technology	

P1-12. Multi-scale Recurrent Residual U-Net for Anomaly Segmentation in Industrial Images.....	455
Haoyu Chen ¹ , Shivani Sanjay Kolekar ¹ and Kyungbaek Kim ¹	
¹ Dept. of Artificial Intelligence Convergence, Chonnam National University, Gwangju, South Korea	
P1-13. LHFAN: Scene Text Recognition Method Based on Multi-level Feature Fusion and Enhancement of Semantic Knowledge.....	463
Ruturaj Mahadshetti ¹ , Guee-Sang Lee ¹ , Hyung-Jeong Yang ¹ and Soo-Hyung Kim ¹	
¹ Department of Artificial Intelligence Convergence, Chonnam National University, Gwangju, South Korea	
P1-14. Preliminary Study on Fish Tracking in Indoor Aquaculture through Deep Learning.....	474
Nguyen-Ngoc Huynh ¹ , Myoungjae Jun ¹ , Hang Thi Phuong Nguyen ¹ , Choonsung Shin ² and Hieyong Jeong ¹	
¹ Department of Artificial Intelligence Convergence, Chonnam National University, Gwangju, Republic of Korea	
² Graduate School of Culture, Chonnam National University, Gwangju, Republic of Korea	
P1-15. Front Cover Image Database of Japanese Manga and Typeface Estimation of their Title.....	482
Shota Ishiyama ¹ , Kosuke Sakai ¹ and Minoru Mori ¹	
¹ Kanagawa Institute of Technology, Atsugi-shi, Kanagawa, JAPAN	
P1-16. Robotics Education under Pandemic Lockdown Situation.....	496
Vicente González ¹ , Kelvin Kung ¹ , Danilo Cáceres-Hernández ^{1,2} and Kang-Hyun Jo ³	
¹ Facultad de Ingeniería Eléctrica, Universidad Tecnológica de Panamá, Panamá, Panamá	
² Sistema Nacional de Investigación (SNI), SENACYT, Panamá, Panamá	
³ Intelligent Systems Laboratory, Graduate School of Electrical Engineering, University of Ulsan	

P1-17. Lane Detection using Canny Edge Detection Algorithm for Real-time Racing Game	504
Sehar Shahzad Farooq ¹ , Hameedur Rahman ² , Samiya Abdul Wahid ² , Iftikhar Ahmad ¹ , Jin Ho Lee ¹ and Soon Ki Jung ¹	
¹ School of Computer Science and Engineering, Kyungpook National University, Daegu, South Korea	
² Department of Computer Games Development, Faculty of Computing and AI, Air University, Islamabad, Pakistan	
P1-18. Influence Analysis of Each Facial Region on Facial Expressions Recognition	516
Minsol Park ¹ and Inseop Na ²	
¹ Dept of Computer Engineering, Chosun University, Gwangju, Korea	
² National Program of Excellence in Software Centre, Chosun University, Gwangju, Korea	
Poster Session 2	
<i>When & Where: February 21st, 12:20 ~ 14:30, 2F Greenwich Hall</i>	
<i>Chair: Prof. Jeong Hieyong</i>	
P2-1. Diffuse Large B-cell Lymphoma Survival Prediction using Encoding Clinical Features	520
Sy-Phuc Pham ¹ , Sae-Ryung Kang ² , Hyung-Jeong Yang ¹ , Deok-Hwan Yang ² , Sudarshan Pant ¹ , Soo-Hyung Kim ¹ and Guee-Sang Lee ¹	
¹ Chonnam National University, Gwangju, South Korea	
² Chonnam National University Hwasun Hospital, Gwangju, South Korea	
P2-2. Robust Data Augmentation for Accurate Human Pose Estimator	528
Tien-Dat Tran ¹ , Xuan-Thuy Vo ¹ , Adri Priadana ¹ , and Kang-Hyun Jo ¹	
¹ School of Electrical Engineering, University of Ulsan, Ulsan, South Korea	
P2-3. Multi-task model for glioma segmentation and isocitrate dehydrogenase status prediction using segmentation boundary	540
Xiaoyu Shi ¹ , Yinhao Li ¹ , Yen-Wei Chen ¹ , Jingliang Cheng ² , Jie Bai ² and Guohua Zhao ²	
¹ Graduate School of Information Science and Engineering, Ritsumeikan University, Shiga, Japan	
² The Affiliated Hospital of Zhengzhou University, Zhengzhou, China	

P2-4. Impression Estimation of Suit Patterns Based on Style Features Using Multi-scale CNN..... 554

Eiki Tsumura¹, KesenkeTobitani², Miyuki Toga¹ and Noriko Nagata¹

¹Kwansei Gakuin University, Gakuen, Sanda-shi, Hyogo, Japan

²University of Nagasaki, Manabino, Nagayo-cho,
Nishi-Sonogi-gun, Nagasaki, Japan

P2-5. A Cascaded structure of Pre-trained Convolutional Neural Network for weed classification..... 569

GwangHyun Yu¹, Dang Thanh Vu¹, JaeCheol Jeong², ChilWoo Lee³
and JinYoung Kim¹

¹Department of ICT Convergence System Engineering,
Chonnam National University, Gwangju , Korea

²Department of Biomedical Engineering,
Chonnam National University Hospital, Korea

³Department of Computer Information and Communication,
Chonnam National University, Gwangju, Korea

P2-6. Two-stream Network for Moving Object Detection_Wisan Dhammatorn..... 577

Dhammatorn Wisan¹, Naoshi Kaneko¹, Seiya Ito¹ and Kazuhiko Sumi¹

¹Aoyama Gakuin University, Japan

P2-7. Multimodal Transformer for Automatic Depression Estimation System..... 585

Dang-Khanh Nguyen¹, Guee-Sang Lee¹, Soo-Hyung Kim¹, Hyung-Jeong Yang¹,
Seung-WonK im¹, MinJhon², and Joo-Wan Kim³

¹Department of AI Convergence, Chonnam National University, Gwangju,
Republic of Korea

²Department of Psychiatry, Chonnam National University Hwasun Hospital,
Gwangju, Republic of Korea

³Department of Psychiatry, Chonnam National University Hospital,
Gwangju, Republic of Korea

P2-8. Motion synthesis for automatic animation of sign language..... 591

Jong Ho Jeong¹, Hee Jae Hwang¹, Hong Nyeom Sung¹ and Chil Woo Lee²

¹Department of Computer Engineering, Chonnam National University,
Korea

²Department of Software Engineering, Chonnam National University, Korea

P2-9. Cattle Action Recognition with Multi-Viewpoint Cameras based on Deep Learning.....	604
Muhammad Fahad Nasir ^{1,2} , Alvaro Fuentes ^{1,2} , Shujie Han ^{1,2} , Jongbin Park ^{1,2} , Sook Yoon ³ and Dong Sun Park ^{1,2}	
¹ Department of Electronics Engineering, Jeonbuk National University, Jeonju, South Korea	
² Core Research Institute of Intelligent Robots, Jeonbuk National University, Jeonju, South Korea	
³ Department of Computer Engineering, Mokpo National University, Muan, South Korea	
P2-10. Convolutional Neural Networks with Particle Swarm Optimization A Reliable Method for SARS-CoV-2 Detection in X-Ray Images.....	614
Waqas Ahmed ¹ , Atif Ali ² , Farrukh Lodhi ¹ , Waqar Ahmed ¹ and Naveed Baloch ³	
¹ Deptdepartment of Computer Science University of Engineering and Technology, Taxila, Pakistan	
² Research Management Centre (RMC), Multimedia University, Cyberjaya, Malaysia	
³ Department of Computer Engineering University of Engineering and Technology, Taxila, Pakistan	
P2-11. Multi-region based radial GCN algorithm for real-time action recognition.....	620
Han-Byul Jang ¹ and Chil-Woo Lee ¹	
¹ Chonnam National University, Gwangju, Republic of Korea	
P2-12. Advanced Machine Learning Techniques To Identify Emotions In Texts.....	633
Atif Ali ¹ and Zulqarnain Farid ²	
¹ Research Management Centre (RMC), Multimedia University, Cyberjaya, Malaysia.	
² Dept of Criminology, University of Karachi, Pakistan	
P2-13. Object Pose Estimation Based on Template-matching Using Attention Module and Residual Block.....	636
Gaeun Noh ¹ and Jong-Il Park ¹	
¹ Department of Computer Science, Hanyang University, Seoul, Republic of Korea	
P2-14. COVID -19 Detection based on CT Scan Images using Deep Learning Methods.....	643
Tuan Le Dinh ¹ , Jae-Hyun Kim ¹ , Suk-Hwan Lee ² and Ki-Ryong Kwon ¹	
¹ Pukyong National University, Busan, South Korea	
² Donga University, Busan, South Korea	

- P2-16. Change Detection Over Multispectral Images A Case Study On RUSHIKONDA..... 649**
- Shaik Fyzulla¹, Chitturi S Pavan Kumar¹,
Chintakayala Pavan Veera Nagendra Kumar¹ and Punukollu Surya Prakash¹
- ¹Department of Information Technology,
Velagapudi Ramakrishna Siddhartha Engineering College,
Vijaywada, Andhra Pradesh, India
- P2-17. Gaussian Process based Illumination Planning for Photometric Stereo..... 659**
- Yuji Oyamada¹
- ¹Tottori University, Japan
- P2-18. Data Generation and Deep Learning network for Micro Defect Detection.... 669**
- Byungjoon Kim¹ and Yongduek Seo¹
- ¹Sogang University, Seoul, South Korea
- P2-19. Classifying Breast Cancer Using Deep Convolutional Neural Network Method_Musfequa Rahman..... 675**
- Musfequa Rahman¹, Kaushik Deb¹ and Kang Hyun Jo²
- ¹Department of Computer Science and Engineering,
Chittagong University of Engineering & Technology (CUET), Chattogram,
Bangladesh
- ²Department of Electrical, Electronic and Computer Engineering,
University of Ulsan
- P2-20. Rough Target Region Extraction with Background Learning..... 687**
- Ryo Nakamura¹, Yoshiaki Ueda¹, Masaru Tanaka¹ and Jun Fujiki¹
- ¹Fukuoka University, Fukuoka Nanakuma, Japan

Hierarchical Image Classification with Conceptual Hierarchies Generated via Lexical Databases

Tomoaki Yamazaki^{*1}, Seiya Ito^{*1}, and Kouzou Ohara¹

Aoyama Gakuin University, Japan

d5620003@aoyama.jp, {s.ito,ohara}it.aoyama.ac.jp

^{*}Equal Contribution

Abstract. Hierarchical image classification is the task of classifying images using hierarchical information. Conventional methods use hierarchies only during training and classify the fine classes in the hierarchy. In this paper, we focus on the hierarchical information utilizing not only during training but also during inference and evaluation. To verify the effectiveness of hierarchical information, we propose a simple architecture that explicitly utilizes it during training and inference. We also introduce hierarchy-based evaluation metrics. The key idea of designing the architecture is to obtain outputs for all nodes of a given hierarchy and integrate them to predict hierarchical classes. To this end, we explore neural network architectures that do not require a hierarchy tailored to the network. In our evaluation metrics, to analyze the characteristic of the hierarchical image classification model in more detail, superior concepts on the hierarchy of fine classes as well as fine classes are subject to evaluation. Furthermore, we present a new method for generating concept hierarchies via a lexical database. We empirically verify the effectiveness of the method that combines the proposed network architecture with the generated concept hierarchy.

Keywords: image classification · hierarchical classification · lexical database · conceptual hierarchy.

1 Introduction

Thanks to deep learning, image classification models have made great strides over the past decade. In general, an image classification model consists of a feature extractor that extracts features from an image and a classifier that recognizes the class as one of the predefined classes. As image features are important for classification, a variety of convolutional neural network- (CNN) [6, 13] and Transformer-based [3, 9] architectures have been proposed for feature extractors. In contrast, simple architectures (e.g., one linear layer) are often used for the classifier. In such a classifier, the classes to be recognized are treated as flat, making it difficult to discriminate among classes with similar image features.

Besides, the similarity of image features does not correspond to semantic similarity. Specifically, even the state-of-the-art method sometimes classifies “couch” as “bed”, even though “chair” and “couch” are semantically close.

To prevent such false classification, several methods have been proposed that utilize the hierarchical structure of the classes [5, 16]. However, since most methods only use hierarchies for training, it is not guaranteed that predictions are performed based on hierarchical information. Moreover, their concept hierarchies are manually constructed to fit the classification models. From this perspective, two questions arise: 1) Is it appropriate not to use hierarchical information during inference? 2) Is it adequate to adjust the conceptual hierarchy according to the classification model architecture?

To address these questions, we propose a network that is simple yet capable of handling complex hierarchies. The key idea is to obtain outputs for nodes of a hierarchy (hierarchical classes) from image features and integrate them along the hierarchy to predict classes. The proposed integration scheme depends only on the number of nodes in the hierarchy. Unlike previous hierarchical classification methods, we do not need to adjust the hierarchy to the model architecture. Thus, we also propose a method to generate a concept hierarchy for classification via a lexical database based on class labels. Since the concept hierarchy in the lexical database contains a lot of information unrelated to classes, we construct a compact concept hierarchy by eliminating unimportant nodes.

We evaluate the proposed method with respect to how it predicts classes along the hierarchy as well as general classification accuracy. The proposed network is capable of predicting classes other than the fine classes as final predictions. In practical situations, it is sometimes more valuable to predict classes at a coarse level that is not wrong than to fail to classify classes at a fine level. For example, there are cases in which it is sufficient to recognize a “chair” or “couch” as a “seat” without distinguishing between them or to recognize “furniture” without even distinguishing between seats and tables. We define metrics focusing on the concepts to which the predictive and correct answer classes belong, and report on the effectiveness of the proposed method.

To summarize, the contributions of this paper are threefold:

- We tackle the task of classifying images based on a conceptual hierarchy and propose a network that can handle complex hierarchical structures.
- We propose a novel method for constructing compact concept hierarchies for hierarchical image classification from fine-grained class labels by referencing a lexical database.
- We present an evaluation method for image classification based on a conceptual hierarchy and show the results of a detailed analysis of hierarchical image classification.

2 Related Work

2.1 Hierarchical Image Classification

Hierarchical image classification is an image classification task in which hierarchical information related to the classes is given in addition to the image [16]. Hierarchy information is generally defined in a tree structure with classes as nodes and can be utilized to improve image classification accuracy with the same classification model architecture. The leaf nodes of the tree have a high level of detail and are called fine classes, while the other nodes are called coarse classes. For example, CIFAR-100 [8], a commonly used benchmark for hierarchical classification, has 20 coarse (super) classes, and each class is associated with five fine classes (i.e., the coarse class “people” has five fine classes: “baby”, “boy”, “girl”, “man”, and “woman”), for a total of 100 fine classes. However, many datasets do not contain hierarchical information, so hierarchical information needs to be manually created in order for hierarchical image classification [16].

Hierarchical image classification models predict coarse as well as fine classes for each image. A typical model is based on the idea that features are represented differently depending on the depth of the layers (blocks) of the CNN and predicts the class of each hierarchy using the features of the blocks according to the depth of the hierarchy. Branch Convolutional Neural Network (B-CNN) [16], one of the representative methods, employs VGG-16 [13] as a feature extractor and divides it into three blocks to predict three hierarchical classes. B-CNN uses separate classifiers to identify each level of hierarchy and improves fine class accuracy by auxiliary use of coarse class classification loss. However, B-CNN is limited in the hierarchical structure it can handle because it is necessary to determine the number of feature extractor blocks according to the hierarchical structure. To seek hierarchy in accordance with model architecture, some hierarchy generation algorithms such as group assignment [7, 12] and hierarchy k -means [4] have been proposed. However, these algorithms do not generate a semantically interpretable tree structure like WordNet [10].

In the literature, the concept closest to our approach was proposed by Guo et al. [5]. They presented a network architecture called CNN-RNN that combines a CNN and a Recurrent Neural Network (RNN). In CNN-RNN, the RNN takes the features of the final layer of the CNN as input and predicts all classes in the hierarchy at each time step once. While CNN-RNN can handle arbitrary hierarchies, it ignores hierarchical structures during prediction. In this paper, we explore methods that can handle arbitrary hierarchies in prediction.

2.2 Hierarchy Generation

Generating hierarchy from a lexical database such as WordNet [10] consists of the following steps [14, 15]:

1. Select words from the collection.
2. Get one hypernym path from each selected word.

3. Build the tree from hypernym paths.
4. Compress the tree by eliminating a parent that has fewer than a specific number of children.

Following the above procedure, we first select words with the appropriate concepts from the class labels in the dataset. In the second step, because the most lexical database is not the tree structure, [14] adopts a heuristic approach that selects the first sense of hypernyms, but we often struggle to determine whether hypernym is appropriate for the dataset. Since the concepts selected in the second step influence the subsequent process of tree construction and compression, it is difficult to choose one hypernym without looking at the tree carefully. In this paper, we determine the importance of nodes based on the tree and eliminate nodes with less importance while updating the tree.

3 Methodology

This study aims to design a hierarchical classifier that can handle arbitrary hierarchies. If the final prediction is not restricted to fine classes but includes hierarchical classes, there are multiple possible ways to predict classes based on hierarchy. To this end, we explore hierarchical classifier architectures. Note that the proposed architecture consists of an arbitrary image feature extractor and a hierarchical classifier as in general image classification models, so a fixed-dimensional feature vector obtained by a feature extractor is an input to the hierarchical classifier.

In this section, we first explain how to construct a concept hierarchy from a lexical database in Sec. 3.1 and then the weighting of hierarchical classes for hierarchical classification in Sec. 3.2. We then discuss the model designs and loss function for any hierarchy in Sec. 3.3.

3.1 Conceptual Hierarchy Generation

We propose a new hierarchy generation algorithm that determines a parent (hypernym) by its importance while comparing it to a tree in generating process. To determine the parent of a node, the importance of a node n is defined as the number of paths N_n^{root} that go through a node n from the leaves to the root (referred to as “root paths”). To avoid deleting important nodes coincidentally, all nodes are sorted by the number of root paths N_n^{root} . Moreover, each time the tree is significantly updated, we recalculate the number of root paths N_n^{root} . The proposed algorithm is as follows:

1. Select a concept (synset) for each class in the dataset.
2. Build the tree, calculate the number of root paths N_n^{root} for each node n , and initialize the target number of root paths r_t as 1.
3. Keep iterating until a r_t is no longer the path of the root.
 - (a) Get target nodes whose root path is equal to r_t and sort the leaves of the tree in descending order of depth.

- (b) Compress the tree by eliminating a parent that has fewer than k children recursively and select a parent for each target node with many root paths.
- (c) Recalculate the number of root paths N_n^{root} for every node.
- (d) Update r_t as $r_t + 1$ if there is no change in the tree through processing the target nodes.

The first step is to manually assign synsets to the dataset if no synsets are assigned. The second step is the initialization that builds the tree from hypernym paths of selected synsets and calculates the root path of all nodes. The root paths of leaves are always one, but in the initialization, the root path of the root is not always the number of classes because some nodes have multiple parents. In the third step, it keeps updating the tree until the target root path r_t reaches the root path of the root. In other words, there is no change in the tree when checking all nodes. In the recursive process, we first obtain target nodes with less important nodes of unprocessed nodes and sort their order by the depth in descending order. Second, we eliminate a parent that has fewer than k children unless the parent is the root. This process is repeated as long as the importance of a parent is the same as that of the target node. At this time, if the node has multiple parents, we select a parent by the number of importance. If parents have the same importance, we select a parent with deeper depth and a smaller synset ID in a lexical database. In image classification, a fine class is not necessarily a tree leaf, as in the case of a chair being considered a piece of furniture. To account for this, if the parent of a fine class is another fine class, it is aligned horizontally and linked to the descendants of the fine class. After updating the tree by processing the target nodes, all root paths are recalculated. If there are no updates, the target number of root paths is incremented so that the root path of the root equals the number of classes eventually.

3.2 Weights for Hierarchical Classes

In a hierarchical structure, all parents inherited from a certain fine class are positive for that fine class (e.g., furniture for a chair). In contrast, siblings of a certain fine class and classes around inherited parents are negative for the fine class (e.g., a desk whose parent is also furniture, but the desk is not a chair).

Based on this idea, we assign weighted positive labels and negative labels to every fine class. Specifically, the positive weights for the parents of the fine class are obtained by a function that takes the distance from the fine class as input. Meanwhile, we compute negative weights for the siblings of a parent, i.e., the children of that parent, by a function with positive weights and the number of children as input. The overall algorithm is shown in Algorithm 1.

In this algorithm, it is necessary to define functions to compute positive and negative weights. Intuitively, the deeper the depth in hierarchical classification, the harder it is to discriminate. For this reason, we define a function that increases the weights with depth:

$$weight_{(+)}(d) = \frac{1}{1 + \log(d + 1)} \quad (1)$$

Algorithm 1 Hierarchical class weighting

Input: \mathcal{N} : nodes that have one parent node and multiple child nodes, n_r : a root node ($n_r \in \mathcal{N}$), n_f : a fine node ($n_f \in \mathcal{N}$)
Output: $\mathbf{w}_{(+)}^f, \mathbf{w}_{(-)}^f$: positive and negative weights

- 1: Initialize positive and negative weights except for the root:
 $\mathbf{w}_{(+)}^f \leftarrow zeros(|\mathcal{N}| - 1)$, $\mathbf{w}_{(-)}^f \leftarrow zeros(|\mathcal{N}| - 1)$
- 2: Initialize target node and depth: $t \leftarrow n_f$, $d \leftarrow 0$
- 3: **while** t is not n_r **do**
- 4: Calculate positive weights: $\mathbf{w}_{(+)}^f(t) \leftarrow weight_{(+)}(d)$
- 5: **for** $c \in children(parent(t)) \setminus t$ **do**
- 6: Calculate negative weights:
 $\mathbf{w}_{(-)}^f(c) \leftarrow weight_{(-)}(weight_{(+)}(d), N_t^{children})$
- 7: **end for**
- 8: $t \leftarrow parent(t)$
- 9: $d \leftarrow d + 1$
- 10: **end while**

Designing a function with negative weights, we note that it may interfere with learning if the weights are larger than the positive weights at the same depth or even their parents' positive weights. Therefore, we calculate the negative weights such that they are less than the positive weights with respect to the positive weights at the same depth:

$$weight_{(-)}(w_{(+)}, s) = w_{(+)} / s \quad (2)$$

where s is a specific number.

3.3 Network Architecture

For cases where the final prediction target is not limited to fine-grained classes but includes hierarchical classes, there are three classifiers for predicting classes based on hierarchy. The first is to predict all hierarchical classes flat as in [5]; the second is to follow the hierarchy top-down; and the third is to follow the hierarchy bottom-up. We hereafter describe how to predict classes using hierarchies. Note that the dimension of the final output is the number of nodes excluding the root (i.e., the total number of hierarchical classes) for all networks.

All-at-Once Classifier The first is an all-at-once classifier, which uniformly predicts all hierarchical classes in a given hierarchy. Let x and f^{all} represent the input image and network, respectively. The probability distribution over hierarchical classes \mathbf{p} are calculated as follow:

$$\mathbf{p} = \sigma(f^{all}(x)) \quad (3)$$

where σ is the softmax function.

Algorithm 2 Top-Down classifier

Input: x : an input image, \mathcal{N} : nodes, n_r : a root node, f^{td} : a network
Output: \mathbf{p} : probability distribution over hierarchical classes

```

1:  $\mathbf{w} \leftarrow f^{td}(x)$ 
2: Initialize target node and depth:  $q \leftarrow \text{queue}()$ ,  $q.\text{push}(n_r)$ 
3: while  $q$  is not empty do
4:    $t \leftarrow q.\text{pop}()$ 
5:   for  $c \in \text{children}(t)$  do
6:      $\mathbf{w}(c) \leftarrow \mathbf{w}(c) + \mathbf{w}(t) * N_c^{root} / N_t^{root}$ 
7:      $q.\text{push}(c)$ 
8:   end for
9: end while
10:  $\mathbf{p} \leftarrow \sigma(\mathbf{w})$ 
```

Top-Down Classifier The second is a top-down classifier, which predicts hierarchical classes by adding the output of the higher level nodes to the lower level nodes. Similar to the all-at-once classifier, the top-down classifier outputs initial scores for all hierarchical classes from image features but updates the scores following Algorithm 2.

Bottom-Up Classifier The third is a bottom-up classifier, which predicts hierarchical classes by adding the output of the lower level nodes to the higher level nodes. This is almost the reverse process of the top-down classifier, as detailed in Algorithm 3. Since a general classifier that predicts only detailed classes can be viewed as a bottom-up classifier, we will show the results of applying Algorithm 3 to a general classifier in our experiments.

Loss Function Using outputs of the hierarchical classifier \mathbf{p} , positive weights $\mathbf{w}_{(+)}$, and negative weights $\mathbf{w}_{(-)}$, the loss function is defined as the sum of the positive and negative errors:

$$\mathcal{L} = \mathcal{L}_{(+)} + \mathcal{L}_{(-)} \quad (4)$$

where

$$\mathcal{L}_{(+)} = -\log(\mathbf{p}) * \mathbf{w}_{(+)} \quad (5)$$

$$\mathcal{L}_{(-)} = -\log(1 - \mathbf{p}) * \mathbf{w}_{(-)} \quad (6)$$

4 Experiments

4.1 Implementation Details

Dataset We evaluate the proposed method on CIFAR-10 and CIFAR-100 [8]. Both datasets consist of 60,000 images of size 32×32 , split into 50,000 for training

Algorithm 3 Bottom-up classifier

Input: x : an input image, \mathcal{N} : nodes, n_r : a root node, f^{bu} : a network
Output: \mathbf{p} : probability distribution over hierarchical classes

```

1:  $\mathbf{w} \leftarrow f^{bu}(x)$ 
2: Initialize target nodes and depth:  $q \leftarrow priority\_queue()$ 
3: for  $n$  in  $\mathcal{N}$  do
4:   if  $is\_fine\_class(n)$  then
5:      $q.push((-depth(n), n))$ 
6:   end if
7: end for
8: while  $q$  is not empty do
9:    $(d, t) \leftarrow q.pop()$ 
10:   $\mathbf{w}(parent(t)) \leftarrow \mathbf{w}(parent(t)) + \mathbf{w}(t)$ 
11:  if  $t$  is not  $n_r$  then
12:     $q.push((-d + 1, parent(t)))$ 
13:  end if
14: end while
15:  $\mathbf{p} \leftarrow \sigma(\mathbf{w})$ 
```

and 10,000 for testing. CIFAR-10 contains 10 classes, including bird, dog, and car, whereas CIFAR-100 has 100 classes, including more fine-grained classes than CIFAR-10. For data augmentation, we apply zero-padding by 4 pixels, random crop of size 32×32 , horizontal flip with 50% probability, and random rotation with the range of $[-15^\circ, 15^\circ]$.

For hierarchical image classification, we prepare three hierarchies for each dataset: one is defined by [16] (manual), and the others are generated by our method described in Sec. 3.1. We basically use WordNet [10] as the lexical database, and for unknown concepts, we refer to BabelNet [11], which is automatically constructed by integrating WordNet and Wikipedia’s lexical and encyclopedic knowledge, as an auxiliary database. For example, in CIFAR-100, the class “aquarium fish” has no counterpart in WordNet. Thus, we referenced BabelNet to obtain a higher level concept for “aquarium fish”, i.e., “fish”¹. In the hierarchy generation process, we set the hyperparameter k for tree compression, which represents the number of children of a node. In our experiments, k was set to 1 and {2, 3} for CIFAR-10 and CIFAR-100, respectively. The generated hierarchies are illustrated in Fig. 1 and Fig. 2.

Training In our experiments, we employ VGG-16 [13] with batch normalization (BN) as an image feature extractor and initialize with pretrained weights by ImageNet [1]. We follow the hyperparameter settings as in [2]. We train all models with 200 epochs with a mini-batch size of 128. We use SGD optimizer with momentum, and the initial learning rate, momentum, and weight decay are set

¹ In fact, the higher level word for “aquarium fish” on BabelNet is fish with the ID “bn:13520875n”, but “bn:13520875n” did not have a link to WordNet. Therefore, we replaced “bn:13520875n” with “bn:00034816n” manually.

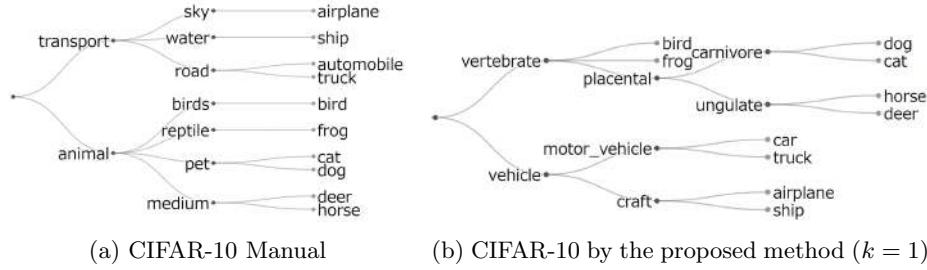


Fig. 1: Conceptual hierarchies on CIFAR-10. (a) Manually constructed by [16] on CIFAR-10. (b) Generated by our method with $k = 1$ on CIFAR-10.

to 0.1, $5e - 4$, and 0.9, respectively. The learning rate is decayed by 0.2 at 60, 120, and 160 epochs.

4.2 Evaluation Metrics

Fine Accuracy Since hierarchical image classification in this study does not necessarily classify into fine classes, we refer to the accuracy in general image classification as fine accuracy (FA). FA is the percentage of matches between the predicted class and the correct class when the predicted class is restricted to one of the fine classes.

Common Concept Ratio When restricting the prediction class to the fine classes, we evaluate whether the prediction class and the correct answer class contain common concepts. In a given conceptual hierarchy, we define the common concept ratio (CCR) as the percentage of cases where at least one node other than the root node is present between the path from the root to the predicted class and the path from the root to the correct class. This is an upper bound for prediction at the appropriate hierarchy and is never less than FA.

Depth-Weighted Hierarchical Classification Accuracy For practical use, it is necessary to determine at which hierarchy to predict hierarchical classes, but the way to do this is nontrivial. To measure how accurately the predictions follow the hierarchy, we define depth-weighted hierarchical classification accuracy (DWHCA):

$$DWHCA(\hat{\mathcal{Y}}, \mathcal{Y}) = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \frac{|\hat{\mathcal{Y}}_i \cap \mathcal{Y}_i|}{|\hat{\mathcal{Y}}_i|} \quad (7)$$

where \mathcal{I} is an evaluation dataset consisting of images and labels, $\hat{\mathcal{Y}}$ is a list that has a set of predicted labels, \mathcal{Y} is a list that has a set of the target fine class and whose hypernyms. This metric is inspired by multilabel classification and implies precision weighted by the depth of the hierarchy.

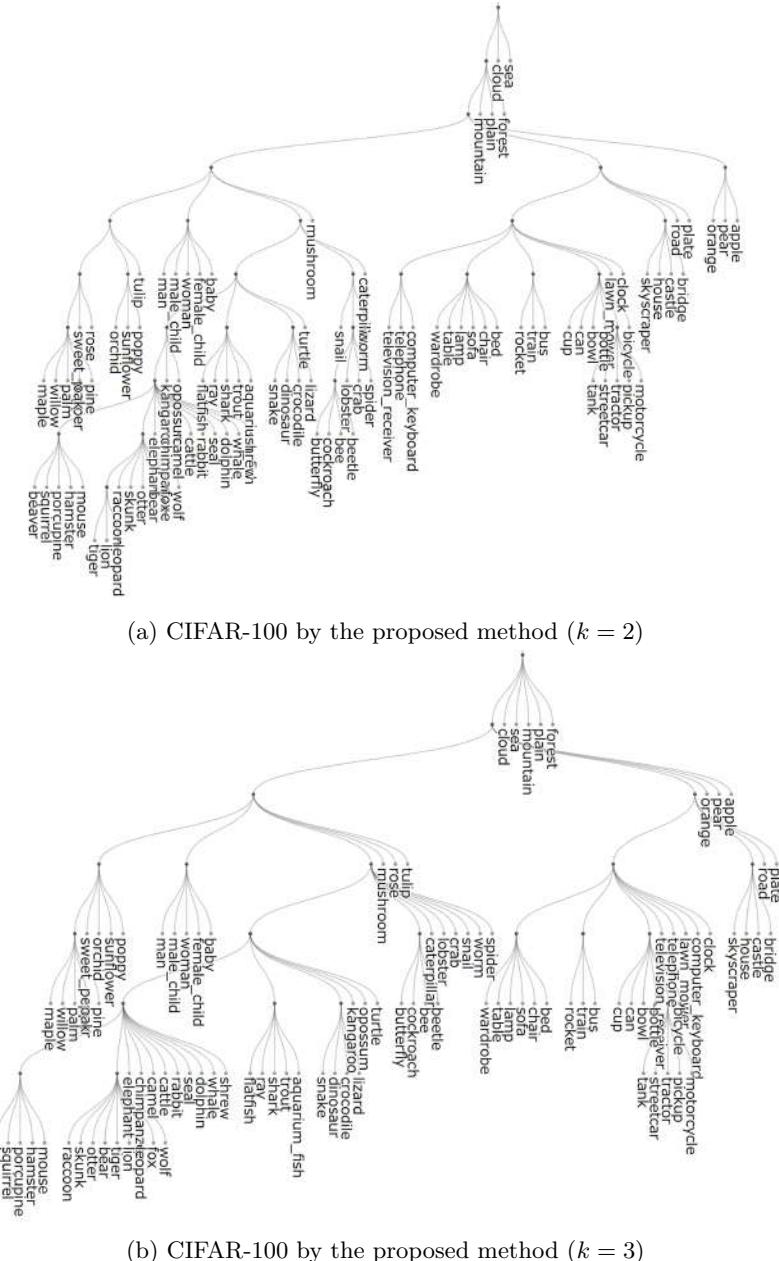


Fig. 2: Conceptual hierarchies on CIFAR-100. (a) and (b) are generated by our method with $k = 2, 3$ on CIFAR-100, respectively.

Table 1: Fine accuracy on CIFAR-10 and CIFAR-100.

Method	CIFAR-10	CIFAR-100
VGG-16 [13]	87.96	63.04
VGG-16 w/ BN* [13]	92.94	72.59
B-CNN [16]	88.22	64.42
CF-CNN [12]	-	65.11
Ours (manual)	93.04	72.86
Ours	93.16	72.81

*Our implementation

4.3 Comparison to Previous Methods

In Table 1, we compare the proposed method with our baseline models [13] and previous methods that use hierarchies [12, 16] with FA. In the proposed method, we show the results of the same manual hierarchy as the B-CNN [16] and the best of the hierarchies generated by the proposed method. While our goal is to predict classes following a given hierarchy, our method achieved results comparable to or better than the baseline in both cases. This suggests that the hierarchy does not need to be constructed manually.

4.4 Ablation Study

We design three hierarchical classifiers and weightings for the hierarchical classes. We conduct ablation experiments on these and report the hierarchical classification performance as shown in Table 2. In Table 2, we denote all-at-once, top-down, and bottom-up classifiers as ALL, TD, and BU, respectively. In the experiment, for each node in the path from the fine class predicted by the hierarchical classifier to the root, the CCR and DWHCA are calculated by determining the most likely hierarchical class among it and its siblings.

The proposed method produces results comparable to or better than the baseline for all metrics. In particular, the method that applies hierarchical class weighting to the bottom-up hierarchical classifier is better overall. However, it is observed in all-at-once and top-down classifiers that the negative weights in the proposed hierarchical label weighting degrade several metrics, meaning that it follows the hierarchy less well than the positive weights alone. This is because negative weights are computed based on a bottom-up scheme, which is helpful for bottom-up classifiers but not compatible with the others.

5 Conclusion

We have explored network architectures for hierarchical image classification along conceptual hierarchies. Unlike previous work, we do not need to construct hierarchies manually, and our hierarchical classifier, which uses the hierarchies

Table 2: Ablation study on network architecture and hierarchical class weighting using CIFAR-100.

Model	manual			Ours ($k = 2$)			Ours ($k = 3$)		
	FA	CCR	DWHCA	FA	CCR	DWHCA	FA	CCR	DWHCA
VGG-16 w/ BN	72.59	88.60	81.11	72.59	98.83	86.79	72.59	97.02	84.87
Ours-ALL	72.45	89.79	81.61	72.65	98.67	87.20	72.81	96.74	85.26
Ours-ALL-w/neg	72.86	89.87	81.80	72.57	98.61	87.08	72.54	96.65	85.20
Ours-TD	72.62	89.89	81.76	72.58	98.64	87.14	72.58	96.64	85.21
Ours-TD-w/neg	72.52	89.52	81.49	72.32	98.69	86.98	72.52	96.72	85.15
Ours-BU	72.23	89.19	81.37	72.16	98.88	86.86	72.45	97.09	85.05
Ours-BU-w/neg	72.77	89.97	81.68	72.23	98.90	87.77	72.77	97.17	85.62

obtained by the proposed hierarchy generation method, achieved results comparable to or even superior to manually constructed hierarchies. Extensive experiments with various backbones and datasets are left for future work.

Acknowledgement

This work was supported by JSPS KAKENHI Grant Number JP22K17978, and a part of this work was conducted as a project of JST SPRING Grant Number JPMJSP2103.

References

1. Deng, J., Dong, W., Socher, R., Li, L., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). pp. 248–255 (2009)
2. Devries, T., Taylor, G.W.: Improved Regularization of Convolutional Neural Networks with Cutout. arXiv preprint arXiv:1708.04552 (2017)
3. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In: Proceedings of the 9th International Conference on Learning Representations (ICLR) (2021)
4. Guo, Y., Xu, M., Li, J., Ni, B., Zhu, X., Sun, Z., Xu, Y.: HCSC: Hierarchical Contrastive Selective Coding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9696–9705 (2022)
5. Guo, Y., Liu, Y., Bakker, E.M., Guo, Y., Lew, M.S.: CNN-RNN: a large-scale hierarchical image classification framework. Multimedia Tools and Applications **77**(8), 10251–10271 (2018)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016)

7. Kim, J., Park, Y., Kim, G., Hwang, S.J.: SplitNet: Learning to Semantically Split Deep Networks for Parameter Reduction and Model Parallelization. In: Proceedings of the 34th International Conference on Machine Learning (ICML). Proceedings of Machine Learning Research, vol. 70, pp. 1866–1874 (2017)
8. Krizhevsky, A.: Learning Multiple Layers of Features from Tiny Images. Tech. rep. (2009)
9. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9992–10002 (2021)
10. Miller, G.A.: WordNet: A Lexical Database for English. Communications of the ACM **38**(11), 39–41 (1995)
11. Navigli, R., Ponzetto, S.P.: BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artificial Intelligence **193**, 217–250 (2012)
12. Park, J., Kim, H., Paik, J.: CF-CNN: Coarse-to-Fine Convolutional Neural Network. Applied Sciences **11**(8), 3722 (2021)
13. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. In: Bengio, Y., LeCun, Y. (eds.) Proceedings of the 3rd International Conference on Learning Representations (ICLR) (2015)
14. Stoica, E., Hearst, M.A.: Nearly-Automated Metadata Hierarchy Creation. In: Proceedings of HLT-NAACL 2004: Short Papers. pp. 117–120. Boston, Massachusetts, USA (2004)
15. Stoica, E., Hearst, M.A., Richardson, M.: Automating Creation of Hierarchical Faceted Metadata Structures. In: Proceedings of the Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL). pp. 244–251 (2007)
16. Zhu, X., Bain, M.: B-CNN: Branch Convolutional Neural Network for Hierarchical Classification. arXiv preprint arXiv:1709.09890 (2017)

Action Recognition for Each Person with Feature Extraction by Large-scale Object Detector

Akira Mitsuoka¹ and Kunihiro Kato¹

¹ Gifu University, 1-1 Yanagase, Gifu City, Gifu 501-1193, Japan
mitsuoka@cv.info.gifu-u.ac.jp

Abstract. In this study, we focus on human-object interaction (HOI) detection. In previous studies of HOI detection, information about person and objects was given at training phase. However, since there are countless objects, it is difficult to identify and teach objects that are relevant to actions. Given this fact, this research aims to be able to detect HOI based only the positions and actions of people. To achieve this, we attempted to introduce knowledge about objects implicitly by re-training the feature extractor of a large-scale object detector.

Keywords: Action Recognition, Human-Object-Interaction, Object Detection

1 Introduction

The advancements in deep learning have led to the resolution of numerous problems in computer vision. However, identifying individuals and recognizing their actions in video images remains underdeveloped. In particular, the recognition accuracy for actions that capture interactions between people and objects needs to be improved [1]. This study focuses on these actions. The ability to accurately identify these actions can lead to advancements in recognition of complex actions [2]. In human-object interaction detection, not only information related to people but also information related to objects is important. Conventional human-object interaction detection assumes that the positions and classes of both people and objects are taught during the learning process. However, due to the vast number of objects, it is difficult to identify and teach action-related objects in practical applications. With this in mind, we aim to detect human-object interactions only from the positions and actions of people.

To address this issue, we propose utilizing the features of a pre-trained large-scale object detector. A large-scale object detector such as Detic [3], which learns object detection from the ImageNet-21k dataset, enables the detection of objects across a wide range of classes. The feature extractor of Detic implicitly knows features that can identify various object positions and shapes. By re-training this feature extractor for person-by-person action detection, we implicitly introduce the knowledge of object positions and shapes and attempt to detect human-object interactions. The results of experiments using the dataset we created show that the feature extractor trained on 21,000 classes with Detic performs better in capturing human-object interactions than other feature extractors and weights. Furthermore, the experiments revealed that the operation of

Global Average Pooling (GAP) applied to the feature map is essential for recognizing human-object interactions.

The contributions of this paper can be summarized into two points:

1. We propose a model that effectively detects human-object interactions by utilizing knowledge of objects possessed by a large-scale object detector, without using information on object positions and classes during the learning process.
2. In the framework above, GAP applied to wide range of feature map is essential for recognition accuracy.

2 Related Works

2.1 Object Detection

Object detection is a problem predicting the position and class of objects in an image. Early detection models [4,5] using deep learning set up several anchor boxes tied to anchors on feature maps. These anchor boxes are considered correct when the IoU with the Bounding Box from the teacher data exceeds a certain threshold, and are used for training. These anchor boxes are generally in the thousands, so the problem was that it was easy to generate overlapping Bounding Boxes. To solve this problem, detection methods that use object keypoints [6,7,8] instead of anchor points were developed. Literature [6,7] embeds information such as object size at the center position of the object. CornerNet [8] embeds information at the object's top left and bottom right corner and groups them. These keypoint-based methods are simple and fast, but they ignore that the optimal keypoint position changes depending on the shape and occlusion of the object. Therefore, research on dynamic assignment methods [9,10,11] has been active in recent years. ATSS [9] assigns GT based on statistical features. OTA [10] approaches the assignment of predicted bounding boxes as an optimal transport problem. TOOD [11] trains so that the position of the embedded object classification branch and object localization branch get closer. Dynamic assignment methods have achieved improved accuracy on datasets such as COCO but require some additional computational cost. Based on the experimental observation that the localization accuracy is sufficient for the intended dataset, we use CenterNet[6], one of the keypoint detectors.

Large-scale object detection is generally defined as detecting more than 1000 object classes. In this problem setting, the class label distribution has a long-tail problem, and many studies are trying to solve it by addressing this problem. EQL [12] calculates the number of classes per total number of images in the dataset and weights the loss function with this value. Seesaw Loss [13] introduces two coefficients: a relaxation coefficient that reduces loss for classes with fewer numbers and a compensation coefficient that works to increase loss for False Positives, and learns by balancing these coefficients. Federated Loss [14] uses random sampling of class sets as learning data for each iteration and ignores the classes that were not sampled. In contrast, Detic tries to solve it from a different perspective by using class labels of images as additional data. In the field of existing research for learning detection from classification labels, there are weakly supervised object detection and semi-supervised object detection. The

distinction between these and Detic is that Detic completely separates the learning of classifiers and object locators by only training the classifier using image data. This is based on the result that the locator itself is sufficiently generalized even for object classes that are not used in training [15]. By separating the training, detection improves accuracy for classes with fewer numbers, which was difficult to detect conventionally.

2.2 Action Recognition

In the context of action recognition, a task that aims to predict a person's location and action class is known as Spatio-Temporal Action Recognition (SAR). Typical methods [16, 17, 18] for SAR apply a pre-trained detector such as Faster-RCNN [4] to the feature maps obtained by a representative 3D CNN such as I3D [19] or SlowFast [20] and use the resulting regions of interest (RoI) for action recognition. Literature [21] argues that, instead of using feature maps, extracting people from the original video is more effective than using RoI. Furthermore, a method using tubelet [22] attempts to solve SAR by connecting detection results for each frame based on the detection score in the temporal direction. Methods that use graph representation [18, 23] generally define the embedding of information about people and objects as nodes and the edges connecting them as actions. While the introduction of graph representation in datasets such as AVA has achieved a certain level of accuracy improvement, there is the problem that the computational cost increases as the number of objects to be detected increases. Approaches that attempt to model the Spatio-temporal relationship between people and objects without using graph representation, such as [1, 24, 25, 26], can be mentioned. Many of these models require additional detection models in addition to the action recognition model. The main point of differentiation between our model and existing research is that our model attempts to perform action detection in a single model.

Human-Object-Interaction detection aims explicitly to capture spatial interactions between people and objects. Existing research in this field typically assumes that information about both person and objects is provided during training. HOI detection methods can be broadly divided into two-stage methods and one-stage methods. Two-stage methods [27,28,29] first extract features related to people and objects using a backbone network, similar to typical methods for Spatio-temporal action recognition. The extracted features are then used in multiple streams, such as object streams, person streams, and streams for capturing relationships, which are then combined to detect HOI. One-stage methods in literature [30,31,32,33] aim to solve this problem by not using external detectors. Literature [30,31] utilizing the mechanism of CenterNet defines the center point of the object, the person and the interaction, and matches these three points for HOI detection. UnionDet [32] detects Bounding Box surrounding the human and object to obtain HOI. QPIC [33] focuses on the expressiveness of DETR [34] and develops HOI detector based on DETR. These HOI detectors also need to predetermine the object related to the action among countless objects.

3 Method

3.1 Detic feature extractor

In order to utilize knowledge of various object positions and shapes, we employ the feature extractor of the large-scale object detector Detic. The feature extractor of Detic is composed of Swin-base [35] and FPN [36]. Swin is a general backbone network for images developed based on the success of ViT [37]. One of the differences between Swin and ViT is that Swin has a hierarchical downsampling structure. In ViT, the scale of tokens in the spatial direction is the same at all levels, but in images, the scale of objects can change due to various factors. Therefore, it may be problematic to fix the scale of tokens when performing object detection or semantic segmentation. By adopting the conventional CNN-like hierarchical downsampling structure, Swin allows changing the spatial scale of tokens and thus handling various object scales. Another difference is the presence of Shifted Window based Self-Attention. In ViT, the calculation of Self-Attention is proportional to the square of the image size, and the size of this calculation is a problem. With Shifted Window-based Self-Attention, the image is divided into a fixed-size window, and Self-Attention is calculated within it. Thus, the calculation is linear for the image size. Also, the connectivity between windows is provided by shifting the window at each layer. FPN is a top-down structure network with horizontal connections, in contrast to the bottom-up structure of Swin. By adopting FPN, feature extraction becomes more scale-invariant. We use only the topmost output of Detic FPN for subsequent action detection.

3.2 Proposed Model

We have constructed an action detection model as shown in figure 1, using the aforementioned feature extractor. Let $\mathbf{X} \in \mathbb{R}^{H \times W \times 3}$ be the input to the model, the output of the Detic feature extractor is $\mathbf{X}' \in \mathbb{R}^{H/8 \times W/8 \times 256}$, where H and W represent the width and height of the input image. \mathbf{X}' is then upsampled to obtain the feature map $\mathbf{F} \in \mathbb{R}^{H/4 \times W/4 \times 256}$. \mathbf{F} is input to the Heatmap Head, Offset Head, Size Head, and Action Head, respectively. The Heatmap Head, Offset Head, and Size Head are parts for person detection, based on CenterNet. The Heatmap Head, Offset Head, and Size Head respectively learn a heatmap representing the object's central position, an offset of the object center caused by downsampling, and the object's size. The Heatmap Head learns by mapping the feature map passed through the Sigmoid function to the teacher heatmap. The Offset Head and Size Head learn by selecting the vector corresponding to the person's center and mapping it to the teacher data. CenterNet is simple and fast, and can achieve sufficient accuracy in the intended detection problem; that is why we use it.

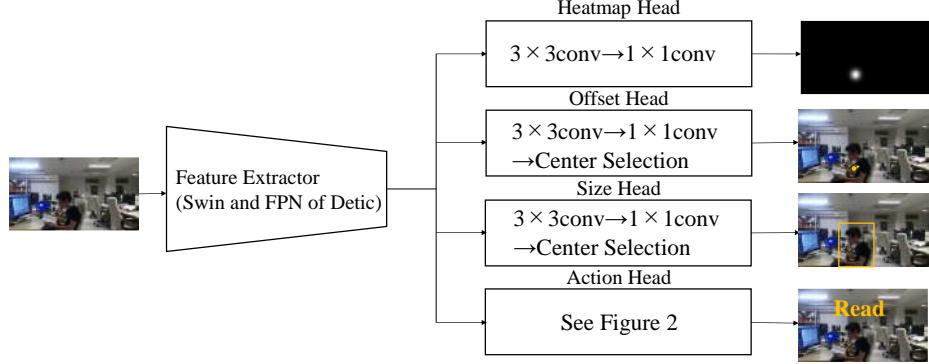
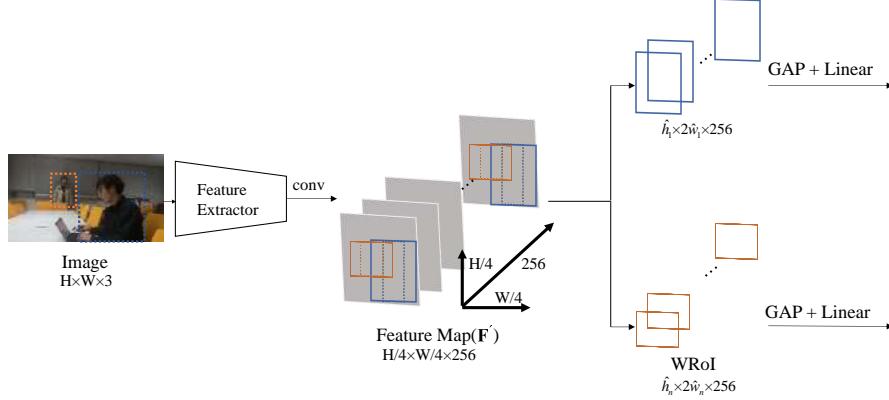


Fig. 1. Overview of the proposed model

The details of the Action Head are shown in Figure 2. In Figure 2, the dotted square represents the human region, while the solid square symbolizes the WRoI region to be described subsequently. Due to empirical observation, the action Head does not use center embeddings. In the Action Head, first, a single 3x3 convolution is applied to the feature map \mathbf{F} to obtain the feature map \mathbf{F}' . We extract detection candidate $\hat{P}_i = \{(\hat{x}_1, \hat{y}_1, \hat{x}_2, \hat{y}_2)\}_{i=1}^N$ using Detection Head for the feature map \mathbf{F}' . $\hat{x}_1, \hat{y}_1, \hat{x}_2$, and \hat{y}_2 are the x -coordinate of the upper left corner, y -coordinate of the upper left corner, x -coordinate of the lower right corner, and y -coordinate of the lower right corner of the bounding box, respectively. N is the number of persons detected. This RoI range often does not include objects related to the HOI, which is a problem during recognition. Therefore, by correcting \hat{x}_1 and \hat{x}_2 using equation (1), we obtain WRoI with a size of $\hat{h} \times 2\hat{w}$ for each person by cropping the feature map. Through the WRoI, the model can consider objects around the person. We obtain the feature map $\mathbf{F}_A \in \mathbb{R}^{N \times C}$ by applying GAP and linear layers to the WRoI, where C is the number of action classes, and \mathbf{F}_A is mapped to the teacher data to train the Action Head. Comparative experiments for Action Head are discussed in more detail in Section 4.3.

$$\hat{w} = \hat{x}_2 - \hat{x}_1, \hat{x}_1 = \max(0, \hat{x}_1 - \frac{\hat{w}}{2}), \hat{x}_2 = \min(\frac{W}{4}, \hat{x}_2 + \frac{\hat{w}}{2}) \quad (1)$$

**Fig.2.** The details of the Action Head

3.3 Loss Function

The loss function is based on Uncertainty Weight Loss [38]. Multi-task learning strongly depends on the relative weighting of the loss for each task, and it is not easy to adjust this weight manually. By using Uncertainty Weight Loss, it is possible to adjust the relative weight according to the task by making the learning parameter. When expressing the loss function of the Heatmap Head, Offset Head, Size Head, and Action Head respectively as L_H , L_O , L_S , and L_A , the total loss function L is represented by the following equation (2). Here, p_D and p_A are learning parameters. We use Focal loss [39] for L_H , L1 Loss for L_O and L_S , and Binary Cross Entropy Loss for L_A

$$L_D = L_H + 0.1L_S + L_O$$

$$L = \frac{1}{2} \left(\frac{1}{e^{p_D}} L_D + \frac{1}{e^{p_A}} L_A + p_D + p_A \right) \quad (2)$$

4 Experiments

To verify the effectiveness of the proposed method in HOI detection, we conducted multiple experiments on the dataset we created. In all experiments, we used Adam as the optimizer and a batch size of 4. The learning rate was explored in the range of $[1 \times 10^{-5}, 1 \times 10^{-4}]$ and the optimal learning rate was applied. The input images were resized to $H=896$, $W=896$ and. Also, we adopted random horizontal flipping and HSV color transformation for data augmentation. We trained on the dataset described later for 20 epochs with this setting. As a note, we conducted experiments in which the learning rate was changed for the feature extractor and other parts and experiments in which the feature extractor part was fixed. However, results that improved accuracy were not obtained.

4.1 Dataset and Evaluation Metrics

In order to create a dataset for experiments, we filmed inside a university laboratory at 1920×1080 and 10fps. We extracted frames from the video data as images, and for each person in each frame, we assigned a Bounding Box and a label of the action class to the person. The action classes selected as typical examples of HOI in the laboratory include "Reading" (Read), "Writing" (Write), "Drinking" (Drink), "Calling" (Call), "Operating a keyboard" (Keyboard), "Operating a mouse" (Mouse), "None of the above classes" (Nothing). Multiple of these action classes may correspond to a person. When collecting data, we made variations in the shooting location and tools used within the dataset to include different variations of the same action class. The details of the action class are shown in Table 1.

In existing action recognition datasets, it is often possible to determine the action class based solely on the pose of the person, the background, or the presence or absence of objects related to the HOI. Figure 3 illustrates an example from the V-COCO dataset [43]. Both Figure 3(a) and Figure 3(b) have labels of the action class "work on computer" for each person in the image. By observing the V-COCO dataset, the label "work on computer" is given if computers exist in the image. Conversely, if the label "work on computer" is not given, it can be confirmed that computers do not exist.

Thus, machine learning models can determine the action class just by identifying the presence or absence of computers in the image, regardless of whether the person is working on the computer. Furthermore, existing datasets often have the person partially cut off, which makes learning for action detection unnecessarily challenging. Figure 3(b) is a prime example, with only the hands of the person in the image. In contrast, our dataset was carefully crafted to ensure that if the person's position, pose, object position, and object class cannot all be accurately determined, the action class will not be appropriately assigned. Figures 4(a) and 4(b) are examples of our dataset. Both images contain a keyboard, but only the individual on the left operating the keyboard is labeled as "Keyboard". Additionally, our dataset was constructed such that person's upper torso is visible throughout our dataset. This avoids complicating the localization aspect unnecessarily and focuses model's attention on HOI.



Fig3(a)



Fig3(b)

Fig.3. Example of V-COCO dataset [43]



Fig4(a)

Fig4(b)

Fig.4. Example of our dataset**Table 1.** The details of our dataset

	Read	Write	Drink	Call	Key-board	Mouse	Nothing
Train	4,816	1,210	204	815	1,016	909	2,519
Val	1,518	260	94	123	616	611	1,109
Test	2,220	279	121	131	920	300	448

As evaluation metrics, we employ Micro-F1-Accuracy and mAP. We use Micro-F1-Accuracy to evaluate only the accuracy of action recognition in our validation data. On the other hand, mAP is a commonly used evaluation metric for object detection and action detection. We use mAP to verify both localization and recognition accuracy on the test data. We use a threshold of 0.7 for F1-Accuracy and 0.5 for IoU in mAP to determine detection as correct.

4.2 Comparison of Feature Extractors

We compared feature extractors to demonstrate the effectiveness of Detic feature extractor in recognizing HOI. The results are shown in Table 2. We used AlexNet [40] and DLA34 [41] for comparison with traditional CNN-based feature extractors, MViT [42] and Swin for comparison with transformer-based feature extractors, and Swin+FPN for comparison with other weights. We used AlexNet and DLA34 with ImageNet-1k pre-trained models, Swin and MViT with ImageNet-21k pre-trained models, Swin+FPN with LVIS pre-trained model, and Detic with LVIS-COCO and ImageNet-21K pre-trained models.

Table 2 shows that the Swin-based models are superior to the other models in terms of accuracy. This result indicates that Swin is an effective feature extractor in images. Next, among the Swin-based models, Detic shows the best accuracy. This result is because Detic has been trained to detect a wide range of object classes and thus has knowledge about object position and shape compared to other feature extractors.

Table 2. Comparison of feature extractors in our validation data

Feature Extractor	F1-Accuracy
AlexNet[40]	0.381
DLA34[41]	0.491
MViT[42]	0.510
Swin[35]	0.558
Swin+FPN	0.575
Detic[3]	0.718

4.3 Comparison of Action Head

In order to investigate the optimal structure in Action Head, we conducted experiments comparing this part by fixing the feature extractor to Detic. The results are shown in Table 3. In the table, Center refers to cases where information is embedded in the center, like CenterNet. GAP (RoI) [16] refers to the result of applying RoI Pooling [4] to the feature map \mathbf{F}' and then applying GAP. ACRN [1] attempts to improve the action classification accuracy without object information. ACRN first obtains the feature map \mathbf{A} by replicating the feature map \mathbf{F}' per individual, and the feature map \mathbf{I} by expanding it to the spatial dimensions of \mathbf{A} following the application of GAP to the RoI. ACRN calculates the concatenation of \mathbf{A} and \mathbf{I} , then applies several convolutions to this feature map and finally applies GAP. NL[26] applies Non-Local Block to feature map \mathbf{F}' and then applies GAP to the resulting feature map.

From Table 3, firstly, it can be seen that Center is unsuitable for HOI recognition. It also shows that GAP (WRoI), which provides a broader area, and GAP (Feature Map), which provides the entire feature map, are more suitable than GAP (RoI), which provides only a part of the feature map. These results suggest that HOI recognition requires looking at a wide area of the image. Also, ACRN and NL, which perform operations on RoI or feature map \mathbf{F}' , did not improve accuracy compared to GAP(RoI) or GAP(Feature Map). This result suggests that the operation of GAP applied to the feature map is essential. Also, it is believed that the reason for not improving the accuracy further is that the Detic feature extractor has already established correlations between individuals and objects.

Table 3. Comparison of Action Head structure in our validation data

Action Head	F1-Accuracy
Center	0.466
ACRN[1]	0.591
NL[26]	0.598
GAP(RoI)[16]	0.611
GAP(WRoI)	0.718
GAP(Feature Map)	0.712

4.4 Detection Result

Figure 5 demonstrates instances in which the proposed model successfully detects HOI. As shown in Figures 5(a), 5(b), and 5(c), the keyboard and mouse, which are objects relevant to HOI, are present, but only in Figure 5(a) where the person is operating the keyboard and mouse does the model output 'Keyboard' and 'Mouse.' This confirms that the proposed model recognizes HOI by considering not only the position of the person but also their pose, the position of the object, and the class of the object. Furthermore, Figures 5(d), 5(e), and 5(f) depict scenes captured from different locations but still demonstrate the ability of the model to recognize HOI accurately. This indicates that the proposed model can generalize to HOI, regardless of the capture location. Additionally, in Figure 5(d), the model can output 'Nothing' when there is no interaction with an object, further demonstrating its capabilities.



Fig. 5. Example of successful HOI detection by our proposed model

Figure 6 shows examples of the proposed model's failure in detecting HOI. Figure 6(a) is an example of over-detection, which is caused by insufficient learning of the detection task for the HOI recognition task. This problem can be resolved by sufficiently training the model, as the positions of the detections are not incorrect in many cases. Figure 6(b) is an example of incorrect detection of HOI, where the model identified incorrectly due to the right hand being close to the mouse and having a pose close to holding the mouse. Figures 6(c) and 6(d) are examples of HOI not being detected. Figure 6(c) fails to detect "Read" and "Keyboard", while Figure 6(d) fails to detect "Drink". This is due to a shortage of training data for scenes of one-handed keyboard operation and scenes with the "Drink" label, which can be resolved by increasing the relevant training data.

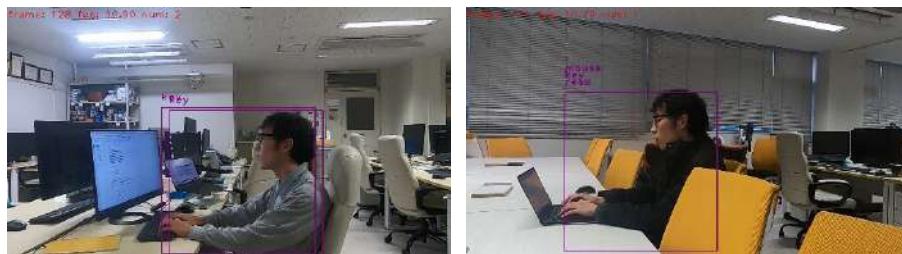


Fig.6(a)

Fig.6(b)



Fig.6(c)

Fig.6(d)

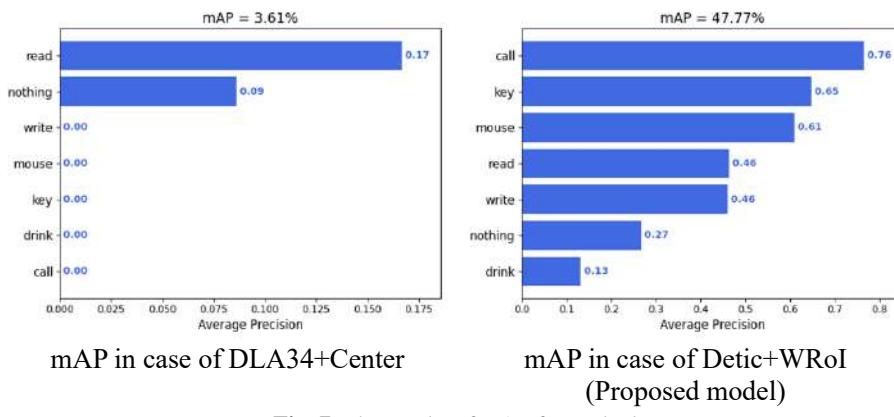
Fig. 6. Example of unsuccessful HOI detection by our proposed model

Table 4 shows the comparison of mAP on the test data. In terms of feature extractors, Detic achieved better results than other models and weights. Furthermore, in the Action Head, it was effective for all feature extractors to apply GAP to WRoI instead of Center. This confirms the effectiveness of the proposed method.

Figure 7 shows the results of mAP for each class. When using the DLA34 feature extractor and the Center action head, the model made random predictions for "Read" and "Nothing" while not predicting any other class. On the other hand, the proposed model could make predictions to some extent for each class. Also, it can be confirmed from the figure that the accuracy of "Drink" is poor throughout the dataset due to the lack of "Drink" samples in the training data. This problem can be addressed by improving the training dataset.

Table 4. Comparison of detection accuracy on our test data

Feature Extractor	Action Head	mAP
DLA34	Center	3.61
DLA34	WRoI	8.94
Swin + FPN	Center	6.92
Swin + FPN	WRoI	35.0
Detic	Center	11.4
Detic	WRoI	47.7

**Fig. 7.** The results of mAP for each class

5 Conclusion

In this study, we proposed to use the knowledge of a large-scale object detector's feature extractor to detect human-object interactions solely based on the position of people and the actions related to them. Through experiments on the created dataset, we confirmed that our proposed method can recognize HOI more accurately than using other feature extractors or weights. We also showed that the operation of GAP applied to the wide range of feature map is essential when recognizing HOI. Furthermore, the proposed model demonstrated the capability of performing individualized action recognition end-to-end.

Actions involve not only the spatial property of how a person interacts with an object but also the temporal property of how the person has acted. In this study, we did not consider the temporal aspect when recognizing actions. As a prospect, we aim to improve the detection of a broader range of action classes by associating HOI over time through tracking and by building a model based on this information.

References

1. Sun, C., Shrivastava, A., Vondrick, C., Murphy, K., Sukthankar, R. and Schmid, C.:Actor-centric relation network. Proceedings of the European Conference on Computer Vision, pp.318-334, (2018)
2. Sugimura, Y., Uchida, D., Suzuki, G. and Endou, T.:Proceedings of the Annual Conference of JSAl JSAI2020 (0).pp.4Rin157-4Rin157, (2020)
3. Zhou, X., Girdhar, R., Joulin, A., Krähenbühl, P. and Misra, I.:Detecting twenty-thousand classes using image-level supervision. Proceedings of European Conference on Computer Vision, pp.350-368, (2022)
4. Ren, S., He, K., Girshick, R. and Sun, J.:Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems (28), (2015)
5. Redmon, J. and Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, (2018)
6. Zhou, X., Wang, D. and Krähenbühl, P.: “Objects as points”, arXiv preprint arXiv:1904.07850. (2019)
7. Tian, Z., Shen, C., Chen, H. and He, T.: Fcos: Fully convolutional one-stage object detection. Proceedings of the IEEE/CVF international conference on computer vision, pp.9627-9636, (2019)
8. Law, H. and Deng, J.: Cornernet: Detecting objects as paired keypoints. Proceedings of the European conference on computer vision, pp.734-750. (2018)
9. Zhang, S., Chi, C., Yao, Y., Lei, Z. and Li, S. Z.: Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp.9759-9768. (2020)
10. Ge, Z., Liu, S., Li, Z., Yoshie, O. and Sun, J.: Ota: Optimal transport assignment for object detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 303-312. (2021)

11. Feng, C., Zhong, Y., Gao, Y., Scott, M. R. and Huang, W.: Tood: Task-aligned one-stage object detection. Proceedings of the IEEE/CVF International Conference on Computer Vision. pp.3490-3499. (2021)
12. Tan, J., Wang, C., Li, B., Li, Q., Ouyang, W., Yin, C. and Yan, J.: Equalization loss for long-tailed object recognition. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp.11662-11671. (2020)
13. Wang, J., Zhang, W., Zang, Y., Cao, Y., Pang, J., Gong, T., ... and Lin, D.: Seesaw loss for long-tailed instance segmentation. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 9695-9704. (2021)
14. Zhou, X., Koltun, V. and Krähenbühl, P.: Probabilistic two-stage detection. arXiv preprint arXiv:2103.07461. (2021)
15. Zareian, A., Rosa, K. D., Hu, D. H. and Chang, S. F.: Open-vocabulary object detection using captions. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,pp. 14393-14402. (2021)
16. Gu, C., Sun, C., Ross, D. A., Vondrick, C., Pantofaru, C., Li, Y. ... and Malik, J.: Ava: A video dataset of spatio-temporally localized atomic visual actions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition pp. 6047-6056. 2018
17. Girdhar, R., Carreira, J., Doersch, C. and Zisserman, A.: Video action transformer network. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp.244-253. (2019)
18. Zhang, Y., Tokmakov, P., Hebert M. and Schmid, C.: A structured model for action detection, Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9975-9984, (2019)
19. Carreira, J. and Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.6299-6308. (2017)
20. Feichtenhofer, C., Fan, H., Malik, J. and He, K.: Slowfast networks for video recognition. Proceedings of the IEEE/CVF international conference on computer vision, pp. 6202-6211. 2019
21. Wu, J., Kuang, Z., Wang, L., Zhang, W. and Wu, G.: Context-aware rcnn: A baseline for action detection in videos. Proceedings of European Conference on Computer Vision, pp.440-456. (2020)
22. Singh, G., Saha, S., Sapienza, M., Torr, P. H. and Cuzzolin, F.: Online real-time multiple spatiotemporal action localisation and prediction. Proceedings of the IEEE International Conference on Computer Vision, pp.3637-3646. (2017)
23. Wang, X. and Gupta, A.: Videos as space-time region graphs. Proceedings of the European conference on computer vision, pp.399-417. (2018)
24. Jiajun, T., Jin, X., Xinzhi, M., Bo, P. and Cewu, L.: Asynchronous interaction aggregation for action detection. Proceedings of European Conference on Computer Vision, pp.71-87, (2020)
25. Pan, J., Chen, S., Shou, M. Z., Liu, Y., Shao, J. and Li, H.: Actor-context-actor relation network for spatio-temporal action localization. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition pp.464-474, (2021)
26. Wang, X., Girshick, R., Gupta, A., and He, K.: Non-local neural networks. Proceedings of the IEEE conference on computer vision and pattern recognition, pp.7794-7803. 2018
27. Chao, Y. W., Liu, Y., Liu, X., Zeng, H. and Deng, J.: Learning to detect human-object interactions. Proceedings of IEEE winter conference on applications of computer vision, pp.381-389. (2018).

28. Gao, C., Xu, J., Zou, Y., and Huang, J. B.: Drg: Dual relation graph for human-object interaction detection. Proceedings of European Conference on Computer Vision, pp.696-712. (2020)
29. Gao, C., Zou, Y., and Huang, J. B.: ican: Instance-centric attention network for human-object interaction detection. arXiv preprint arXiv:1808.10437. (2018)
30. Liao, Y., Liu, S., Wang, F., Chen, Y., Qian, C. and Feng, J.: Ppdm: Parallel point detection and matching for real-time human-object interaction detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.482-490. (2020)
31. Wang, T., Yang, T., Danelljan, M., Khan, F. S., Zhang, X. and Sun, J.: Learning human-object interaction detection using interaction points. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.4116-4125. (2020)
32. Kim, B., Choi, T., Kang, J. and Kim, H. J.: Uniondet: Union-level detector towards real-time human-object interaction detection. Proceedings of European Conference on Computer Vision, pp. 498-514. (2020)
33. Tamura, M., Ohashi, H., and Yoshinaga, T.: Qpic: Query-based pairwise human-object interaction detection with image-wide contextual information. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.10410-10419. (2021)
34. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A. and Zagoruyko, S.: End-to-end object detection with transformers. In European conference on computer vision, pp. 213-229. (2020)
35. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., ... and Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10012-10022. (2021)
36. Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B. and Belongie, S.: Feature pyramid networks for object detection. Proceedings of the IEEE conference on computer vision and pattern recognition, pp.2117-2125. (2017)
37. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... and Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929. (2020)
38. Kendall, A., Gal, Y. and Cipolla, R.: Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp.7482-7491. (2018)
39. Lin, T. Y., Goyal, P., Girshick, R., He, K. and Dollár, P.: Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision pp. 2980-2988. (2017)
40. Krizhevsky, A., Sutskever, I. and Hinton, G. E. Imagenet classification with deep convolutional neural networks. Communications of the ACM, pp.84-90, (2017)
41. Yu, F., Wang, D., Shelhamer, E. and Darrell, T. Deep layer aggregation. Proceedings of the IEEE conference on computer vision and pattern recognition pp. 2403-2412. (2018)
42. Fan, H., Xiong, B., Mangalam, K., Li, Y., Yan, Z., Malik, J. and Feichtenhofer, C.: Multiscale vision transformers. Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6824-6835. (2021)
43. Saurabh, G and Jitendra, M.: “Visual semantic role labeling”, arXiv preprint arXiv:1505.04474, (2015)

Structural Point Cloud Data Recovery to Learning 3D Feature Representation

Ryosuke Yamada^{1,2}, Ryu Tadokoro², Yue Qiu²,
Hirokatsu Kataoka², and Yutaka Satoh^{1,2}

¹ University of Tsukuba, Japan

² National Institute of Advanced Industrial Science and Technology (AIST), Japan

{ryosuke.yamada, ryu.tadokoro, qiu.yue
hirokatsu.kataoka, yu.satou}@aist.go.jp

Abstract. Can we obtain 3D feature representations by reconstructing structural point clouds without real data? This paper aims to develop Point Cloud Perlin Noise (PCPN) for pre-training for 3D object recognition. PCPN is automatically generated based on the natural 3D structure in the real world. Using a simple formula that is based on Perlin noise, our proposed method can automatically generate more 3D patterns than conventional 3D datasets. In addition, we apply Point-MAE [1] to PCPN in order to construct a pre-trained model for improving the performance of downstream tasks. Through experiments, we demonstrate that our proposed method improves performance by 1.5% for ModelNet40 compared to the conventional 3D datasets used for pre-training. Our proposed pre-training strategy has also revealed the ability to achieve effective pre-training in 3D object recognition without real data and supervised labels.

Keywords: 3D object recognition, Point cloud, Self-supervised learning

1 Introduction

3D object recognition using point clouds has been extensively studied for autonomous driving and robotics applications. Point clouds can better represent real-world environments than 2D images, as it is not dependent on the camera's viewpoint. We can capture rich 3D data using high-precision sensors such as LiDAR systems, and therefore, 3D object recognition has received considerable attention [2,3,4].

However, one challenge associated with the construction of 3D point cloud datasets is the considerable amount of human effort required for the collection of 3D data and for annotation. Collecting 3D data requires 3D scanning or reconstruction from 2D images and annotation considering position and orientation in 3D space. Thus, constructing a 3D dataset demands more human effort than 2D datasets. In addition, missing 3D data collected from the real world often adversely affects the learning of 3D feature representation. Previous research has used 3D Computer Aided Design (CAD) models to solve these issues for

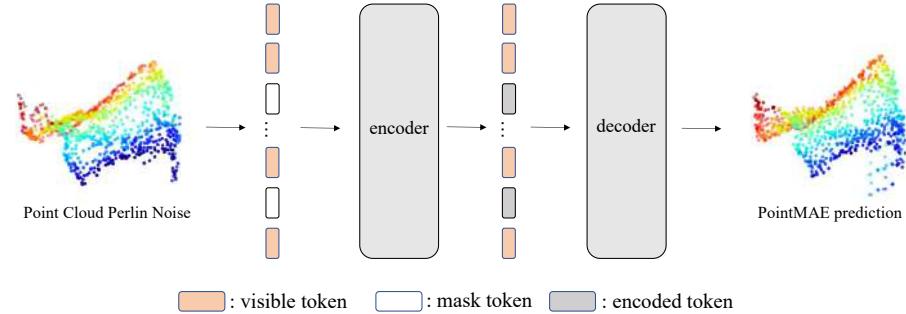


Fig. 1. Overview of the point cloud Perlin noise pre-training process. The pre-training converts the input point clouds with a random embedding (e.g., 60%) as mask tokens and other embeddings as visible tokens. The encoder processes only visible tokens. The mask tokens are added to the input sequence of the decoder to predict token and reconstruct the point clouds. The parameters learned in the pre-training are set as initial parameters for finetuning the downstream tasks.

training [5]. The advantage of this approach is the annotation is automatic and does not require data collection. However, we expect that increasing the size of a dataset in the future will be pretty costly.

Recently, increasing attention has been given to the use of artificially generated images and automatically generated labels for learning feature representations [7]. For example, the research strategy in “Learning to See by Looking at Noise” involves generating images according to specific rules and using self-supervised learning to learn visual feature representations from the images. Researchers have conducted studies using simple shapes with “Dead Leaves” and natural images generated with StyleGAN [6] and have successfully acquired superior feature representations compared to learning from random parameters. This framework can also help acquire feature representations in 3D object recognition and not just for 2D images.

In this paper, we propose an automatic 3D dataset construction method called Point Cloud Perlin Noise (PCPN), as see Figure 1. Which pre-training tasks should be designed using the PCPN? In 2023, a self-supervised learning method called point-MAE [1] was proposed. point-MAE is based on BERT [8], which is a pre-text task proposed in natural language processing. BERT masks words in sentences and restores the original sentences, whereas point-MAE deliberately creates masked data from input 3D point clouds and reconstructs the original data. Using Perlin Noise, it is possible to generate structured 3D data based on predefined rules. Therefore, we can acquire general 3D feature representations more easily than those observed in the real world. Because 3D scans from the real world initially include missing parts in the 3D data, these could be better for reconstruction tasks, such as those that use Point-MAE. However, since the 3D data generated from equations do not have missing parts in the 3D structure, the pre-training strategy of point-MAE would be optimized for PCPN.

2 Related Work

2.1 3D object recognition with point cloud

The notion of 3D object recognition using point clouds has attracted attention since the emergence of PointNet [2], which made it possible to input point clouds into neural networks directly. Since then, a large number of 3D object recognition methods that used Graph Neural Networks (GNN) [11,12] and Convolutional Neural Networks (CNN) [9,10] have been proposed to improve recognition accuracy. Transformer models have been drawing attention in 2023; however, they require a larger amount of training data because of their lower inductive bias. In 2D image recognition, Vision Transformer [13] achieved state-of-the-art performance by pre-training on the JFT-300M [14], which is a dataset with 300 million images and is not available to the public. However, constructing a 3D dataset is costlier than building a 2D image dataset. For 3D object recognition using point clouds, there are no large-scale datasets similar to JFT-300M. In this paper, we propose a framework for generating a theoretically infinite amount of 3D data based on the regularity of natural phenomena. Our proposed method improves the pre-training effectiveness of the Transformer-based network. We believe it is also possible to construct a large-scale pre-training dataset that can become a de-facto standard.

2.2 Self-supervised learning

The construction of a 3D dataset requires a huge amount of human effort. Therefore, supervised labels and training data are expected to be limited when applied to applications. From the perspective of real-world applications, it would be ideal for improving the performance of 3D object recognition with limited data. Self-supervised learning learns pre-text tasks from unlabeled 3D data to learn 3D feature representations while the annotation cost is reduced. In particular, self-supervised learning with 3D point clouds can be classified into two main categories: reconstruction tasks based on generative models [15,16] and contrast learning based on the similarity of the input data [17,18]. In the case of reconstruction tasks, the network is trained to reconstruct the parts of 3D objects that have been deliberately masked using an Encoder-Decoder network. In the case of contrastive learning, utilizes different viewpoints of 3D scenes, learning to identify correspondences between two point clouds. A pre-trained model learned by reconstruction tasks is used for object classification or segmentation tasks. Contrariwise, the contrastive learning approach is used for 3D object detection and 3D scene segmentation in 3D scenes.

3 Point Cloud PerlinNoise

In this section, we explain the automatic construction method of PCPN that is based on Perlin noise in Section 3.1. We describe the pre-training strategy that uses point-MAE in Section 3.2.

3.1 Point Cloud Perlin Noise

In this paper, we propose Point Cloud Perlin Noise (PCPN) based on the simplex noise method [22]. To construct the PCPN, we take two steps: (i) generating Perlin noise and (ii) projecting into a 3D space.

Generating Perlin noise: To generate the Perlin noise map, we divide a 2D space into a grid of equilateral triangles with a side length of 1. For a given coordinate (x, y) , we refer to the three vertices of the equilateral triangle grid that surround (x, y) as $P = \{(x_i, y_i)\}_{i=1}^3$. We use a hash function H to calculate the pseudo-random gradient vectors $(Gradx_k, Grady_k)$ for each vertex as follows:

$$(Gradx_k, Grady_k) = H(x_k, y_k) \quad (1)$$

We then define a function C that depends on the gradient at a coordinate (x_k, y_k) as follows:

$$f(x_k, y_k) = \max \left(0, \frac{1}{2} (1 - 2x_k^2 - 2y_k^2) \right)^4 \quad (2)$$

$$C(x_k, y_k) = (x_k Gradx_k + y_k Grady_k) f(x_k, y_k) \quad (3)$$

The noise N at a coordinate (x, y) in the 2D space is calculated as follows.

$$N(x, y) = \sum_{i=1,2,3} C(x_i, y_i) \quad (4)$$

Projecting into a 3D space: To project the Perlin noise map into a 3D space, we use the noise value at each coordinate on the 2D Perlin noise as the value on the z -axis and map the points in the 3D space. In this way, we can generate 3D point cloud based on Perlin noise.

3.2 Pre-training with Point-MAE

In this paper, we develop a pre-trained model by applying Point-MAE to PCPN. The Point-MAE reported the highest accuracy in self-supervised learning for 3D object recognition. We consider that PCPN can theoretically generate data infinitely and has no missing data, making it suited for reconstruction tasks such as Point-MAE. In the following, we introduce PointMAE's pre-training method. The pre-training process for Point-MAE involves three steps: (i) dividing into patches, (ii) masking and patch embedding, and (iii) autoencoder pre-training.

projecting into a 3D space: We downsample point clouds by using farthest point sampling and determine m center points. For each center point, we select k nearest neighbor points using the k-nearest neighbor (KNN) algorithm and group them into patches $PT \in \mathbb{R}^{m \times k \times 3}$. The point clouds in each patch overlap, and the coordinates within each patch are normalized by the center point.

Masking and patch embedding: We randomly select n out of the m patches created during Stage (i) and create masked patches $PT_{mask} \in \mathbb{R}^{n \times k \times 3}$. For unmasked patches $PT_{unmasked} \in \mathbb{R}^{(m-n) \times k \times 3}$, we use the lightweight Point-Net to convert them into $E_{unmasked} \in \mathbb{R}^{(m-n) \times C}$. C is the number of dimensions

of the tokens. The mask token is the shared weight token $TK_{mask} \in \mathbb{R}^{(m-n) \times C}$. The encoder and decoder are composed of standard transformer blocks. The positional embedding adopts the coordinates of the center points of each patch converted by a multi-layer perceptron (MLP), which is added to each transformer block in the encoder and decoder. In the encoder, $E_{unmasked}$ is the input, and the encoded token $TK_{unmasked} \in \mathbb{R}^{(m-n) \times C}$ is the output. Further, in the decoder, the output of the encoder $TK_{unmasked}$ and the masked token T_{mask} are combined as the input, and only T_{mask} is decoded to output the token $F_{mask} \in \mathbb{R}^{n \times C}$. By inputting F_{mask} into the MLP as the head, we obtain the reconstructed patch $PT_{pred} \in \mathbb{R}^{n \times k \times 3}$. During training, we minimize the loss function shown in the following equation:

$$L = \frac{1}{|PT_{pred}|} \sum_{p \in PT_{pred}} \min_{q \in PT_{mask}} \|p - q\|_2^2 + \frac{1}{|PT_{mask}|} \sum_{q \in PT_{mask}} \min_{p \in PT_{pred}} \|p - q\|_2^2 \quad (5)$$

4 Experiments

In this section, we verify the effectiveness of PCPN pre-training in 3D object recognition through comparative experiments. In Section 4.1, we explain the experimental setup. In Section 4.2, we describe the fine-tuning experiments for each downstream task.

4.1 Experiment setting

Pre-training setup. In this experiment, we follow Point-MAE and adopt the PointCloud Transformer [4] as the network. In this paper, we use PCPN and ShapeNet to pre-train Point-MAE. We divide PCPN into three subsets of data $\{1,000, 10,000, 100,000\}$ and evaluate them for fine-tuning performance. ShapeNet is a commonly used dataset in conventional self-supervised learning and consists of 55 object categories and 51,300 3D CAD models. We use AdamW as the optimizer and the cosine learning rate decay during pre-training. We set the initial learning rate to 0.001, weight decay to 0.05, and batch size to 256 and conducted the training for 300 epochs.

Fine-tuning setup. In the downstream task, we evaluate the pre-trained model with regard to object classification, few-shot learning, and part segmentation. For object classification, we use ModelNet40 [20] and ScanObjectNN [21] as the evaluation datasets. ModelNet40 is a single-object dataset consisting of 40 categories, 9,843 samples for train, and 2,468 samples for test. ScanObjectNN is a real-world dataset consisting of 15 categories, 2,312 samples for train, and 581 samples for test. We verify our proposed method on three subsets $\{\text{OBJ-BG}, \text{OBJ-ONLY}, \text{PB-T50-RS}\}$ of ScanObjectNN. The OBJ-BG contains the background around the object, while OBJ-ONLY is the 3D object without a background. The PB-T50-RS is a subset with translation, rotation (about the

Table 1. The comparison for PCPN and ShapeNet on the object classification.

Pre-train	ModelNet40	ScanObjectNN		
		OBJ-BG	OBJ-ONLY	PB-T50-RS
From scratch [16]	91.4	79.8	80.5	77.2
ShapeNet	92.1	83.5	86.9	87.4
PCPN-1k	92.8	84.2	87.2	82.5
PCPN-10k	92.9	80.2	81.9	82.4
PCPN-100k	92.5	80.7	81.4	82.2

Table 2. The comparison for PCPN and ShapeNet on few-shot learning.

Pre-train	5-way, 10-shots	5-way, 20-shots	10-way, 10-shots	10-way, 20-shots
From scratch [16]	87.8 \pm 5.2	93.3 \pm 4.3	84.6 \pm 5.5	89.4 \pm 6.3
ShapeNet	97.3 \pm 1.9	96.8 \pm 2.1	91.7 \pm 4.3	93.4 \pm 2.7
PCPN-100k	95.0 \pm 3.0	95.3 \pm 5.6	89.6 \pm 4.1	92.6 \pm 2.8

gravity axis), and scaling for all 3D objects. For few-shot learning, we randomly select K classes from ModelNet40 and sample $N + 20$ objects for each class. We train the model using the K -way \times N -shots subset and evaluate it using the 20K samples. In this study, we prepare ten subsets with $K=\{5, 10\}$ and $N=\{10, 20\}$ and evaluate the performance by averaging the highest accuracy for each subset. Part segmentation is a difficult task identifying finer class labels for all points of 3D models. We evaluate it using ShapeNetPart [19], which includes 16,881 models from 16 categories. Following previous studies, we sample 2,048 points as the input for each object and generate 128-point patches. We evaluate the performance using the mean Intersection over Union (mIoU) for all instances and the IoU for each category.

4.2 Downstream tasks

In this section, we explain the experimental results of the downstream tasks. We evaluate our proposed method by verifying it using benchmark datasets for object classification, few-shot learning, and part segmentation.

Object Classification. We investigate and evaluate the transfer learning performance by setting ModelNet40 and ScanObjectNN as the downstream datasets. Table 1 shows the experimental results for object classification. As can be seen in the table, an increase in the amount of data in PCPN leads to improved classification accuracy for ModelNet40 and ScanObjectNN. Furthermore, when comparing the PCPN-10k pre-trained model with scratch training, the classification accuracy improves by +1.5% for ModelNet40. This result confirms the effect of pre-training with PCPN. The classification accuracy was also improved over the scratch in ScanObjectNN. Furthermore, the performance is comparable to that of the ShapeNet pre-trained model. The results show that our proposed pre-training model is applicable to both synthetic and real data in the object classification task.

Table 3. Part segmentation results on the ShapeNetPart dataset. We report the mean IoU across all part categories mIoUC (%) and the mean IoU across all instances mIoUI (%), as well as the IoU (%) for each category.

	mIoUC	mIoUI	aero	bag	cap	car	chair	earphone	guitar	knife
			lamp	laptop	motor	mug	pistol	rocket	skateboard	table
From scratch [16]	83.4	85.1	82.9	85.4	87.7	78.8	90.5	80.8	91.1	87.7
			85.3	95.6	73.9	94.9	83.5	61.2	74.9	80.6
ShpeNet	85.5	83.1	83.1	85.1	87.0	77.9	90.6	79.8	91.1	86.3
			85.6	94.8	71.8	93.7	83.4	61.8	72.9	82.0
PCPN-100k	83.6	85.8	84.1	83.2	87.7	79.9	91.2	77.8	91.4	87.0
			85.7	95.9	73.8	94.6	83.8	61.6	76.6	82.0

Few-shot Learning. In the experimental results for object classification, we evaluate the PCPN-100k pre-trained model with regard to few-shot learning. As shown in Table 2, the PCPN-100k pre-trained model also demonstrates effectiveness for few-shot learning. Specifically, we compare it to learning from scratch in four subsets and confirm performance improvements of 7.2%, 2.0%, 5.0%, and 3.2%, respectively. This result suggests that the PCPN model pre-trained with PointMAE acquires a universal 3D feature representation that can quickly adapt to new downstream tasks without the use of any real data or manual annotations. However, the results were inferior for ShapeNet pre-trained model. We speculate that the factor is that ShapeNet is created from a 3D CAD model, which is the same 3D data domain as ModelNet40 for fine-tuning.

Part Segmentation. In this experiment, we evaluate the pre-trained model using the ShapeNetPart, which contains 16,881 objects from 16 categories. As shown in Table 3, the PCPN-100k pre-trained model achieved a mIoUI of 85.8%, outperforming the result from scratch training by 0.7% mIoU. Furthermore, the performance is equivalent to the ShapeNet pre-trained model. Based on these results, it can be inferred that the PCPN pre-trained model is also effective for more challenging tasks such as part segmentation.

5 Conclusion

This paper proposes a method for automatically constructing a 3D point cloud dataset (PCPN) without collecting 3D data or requiring human labeling. The experimental results show that the PCPN pre-trained model achieved a performance that was equivalent to the datasets used in conventional pre-training methods. Furthermore, our proposed method could improve the shortage of 3D data and the diversity of 3D datasets. Based on these results, we can confirm the effectiveness of the PCPN pre-training strategy. In this paper, we proposed PCPN to create a pre-trained model method as an initial consideration. However, we believe that designing the most effective learning method for PCPN will further affect pre-training in the future.

References

1. Pang, Yatian, et al., “Masked Autoencoders for Point Cloud Self-supervised Learning,” in *European Conference on Computer Vision (ECCV)*, 2022.
2. Qi, Charles R., et al., “Pointnet: Deep Learning on Point Sets for 3D Classification and Segmentation,” in *Computer Vision and Pattern Recognition (CVPR)*, 2017.
3. Qi, Charles R., et al., “Deep Hough Voting for 3D Object Detection in Point Clouds,” in *International Conference on Computer Vision (ICCV)*, 2019.
4. Zhao, Hengshuang, et al., “Point Transformer,” in *International Conference on Computer Vision (ICCV)*, 2019.
5. Yongming Rao, et al., “RandomRooms: Unsupervised Pre-training from Synthetic Shapes and Randomized Layouts for 3D Object Detection,” in *International Conference on Computer Vision (ICCV)*, 2021.
6. Karras, Tero, et al., “A Style-Based Generator Architecture for Generative Adversarial Networks,” in *Computer Vision and Pattern Recognition (CVPR)*, 2019.
7. Baradad Jurjo, Manel, et al., “Learning to See by Looking at Noise,” in *Neural Information Processing Systems (NeurIPS)*, 2021.
8. Devlin, Jacob, et al., “Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in *arXiv* , 2018.
9. Li, Yangyan, et al., “PointCNN: Convolution On X -Transformed Points,” in *Neural Information Processing Systems (NeurIPS)*, 2021.
10. Xu, Yifan, et al., “SpiderCNN: Deep Learning on Point Sets with Parameterized Convolutional Filters,” in *European Conference on Computer Vision (ECCV)*, 2018.
11. Shi, Weijing, et al., “Point-GNN: Graph Neural Network for 3D Object Detection in a Point Cloud,” in *Computer Vision and Pattern Recognition (CVPR)*, 2020.
12. Landrieu, Loic, et al., “Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs,” in *Computer Vision and Pattern Recognition (CVPR)*, 2018.
13. Dosovitskiy, Alexey, et al., “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” in *International Conference on Learning Representations (ICLR)*, 2021.
14. Sun, Cen, et al., “Revisiting Unreasonable Effectiveness of Data in Deep Learning Era,” in *International Conference on Computer Vision (ICCV)*, 2019.
15. Wang, Hanchen, et al., “Unsupervised Point Cloud Pre-training via Occlusion Completion,” in *International Conference on Computer Vision (ICCV)*, 2021.
16. Yu, Xumin, et al., “Point-bert: Pre-training 3D Point Cloud Transformers with Masked Point Modeling,” in *Computer Vision and Pattern Recognition (CVPR)*, 2022.
17. Xie, Saining, et al., “PointContrast: Unsupervised Pre-training for 3D Point Cloud Understanding,” in *European Conference on Computer Vision (ECCV)*, 2020.
18. Hou, Ji, et al., “Exploring Data-Efficient 3D Scene Understanding with Contrastive Scene Contexts,” in *Computer Vision and Pattern Recognition (CVPR)*, 2021.
19. Chang, X., Angel, et al., “ShapeNet: An Information-Rich 3D Model Repository,” in *arXiv*, 2015.
20. Wu, Z., et al., “3D ShapeNets: A Deep Representation for Volumetric Shapes,” in *Computer Vision and Pattern Recognition (CVPR)*, 2015.
21. Angelina, Mikaela, Uy, et al., “Revisiting Point Cloud Classification: A New Benchmark Dataset and Classification Model on Real-World Data,” in *International Conference on Computer Vision (ICCV)*, 2019.
22. Perlin, K. 2002. Noise hardware. In Course Notes from Siggraph 2002, Course 36: Real-Time Shading Languages, SIGGRAPH 2002, ACM.

Point Cloud Based Deep Molecular Pose Estimation for Structure-Based Virtual Screening

Ken Kariya¹[0000-0002-8981-7046], Go Irie¹[0000-0002-4309-4700], Ryosuke Furuta²[0000-0003-1441-889X], Yota Yamamoto¹[0000-0002-1679-5050], Shin Aoki¹[0000-0002-4287-6487], and Yukinobu Taniguchi¹[0000-0003-3290-1041]

¹ Tokyo University of Science, Tokyo, Japan
 4621508@ed.tus.ac.jp, goirie@ieee.org, {yy-yamamoto, shinaoki,
 taniguchi.yukinobu}@rs.tus.ac.jp
² The University of Tokyo, Tokyo, Japan
 furuta@iis.u-tokyo.ac.jp

Abstract. Structure-Based Virtual Screening (SBVS) is a computer-based simulation technique to streamline the drug design process by identifying candidate inhibitor structures that are likely to bind to a target protein based on the shape similarity of molecular structures. Specifically, the molecular surface is acquired as a point cloud and then the point cloud alignment method is applied; this method continues to be actively studied in computer vision. Since the protein binding site and the inhibitor are not the same objects, their shapes do not match perfectly with numerous gaps between them. Therefore, conventional point cloud alignment methods may provide unsatisfactory performance. This paper proposes an SBVS method based on shape features. Specifically, the proposed method comprises (i) docking simulation based on a learning-based alignment model that simultaneously estimates pose and expansion parameters and (ii) scoring by Truncated Chamfer Distance with expansion transformation. Experiments show that the proposed method yields faster and more accurate SBVS processing than previous methods using optimization of chemical properties.

Keywords: Inhibitor Retrieval · Structure-Based Virtual Screening · Point Cloud · Point Cloud Alignment

1 Introduction

Many diseases are triggered by the binding of a specific substrate (ligand) to an enzyme protein, causing a harmful chemical reaction. Drugs work by preemptively bind another ligand (inhibitor) to the enzyme protein to suppress the harmful chemical reaction. Therefore, identifying inhibitors that are likely to bind to a target enzyme protein is critical in the drug design process. However, conducting chemical experiments to identify candidate inhibitors from the huge library of possible ligands is highly inefficient. The experimental confirmation of binding to proteins is labor-intensive and expensive. To reduce this labor and

cost, computer-based simulations (virtual screening) is widely used. The goal of virtual screening is to streamline ligand identification through computer-based simulations.

There are two major types of virtual screening [12]. Ligand-Based Virtual Screening (LBVS) [15, 27, 6] and Structure-Based Virtual Screening (SBVS) [25, 1, 11]. LBVS searches for inhibitors that are similar to known ligands that bind to the target enzyme protein, whereas SBVS evaluates the likelihood of binding by the molecular structural similarity between the target enzyme protein and the ligands. In this paper, we focus on SBVS, which can be applied without prior knowledge of which ligands bind to the target enzyme protein.

There are several SBVS methods [16]. DOCK [1] has the longest history and uses scores based on physicochemical values such as van der Waals forces. In addition, empirical weighted scores and systematic search algorithms that avoid trying to search solutions in spaces known to lead to the wrong solution have been developed and used in Glide [11]. Recently, AutoDock Vina [25], which use an even more improved score function and a genetic algorithm, has become one of the most widely used methods. These methods compute scores for a large number of randomly generated poses during structure exploration, but the calculation time needed is excessive. In addition, shape similarity is evaluated using chemical properties such as van der Waals forces. However, using chemical properties may negatively affect the score, and examples include slight collisions between the inhibitor and the pocket that may cause score errors [26].

In this paper, we apply the point cloud alignment method, actively studied in computer vision, to SBVS to reduce computation time and to better evaluate shape similarity. We represent the molecular surfaces of the pocket (binding site) and the candidate inhibitor as a point cloud and align them by predicting the binding pose. Then we evaluate shape similarity by calculating the point cloud displacement needed to attain the predicted pose. To the best of our knowledge, this is the first proposed method for SBVS based only on shape from docking to scoring using point clouds. The proposed method can find the pose in which the inhibitor can fit into the pocket faster and achieve a higher accuracy in virtual screening. Since point cloud alignment has been extensively studied in recent years, we choose the point cloud as the shape representation.

There are two differences between the general point cloud alignment problem and point cloud alignment in SBVS (Fig. 2). The first difference is that SBVS alignment targets different objects, not the same object adopted in general point cloud alignment. The second difference is that gaps occur even when the inhibitor and pocket are bound in the correct position. This is because not only geometrical similarity but also chemical binding forces such as electrostatic forces contribute to the binding of the inhibitor and the protein. To adapt the point cloud-based approach to SBVS, we propose (i) a learning-based point cloud alignment method that simultaneously estimates the pose at the merge and the amount of expansion needed to fill the gap, and (ii) a scoring method using Truncated Chamfer Distance with expansion transformation. Experiments

on the DUD-E dataset [18] to evaluate virtual screening yield analytical results showing that the proposal is effective and superior to existing SBVS methods.

2 Related Work

2.1 Structure-Based Virtual Screening

Fig. 1 shows the processing pipeline of SBVS. (i) docking simulation: specify a query protein pocket and a candidate inhibitor, then predict the pose when the inhibitor binds to the pocket; (ii) scoring: calculate a score representing the stability of the predicted binding state; (iii) ranking: order the inhibitors based on their scores.

DOCK [1], Glide [11], and AutoDock Vina [25] are all methods that predict binding by solving a chemical energy minimization problem focusing on chemical

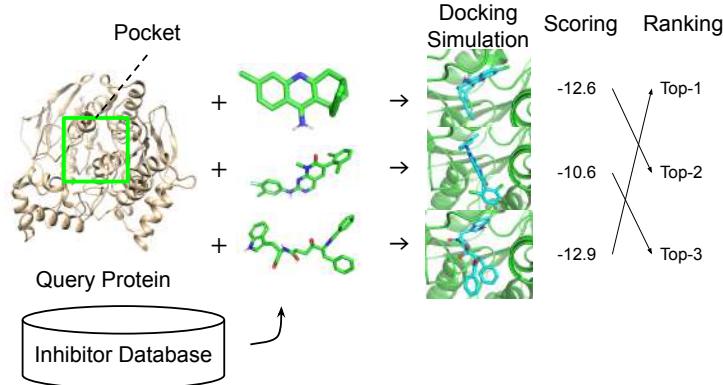
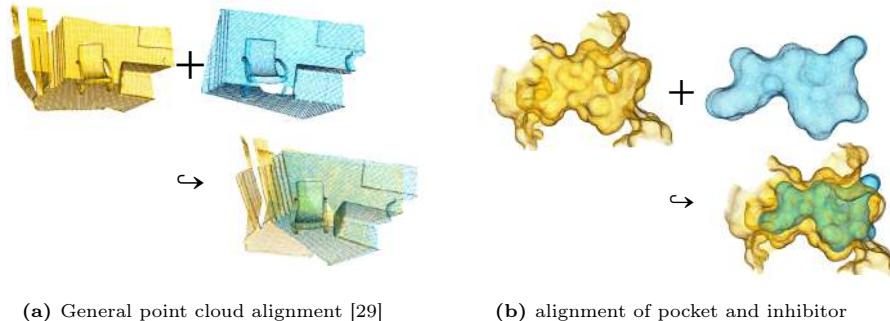


Fig. 1: Illustration of Structure-Based Virtual Screening (SBVS) method



(a) General point cloud alignment [29]

(b) alignment of pocket and inhibitor

Fig. 2: Comparison of point cloud alignments

affinity. In the latest benchmark CASF-2016 [24], which is a comparative evaluation of scoring capabilities, AutoDock Vina obtained the highest docking power among all docking tools. In recent years, several methods using deep learning have been proposed. EquiBind [23] applies graph neural networks to docking. GNINA [17] scores the binding stabilities using deep learning. DiffDock [7] uses a diffusion generative model for docking.

Since AutoDock Vina was improved in 2021 [9] and has a successful track record in developing clinically approved HIV-1 integrase inhibitors [13], it is used to benchmark the proposed method.

2.2 Point Cloud Alignment / Pose Estimation

The proposed method in this paper uses point cloud alignment to determine docking. Point cloud alignment creates a rigid transformation from one point cloud to the other with the goal of better fitting the two point clouds. Here, we discuss related methods for point cloud alignment.

Iterative Closest Point (ICP) [3] is one of the most common methods for point cloud alignment, but it has a problem in its dependence on the initial relative poses of the point clouds. As a method that does not depend on initial values, the Fast Point Feature Histograms (FPFH) [21] uses features to describe each point's local characteristics, and the points in the two point clouds with similar FPFH features are registered as corresponding points. Then, the point clouds are aligned so that the corresponding points are closest to each other. Here, finding the corresponding points is called registration.

PointNet [19] is a representative neural network model for point cloud processing and has shown to be helpful in segmentation and classification tasks. Guo et al. [14] proposed the Point Cloud Transformer (PCT) based on Transformer, which has shown great success in natural language processing and great potential in image processing. In addition, they proposed an input embedding method that performs feature aggregation of neighboring points by farthest point sampling and nearest neighbor search to better capture the local context within a point cloud. PCT achieves state-of-the-art performance on shape classification, part segmentation, semantic segmentation, and the estimation of normals tasks. Several extent works apply learned 3D descriptors for point cloud alignment. Aoki et al. [2] proposed a deep learning model PointNetLK that solves the alignment problem by minimizing the distance between the fixed-length global descriptors generated by PointNet. Sarode et al. [22] proposed a deep learning model, PCRNet, which similarly uses PointNet to extract features and a Fully Connected (FC) layer to solve the alignment problem. We use PointNetLK and PCRNet as baseline models and compare their accuracy with that of the alignment model proposed in this paper.

Some studies have adapted point cloud alignment to chemical pose estimation. Douguet et al. [8] proposed a point cloud method for comparing shapes and partial shapes between molecules. The van der Waals surface is represented as a point cloud, and point cloud alignment is performed by a registration-based

method using FPFH features, after which the alignment is improved by optimizing the matching of the colored points. This study suggests the possibility of assessing molecular shape similarity. Eguida et al. [10] proposed fragment-based virtual screening using 3D point clouds. Fragment-based virtual screening is a method in which inhibitors are created from small blocks of fragmented molecules called "fragments." Alignment by a registration-based method using FPFH features enables the pockets to be filled with fragments. Their study suggested that point cloud-based computer vision approaches to the protein-ligand docking problem can be developed. In this paper, we adopt a registration-based method that uses the FPFH features of those works for docking and compare its accuracy with the proposed method.

3 Proposed Method

We propose a shape-based SBVS method shown in Fig.3. First, point clouds of pockets and candidate inhibitors are generated. Next, virtual screening is performed in three steps as in general SBVS: (i) docking simulation, (ii) scoring, and (iii) ranking. In this paper, in step (i), docking is treated as an alignment problem to quickly find the pose in which the inhibitor fits into the pocket. Unlike the usual point cloud alignment, there is a certain gap between the point clouds of the inhibitor and the pocket, and the shapes do not perfectly match. To solve the problem, we propose a learning-based alignment model that simultaneously estimates the pose and expansion parameters. In step (ii), the error score between the pocket and the inhibitor is calculated to evaluate how well the estimated pose of the inhibitor fits into the pocket. To better utilize the expansion obtained from docking, we propose a scoring method that uses truncated chamfer distance with expansion transformation.

3.1 Deep Learning-Based Point Cloud Alignment

As shown in Fig. 2b, because the pockets and inhibitor shapes do not perfectly match, registration-based alignment using FPFH, as used by Douguet et al. [8] and Eguida et al. [10], does not provide high accuracy. In this paper, we use a learning-based alignment model to solve this problem.

Let inhibitor surface point cloud be $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots\}$ ($\mathbf{x}_i \in \mathbb{R}^3$), and the pocket surface point cloud be $Y = \{\mathbf{y}_1, \mathbf{y}_2, \dots\}$ ($\mathbf{y}_i \in \mathbb{R}^3$). The architecture of the proposed alignment model is shown in Fig.4. The model takes the inhibitor point cloud X and pocket point cloud Y as input, and outputs the pose parameter $\mathbf{z}_{\text{pose}} \in \mathbb{R}^7$, which consists of rotation quaternion $\mathbf{q} \in \mathbb{R}^4$ and translation vector $\mathbf{t} \in \mathbb{R}^3$, and the expansion parameter $a \in \mathbb{R}$.

Input Embedding. We convert the input point clouds to 256-dimentional features using the input embedding module proposed by Guo et al. in PCT [14]. The module reduces the size of inhibitor point cloud X and pocket point cloud Y to 256 points, and creates a new point cloud X' , Y' . The module extracts, simultaneously, features $\mathbf{f}_x \in \mathbb{R}^{256 \times |X'|}$ and $\mathbf{f}_y \in \mathbb{R}^{256 \times |Y'|}$.

Self-Attention. We use the self-attention layer proposed in PCT to capture global feature. When the input is $\mathbf{f}_{\text{in}} \in \mathbb{R}^{256 \times N}$, the self-attention layer outputs $\mathbf{f}_{\text{out}} = \text{LBR}(\text{Scale}(\text{Softmax}(\mathbf{Q}\mathbf{K}^T)\mathbf{V})) + \mathbf{f}_{\text{in}} \in \mathbb{R}^{256 \times N}$ that captures the context of the entire point cloud. N is the number of points, LBR is Linear, BatchNorm, and ReLU layer, Scale is the normalization by $l1$ -norm for the second dimension, $\mathbf{Q} \in \mathbb{R}^{64 \times N}, \mathbf{K} \in \mathbb{R}^{64 \times N}, \mathbf{V} \in \mathbb{R}^{256 \times N}$ are the query, key and value matrices, respectively. The four Self-Attention layers yield features $\mathbf{f}_x^i \in \mathbb{R}^{256 \times |X'|}$ and

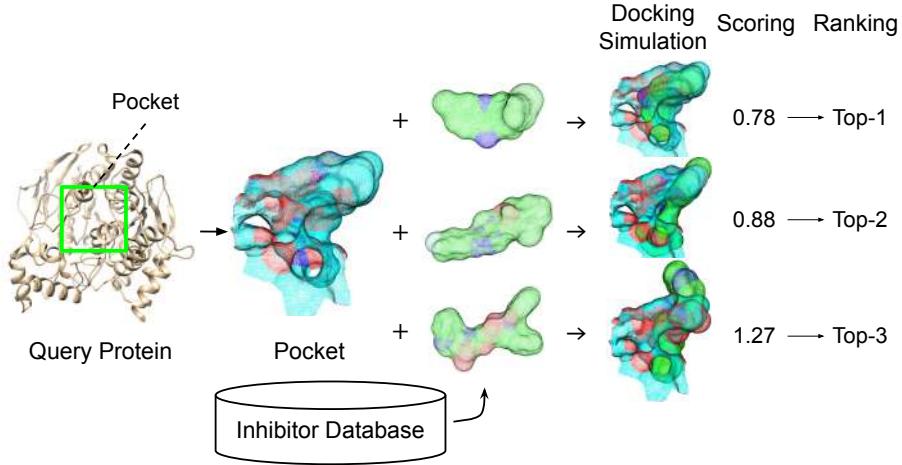


Fig. 3: Illustration of SBVS

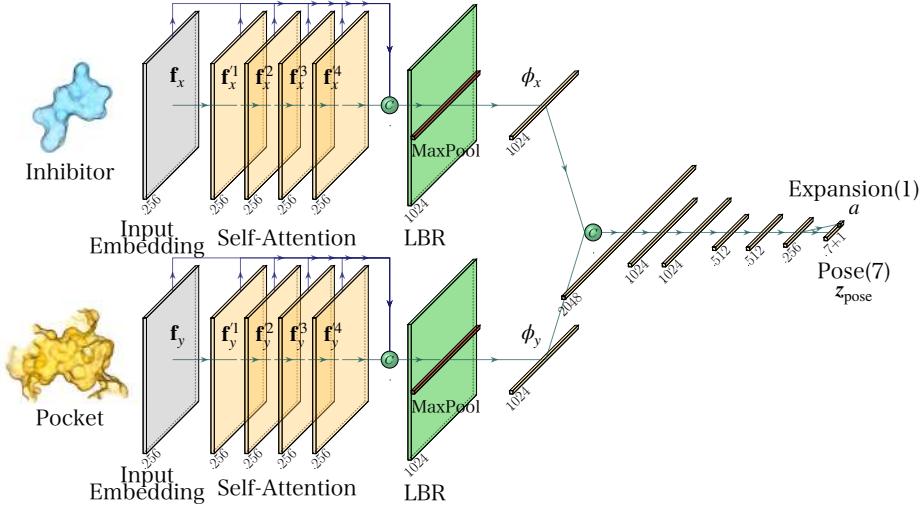


Fig. 4: Proposed alignment model

$\mathbf{f}'_y^i \in \mathbb{R}^{256 \times |Y'|}$, $i = 1, 2, 3, 4$ from features \mathbf{f}_x and \mathbf{f}_y . Since the pocket point cloud and the candidate inhibitor point cloud are different objects, we learn different network parameters (weights) in the self-attention layers for pockets and inhibitors although the network parameters in the input embedding module are shared between two streams.

Pose and Expansion Parameter Prediction. Similar to PCT, we concatenate the 256-dimensional local features yielded by the input embedding module and the 256-dimensional global features yielded by each layer of the four self-attention layers. As a result, the dimension of each concatenated feature increases to 1,280. The concatenated features $\text{concat}(\mathbf{f}_x, \mathbf{f}'_x^1, \mathbf{f}'_x^2, \mathbf{f}'_x^3, \mathbf{f}'_x^4) \in \mathbb{R}^{1280 \times |X'|}$ and $\text{concat}(\mathbf{f}_y, \mathbf{f}'_y^1, \mathbf{f}'_y^2, \mathbf{f}'_y^3, \mathbf{f}'_y^4) \in \mathbb{R}^{1280 \times |Y'|}$ are passed through the LBR layer and max pooling, and then we extract two global feature vectors $\phi_x \in \mathbb{R}^{1024}$ and $\phi_y \in \mathbb{R}^{1024}$. ϕ_x and ϕ_y are input to the FC layers to predict pose parameter z_{pose} and expansion parameter a . Here, the weights of LBR layer are shared between two streams.

3.2 Scoring

In order to rank the candidate inhibitors for virtual screening, we need to define a score to evaluate how well the inhibitor with the estimated pose fits the pocket. However, the gap between the inhibitor and the pocket has a harmful effect when calculating the score. To solve this problem, we expand the inhibitor to narrow the gap before calculating the score.

Expansion of Molecular Surface Representation. We estimate the expansion of the inhibitor and allow recourse to the general point cloud alignment problem. The molecular surface points $\mathbf{x}_i \in \mathbb{R}^3$ are expanded in the normal direction to yield new molecular surface points

$$\mathbf{x}'_i(a) = \mathbf{x}_i + a\mathbf{n}_i, \quad (1)$$

where \mathbf{n}_i is the normal at $\mathbf{x}_i \in X$ and a is the expansion parameter.

Score. If we apply a dissimilarity metric that is popularly used for calculating the distance between two point clouds (e.g., chamfer distance) as the score, the points in the pocket that are irrelevant to the binding with the inhibitor (far from the inhibitor) inappropriately increase the score. To prevent that, we propose Truncated chamfer distance as score. Truncated chamfer distance with expansion transformation of point cloud X and Y is given by

$$\begin{aligned} \text{TruncatedCD}(X, Y, a_{\text{est}}, R_{\text{est}}, \mathbf{t}_{\text{est}}) \\ = & \frac{1}{2} \left(\frac{1}{|X|} \sum_{\mathbf{x} \in X} \min \left(\min_{\mathbf{y} \in Y} \|R_{\text{est}}\mathbf{x}' + \mathbf{t}_{\text{est}} - \mathbf{y}\|_2, d_{\max} \right) \right. \\ & \left. + \frac{1}{|Y|} \sum_{\mathbf{y} \in Y} \min \left(\min_{\mathbf{x} \in X} \|R_{\text{est}}\mathbf{x}' + \mathbf{t}_{\text{est}} - \mathbf{y}\|_2, d_{\max} \right) \right), \end{aligned} \quad (2)$$

where \hat{a}_{est} is the estimated expansion parameter, R_{est} is the estimated rotation matrix transformed from quaternion \mathbf{q} in z_{pose} , and \mathbf{t}_{est} is the estimated

translation vector from output \mathbf{z}_{pose} by the alignment model. $d_{\max}(= 2.0)$ is the threshold value. In Eq. 2, each point in the expanded inhibitor is rotated and shifted to the estimated pose, and the distance to the closest point in the pocket is accumulated, but distances more than d_{\max} are truncated at that time.

3.3 Ranking

Finally, we obtain the ranking of the candidate inhibitors based on the scores, which indicates how likely the inhibitors bind to the pocket.

3.4 Training of the Proposed Model

This section describes the training method of the proposed model.

Ground Truth for Expansion Parameters. To train the alignment model, we prepare the ground truth (GT) of the expansion parameter a_{gt} . We define it as the expansion that minimizes the distance between the point cloud subset X_{ad} and the pocket Y as follows:

$$a_{\text{gt}} = \arg \min_a \frac{1}{|X_{\text{ad}}|} \sum_{\mathbf{x} \in X_{\text{ad}}} \min_{\mathbf{y} \in Y} \|\mathbf{x}' - \mathbf{y}\|_2, \quad (3)$$

where X_{ad} is the point cloud subset of inhibitor X whose distances to the closest point $y \in Y$ is less than a threshold, d , and is defined as

$$X_{\text{ad}} = \left\{ \mathbf{x} \in X \mid \min_{\mathbf{y} \in Y} \|\mathbf{x} - \mathbf{y}\|_2 < d \right\}. \quad (4)$$

Loss function. The loss function is written as

$$L = L_{\text{CD}} + L_{\text{rot}} + L_{\text{trans}} + L_{\text{expand}}, \quad (5)$$

where L_{CD} is the chamfer distance between the converted (expanded, rotated, and shifted with the estimated parameters) points in the inhibitor and the pocket point cloud.

$$\begin{aligned} L_{\text{CD}}(X, Y, a_{\text{est}}, R_{\text{est}}, \mathbf{t}_{\text{est}}) \\ = \frac{1}{2} \left(\frac{1}{|X|} \sum_{\mathbf{x} \in X} \min_{\mathbf{y} \in Y} \|R_{\text{est}} \mathbf{x}' + \mathbf{t}_{\text{est}} - \mathbf{y}\|_2 + \frac{1}{|Y|} \sum_{\mathbf{y} \in Y} \min_{\mathbf{x} \in X} \|R_{\text{est}} \mathbf{x}' + \mathbf{t}_{\text{est}} - \mathbf{y}\|_2 \right). \end{aligned} \quad (6)$$

We also give direct supervision to both pose and expansion estimation. L_{rot} , L_{trans} is the error in rotation and translation parameters

$$L_{\text{rot}} = |R_{\text{est}} R_{\text{gt}}^{-1} - I|_F, \quad (7)$$

$$L_{\text{trans}} = \|\mathbf{t}_{\text{est}} - \mathbf{t}_{\text{gt}}\|_2, \quad (8)$$

where R_{gt} is the GT rotation matrix, I is the identity matrix and $|\cdot|_F$ is the Frobenius norm, and \mathbf{t}_{gt} is the GT translation vector.

L_{expand} is the error in the expansion parameter

$$L_{\text{expand}} = \begin{cases} (1+k)|a_{\text{est}} - a_{\text{gt}}| & \text{if } a_{\text{est}} - a_{\text{gt}} > 0, \\ (1-k)|a_{\text{est}} - a_{\text{gt}}| & \text{if } a_{\text{est}} - a_{\text{gt}} \leq 0, \end{cases} \quad (9)$$

where a_{est} is the estimated value of the expansion parameter and a_{gt} is the GT value of the expansion parameter. We apply weights when the estimated expansion parameter is larger than the GT value to prevent the inhibitor point cloud from expanding so much that it overflows the pocket point cloud. k is a constant for weighting and here is set to $k = 0.5$.

4 Experimental Setting

4.1 Datasets

We constructed a dataset of protein-inhibitor pairs for training and validation. First, we listed protein-inhibitor complexes published in ten biochemical journals. Next, we obtained the 3-dimensional structures of the complexes from the Protein Data Bank (PDB) [4] and generated the point clouds of their molecular surfaces. The number of protein-inhibitor pairs obtained was 3,512 for training and 386 for validation. We used the DUD-E dataset [18] for testing, the point clouds of which were obtained in the same way. The number of pairs is 102. The obtained point clouds were sampled to 2,048 points by farthest point sampling [20].

4.2 Training

We used transfer learning from a model pre-trained by ModelNet40 [5]. For pre-training, we generate source and template point clouds from ModelNet40. The template is rotated by three random Euler angles in the range of $[-45^\circ, 45^\circ]$ from the same pose with the source. We train the alignment module by the alignment task between the source and template point clouds. After pre-training, the model was trained by the alignment task between the inhibitor and pocket point clouds, where the initial pose of the inhibitor point cloud was given by rotating it with three random Euler angles $[-45^\circ, 45^\circ]$ from the binding pose.

In the first half of training, we used the loss function without the expansion parameter $L = L_{\text{CD}} + L_{\text{rot}} + L_{\text{trans}}$ to stabilize learning. In the second half of the learning process, the loss function included the error of expansion (Eq.5). When learning the expansion parameter, we apply data augmentation to shrink the inhibitor so that the expansion parameter's GT values become uniformly distributed. The networks were trained for 200 epochs, using a learning rate of 10^{-3} , an exponential decay rate of 0.3 every 50 epochs, and a batch size of 32 in the first and second half of the training. The d_{max} in TruncatedCD was set as the maximum value of the GT expansion parameter, a_{gt} , rounded up to the nearest whole number, $d_{\text{max}} = 2.0$.

5 Alignment and Expansion Evaluation Experiments

In this section, we compared the alignment and expansion estimation accuracy of the proposed and previous methods on the DUD-E dataset.

5.1 Experimental Setting

The pocket point clouds were obtained from the pocket surfaces around the known bound ligands. The binding pose of the pocket point cloud and the inhibitor point cloud was obtained, and the inhibitor point cloud was rotated by three random Euler angles in the range $[-45^\circ, 45^\circ]$ as input. We compared the following methods: registration-based methods using FPFH and RANSAC [21], FPFH and TEASER [28], and feature learning-based methods PointNetLK [2], PCRNet [22], and the proposed method. For the proposed method, we evaluated the impact of the weight sharing in the input embedding module and self-attention layers in terms of accuracy.

The accuracy of alignment was calculated as the rotational error e_{rot} and the translational error e_{trans}

$$e_{\text{rot}} = \theta(R_{\text{est}} R_{\text{gt}}^{-1}), \quad (10)$$

$$e_{\text{trans}} = \|t_{\text{est}} - t_{\text{gt}}\|, \quad (11)$$

where $\theta(M)$ is a function that calculates the rotation angle around the rotation axis by considering M as the representation matrix of Rodrigues' rotation formula. R_{est} is the estimated rotation matrix and R_{gt} is the GT rotation matrix.

The accuracy of the expansion was calculated as the expansion error

$$e_{\text{expand}} = |a_{\text{est}} - a_{\text{gt}}|. \quad (12)$$

5.2 Results

Table 1 lists the alignment and expansion estimation accuracy results. We observe that the proposed method had smaller rotational error e_{rot} as well as translational error e_{trans} than the methods compared. Similarly, the expansion error of the proposed method was the smallest without sharing the weights of self-attention layers.

Analysis of Weight Sharing. The proposed method that shared only the weights of input embedding (i.e., did not share the self-attention weight) had the smallest error. Although the pocket point cloud and the inhibitor candidate point cloud are different objects, their shapes are similar if we focus on the local parts that are in contact with each other. That may be the reason for the performance improvement achieved by sharing the weights of the shallow layers (i.e., the input embedding) which capture local patterns rather than the self-attention weights, which capture global shapes.

Table 1: Accuracy of alignment and expansion estimation

Pose estimate method type	method	Shared weights		e_{rot}	e_{trans}	e_{expand}
		Input	Self- embedding attention			
Registration- Base	FPFH+RANSAC	-	-	120.4	1.504	-
	FPFH+TEASER	-	-	126.9	1.590	-
Learning- Base	PointNetLK	✓	-	32.3	0.721	-
	PCRNet	✓	-	31.5	0.095	-
	Ours	✗	✗	21.6	0.091	0.25
			✓	20.6	0.091	0.12

6 SBVS Evaluation Experiments

This section uses the DUD-E dataset to compare the SBVS performance of the proposed and previous methods. In addition, we use ICP to fine-tune the alignment estimated by the proposed method and assess the change in accuracy.

6.1 Experimental Setting

SBVS was performed in the following steps. We docked all candidate inhibitors $X_i (i = 1, 2, \dots)$ against query pocket Y_j and calculated scores. Then, we obtained retrieval (ranking) results of the inhibitors based on the calculated scores, regarding the pockets as queries. The above retrievals were processed for all query pockets $j = 1, 2, \dots$. We evaluated the performance of SBVS using the metrics of top- k accuracy and mean average precision at one (mAP@1) defined as

$$\text{mAP@1} = \frac{1}{N_{\text{query}}} \sum_{i=1}^{N_{\text{query}}} \frac{1}{k_i} \quad (13)$$

and processing times. Here, N_{query} is the total number of queries, and k_i is the rank of the correct inhibitor retrieved for the i th query. Processing time is the retrieval time per query. We used six alignment methods (FPFH+RANSAC, FPFH+TEASER, PointNetLK, PCRNet, the proposed method, and the proposed method+ICP) for docking and compared them. For score calculation, we used chamfer distance (CD) and the proposed TruncatedCD. We also compared the SBVS performance with AutoDock Vina, which is based on chemical property optimization.

6.2 Result

Table 2 shows the evaluation results of SBVS accuracy. The proposed model yielded the highest accuracy. Although its processing time was longer than those of PointNetLK and PCRNet; however, its processing time was reduced by about 14/15 compared to AutoDock Vina.

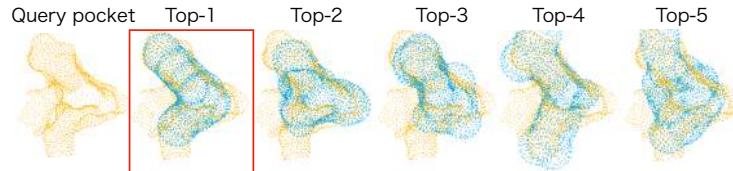
Table 2: Accuracy of SBVS

Method	Score Function	Expansion Estimation	Top-1	Top-5	Top-10	mAP@1	CPU Run
			Acc.	Acc.	Acc.		Time[s]
AutoDock Vina	-	-	0.28	0.50	0.58	0.39	1,411
FPFH+RANSAC	CD	-	0.01	0.06	0.09	0.06	347
FPFH+TEASER	CD	-	0.01	0.05	0.10	0.06	52
PointNetLK	CD	-	0.03	0.12	0.25	0.10	13
PCRNet	CD	-	0.08	0.024	0.39	0.17	10
Ours	CD	✗	0.10	0.25	0.44	0.20	25
		✓	0.33	0.65	0.77	0.48	25
Ours+ICP	TruncatedCD	✗	0.11	0.22	0.37	0.19	25
		✓	0.51	0.75	0.78	0.61	25
Ours+ICP	CD	✗	0.10	0.22	0.34	0.18	45
		✓	0.41	0.67	0.76	0.52	43
Ours+ICP	TruncatedCD	✗	0.04	0.14	0.25	0.12	44
		✓	0.57	0.75	0.81	0.65	44

Fig.5 shows an example of a successful SBVS retrieval using the proposed method that yielded the highest accuracy. The correct pocket-inhibitor correspondence occupied the top-1 rank because of the synergism between estimated expansion transformations and alignment.

Fig.6 shows an example of a SBVS retrieval failure. Fig. 6b and 6c show the ground-truth pose and the output of the proposed model with the correct pocket-inhibitor in Fig.6. We can see that the gap size is not uniform (although the bottom part of the inhibitor is touching the pocket, there are open spaces between the inhibitor and the pocket in other parts in Fig. 6b.). Because the proposed method can deal with only globally uniform expansions, the estimated pose did not fit the pocket, and the score inappropriately increased. Extending the proposed method to support more flexible expansion will be one of our future works.

Ablation Study. As shown in Table 2, the proposed method with ICP, TruncatedCD, and expansion estimation achieved the highest accuracy. We can see that the accuracy of the proposed method with TruncatedCD is higher than that with CD. This is because the truncation removes the score (distance) cal-

**Fig. 5:** An example of a successful SBVS retrieval

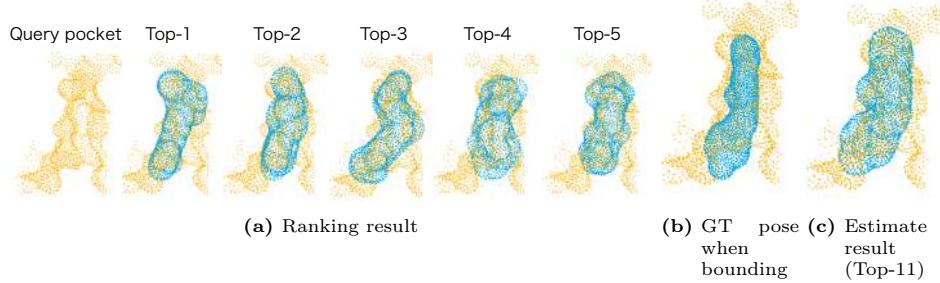


Fig. 6: An example of SBVS retrieval failure

culation between the points that are too far from each other (i.e., irrelevant to the docking) as discussed in Sec. 3.2. In addition, incorporating ICP improved the accuracy of the proposed method because the estimated poses are fine-tuned by ICP and become more accurate.

Fig. 7 shows the intermediate results of the correct inhibitor retrieved in Fig. 5. Fig. 7a, 7b, 7c, and 7d show the ground-truth pose, the alignment estimated by the proposed alignment model, after expansion transformation, and fine-tuning by ICP, respectively.

7 Conclusions

In this paper, we proposed a fast SBVS method based on shape features. The proposed method employs docking simulation based on a learning-based alignment model that simultaneously estimates pose and expansion parameters, and scoring based on Truncated Chamfer Distance with expansion transformation. The proposed method achieved more accurate estimation results compared to previous methods, including chemical property optimization, in less time. Thus, the results suggest that SBVS can be performed quickly and with sufficient accuracy by focusing only on shape features.

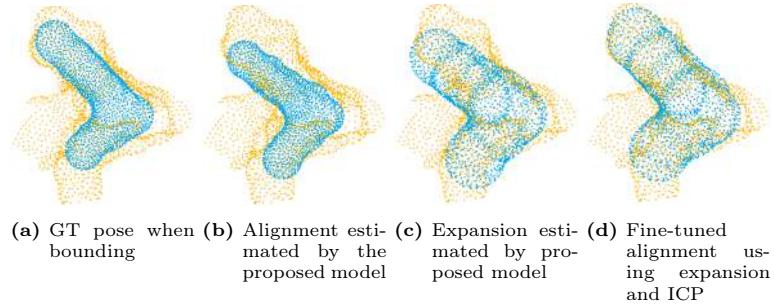


Fig. 7: An example of expansion estimation and alignment fine-tuning

However, newer SBVS methods using graph neural networks and diffusion models have been proposed in recent years. Their performance will need to be compared with that of the proposed method in the future.

References

- Allen, W.J., Balius, T.E., Mukherjee, S., Brozell, S.R., Moustakas, D.T., Lang, P.T., Case, D.A., Kuntz, I.D., Rizzo, R.C.: DOCK 6: Impact of new features and current docking performance. *Journal of computational chemistry* **36**(15), 1132–1156 (2015)
- Aoki, Y., Goforth, H., Srivatsan, R.A., Lucey, S.: PointNetLK: Robust & Efficient Point Cloud Registration Using PointNet. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7163–7172 (2019)
- Arun, K.S., Huang, T.S., Blostein, S.D.: Least-squares fitting of two 3-d point sets. *IEEE Transactions on pattern analysis and machine intelligence* (5), 698–700 (1987)
- Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E.: The Protein Data Bank. *Nucleic Acids Research* **28**(1), 235–242 (2000)
- Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: ShapeNet: An Information-Rich 3D Model Repository. arXiv preprint arXiv:1512.03012 (2015)
- Cheeseright, T.J., Mackey, M.D., Melville, J.L., Vinter, J.G.: FieldScreen: Virtual Screening Using Molecular Fields. Application to the DUD Data Set. *Journal of Chemical Information and Modeling* **48**(11), 2108–2117 (2008)
- Corso, G., Stärk, H., Jing, B., Barzilay, R., Jaakkola, T.: DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking. arXiv preprint arXiv:2210.01776 (2022)
- Douguet, D., Payan, F.: Sensaas (sensitive surface as a shape): utilizing open-source algorithms for 3d point cloud alignment of molecules. arXiv preprint arXiv:1908.11267 (2019)
- Eberhardt, J., Santos-Martins, D., Tillack, A.F., Forli, S.: AutoDock Vina 1.2. 0: New docking methods, expanded force field, and python bindings. *Journal of Chemical Information and Modeling* **61**(8), 3891–3898 (2021)
- Eguida, M., Rognan, D.: A Computer Vision Approach to Align and Compare Protein Cavities: Application to Fragment-based Drug Design. *Journal of Medicinal Chemistry* **63**(13), 7127–7142 (2020)
- Friesner, R.A., Murphy, R.B., Repasky, M.P., Frye, L.L., Greenwood, J.R., Halgren, T.A., Sanschagrin, P.C., Mainz, D.T.: Extra Precision Glide: Docking and Scoring Incorporating a Model of Hydrophobic Enclosure for Protein – Ligand Complexes. *Journal of Medicinal Chemistry* **49**(21), 6177–6196 (2006)
- Gimeno, A., Ojeda-Montes, M.J., Tomás-Hernández, S., Cereto-Massagué, A., Beltrán-Debón, R., Mulero, M., Pujadas, G., García-Vallvé, S.: The Light and Dark Sides of Virtual Screening: What Is There to Know? *International Journal of Molecular Sciences* **20**(6), 1375 (2019)
- Goodsell, D.S., Sanner, M.F., Olson, A.J., Forli, S.: The AutoDock suite at 30. *Protein Science* **30**(1), 31–43 (2021)
- Guo, M.H., Cai, J.X., Liu, Z.N., Mu, T.J., Martin, R.R., Hu, S.M.: PCT: Point cloud transformer. *Computational Visual Media* **7**(2), 187–199 (2021)

15. Hawkins, P.C., Skillman, A.G., Nicholls, A.: Comparison of Shape-Matching and Docking as Virtual Screening Tools. *Journal of Medicinal Chemistry* **50**(1), 74–82 (2007)
16. Maia, E.H.B., Assis, L.C., De Oliveira, T.A., Da Silva, A.M., Taranto, A.G.: Structure-Based Virtual Screening: From Classical to Artificial Intelligence. *Frontiers in chemistry* **8**, 343 (2020)
17. McNutt, A.T., Francoeur, P., Aggarwal, R., Masuda, T., Meli, R., Ragoza, M., Sunseri, J., Koes, D.R.: GNINA 1.0: molecular docking with deep learning. *Journal of cheminformatics* **13**(1), 1–20 (2021)
18. Mysinger, M.M., Carchia, M., Irwin, J.J., Shoichet, B.K.: Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking. *Journal of medicinal chemistry* **55**(14), 6582–6594 (2012)
19. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 652–660 (2017)
20. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *Advances in neural information processing systems* **30** (2017)
21. Rusu, R.B., Blodow, N., Beetz, M.: Fast Point Feature Histograms (FPFH) For 3D Registration. In: 2009 IEEE international conference on robotics and automation. pp. 3212–3217. IEEE (2009)
22. Sarode, V., Li, X., Goforth, H., Aoki, Y., Srivatsan, R.A., Lucey, S., Choset, H.: PCRNet: Point Cloud Registration Network using PointNet Encoding. arXiv preprint arXiv:1908.07906 (2019)
23. Stärk, H., Ganea, O., Pattanaik, L., Barzilay, R., Jaakkola, T.: EquiBind: Geometric Deep Learning for Drug Binding Structure Prediction. In: International Conference on Machine Learning. pp. 20503–20521. PMLR (2022)
24. Su, M., Yang, Q., Du, Y., Feng, G., Liu, Z., Li, Y., Wang, R.: Comparative Assessment of Scoring Functions: The CASF-2016 Update. *Journal of Chemical Information and Modeling* **59**(2), 895–913 (2019)
25. Trott, O., Olson, A.J.: AutoDock Vina: Improving The Speed and Accuracy of Docking with A New Scoring Function, Efficient Optimization, and Multithreading. *Journal of computational chemistry* **31**(2), 455–461 (2010)
26. Wang, X., Ramírez-Hinestrosa, S., Dobnikar, J., Frenkel, D.: The Lennard-Jones potential: when (not) to use it. *Physical Chemistry Chemical Physics* **22**(19), 10624–10633 (2020)
27. Wolber, G., Langer, T.: Ligandscout: 3-d pharmacophores derived from protein-bound ligands and their use as virtual screening filters. *Journal of Chemical Information and Modeling* **45**(1), 160–169 (2005)
28. Yang, H., Shi, J., Carlone, L.: TEASER: Fast and Certifiable Point Cloud Registration. *IEEE Transactions on Robotics* **37**(2), 314–333 (2020)
29. Zhou, Q.Y., Park, J., Koltun, V.: Open3D: A modern library for 3D data processing. arXiv:1801.09847 (2018)

Efficient Multi-Receptive Pooling for Object Detection on Drone

Jinsu An¹, Muhamad Dwisnanto Putro², Adri Priadana¹, and Kang-Hyun Jo¹

¹ Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Ulsan, South Korea

² Department of Electrical Engineering, Universitas Sam Ratulangi, Manado, Indonesia

jinsu5023@islab.ulsan.ac.kr, dwisnantomputro@unsrat.ac.id,
priadana3202@mail.ulsan.ac.kr, acejo@ulsan.ac.kr

Abstract. Object detection is the most fundamental and important research in computer vision to discriminate the location and class of the object in the image. This technology has been continuously researched for the past few years. Recently, with the development of hardware such as GPU computing power and cameras, object detection technology is gradually improving. However, there are many difficulties in utilizing GPUs on low-cost devices such as drones. Therefore, efficient deep learning technology that can operate on low-cost devices is needed. In this paper, we propose a deep learning model to enable real-time object detection on a low-cost device. We experiment to reduce the amount of computation and improve speed by modifying the CSP Bottleneck and SPPF parts corresponding to the backbone of YOLOv5. The model has been trained on MS COCO and VisDrone datasets, and the mAP values are measured at 0.364mAP and 0.19mAP, which are about 0.07 and 0.04 higher than Refinedetlite and Refinedet, respectively. The speed is 23.010 frames per second on the CPU configuration, which is enough for real-time object detection.

Keywords: Object Detection · Drone Vision · Convolutional Neural Network (CNN) · Efficient Module · Attention Modules.

1 Introduction

Nowadays, drone technology has developed rapidly and guided to widespread use for many purposes. Drones, equipped with cameras, can capture images or videos and generate a variety of beneficial application scenarios, such as video surveillance [2], monitoring [34,11], tracking [41] and searching [33,31]. A drone even can enter difficult or dangerous areas that are impossible for humans to perform these works. This approach can also reduce the possibility of risks incurred.

Advances in computer vision have dramatically enhanced drone vision technology. Many works, such as object detection and classification, can be conducted based on video captured by the drone to support the intelligence system. It leads

the drone to localize and classify the objects based on its vision with high accuracy. It can even perform over enormous areas because the drone can capture extensive coverage only in a short period. It pushes drone vision technology to become increasingly popular.

Recently, the rapid development of Convolutional Neural Networks (CNNs) has improved object detection and classification tasks, providing improved results. Many researchers are developing deeper networks to achieve higher performance [20,30,28]. Unfortunately, it guides the architecture to produce enormous parameters and operate inefficiently. A drone practically uses a low-cost device to run its system. Therefore, it requires an efficient model to perform, especially in real-time.

The field of object detection has evolved over the past 20 years. it is generally divided into two methods. It is a traditional image processing method and a deep learning method. The deep learning method is also divided into two types, one-stage, and two-stage. The network proposed in this paper is an Improved one-stage YOLO(You Only Look Once) network. One-stage based YOLO has been presented as superior real-time object detection and brought much attention. YOLOv5 [12] appeared, which applies a Cross Stage Partial (CSP) [36] block with a bottleneck mechanism to make the network more efficient. This method offers many types based on size, which have various performances. Although the framework provides small versions with fewer parameters, the detector still suffers from infeasible results.

CNN architecture creates feature maps at different levels in each layer. The initial layer creates low-level features representing simple shapes, and as the layer deepens, mid and high-level features representing complex features are extracted. In general, small, medium, and large size objects are detected using low, mid, and high-level features. However, even when detecting large objects, for example, low-level features that respond strongly to edges or small instances are needed. We also need a high-level feature that captures the context of the image to detect small objects. To this end, it is possible to more accurately localize by effectively utilizing low, medium, and large features. In order to detect an object, these various feature information are essential. The existing Feature Pyramid Network (FPN) goes through more than 100 layers to deliver low-level information to high-level, but about 10 layers are sufficient in PANet. The detector used in YOLOv5 is applied to three layers of 80, 40, and 20 sizes of PANet. This layer is upsampled from the last layer of the backbone feature map and merged with the previous level feature map of the same size.

In this work, we adjusted the C3 and SPPF [10] layers to operate the object detection algorithm in real-time, the number of parameters of the network must be reduced. The C3 layer and SPPF layer used in the original YOLOv5 are lightened, and the C3 layer is composed of a bottleneck and 3 convolution layers as CSP bottleneck with 3 convolutions. The C3 layer is lightened by adjusting the convolution of the C3 layers from three to two and changing the order of the concatenation and addition operations of the feature map.

The SPPF layer consists of two convolution layers and three max-pooling layers. To lighten the layer, we reduced one max-pooling layer and added an addition operation. The contributions of this work are summarized as follows:

1. A real-time object detection method is proposed to localize the specific object quickly that can be operated on a low-cost device.
2. A new structure of the convolutional block is introduced by modifying the fusion operation on the CSP bottleneck module.
3. SPPF layer is improved to be more efficient. It supports the network to operate on a low-cost device without compromising its accuracy.

2 Related Work

CNN architectures as a backbone have been employed and developed to perform object detection and classification. It has offered outstanding results in extracting features equipped with many techniques to predict object locations with various sizes. Faster R-CNN [30] came to refine the previous version, R-CNN [9] and Fast R-CNN [8], proposed a Region Proposal Network (RPN) to locate the Region of Interest (RoI) and identify the class of objects. Another work, RetinaNet, offered a novel loss called Focal Loss to deal with the class imbalance problem. Meantime, YOLOv3 [25], YOLOv4 [3], and YOLOv5 [12] utilized the Feature Pyramid Network (FPN) [18] strategy to combine features with various levels.

Many researchers designed various efficient CNN architectures as a backbone to perform object detection. Fast-PdNet [27] offered a lightweight CNN architecture with multi-level contextual blocks that produce fewer parameters than general detectors. The detector is specially designed to perform person detection in supporting assistive robots. Another work [1] adjusted C3 module with a residual bottleneck mechanism on YOLOv5 [12] to make the model more efficient.

Several works modified the YOLO framework to perform efficient object detection applied in supporting drone vision. Pruned-YOLOv3/v5 [39] proposed an iterative channel pruning mechanism to design a lightweight network for YOLOv3 and YOLOv5. It gains a satisfactory balance between efficiency and accuracy on MS-COCO and VisDrone datasets. ECAP-YOLO [13], modified from YOLOv5 [12], offered an efficient channel attention pyramid method to deal with small object problems in aerial images. SPB-YOLO [38] also adjusted YOLOv5 [12] with Strip Bottleneck (SPB) module to build an efficient real-time detector for a drone. It achieves a good trade-off between speed and accuracy.

3 The Proposed Method

The proposed architecture has two main modules as shown in Fig. 1. Both are used in the backbone of YOLOv5, which corresponds to the baseline. The first is Efficient Residual Bottleneck (ERB), and the second is Efficient Multi-Receptive Pooling (EMRP).

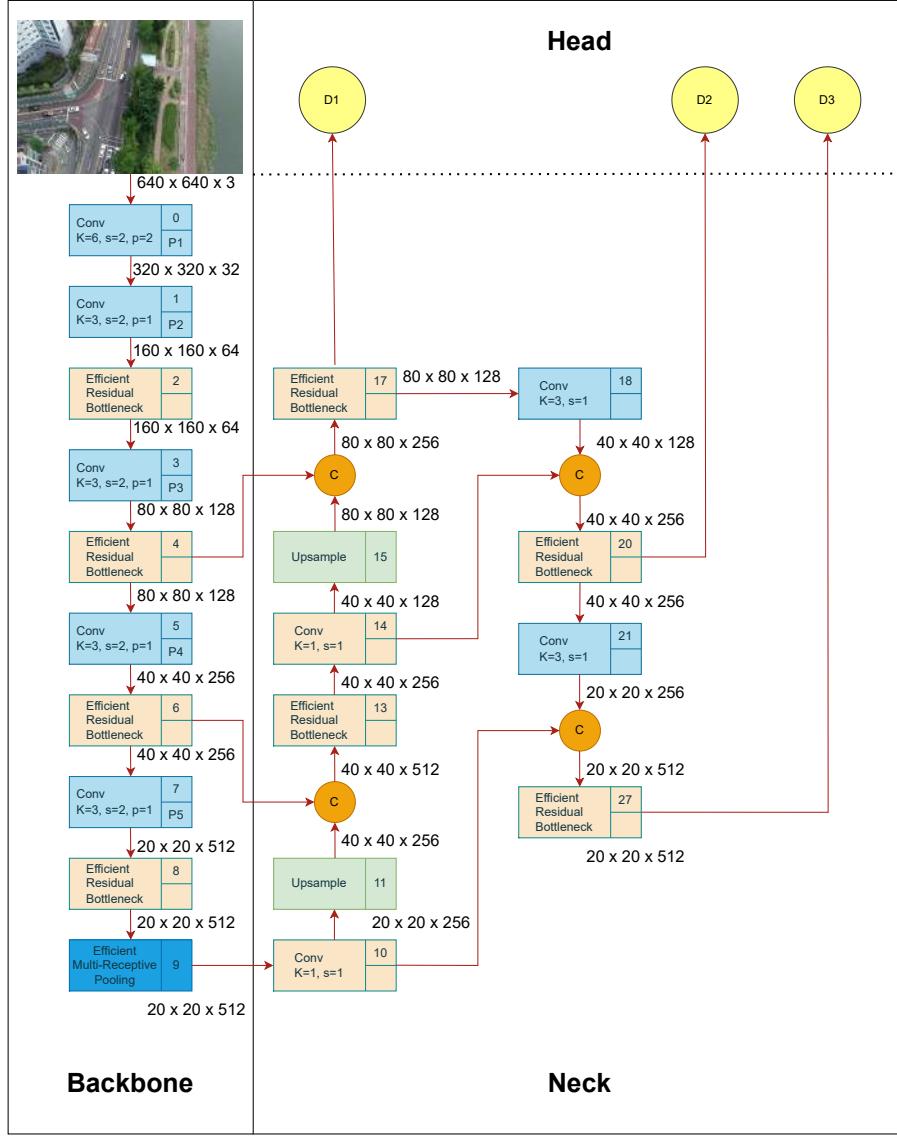


Fig. 1. The proposed architecture. A backbone module is used to extract object features with the proposed efficient methods. Besides, the PANet(Neck) and detection(Head) modules help the detector identify the location of the object in multi-scale variants.

3.1 The Backbone

The framework of YOLOv5 has three main components. It consists of Backbone, Neck, and Head. The Backbone extracts the features of the image and

transfers them to the Head through the Neck. Neck creates a feature pyramid by collecting feature maps extracted from the Backbone. Finally, it is composed of an output layer that detects objects in the Head. CSPDarknet53 [35] is used as the backbone, PANet(Path Aggregation Network) [23] is used for the Neck, and $B \times (5+C)$ output layer is used for the Head. B is the number of bounding boxes, and C is the class score. Among them, the C3 layer and SPPF [10] layer of CSPDarknet53 used in the backbone are modified to lighten the deep learning object detection model.

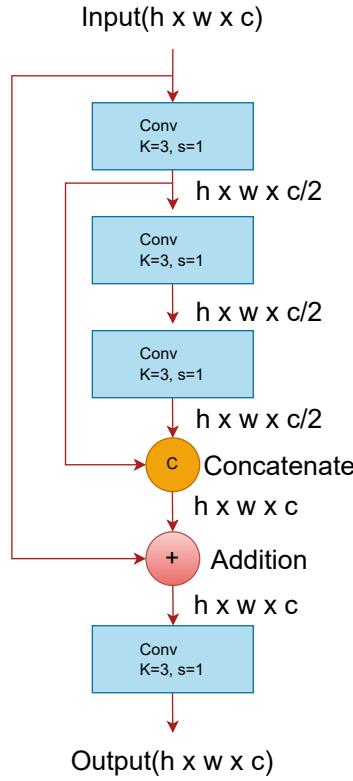


Fig. 2. Efficient Residual Bottleneck.

3.2 Efficient Residual Bottleneck

Efficient Residual Bottleneck (ERB) is an improved layer of the C3 layer used in YOLOv5. The C3 layer is CSP Bottleneck with 3 convolutions and consists of a bottleneck and 3 convolution layers. In order to operate the object detection algorithm in real-time on drones using low-cost devices, the number of parameters

of the deep learning object detection network must be reduced. To decrease the number of parameters, the convolution of the C3 layer is adjusted from three to two, and the order of concatenation and addition operations of the feature map is changed. The proposed network offers an improved backbone that extracts the object features and discriminates the essential elements from the background. It applies a set of convolution layers sequentially using an efficient module. Light blocks apply residual techniques to maintain the quality of the feature map to push high performance in the final prediction. To avoid gradient performance degradation and prevent saturation of the training process, SiLU activation and Batch Normalization are employed sequentially in each convolution operation.

3.3 Efficient Multi-Receptive Pooling

Improved from [10], the efficient multi-receptive pooling is introduced to capture the difference of spatial information that employs a cascade pooling and a simple convolution. It applies convolutional and two sequential pooling to provide various receptive areas. It can increase the options of feature selection from multi-perspective combinations. It uses simple convolution to obtain one spatial area. Two pooling with window size of 5x5 is employed sequentially to capture the maximum value of the features. Combining features from different receptive areas will increase the variety of information so that the network will learn more about the feature type. Then, it applies a convolution operation to mix the various information. The residual technique is used in this module to ensure that the different feature pooling results obtain the expected quality and reduce the error rate of the filtering process.

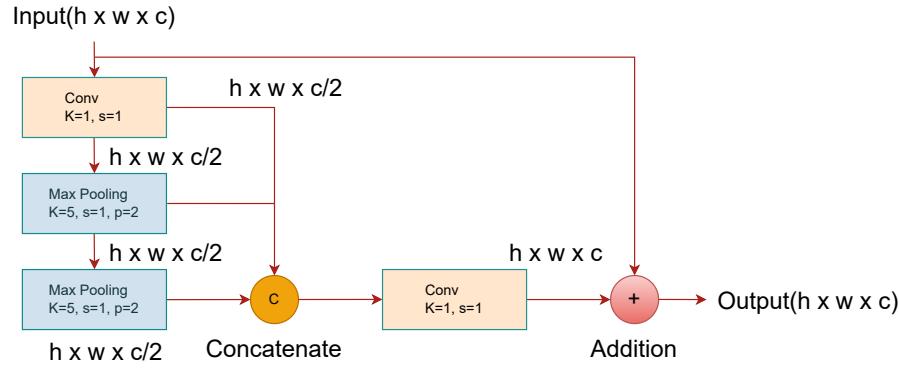


Fig. 3. Efficient Multi-Receptive Pooling, less complexity by double receptive pooling addition path ways

3.4 Loss Function

In YOLOv5, IoU loss, binary cross-entropy, and confidence loss were used as loss functions. Bounding-box regression is the most widely used method in object detection algorithms used to predict the position of an object to be detected using a bounding box. This method aims to correct the position of the predicted bounding box. Bounding box regression uses an overlapping region of the box of the real object and the predicted box location, called Intersection over Union (IOU). First, the IoU loss evaluates the difference between the predicted box position and the actual object's box's intersection, centroid distance, and aspect ratio. Second, we apply a confidence loss to evaluate whether or not there is an object in each cell. Finally, we use binary cross-entropy to measure the probability error of the predicted object class. Binary cross entropy is very effective for training models to solve many classification problems simultaneously. Combining the above three loss functions, the multi-box loss is expressed as:

$$L_{MB} = \lambda_{coord} \sum_{g=1}^{G^2} \sum_{a=1}^A g_{ga}^{obj} L_{coord} + \lambda_{obj} \sum_{g=1}^{G^2} \sum_{a=1}^A 1_{ga}^{obj} L_{obj} + \lambda_{cls} \sum_{g=1}^{G^2} \sum_{a=1}^A 1_{ga}^{obj} L_{cls} \quad (1)$$

4 Implementation Details

In this section, experiments with MS COCO [22] and VisDrone [41] datasets are described through the proposed architecture. As an experimental environment, the model is implemented using PyTorch in a Linux environment. When training the deep learning model, training is conducted using Intel Xeon Gold CPU and Nvidia Tesla V100 32GB GPU.

5 Experimental Results

5.1 Evaluation on Datasets

The proposed method tested the object detection performance on MS COCO 2017, VisDrone dataset. There are a total of 80 different classes in the COCO dataset, and it consists of a total of 143,575 image data. The COCO dataset contains objects of various sizes, complex backgrounds, and many obstacles, and the proposed model is trained with 118,287 image data. The model is evaluated with 5000 images, and the model is tested with the remaining 20,288 images. The VisDrone dataset consists of 288 video clips (261,908 images) and 10,209 static photos were collected from multiple cameras mounted on drones and has a total of 10 classes (pedestrian, people, bicycle, car, van, truck, tricycle, awning-tricycle, bus, motor). Among them, the proposed model is trained with 6,471 image data, and the model is evaluated with 1,610 images and tested with 548

Table 1. Detection Result Comparisons on MS COCO Dataset, where Time@CPU1 and Time@CPU2 mean Running Time Tested on Intel I7-6700@3.40GHZ and Intel I5 6600@3.30GHZ, respectively.

Model	mAP 0.5:.95	Backbone	Time@CPU1	Time@CPU2
SSD[24]	0.193	MobileNet	128ms	-
SSDLite[32]	0.222	MobileNet	125ms	-
SSDLite[32]	0.221	MobileNetV2	120ms	-
Pelee[37]	0.224	PeleeNet	140ms	-
Tiny-DSOD[15]	0.232	DDB-Net+D-FPN	180ms	-
SSD[24]	0.251	VGG	1250ms	-
SSD[24]	0.28	ResNet101	1000ms	-
YOLOv3[29]	0.282	DarkNet53	1300ms	-
RefineDetLite[7]	0.268	Res2NetLite72	130ms	-
RefineDetLite++[7]	0.296	Res2NetLite72	131ms	-
YOLOv5s-ERB	0.367	Improved CSPDarknet53	-	43ms
YOLOv5s-ERB_wosppf	0.334	Improved CSPDarknet53	-	36ms
YOLOv5s-ERB_conv3	0.366	Improved CSPDarknet53	-	40ms
YOLOv5s-ERB_EMRP	0.364	Improved CSPDarknet53	-	-

Table 2. Detection Result Comparisons on VisDrone Dataset.

Model	mAP 0.5:.95	Backbone
Cascade R-CNN+[5]	0.183	SEResNeXt-50
EnDet	0.178	ResNet101-fpn
DCRCNN[6]	0.178	ResNeXt-101
Cascade R-CNN++[5]	0.177	ResNeXt-101
ODAC	0.174	VGG
DA-RetianNet[26]	0.171	ResNet101
MOD-RETINANET	0.169	ResNet50
DBCL	0.168	Hourglass-104
ConstraintNet	0.161	Hourglass-104
CornetNet*[14]	0.174	Hourglass-104
Light-RCNN*[16]	0.165	ResNet101
FPN*[19]	0.165	ResNet50
Cascade R-CNN*[4]	0.161	ResNeXt-101
DetNet59*[17]	0.153	ResNet50
RefineDet*[40]	0.149	ResNet101
RetinaNet*[21]	0.118	ResNet101
YOLOv5s-vis-c3	0.195	Improved CSPDarknet53
YOLOv5s-vis-esppf	0.193	Improved CSPDarknet53
YOLOv5s-vis-c3esppf	0.190	Improved CSPDarknet53

images. An object detection model is evaluated through a dataset by extracting and learning the features of various objects included in the dataset. To evaluate the model, we use Average Precision (AP) to measure the accuracy of the predicted bounding box, derive AP for each class, and finally calculate the mean Average Precision (mAP) value for all classes. As a result, the mAP values of the proposed method are calculated as 0.364 and 0.190, respectively.

6 Conclusion

This paper proposes efficient residual bottleneck and efficient multi-receptive pooling for a deep learning algorithm capable of real-time object detection. In order to reduce the complexity, the existing CSP Bottleneck and SPPF are improved. And the proposed network is trained on MS COCO and VisDrone datasets. The mAP value on the MS COCO dataset is measured at 0.364, and when compared to RefineDetLite++, the performance increased by about 0.07 mAP difference. The mAP value on the VisDrone dataset is measured at 0.190, and when compared to RefineDet+, the value is about 0.04 mAP higher. In the future, we plan to use the additional detector to increase the object detection rate. As the number of layers in the network increases, the number of parameters required for computation increases. It is expected that the method proposed in this paper can be used to reduce the number of parameters and increase the object detection rate by using additional detectors.

7 Acknowledgement

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the government(MSIT).(No.2020R1A2C2008972)

References

1. An, J., Putro, M.D., Jo, K.H.: Efficient residual bottleneck for object detection on cpu. In: 2022 International Workshop on Intelligent Systems (IWIS). pp. 1–4. IEEE (2022)
2. Bera, B., Das, A.K., Garg, S., Piran, M.J., Hossain, M.S.: Access control protocol for battlefield surveillance in drone-assisted iot environment. IEEE Internet of Things Journal **9**(4), 2708–2721 (2021)
3. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020)
4. Cai, Z., Vasconcelos, N.: Cascade R-CNN: delving into high quality object detection. CoRR **abs/1712.00726** (2017), <http://arxiv.org/abs/1712.00726>
5. Cai, Z., Vasconcelos, N.: Cascade r-cnn: High quality object detection and instance segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **43**(5), 1483–1498 (2021). <https://doi.org/10.1109/TPAMI.2019.2956516>
6. Chakraborty, S., Aich, S., Kumar, A., Sarkar, S., Sim, J.S., Kim, H.C.: Detection of cancerous tissue in histopathological images using dual-channel residual convolutional neural networks (dcrenn). In: 2020 22nd International Conference on Advanced Communication Technology (ICACT). pp. 197–202 (2020). <https://doi.org/10.23919/ICACT48636.2020.9061289>
7. Chen, C., Liu, M., Meng, X., Xiao, W., Ju, Q.: Refinedetlite: A lightweight one-stage object detection framework for cpu-only devices. CoRR **abs/1911.08855** (2019), <http://arxiv.org/abs/1911.08855>
8. Girshick, R.: Fast r-cnn. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 1440–1448. IEEE (2015)



Fig. 4. Visualization of the Detection Result on VisDrone dataset.

9. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. pp. 580–587. IEEE (2014)
10. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence* **37**(9), 1904–1916 (2015)
11. Ikshwaku, S., Srinivasan, A., Varghese, A., Gubbi, J.: Railway corridor monitoring using deep drone vision. In: Computational Intelligence: Theories, Applications and Future Directions-Volume II, pp. 361–372. Springer (2019)
12. Jocher, G., Stoken, A., Borovec, J.: ultralytics/yolov5: v3.0, <https://doi.org/10.5281/zenodo.3983579>
13. Kim, M., Jeong, J., Kim, S.: Ecap-yolo: Efficient channel attention pyramid yolo for small object detection in aerial image. *Remote Sensing* **13**(23), 4851 (2021)
14. Law, H., Deng, J.: CornerNet: Detecting objects as paired keypoints. CoRR [abs/1808.01244](https://arxiv.org/abs/1808.01244) (2018), <http://arxiv.org/abs/1808.01244>
15. Li, Y., Li, J., Lin, W., Li, J.: Tiny-dsod: Lightweight object detection for resource-restricted usages. CoRR [abs/1807.11013](https://arxiv.org/abs/1807.11013) (2018), <http://arxiv.org/abs/1807.11013>

16. Li, Z., Peng, C., Yu, G., Zhang, X., Deng, Y., Sun, J.: Light-head R-CNN: in defense of two-stage object detector. CoRR **abs/1711.07264** (2017), <http://arxiv.org/abs/1711.07264>
17. Li, Z., Peng, C., Yu, G., Zhang, X., Deng, Y., Sun, J.: Detnet: A backbone network for object detection. CoRR **abs/1804.06215** (2018), <http://arxiv.org/abs/1804.06215>
18. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 936–944. IEEE (2017)
19. Lin, T., Dollár, P., Girshick, R.B., He, K., Hariharan, B., Belongie, S.J.: Feature pyramid networks for object detection. CoRR **abs/1612.03144** (2016), <http://arxiv.org/abs/1612.03144>
20. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **42**(2), 318–327 (2018)
21. Lin, T., Goyal, P., Girshick, R.B., He, K., Dollár, P.: Focal loss for dense object detection. CoRR **abs/1708.02002** (2017), <http://arxiv.org/abs/1708.02002>
22. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
23. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8759–8768. IEEE (2018)
24. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S.E., Fu, C., Berg, A.C.: SSD: single shot multibox detector. CoRR **abs/1512.02325** (2015), <http://arxiv.org/abs/1512.02325>
25. Murthy, C.B., Hashmi, M.F.: Real time pedestrian detection using robust enhanced yolov3+. In: 2020 21st International Arab Conference on Information Technology (ACIT). pp. 1–5. IEEE (2020)
26. Pasqualino, G., Furnari, A., Signorello, G., Farinella, G.M.: An unsupervised domain adaptation scheme for single-stage artwork recognition in cultural sites. Image and Vision Computing p. 104098 (2021). <https://doi.org/https://doi.org/10.1016/j.imavis.2021.104098>
27. Putro, M.D., Nguyen, D.L., Priadana, A., Jo, K.H.: Fast person detector with efficient multi-level contextual block for supporting assistive robot. In: 2022 IEEE 5th International Conference on Industrial Cyber-Physical Systems (ICPS). pp. 1–6. IEEE (2022)
28. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 779–788. IEEE (2016)
29. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. CoRR **abs/1804.02767** (2018), <http://arxiv.org/abs/1804.02767>
30. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(6), 1137–1149 (2016)
31. Sambolek, S., Ivasic-Kos, M.: Automatic person detection in search and rescue operations using deep cnn detectors. IEEE Access **9**, 37905–37922 (2021)
32. Sandler, M., Howard, A.G., Zhu, M., Zhmoginov, A., Chen, L.: Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. CoRR **abs/1801.04381** (2018), <http://arxiv.org/abs/1801.04381>

33. Sibanyoni, S.V., Ramotsoela, D.T., Silva, B.J., Hancke, G.P.: A 2-d acoustic source localization system for drones in search and rescue missions. *IEEE Sensors Journal* **19**(1), 332–341 (2018)
34. Sun, W., Dai, L., Zhang, X., Chang, P., He, X.: Rsod: Real-time small object detection algorithm in uav-based traffic monitoring. *Applied Intelligence* **52**(8), 8448–8463 (2022)
35. Wang, C.Y., Liao, H.Y.M., Wu, Y.H., Chen, P.Y., Hsieh, J.W., Yeh, I.H.: Cspnet: A new backbone that can enhance learning capability of cnn. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 1571–1580. IEEE (2020)
36. Wang, C., Liao, H.M., Yeh, I., Wu, Y., Chen, P., Hsieh, J.: Cspnet: A new backbone that can enhance learning capability of CNN. CoRR **abs/1911.11929** (2019), <http://arxiv.org/abs/1911.11929>
37. Wang, R.J., Li, X., Ao, S., Ling, C.X.: Pelee: A real-time object detection system on mobile devices. CoRR **abs/1804.06882** (2018), <http://arxiv.org/abs/1804.06882>
38. Wang, X., Li, W., Guo, W., Cao, K.: Spb-yolo: An efficient real-time detector for unmanned aerial vehicle images. In: 2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC). pp. 099–104. IEEE (2021)
39. Zhang, J., Wang, P., Zhao, Z., Su, F.: Pruned-yolo: Learning efficient object detector using model pruning. In: International Conference on Artificial Neural Networks. pp. 34–45. Springer (2021)
40. Zhang, S., Wen, L., Bian, X., Lei, Z., Li, S.Z.: Single-shot refinement neural network for object detection. CoRR **abs/1711.06897** (2017), <http://arxiv.org/abs/1711.06897>
41. Zhu, P., Wen, L., Du, D., Bian, X., Fan, H., Hu, Q., Ling, H.: Detection and tracking meet drones challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **44**(11), 7380–7399 (2021)

Robust Scene Text Detection under Occlusion via Multi-Scale Adaptive Deep Network

My-Tham Dinh, Minh-Trieu Tran^[0000-0002-5015-5604], Quang-Vinh Dang, and Guee-Sang Lee^[0000-0002-8756-1382]

Department of Artificial Intelligence Convergence, Chonnam National University,
Gwangju, South Korea
 thamdinh.dmt@gmail.com, tmtvaa@gmail.com, quangvinh242003@yahoo.com,
 gslee@jnu.ac.kr

Abstract. Detecting text under occlusion in natural images is a challenge in scene text detection, which is severely sensitive and dramatically affects the performance of this field. Although some papers mention the solutions for the missing text problem, i.e., occlusion, they still fail on word regions with text bounding boxes splitting by the occlusion phenomena. In this paper, we first exploit the salient attention maps from Gradient Class Activation Maps Plus Plus (Grad-CAM++) on ImageNet to obtain knowledge of the important regions in the images. Moreover, to capture the diversity sizes of text instances and robustly enrich feature representations, we create a Multi-scale adaptive Deep network (MTD). In addition to this task, from ICDAR 2015 benchmark, we build occluded text, namely Realistic Occluded Text Detection dataset (ROTD), and then combine a part of this new dataset with the ICDAR 2015 dataset for the training process to capture occluded text perception. Through these works, our model significantly improves the accuracy of text detection containing partially occluded text in natural scenes. Our proposed method achieves state-of-the-art results on partial occlusion text detection with $F1 - score$ of 69.6% on ISTD-OC, 78.7% on our ROTD, and validates competitive performance $F1 - score$ of 82.4% on ICDAR 2015 benchmark.

Keywords: Scene Text Detection · Multi-Scale Adapter Network · Grad-CAM++ · Occluded Text · Deep Learning.

1 Introduction

With the development of deep learning, scene text detection [1, 3], scene text segmentation [27, 29, 32], scene text recognition [31], and text spotting [30] have many achievements in scene text reading. As a key prior component of this field, text detection in natural scenes has played an essential role in computer vision, signal, and image processing. However, due to the variety of orientations, shapes, or sizes, it is still a challenging task, although many existing methods have achieved noticeable breakthroughs [1–3]. For example, DBNet++ [2] achieves consistently state-of-the-art accuracy and speed on five benchmarks of scene



Fig. 1. Several examples of failure text detection with text bounding boxes splitting by the occlusion phenomena of previous deep network architectures (a), and solved by our method (b) on ISTD-OC dataset.

text detection. Furthermore, TextPMs [3] obtains state-of-the-art performance in terms of detection accuracy both on polygonal and on quadrilateral datasets.

Unlike the previous prevalent problems in scene text detection, few researchers work on addressing partially occluded text problems [7, 8], which can significantly affect detection performance. For instance, [7] is detection and recognition task that can also achieve effective detection by restoring missing text. However, this method only assumes to detect text with few character-based distortion. Besides, in [8], the main task is to create ISTD-OC text occlusion dataset, involving different occlusion levels (from 0% to 100%), and evaluates the efficiency of state-of-the-art deep learning frameworks on ISTD-OC. Nevertheless, these frameworks are sensitive to occlusions and fail on text regions detection with text bounding boxes splitting by the occlusion phenomena, as in Figure 1.

In this paper, we design an approach to address this problem more efficiently. We apply a transfer learning Guided Grad-CAM++ Attention maps relying on Grad-CAM++ pre-trained on ImageNet [15] to obtain salient text regions. In addition, we give more robust feature representations with various sizes by exploring MTD. Our core contributions are as follows:

- We take advantage of the pre-trained from Grad-CAM++ from ImageNet [15] to gain the Guided Attention salient maps as one kind of specific information for training process, after that, we transfer the attention knowledge to our text detection under occluded text task.
- Our model MTD enhances both receptive fields by obtaining diverse scales and feature representation capability by learning multi-level information of features. Additionally, we adopt CBAM attention to improve the channel and spatial awareness abilities. Hence, our method is able to get richer feature representations.
- We also build our own occluded dataset ROTD based on ICDAR 2015 benchmark and combine only 10% as experimental results in Figure 6 with the

original one during training phase to learn more efficiently and accurately about occluded text awareness.

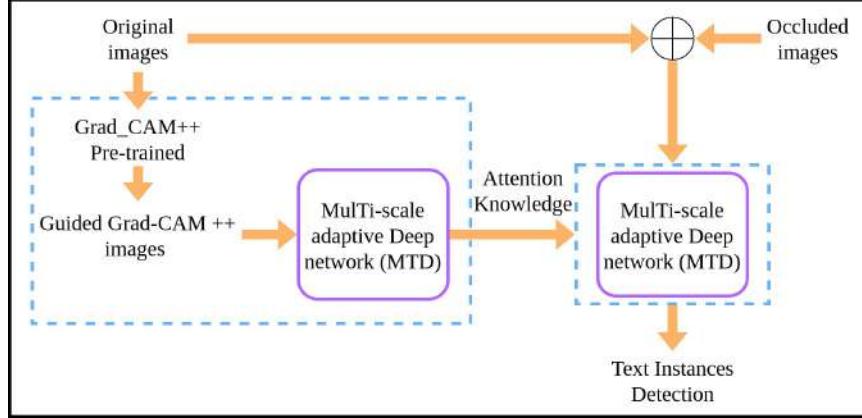


Fig. 2. This figure illustrates the overall architecture of the proposed method. Our approach includes three steps: Firstly, applying Guided Grad-CAM++ Attention maps for training process, next, transferring the learning knowledge to the main task, and finally, predicting text instances detection under occlusion.

2 Related Work

2.1 Scene Text Detection

Text detectors in natural images have achieved many remarkable results by many methods. Most of them are roughly divided into two phases, regression-based, and segmentation-based.

In the first category, several impressive research employed regression-based method [3, 17] that regresses directly bounding boxes of the text instances. EAST [13] could predict score maps from the fully convolutional network and multi-oriented text instances. Similarly, Deep-Reg [18] designed a per pixel-regression approach to detect multi-oriented tasks. However, it can be noted that these models are inadequate to cope with the occluded text challenge.

Additionally, another attractive method is segmentation-based [1, 11] that usually locates text regions following pixel-level prediction with post-processing algorithms. A progressive scale expansion algorithm is exploited in PSENet [11] to expand the detection areas with whole text instances. PAN [1] detected scene text instances and tackled overlap problem by clustering and aggregating text pixels by predicted similarity vectors.

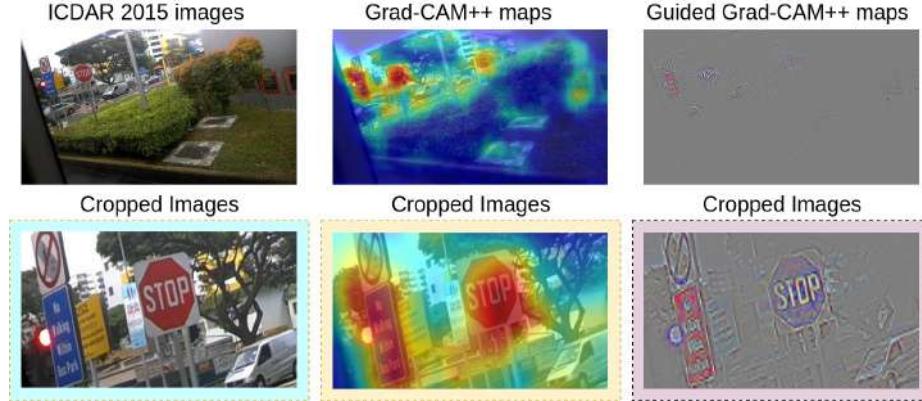


Fig. 3. Visualization of attention maps: Grad-CAM++ and Guided Grad-CAM++ from ICDAR 2015 images.

2.2 Occluded Text Detection

Partially occluded text in scene images, which may threaten prediction accuracy, is still a difficult challenge in scene text detection. [8] proved that several models from PAN [1], PSENet [11], CRAFT [12], and EAST [13] are still ineffective in detecting occluded text instances. In the same way, [7] handled the missing text issue by inheriting the strength of the characteristic Discrete Cosine Transform. Nevertheless, these methods have still failed significantly on text instances with text bounding boxes splitting by the occlusion phenomena. Therefore, this paper refines the capability of occlusion perception for scene text detection.

3 Methodology

Our overall architecture is illustrated in Figure 2. Firstly, we create Guided Grad-CAM++ maps from Grad-CAM++’s pre-trained on ImageNet [26] as in Figure 3, which focus on attention information and mitigate complex backgrounds. By learning those attention information in scene images, our approach can capture the spatial context of text instances. And then, passing them over MTD, including a ResNet18 [22] backbone, CBAM attention, a Multi-scale FEN (MFEN) [14], and features fusion. Due to the robustness of learning both extracted features with different scales and multi-level information features with less computation, our model enlarges the receptive fields and enhances the feature representation capabilities. After that, we follow as a PAN post-processing [1] of prediction network to detect bounding boxes of text instances as in Figure 4. Finally, transferring the learning attention knowledge of Guided Grad-CAM++ maps to the main task: Text detection containing occluded text. To help our model understand apparently occlusion knowledge from occluded images, we also employ a novel realistic occluded dataset (ROTD) by the OpenCV tool in Algorithm 1,

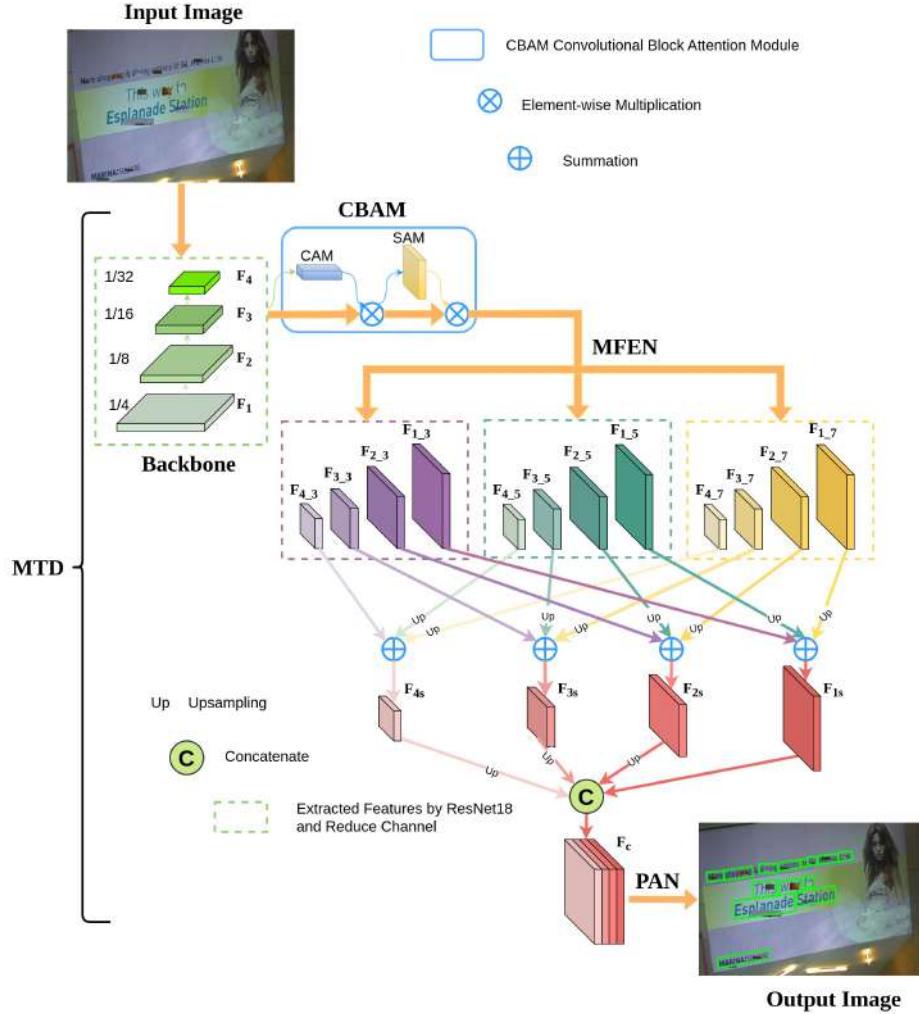


Fig. 4. Illustration of our proposed MTD network for occluded scene text detection.

and fuse a few number of ROTD images combined with the original ICDAR 2015 benchmark for training in the main phase.

3.1 Transfer Learning Knowledge from Guided Grad-CAM++ Attention

Gradient Class Activation Maps Plus Plus (Grad-CAM++) [15] improves the localization of both single, multiple instances, and produced perfect results for object classification and localization in the current state-of-the-art methods. Guided Grad-CAM++ Attention is carried out pointwise multiplication by

salient maps of Grad-CAM++ with pixel-space visualization by Guided Back-propagation. Therefore, to learn the important context in features, we apply pre-trained Grad-CAM++ on ImageNet [26] to obtain those salient maps containing the attention knowledge, then through them to MTD to get the valuable knowledge on ICDAR 2015. After that, transfer these weights to our main task: scene text detection under occlusion in scene images.

3.2 Multi-Scale Adaptive Deep Network

As illustrated in Figure 2, the input image (736x736) is fed into the feature extraction by ResNet18 [22] with pixel ratios 1/4, 1/8, 1/16, 1/32 corresponding F_1 , F_2 , F_3 , and F_4 . To reduce the time-consuming while keeping the general feature information, as PAN [1], we reduce the number of channels of each feature map to 128 by convolutional kernel 1x1. However, due to a lightweight backbone, features are often weak representation capabilities, so we apply CBAM attention [16], which perfectly illustrates the effectiveness of capturing spatial attention (SAM) along with channel attention (CAM). Thus, we can obtain richer feature representations. Then, MFEN is capable of the receptive field enhancement and different resolutions of text regions perception due to simultaneously progress with three different scales convolution kernels 3x3, 5x5, and 7x7. In the details, the structure of each MFEN is based on MobileNetv2 [23], which uses depthwise separable convolution (depthwise convolution 3x3 and pointwise convolution 1x1). Thus, the spatial information on features F_{1_n} , F_{2_n} , F_{3_n} , F_{4_n} (n is kernel size) are captured more adequately. Afterward, to prepare for predicting task, features of different depths are integrated into an enriched feature F_c by upsampling and concatenating extracted features. Finally, we inherited the post-processing of PAN [1] that detects text instances followed by pixel aggregation algorithms. This method clusters the neighbor pixels and merges them in the iterating process; consequently, text kernel is gradually expanded to text region.

$$F_{2_n} = \text{Upsample}(F_{2_n}|F_{1_n}) \quad (1)$$

$$F_{3_n} = \text{Upsample}(F_{3_n}|F_{1_n}) \quad (2)$$

$$F_{4_n} = \text{Upsample}(F_{4_n}|F_{1_n}) \quad (3)$$

$$F_c = \text{Concatenate}((F_{1_n}, F_{2_n}, F_{3_n}, F_{4_n})|1) \quad (4)$$

3.3 Loss Function

Our loss function L can be formulated as a weighted sum of the loss for text region, text kernel and sum of loss for similarity vector by segmentation network:

$$L = L_{reg} + \alpha L_{ker} + \beta(L_{agg} + L_{disc}) \quad (5)$$

where L_{reg} , and L_{ker} define loss of text regions and text kernels as Eq. 6, Eq. 7, respectively. L_{agg} , and L_{disc} are aggregation loss and discrimination loss of

post-processing stage as in PAN in Eq. 8, Eq. 9. According to the numeric values of the losses, $\alpha = 0.5$, $\beta = 0.25$ are two constants selecting to keep the balance among these losses.

In more details, prediction of text regions and text kernels are basically a pixel-wise classification text or non-text problem, so we apply dice loss [24] to handle these works.

$$L_{reg} = \sum_i Dice(P_{reg}, G_{reg}) \quad (6)$$

$$L_{ker} = \sum_i Dice(P_{ker}, G_{ker}) \quad (7)$$

where P_{reg} , G_{reg} are the prediction and ground truth of text region, respectively. P_{ker} , G_{ker} are the prediction and ground truth of text kernel.

Besides, in post-processing, we adopt loss function from PAN by using aggregation loss L_{agg} and L_{dis} as shown below:

$$L_{agg} = \frac{1}{N} \sum_{j=1}^N \frac{1}{|T_j|} \sum_{pix \in T_j} \ln(D(pix, T_{ker_j}) + 1) \quad (8)$$

where N , T_j define the number and j th of text instances. The distance between text pixel pix and kernel j th T_{ker_j} of the same instance should be small, which is denoted by $D(pix, T_{ker_j})$. This function is calculated by maximum of similarity vector between pix and T_{ker} . This function is set with a constant 0.5 as PAN experimentally.

In addition to this, to reduce overlap among text regions, the text instance vectors should keep apparently discrimination. In training phase, PAN implemented this discrimination loss L_{disc} as below:

$$L_{disc} = \frac{1}{N_j N_k} \sum_{j=1}^N \sum_{k=1}^N \ln(D(T_{ker_j}, T_{ker_k}) + 1) \quad (9)$$

Similar to aggregation loss, where N , $D(T_{ker_j}, T_{ker_k})$ define the number of text instances, the distance between the text kernel T_{ker_j} and the text kernel T_{ker_k} , respectively, corresponding j th, k th.

4 Experimental Results

ICDAR 2015 [25] is the incidental scene text of challenge four on the website <https://rrc.cvc.uab.es/?ch=4>. It consists of 1000 incidental natural images for training process and 500 images for testing set. ICDAR 2015 dataset is one of the popular datasets for scene text detection, including word-level text instances with multi-oriented texts.

ISTD-OC [8], named Incidental Scene Text Dataset - Occlusion, was conducted for occluded text detection and recognition task in workshop CBDAR

Algorithm 1 Realistic Occluded Text Detection (ROTD) dataset

Input: ICDAR 2015 images
Method: OpenCV
Output: Occluded text images

```

1: while Bounding box (bbox) has described-text: do
2:   Find location of bbox with described-text ( $x_a, y_a$ ).
3:   Get color value (0 – 255) of a random pixel inside the bounding box above.
4:   if pixel=255 then
5:     pixel=0
6:   else
7:     pixel=pixel/255
8:   end if
9:   Initial: Set n, r, N
10:  Draw shape with initial parameters and the color value got from Step 1.
11:  Save occluded images.
12: end while
```

2021. ISTD-OC contains the rectangle shape for ICDAR 2015 benchmark with different levels of occlusions from zero to a hundred percent. The number of images is 1500 occluded images for detection, but only 500 evaluation images are published.

We create a novel Realistic Occluded Text Detection (ROTD) dataset in Algorithm 1 with two steps: The first, find the color of a random pixel inside the box, providing localization in Ground Truth of ICDAR 2015. In the second step, draw the arbitrary shape inside the bounding box with the color value obtained above. The arbitrary shape is initialized with the number of possibly sharp edges $n = 7$, the magnitude of the perturbation from unit circle $r = 0.7$, and the number of points in the path $N = n*3+1$, experimentally. The difference between our proposed dataset and ISTD-OC is incidental shape and color, making the part of missing texts look real as in Figure 5. As the number of ICDAR 2015 images, our proposed dataset includes 1000 training and 500 testing images.

In this work, to help model understand deeply context of texts and even occlusion texts, we choose random only 10% training images following our experimental results (the highest F1-score performance 67.21%) in Figure 6 and associate with original ICDAR 2015 training set during training stage, totally 1100 images (1000 original ICDAR 2015 images and 100 occluded text ROTD images). Additionally, to compare fairly with ISTD-OC, as proved in paper [8], we selected 70% occlusion on ISTD-OC as a standard testing set for our evaluation.

The comparison results with previous methods on occluded text ISTD-OC, ICDAR 2015 benchmark, and our own ROTD dataset are demonstrated in Table 1, Table 2, and Table 3, respectively. As shown, our method is superior for 70% occluded text detection on ISTD-OC by 69.6%, 78.7% on ROTD dataset, and

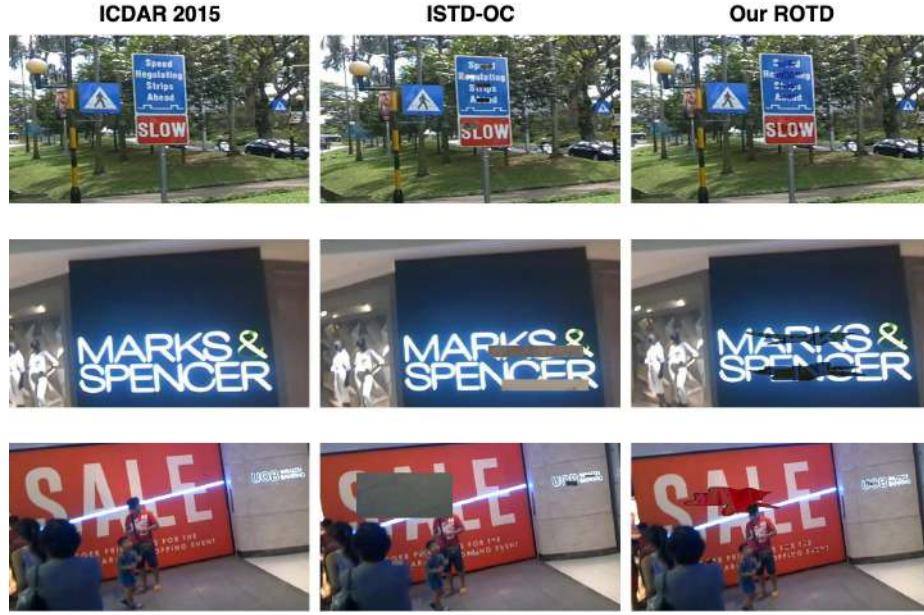


Fig. 5. ICDAR 2015 is represented for normal scene texts, ISTD-OC is occluded text images with rectangle shape, our ROTD is occluded text images with arbitrary shape.

performs better in detecting the text instances on ICDAR 2015 by 82.4%. Several visualizations are shown in Figure 7.

Table 1. Comparison of Occluded Text Detection on ISTD-OC. *≈ is represented the results from the mentioned graph performances in [8].

Method	Precision	Recall	F-score
PAN [1]	≈ 64	≈ 44	≈ 61
PSE-Net [11]	≈ 58	≈ 52	≈ 62
EAST [13]	≈ 43	≈ 51	≈ 60
PAN [1] (1100 images)	78.5	57.7	66.5
Ours (1100 images)	77.7	63.0	69.6

5 Conclusion

In this paper, we have presented a method for addressing text bounding boxes splitting by the occlusion phenomena problem with three stages: We first focus on perceiving attention information from Guided Grad-CAM++ maps to prepare knowledge for the main task. Next, we employ a novel MTD network, which

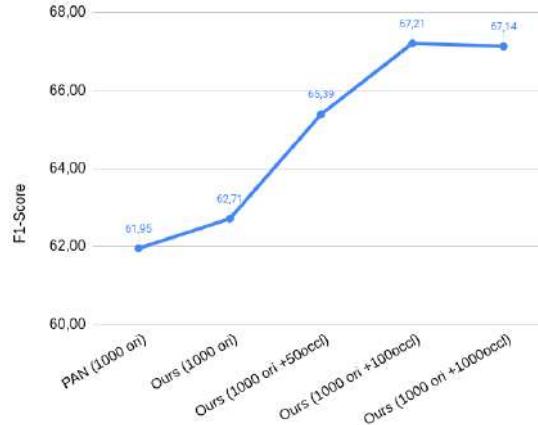


Fig. 6. This graph shows the comparison of F1-score performances of proportional scene text occlusion images.

Table 2. Comparison of state-of-the-art scene text detection on ICDAR 2015 without external data.

Method	Precision	Recall	F-score
PAN [1]	82.9	77.8	80.3
PSE-Net [11]	81.5	79.7	80.6
EAST [13]	83.6	73.5	78.2
MFEN [14]	84.5	79.7	82.0
Ours (1100 images)	85.8	79.3	82.4

Table 3. Comparison of our model with validation on ROTD dataset.

Method	Precision	Recall	F-score
PAN [1]	70.1	50.2	58.5
Ours (1000 images)	72.2	55.0	62.4
Ours (1100 images)	82.1	75.6	78.7

is capable of enlarging the receptive fields and enriching feature representations while bringing minor extra computation. Finally, to aware text occlusion knowledge, we conduct a new dataset ROTD, and combine a part of them with original images (ICDAR 2015) for training process. Therefore, the proposed method outperforms the state-of-the-art on ISTD-OC dataset. Extensive experiment on ICDAR 2015 shows the competitive result compare to other recent methods.

6 Acknowledgements

In the future, This research was supported by the Bio & Medical Technology Development Program of the National Research Foundation (NRF) & funded by the Korean government (MSIT) (NRF-2019M3E5D1A02067961) and by Basic

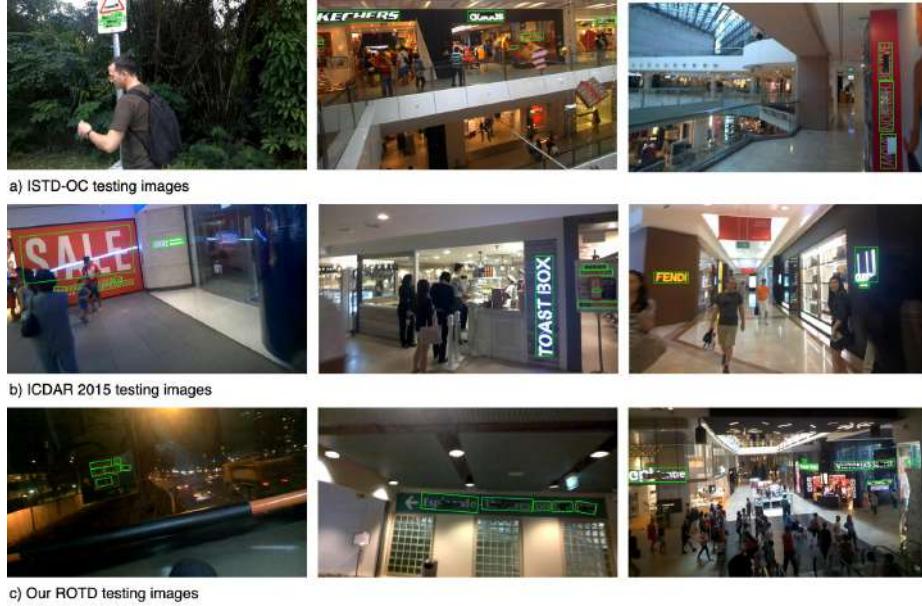


Fig. 7. Visual comparisons of bounding box representations for text detection on three sets: ISTD-OC, ICDAR 2015 and our ROTD.

Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2018R1D1A3B05049058 & NRF-2020R1A4A1019191).

References

1. Wang Wenhai, Xie Enze, Song Xiaoge, Zang Yuhang, Wang Wenjia, Lu Tong, Yu Gang, Shen Chunhua, “Efficient and accurate arbitrary-shaped text detection with pixel aggregation network,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8440–8449, 2019.
2. Liao Minghui, Zou Zhisheng, Wan Zhaoyi, Yao Cong, Bai Xiang, “Real-time scene text detection with differentiable binarization and adaptive scale fusion,” in IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022.
3. Zhang Shi-Xue, Zhu Xiaobin, Chen Lei, Hou Jie-Bo, Yin Xu-Cheng, “Arbitrary Shape Text Detection via Segmentation with Probability Map,” in IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022.
4. Tang Jingqun, Zhang Wenqing, Liu Hongye, Yang MingKun, Jiang Bo, Hu Guanglong, Bai Xiang, “Few Could Be Better Than All: Feature Sampling and Grouping for Scene Text Detection,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4563–4572, 2022.
5. Yin Xu-Cheng, Yin Xuwang, Huang Kaizhu, Hao Hong-Wei, “Robust text detection in natural scene images,” in IEEE transactions on pattern analysis and machine intelligence, pp. 970–983, 2013.

6. Chen Zhe, Wang Wenhai, Xie Enze, Yang ZhiBo, Lu Tong, Luo Pin, “FAST: Searching for a Faster Arbitrarily-Shaped Text Detector with Minimalist Kernel Representation,” in arXiv preprint arXiv:2111.02394, 2021.
7. Mittal Ayush, Shivakumara Palaiahnakote, Pal Umapada, Lu Tong, Blumenstein Michael, “A new method for detection and prediction of occluded text in natural scene images,” in Signal Processing: Image Communication, pp. 116512, 2022.
8. Geovanna Soares Aline, Leite Dantas Bezerra Byron, Baptista Lima Estanislau, “How Far Deep Learning Systems for Text Detection and Recognition in Natural Scenes are Affected by Occlusion?,” in International Conference on Document Analysis and Recognition, pp. 198–212, 2021.
9. Zhou Bolei, Khosla Aditya, Lapedriza Agata, Oliva Aude, Torralba Antonio, “Learning deep features for discriminative localization,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2921–2929, 2016.
10. Mittal Ayush, Shivakumara Palaiahnakote, Pal Umapada, Lu Tong, Blumenstein Michael, Lopresti Daniel, “A new context-based method for restoring occluded text in natural scene images,” in International Workshop on Document Analysis Systems, pp. 466–480, 2020.
11. Wang Wenhai, Xie Enze, Li Xiang, Hou Wenbo, Lu Tong, Yu Gang, Shao Shuai, “Shape robust text detection with progressive scale expansion network,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9336–9345, 2019.
12. Baek Youngmin, Lee Bado, Han Dongyoon, Yun Sangdoo, Lee Hwalsuk, “Character region awareness for text detection,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9365–9374, 2019.
13. Zhou Xinyu, Yao Cong, Wen He, Wang Yuzhi, Zhou Shuchang, He Weiran, Liang Jiajun, “East: an efficient and accurate scene text detector,” in Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp. 5551–5560, 2017.
14. Dinh My-Tham and Lee Guee-Sang, “Arbitrary-shaped Scene Text Detection based on Multi-scale Feature Enhancement Network,” in Korea Computer Congress, pp. 669–671, 2022.
15. Chattopadhyay Aditya, Sarkar Anirban, Howlader Prantik, Balasubramanian Vireeth N , “Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks,” in 2018 IEEE winter conference on applications of computer vision (WACV), pp. 839–847.
16. Woo Sanghyun, Park Jongchan, Lee Joon-Young, Kweon In So, “Cbam: Convolutional block attention module,” in Proceedings of the European conference on computer vision (ECCV), pp. 3–19, 2018.
17. Dai Pengwen, Zhang Sanyi, Zhang Hua, Cao Xiaochun, “Progressive contour regression for arbitrary-shape scene text detection,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 7393–7402, 2021.
18. He Wenhao, Zhang Xu-Yao, Yin Fei, Liu Cheng-Lin, “Deep direct regression for multi-oriented scene text detection,” in Proceedings of the IEEE international conference on computer vision, pp. 745–753, 2017.
19. Sheng Tao, Chen Jie, Lian Zhouhui, “Centripetaltext: An efficient text instance representation for scene text detection,” in Advances in Neural Information Processing Systems, pp. 335–346, 2021.
20. Tian Zhuotao, Shu Michelle, Lyu Pengyuan, Li Ruiyu, Zhou Chao, Shen Xiaoyong, Jia Jiaya, “Learning shape-aware embedding for scene text detection,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 4234–4243, 2019.

21. Selvaraju Ramprasaath R, Cogswell Michael, Das Abhishek, Vedantam Ramakrishna, Parikh Devi, Batra Dhruv, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in Proceedings of the IEEE international conference on computer vision, pp. 618–626, 2017.
22. He Kaiming, Zhang Xiangyu, Ren Shaoqing, Sun Jian, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778, 2016.
23. Howard Andrew G, Zhu Menglong, Chen Bo, Kalenichenko Dmitry, Wang Weijun, Weyand Tobias, Andreetto Marco, Adam Hartwig, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," in arXiv preprint arXiv:1704.04861, pp. 770–778, 2017.
24. Sudre Carole H, Li Wenqi, Vercauteren Tom, Ourselin Sebastien, Jorge Cardoso M, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," in Deep learning in medical image analysis and multimodal learning for clinical decision support, pp. 240–248, 2017.
25. Karatzas Dimosthenis, Gomez-Bigorda Lluis, Nicolaou Anguelos, Ghosh Suman, Bagdanov Andrew, Iwamura Masakazu, Matas Jiri, Neumann Lukas, Chandrasekhar Vijay Ramaseshan, Lu Shijian and others, "ICDAR 2015 competition on robust reading," in 2015 13th international conference on document analysis and recognition (ICDAR), pp. 1156–1160, 2015.
26. Deng Jia, Dong Wei, Socher Richard, Li Li-Jia, Li Kai, Fei-Fei Li, "Imagenet: A large-scale hierarchical image database," in 2009 IEEE conference on computer vision and pattern recognition, pp. 248–255, 2009.
27. Dang Quang-Vinh, Lee Guee-Sang, "Document image binarization with stroke boundary feature guided network," in IEEE Access, pp. 36924–36936, 2021.
28. Wu Yonghui, Schuster Mike, Chen Zhifeng, Le Quoc V, Norouzi Mohammad, Macherey Wolfgang, Krikun Maxim, Cao Yuan, Gao Qin, Macherey Klaus and others, "Google's neural machine translation system: Bridging the gap between human and machine translation," in arXiv preprint arXiv:1609.08144, 2016.
29. Dang Quang-Vinh, Lee Guee-Sang, "Document Image Binarization by GAN with Unpaired Data Training," in International Journal of Contents, pp. 8–18, 2020.
30. Wang Wenhai, Xie Enze, Li Xiang, Liu Xuebo, Liang Ding, Yang Zhibo, Lu Tong, Shen Chunhuag, "Pan++: Towards efficient and accurate end-to-end spotting of arbitrarily-shaped text," in IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 5349–5367, 2021.
31. Aberdam Aviad, Litman Ron, Tsiper Shahar, Anschel Oron, Slossberg Ron, Mazor Shai, Manmatha R, Perona Pietro, "Sequence-to-sequence contrastive learning for text recognition," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15302–15312, 2021.
32. Xu Xingqian, Zhang Zhifei, Wang Zhaowen, Price Brian, Wang Zhonghao, Shi Humphrey, "Rethinking text segmentation: A novel dataset and a text-specific refinement approach," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12045–12055, 2021.
33. Deng Dan, Liu Haifeng, Li Xuelong, Cai Deng, "Pixellink: Detecting scene text via instance segmentation," in Proceedings of the AAAI Conference on Artificial Intelligence, 2018.

Detection and Tracking of Flying Small Bats under Complex Backgrounds

Kakeru Sugimoto¹, Kazusa Usio², Ryota Sugimori², Emyo Fujioka³,
Hiroaki Kawashima⁴, Shizuko Hiryu⁵, and Hitoshi Habe^{6,7}

¹ Graduate School of Science and Engineering, Kindai University, Japan

² Graduate School of Life and Medical Sciences, Doshisha University, Japan

³ Organization for Research Initiatives and Development, Doshisha University, Japan

⁴ School of Social Information Science, University of Hyogo, Japan

⁵ Faculty of Life and Medical Sciences, Doshisha University, Japan

⁶ Department of Informatics, Faculty of Informatics, Kindai University, Japan

⁷ Cyber Informatics Research Institute, Kindai University, Japan

Abstract. Bats emit ultrasonic waves during their flight and listen to the echoes to understand their surroundings. To understand the unique ecology of bats, various efforts have been made. Among them, computer-aided automatic detection and tracking would play an important role. This enables us to analyze the movement precisely. Bats are nocturnal, small in size, and move at high speeds, which are pretty unfavorable conditions for detection and tracking. However, the state of the art of multiple object tracking (MOT) methods yields better performance in object detection and tracking. In this paper, we use YOLOv7, a new version of YOLO, for object detection and OC-SORT, a kind of MOT method, for object tracking and compare the accuracy of each method with other existing methods. The video images used in this study have complex backgrounds, and the accuracy of detection could be low because bats are assimilated into the background. Additionally, the shadows of bats are miss-detected as bats. To cope with such difficult situations, we first calculate the inter-frame differences to extract moving objects clearly and then detect the shadows of bats as another object class to avoid the miss-detection of bat shadows as bats. We finally compare the difference in performance with the actual video of flying bats.

Keywords: Object Detection · Moving Object Tracking · YOLO · SORT · Small Object · Complex Background

1 Introduction

Bats are the only mammals that can fly. Bats, which are not very visible, emit ultrasonic waves themselves and listen to their echoes to understand their surroundings and targets. This feature is called echolocation, and efforts are underway to elucidate its mechanism[1].

However, because bats are nocturnal, the images are often dark. Additionally, the background is often complex, and because bats are small and move at high

2 Authors Suppressed Due to Excessive Length

speed, they may blend in with the background or become blurred. Therefore, it is difficult and time-consuming for the human eyes to distinguish between bat images.

In addition, when capturing video outside, we need a light source to lit bats under dark conditions. This often results in the shadow of the bat, making it difficult to distinguish it from the actual bat. The sample of the actual video used in this study is shown in Fig. 1.

The goal of this study is to improve the accuracy of bat detection and tracking. Our method is based on YOLOv7[2] developed by Chien-Yao Wang et al. for detection and OC-SORAT[3] developed by Jinkun Cao et al. for tracking. To achieve the goal, we first extract bat regions under complex backgrounds using inter-frame differences. Then, we apply two-class detection, not only detecting actual bats but also their shadows as another class of objects. This enables us to reduce the false detection of shadows as bats. We think that these methods can be applied to any data if they can correctly detect and track complex images that are small, dark, and fast, such as the images used in this study.

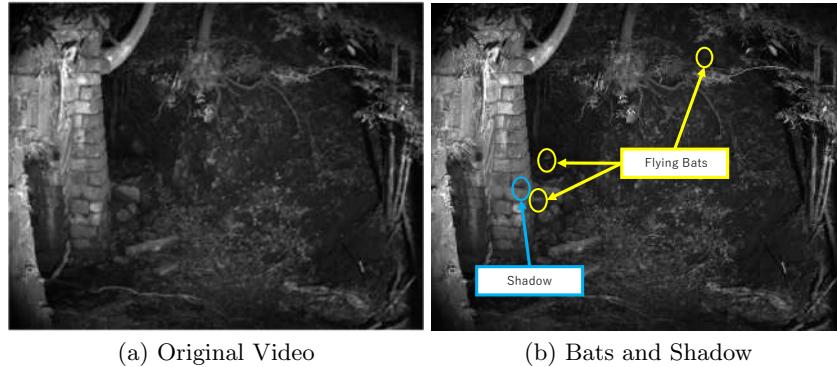


Fig. 1. Example of Actual Video

2 Related Work

2.1 Measurement with Microphone Arrays

Fujioka et al. [4] simultaneously measured the flight routes of wild bats and the direction of ultrasonic pulse emission using microphone arrays to elucidate the mechanism of echolocation, but the microphone arrays have limitations in their observable environment, such as ultrasonic interference when multiple bats are flying and limited space for observation.



Fig. 2. Overview of the proposed method

2.2 Object detection by Background Subtraction

Background subtraction[5–7] can be used for separating the background and foreground portions. Additionally, background subtraction[8][9][10] using a mixed Gaussian model can incorporate complex background and adaptively updates the model.

In our previous study, we extracted and detected moving objects by background subtraction for bat videos and then tracked them using nearest neighbor search and motion model[11]. However, this method was not sufficiently accurate in some areas and required tuning for each application environment.

3 Proposed Method

This section describes our proposed method.

3.1 Overview

As described earlier, our method is based on YOLO and OC-SORT. In addition to those methods, we apply inter-frame subtraction to extract small moving objects, and we perform two-class detection: actual bats and their shadows. This would decrease the false detection of shadows as actual bats. The flow of this method is as follows and shown in Fig. 2:

1. Inter-frame subtraction
2. Two-class bat detection by YOLO
3. Bat tracking and ID association by OC-SORT

Each process will be described in the subsequent sections.

3.2 Inter-frame Subtraction

Inter-frame subtraction is used to extract the position of flying bats, to improve the detection accuracy of bats. More specifically, at time t , the difference between

the images at t and $t - 1$ and the difference between the images at $t + 1$ and t are computed from three consecutive images. Then, the disjunction of the two subtracted images is calculated to obtain an image in which the moving object is emphasized. Since the resulting image generates salt and pepper noise, the salt and pepper noise is removed by shrinking and expanding the image. Finally, the brightness of the original image is increased by 30 at pixels where moving objects are detected and decreased by 100 at other pixels. This image is used in the subsequent detection process.

3.3 Two-class Bat Detection by YOLO

YOLOv7 is a detection method in YOLO, developed in 2022 by the same group as YOLOv4, Scaled-YOLOv4, and YOLOR. This method is reported to outperform existing methods on the MS COCO dataset significantly[2]. We expect it also yields good results in this study. We also use YOLOv5 for detecting bats. YOLOv5 is an older method but is widely used for detection.

As mentioned earlier, we perform two-class detection: actual bats and their shadows. As shown in Fig. 1(b), flying bats and their shadows are quite similar in their appearance. If we train the object detection model so that only the bat region will be detected, may false detection and miss detection would happen. To cope with this issue, we train the model to detect the actual bats and shadows as different classes. This makes the detection model able to distinguish between the two objects and suppress false detections.

3.4 Bat tracking and ID association by OC-SORT

Observation-Centric SORT (OC-SORT), created by Cao et al. in 2022, is an object tracking method that is robust to occlusion and nonlinear motion while maintaining the simple, real-time, online approach characteristic of traditional SORT [3]. OC-SORT solves the problems of traditional SORT[12] and achieves state-of-the-art performance in the latest MOT benchmarks. OC-SORT created a robust model for occlusion and nonlinear motion with the following three techniques: Observation-centric Online Smoothing, Observation-Centric Momentum, and Observation-Centric Recovery.

4 Experiments

We conduct three kinds of experiments to evaluate and compare each component of the proposed method. The details and results of each experiment will be described in the followings.

The training data in all experiments contains 300 images of 10 seconds at the same scene. The number of epochs in training was 300.

4.1 Exp1: Inter-Frame Subtraction

In the first experiment, we examine the effect of inter-frame subtraction. To this end, we compare the accuracy of bat detection with inter-frame subtraction and without it. The object detection method is YOLOv5[13].

Fig.3 shows an example of bat detection results by YOLOv5, with and without inter-frame subtraction. Table 1 summarizes the evaluation results. As shown in Table 1, the inter-frame difference improved both recall and mAP by about 16 points. However, the recall value is still low at 56%. The main reason for this is that the value of the inter-frame difference was smaller for bats with small movements and bats moving in the depth direction.

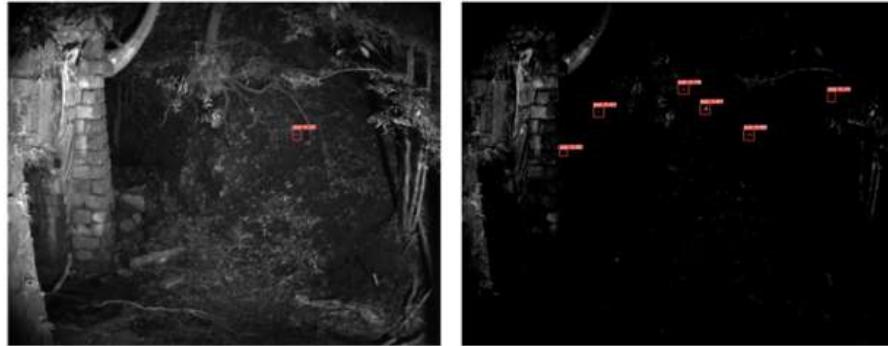


Fig. 3. Exp1: Detection results

Table 1. Exp1: Performance comparison of inter-frame subtraction(YOLOv5)

	Precision	Recall	mAP50
w/o inter-frame subtraction	88.0%	39.9%	45.5%
w/ inter-frame subtraction	83.1%	55.9%	61.4%

4.2 Exp2: Two-Class Bat Detection

In the second experiment, we see the effect of the two-class bat detection, i.e., bats and shadows are treated as different classes. We will compare the performance of bat detection using the two-class setting and the standard single-class setting, which aim to detect bats only.

Also, we compare two object detection models: YOLOv5 and YOLOv7. Although YOLOv7 is newer model than YOLOv5, it is worth evaluating which is better for our application domain.

Table 2 summarizes the results. The results show that the two-class bat detection strategy is effective, as both models give higher performance. Especially for YOLOv7, all evaluations were more than 10 points higher than in the one-class setting. However, this may not be true for all results since only the recall value dropped in YOLOv5.

Also, when comparing the two detection models: YOLOv5 and YOLOv7, contrary to expectations, YOLOv5 gives better results. This implies that YOLOv7 shows better results for standard data, such as people, dogs, and cars, but not for all data.

Table 2. Exp2: Effect of two-class bat detection

Model	Detection Method	Precision	Recall	mAP50
YOLOv5	One-Class	83.1%	55.9%	61.4%
	Two-Class	83.8%	55.5%	65.6%
YOLOv7	One-Class	55.8%	47.1%	45.1%
	Two-Class	79.1%	57.0%	61.4%

4.3 Exp3: Bat Tracking

In the third experiment, we will evaluate the results of tracking bats using OC-SORT. For OC-SORT we use the YOLOv5 model which gives the best detection performance among all the detection results we have tested. The comparison is made between the ByteTrack[14] and OC-SORT.

The observations from the experimental results are (1) OC-SORT reduced the number of false track detections of overlapping bats, and is more accurate than ByteTrack, (2) OC-SORT is highly dependent on detection same as ByteTrack. Hence, if there is a missing detection in the middle of a trajectory, it may be regarded as a different individual, or terminate the tracking. One example of the individuals tracked by OC-SORT is shown in Fig.4. The coordinate axes of the graph are aligned with the actual image size.

5 Conclusions

To detect and track flying bats under complex backgrounds, we apply YOLO and OC-SORT. In addition to the standard method, we perform the inter-frame subtraction to extract flying bats even when the bats are small and fast and the backgrounds are complex. Also, we conduct two-class bat detection: bats

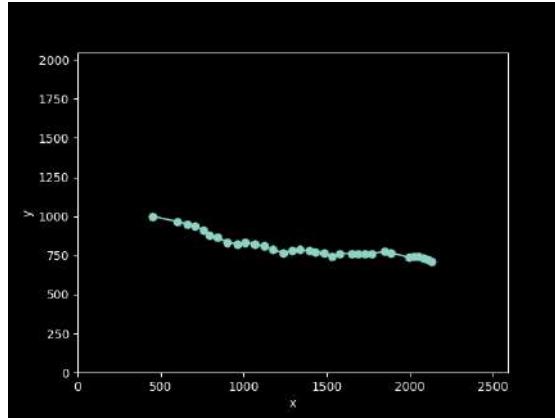


Fig. 4. An example of tracking results

and their shadows are differently detected. Experimental results demonstrate the proposed framework works well for the actual video containing flying bats.

Future work includes improving the detection so that bats with small movements. At the tracking stage, it is necessary to cope with missing detection in the middle of a trajectory.

This work was partly supported by JSPS KAKENHI JP21H05302.

References

1. K.Hase et al., Bats enhance their call identities to solve the cocktail party problem, Communications Biology, volume1, Article number: 39 (2018)
2. Chien-Yao Wang, Alexey Bochkovskiy, Hong-Yuan Mark Liao "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors" arXiv:2207.02696 (2022).
3. Cao, Jinkun, Weng, Xinshuo, Khirodkar, Rawal, Pang, Jiangmiao, Kitani, Kris. "Observation-centric sort: Rethinking sort for robust multi-object tracking" arXiv preprint arXiv:2203.14360 (2022).
4. E. Fujioka, I. Aihara, M. Sumiya, K. Aihara, and S. Hiryu. Echolocating bats use future-target information for optimal foraging. In PNAS, 4 (2016).
5. A.Elgammal, D.Harwood, and L.S.Davis, "Non-parametric background model for background subtraction" In Proceedings6th ECCV, (2000).
6. C.Stauffer, W.E.L.Grimson "Adaptive background mixture models for real-time tracking" Proceedings 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (1999).
7. O.Barnich, M.V.Droogenbroeck "ViBe: A Universal Background Subtraction Algorithm for Video Sequences" IEEE Transactions on Image Processing, Vol. 20, Issue .6, June (2011)
8. P. KaewTraKulPong, R. Bowden "An Improved Adaptive Background Mixture Model for Realtime Tracking with Shadow Detection" In: P.RemagninoGraeme, A.JonesNikos, ParagiosCarlo S. Regazzoni(eds.) Video-Based Surveillance Systems, pp.135-144

8 Authors Suppressed Due to Excessive Length

9. Z.Zivkovic "Improved adaptive Gaussian mixture model for background subtraction", Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.
10. Z.Zivkovic, F.Heijden "Efficient adaptive density estimation per image pixel for the task of background subtraction" In: T.K.Ho, Murray Hill, G. Sanniti et.al, Pattern Recognition, Letters Volume 27, Issue 7, May 2006, Pages.773-780
11. E. Fujioka, M. Fukushiro, K.Ushio, K. Kohyama, H. Habe, S. Hiryu, "Three-Dimensional Trajectory Construction and Observation of Group Behavior of Wild Bats during Cave Emergence." Journal of Robotics and Mechatronics 33 (3): pp. 556–63, 2021.
12. Bewley, Alex, Ge, Zongyuan, Ott, Lionel, Ramos, Fabio, Upcroft, Ben. "simple online and realtime tracking". In 2016 IEEE International Conference on Image Processing (ICIP), pp. 3464–3468, (2016).
13. Ultralytics YOLOv5: <https://github.com/ultralytics/yolov5>. Last accessed 30 Nov 2022
14. Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, Xinggang Wang. "ByteTrack: Multi-Object Tracking by Associating Every Detection Box" arXiv 2110.06864 (2022)

Facial Depth and Normal Estimation using Single Dual-Pixel Camera*

Minjun Kang^{1†} Jaesung Choe¹ Hyowon Ha⁴ Hae-Gon Jeon²
Sunghoon Im³ In So Kweon¹ Kuk-Jin Yoon¹

¹KAIST ²GIST ³DGIST ⁴Meta Reality Labs
†kmmj2005@kaist.ac.kr

Abstract. Recently, Dual-Pixel (DP) sensors have been adopted in many imaging devices. However, despite their various advantages, DP sensors are used just for faster auto-focus and aesthetic image captures, and research on their usage for 3D facial understanding has been limited due to the lack of datasets and algorithmic designs that exploit parallax in DP images. In this paper, we introduce a DP-oriented Depth/Normal estimation network that reconstructs the 3D facial geometry. In addition, to train the network, we collect DP facial data with more than 135K images for 101 persons captured with our multi-camera structured light systems. It contains ground-truth 3D facial models including a depth map and surface normal in metric scale. Our dataset allows the proposed network to be generalized for 3D facial depth/normal estimation. The proposed network consists of two novel modules: Adaptive Sampling Module (ASM) and Adaptive Normal Module (ANM), which are specialized in handling the defocus blur in DP images. Finally, our proposed method achieves state-of-the-art performances over recent DP-based depth/normal estimation methods.

Keywords: Dual-Pixel · Depth/Normal estimation · Face Reconstruction

1 Introduction

A huge number of facial images are posted every day on social media [16, 17]. Accordingly, acquiring facial geometry from images has emerged as an interesting research topic, since 3D facial geometry can be used for various applications [22, 5, 52, 33, 54, 50]. 3D facial geometry can be obtained by either using multiple cameras [14, 4] or active sensing devices [28, 26]. However, these methods often suffer from uncontrolled lighting conditions or hardware synchronization.

Recently, **Dual-Pixel (DP)** sensors get noticed due to their popularity in being installed in many portable imaging devices such as the iPhone13 ProMax and Samsung Galaxy 22 and their strengths of capturing two images perfectly synchronized with the same exposure, white balance, and geometric rectification. Based on these properties, currently, there have been few studies that explore the possibility of DP sensors for scene depth estimation [15, 36, 53, 35, 49]. Usually,

* This paper is the short version of ECCV'22 and is NEVER considered an official publication.

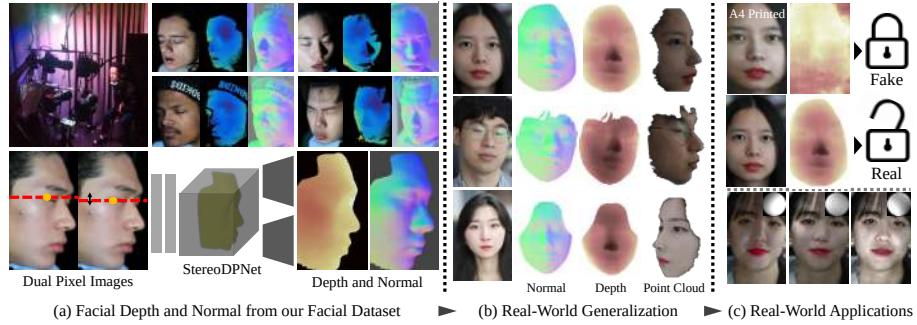


Fig. 1. Our method aims at the generalized estimation of unmet facial geometry, which can be used for various applications, such as face spoofing or relighting.

these studies regard DP images as extremely narrow-baseline stereo images having different defocus-blur to infer depth maps. Although the DP sensors are actively used to take face pictures, there has been a limited study [48] that recovers facial geometry using a Dual-Pixel camera. Previous methods have difficulty in facial geometry estimation, which is due to the lack of a facial DP dataset with precise 3D geometry and an appropriate algorithm for generalized estimation.

To address the issue, we present a DP-oriented 3D facial dataset and a depth/normal estimation network toward high-quality facial geometry reconstruction with DP cameras. We represent the 3D facial geometry not only with the depth map but also with the normal map for various applications such as face relighting. Our dataset involves 135,744 face data for 101 persons consisting of DP images and their corresponding depth maps and surface normal maps, which are captured by our structured light camera system. Based on these data, we train our depth/normal estimation network, called stereoDPNet, to infer 3D facial information from DP images. In particular, our stereoDPNet is fully oriented from the properties of dual-pixel images that have an extremely small range of disparity with defocus-blur. Our network design carefully treats these distinctive properties through our Adaptive Sampling Module (ASM) and Adaptive Normal Module (ANM). Finally, the contributions are as follows:

- DP-oriented 3D facial dataset with more than 135K DP images and their corresponding high-quality 3D models.
- Novel depth/normal estimation network for facial 3D reconstruction from a DP image with better generalization.

2 Proposed Method

This paper covers dual-pixel based facial understanding: from data acquisition (Section 2.2) to general estimation by stereoDPNet (Section 2.3). Different from natural images from typical cameras, dual-pixel sensors capture images having an extremely small range of disparity as well as defocus-blur, as shown in Figure 2-(c). Through our carefully designed dataset and network, we design a well-generalized methodology that even can infer facial geometry from unmet DP facial images.

2.1 Preliminaries

Depth Estimation from Dual-Pixel. Dual-Pixel images can be considered as a pair of stereo images since the DP camera captures two sub-aperture images with small parallax. There exists an extremely small range of pixel discrepancy from the same scene point between these two images ($-4\text{px} \sim +4\text{px}$) and pioneer works [45, 15] introduce the affine relationship between metric depth and the defocus-disparity driven from the paraxial and thin-lens approximations.

$$\begin{aligned} d(x, y) &= \alpha \bar{b}(x, y) \\ &\approx \alpha \frac{Lf}{1 - f/g} \left(\frac{1}{g} - \frac{1}{Z(x, y)} \right) \\ &\triangleq A(L, f, g) + \frac{B(L, f, g)}{Z(x, y)}, \end{aligned} \quad (1)$$

Based on this property, there have been several works that estimate scene depth from Dual-Pixel by adopting simple U-Net [15], using additional stereo camera [53], parametrized point spread functions [36, 35], and multiplane image representation [49]. Compared to these works, our proposed cost-volume-based network provides better geometry by capturing this narrow range of defocus-disparity and converting to metric depth by using Equation (1).

Monocular Face Reconstruction.

In general, it is only available to reconstruct faces from monocular images with a limited assumption [46], many 3D face regression methods [39, 13, 18] rely on the prior knowledge of face morphable model [44, 7], facial keypoints/landmarks [13, 41, 6], and symmetric assumption [47]. Recently, generative model-based methods [9, 8] are also actively explored. However, these methods lead to failure with unmet conditions (*e.g.* extreme poses), and the biased result to the prior knowledge/training dataset.

2.2 Dual-Pixel Facial Dataset

Dataset Configuration. Given an array of multiple DP cameras, we capture various human faces with different expressions and light conditions. The dataset consists of 135,744 photos, which are a combination of 101 people, eight cameras, seven different lighting condition, four facial heading directions (left, right, center and upward), three facial expressions (normal, open mouth and frown), and two fixed distances of subjects from the camera array, as illustrated in Figure 2-(b). The distances between the camera array and subjects range from 80 cm to 110 cm.

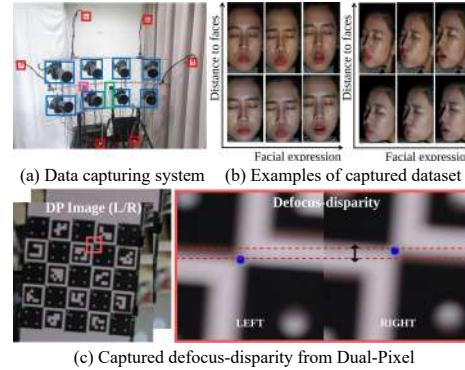


Fig. 2. Capturing face with Dual-Pixel. (a) (2×4 multi-camera array (blue), 6 LEDs (red), a projector (green), and a LED controller (magenta). (b) Examples of the captured facial dataset. (c) Example of captured defocus-disparity in DP images.

Since the focus distance is about 97 cm, our captured images contain both front focused and back focused cases. Our dataset includes 44,352 female photos as well as 91,392 male photos, ages range from 19 to 45. In main experiments, we use 76 people (76%) as a train set and the others (24%) as a test/validation set without any overlap with the train set.

Ground Truth Data Acquisition.

Structured light systems are designed for high-quality 3D geometry acquisition under controlled environments by projecting pre-defined patterns on surfaces of objects [40, 20, 12] and by analyzing the projected patterns to measure 3D shapes of the objects. It is extensively used for ground-truth depth maps in stereo matching benchmarks [24, 1, 43] and shape from shading [21]. In this work, we tailor the structured light-based facial 3D reconstruction method [19] with our well-synchronized multi-camera system. Thanks to our capturing system and structured light-based reconstruction method, we obtain dense, high-quality facial 3D corresponding to high-resolution DP images in Figure 1-(a). Moreover, we calibrate point light directions by using a chrome ball and applying a photometric stereo in [34] to obtain accurate surface normal maps of subjects' faces in Figure 3(d). We utilize the RANSAC algorithm in obtaining both surface normal and albedo for robust estimation by excluding severe specular reflection. By using the surface normals, initial depth is refined by conforming the initial facial depth and the surface normal [34], as illustrated in Figure 3(a), (b), and (c).

2.3 Facial Depth and Normal Estimation

Overall Architecture. Given DP images with left I^L and right I^R , stereoDPNet is trained to infer a disparity map \hat{d} and a surface normal map \hat{n} . To do so, first, the feature extraction layer infers DP image features F^L and F^R , respectively. Second, using F^L and F^R , the proposed ASM captures an amount of spatially varying blur in dynamic ranges, and then adaptively samples the features. Then, the sampled features G^L and G^R are stacked into a cost volume \mathcal{V} . Third, the cost volume is aggregated through three stacked hourglass modules to infer the aggregated cost volume C_A . Lastly, this aggregated volume C_A is used to regress a disparity map following the baseline and infer a surface normal map by ANM.

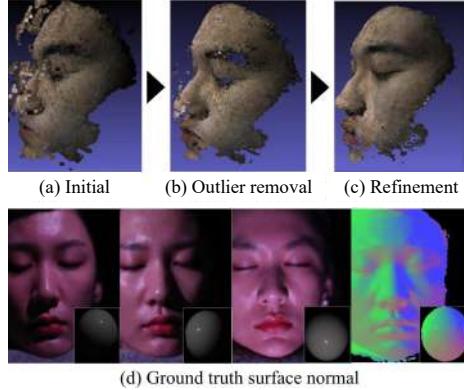


Fig. 3. Ground-truth depth and surface normal acquisition. (a) Initial depth from the structured light. (b) Depth after removing outliers. (c) Depth via fusion of the initial depth and the surface normal obtained from the photometric stereo in (c).

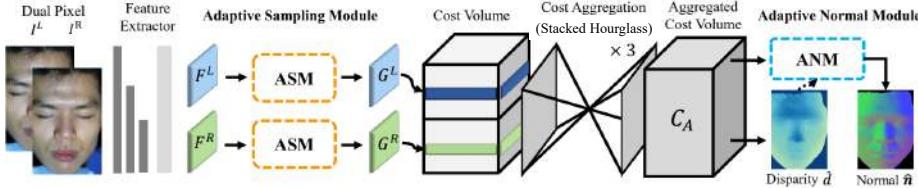


Fig. 4. Architecture of StereoDPNet. Given DP images, our network is trained to infer facial depth/normal maps. Our two key modules, Adaptive Sampling Module and Adaptive Normal Module overcome the extremely narrow baseline in DP images by capturing disparities in blurry regions. Note that pre-calibrated disparity to depth conversion using Equation (1) is used to get metric-scale depth and normal.

Adaptive Sampling Module. To cope with narrow disparity range and defocus-blur in DP, we design ASM in Figure 5 inspired by defocus blur matching method [11] and depth from light-field image [25].

According to Jeon *et al.* [25], the sub-pixel shift from different sampling strategies to construct cost volume for matching provides varying results depending on the local scene configurations. To take advantage of various conventional sampling methods, we incorporate them into ASM. The dynamic sampling layer in ASM is designed with a combination of nearest-neighbor, bilinear, and phase-shift interpolation, which can have various receptive fields to find varying blur sizes and can obtain subpixel-level shifted features of F^L and F^R . To this end, the shifted features from the three different sampling strategies are concatenated into a volumetric feature \mathcal{V} .

To extract useful features from \mathcal{V} , we design a self-3D attention layer. Our self-3D attention layer adaptively selects sampling strategies using attention map \mathcal{W} to include prominent texture information in an extracted feature map. Finally, the sampled features with the sub-pixel shift, G^L and G^R , are obtained by averaging the sampled volume \mathcal{V}_S . The matching cost volume, constructed from the selected feature maps (G^L, G^R), contains rich texture information with relative blur and performs effective matching in homogeneous regions as well [11].

Adaptive Normal Module. We design ANM in Figure 6 to produce a surface of human faces complementary to an estimated defocus-disparity map.

According to [29], an accurately aggregated cost volume contains an implicit function representation of underlying surfaces for depth estimation. Since the surface normal mainly depends on the shape of the local surface,

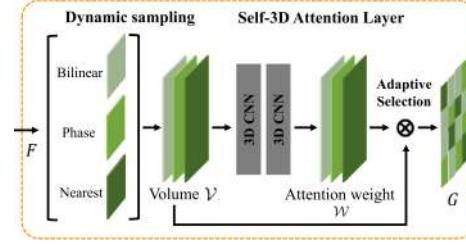


Fig. 5. Adaptive Sampling Module (ASM) consists of a dynamic sampling and a self-3D attention layer.

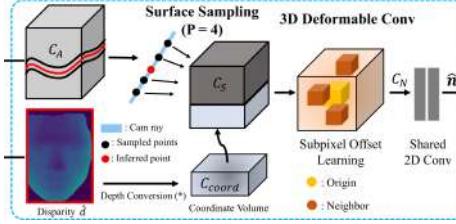


Fig. 6. Adaptive Normal Module (ANM) regresses surface normal by surface sampling and a 3D deformable CNN.

it is redundant to use all voxel embeddings in C_A for facial normal estimation. We thus sample the P candidates of hypothesis planes among M planes from the aggregated volume C_A using the estimated disparity map (Equation (2)). Since the surface normal is defined with the metric scale depth, we convert disparity to a depth map using pre-calibrated Equation (1) and provide this volumetric information with our network denoted as coordinate volume C_{coord} .

Since a human face has a variety of curved local surfaces, we need to consider dynamic ranges of neighbors to extract a local surface from the sampled hypothesis planes C_S in the previous stage. To do this, we follow the assumption of local plane in [31, 37, 32] and form local planes by a small set of neighbor points. Since these local patches have arbitrary shapes and sizes composed with its sampled neighboring points, we use 3D deformable convolutions [51] to consider the neighboring points within the dynamic ranges. The learnable offsets of the deformable convolution in 3D space allow us to adaptively sample neighbors and find the best local plane. The final feature volume C_N is predicted after passing two 3D deformable convolution layers to extract surface normal $\hat{\mathbf{n}}$.

2.4 Loss Functions

The aggregated volume C_A passes through a classifier to produce a final matching cost \mathcal{A} , and the softmax function $\sigma(\cdot)$ is applied to regress the defocus-disparity \hat{d} . Accordingly, we compute the disparity as follows:

$$\hat{d}_{u,v} = \sum_{m=1}^M d^m \cdot \sigma(\mathcal{A}_{u,v}^m), \quad (2)$$

where $\hat{d}_{u,v}$ is the defocus-disparity and $\mathcal{A}_{u,v}$ is the final matching cost at a pixel (u, v) . M and d^m are the range of defocus-disparity, and predefined discrete disparity levels, respectively. Following [10], we minimize a disparity loss $\mathcal{L}_{\text{disp}}$ using a smooth L_1 loss as follows:

$$\mathcal{L}_{\text{disp}} = \frac{1}{H \cdot W} \sum_{u=1}^W \sum_{v=1}^H \mathcal{M}_{u,v} \cdot \text{smooth}_{L_1}(d_{u,v} - \hat{d}_{u,v}), \quad (3)$$

where $d_{u,v}$ is a ground-truth defocus-disparity at a pixel (u, v) converted from the ground-truth metric scale depth and $\mathcal{M}_{u,v}$ is the facial mask in Section 2.2.

For the surface normal estimation, shared 2D convolutions are applied to the feature volume C_N to regress a surface normal. The final convolutional layers follow the same structure of the baseline architecture in [29]. Finally, we train ANM by minimizing a cosine similarity normal loss $\mathcal{L}_{\text{normal}}$ as:

$$\mathcal{L}_{\text{normal}} = \frac{1}{H \cdot W} \sum_{u=1}^W \sum_{v=1}^H \mathcal{M}_{u,v} \cdot (1 - \mathbf{n}_{u,v} \cdot \hat{\mathbf{n}}_{u,v}), \quad (4)$$

where $\mathbf{n}_{u,v}$ and $\hat{\mathbf{n}}_{u,v}$ are a ground-truth, and a predicted normal at a pixel (u, v) . Finally, our StereoDPNet is fully supervised by our constructed dataset in Section 2.2 and minimizing the combination of Equation (3) and Equation (4).

Method	Task	Absolute error metric [mm] ↓					Affine error metric [px] ↓			Accuracy metric ↑	
		AbsRel	AbsDiff	SqRel	RMSE	RMSElog	WMAE	WRMSE	$1-\rho$	$\delta < 1.01^2$	$\delta < 1.01^2$
PSMNet [10]	ST	0.006	5.314	0.054	6.770	0.008	0.093	0.126	0.054	0.818	0.983
StereoNet [27]	ST	0.005	4.306	0.038	5.811	0.006	0.112	0.150	0.087	0.903	0.991
DpNet [15]	DP	0.008	7.175	0.092	8.833	0.010	0.110	0.148	0.086	0.688	0.959
MDD [36]	DP	-	-	-	-	-	1.830	2.348	0.575	-	-
BTS [30]	M	0.007	6.575	0.081	8.102	0.009	0.111	0.150	0.077	0.731	0.964
NNet [29]	DN	0.004	3.608	0.027	4.858	0.005	0.073	0.102	0.048	0.934	0.995
Ours	DN	0.003	2.864	0.019	3.899	0.004	0.064	0.091	0.034	0.966	0.995

Table 1. Depth Benchmark Results. Our proposed method outperforms the existing stereo matching methods [10], [27], DP-oriented state-of-the-art methods [15], [36], monocular depth estimation [30], and depth/normal network for stereo matching [29]. Note that MDD [36] cannot be measured by absolute metrics since it adopts different defocus-disparity geometry of Equation (1). ST, DP, M, and DN denotes “Stereo Matching”, “DP-oriented method”, “Monocular”, and “Depth and Normal”, respectively.

Method	ANM		Absolute [mm] ↓			Affine [px] ↓			Accuracy ↑		Normal [deg] ↓
	SS	D3D	AbsDiff	RMSE	WMAE	WRMSE	$1-\rho$	$\delta < 1.01$	$\delta < 1.01^2$	MAE	RMSE
ASM Only	4.895	6.223	0.095	0.127	0.056	0.850	0.992	-	-	-	-
NNet [29]	3.608	4.858	0.073	0.102	0.048	0.934	0.995	9.634	11.877	-	-
ASM + NNet	3.271	4.434	0.064	0.090	0.033	0.947	0.997	9.072	11.045	-	-
ASM + NNet ✓	3.214	4.519	0.062	0.089	0.037	0.943	0.990	8.894	10.837	-	-
StereoDPNet ✓	✓	✓	2.864	3.899	0.064	0.091	0.034	0.966	0.995	7.479	9.386

Table 2. Normal Benchmark Results with Ablation Study of ANM. NNet [29] is a baseline model of our overall architecture. We compare the performance of depth and surface normal estimation by adding each component. SS denotes “Surface Sampling” and D3D denotes “Deformable 3D convolution” of ANM respectively.

3 Experiments

To evaluate the effectiveness and the robustness of our work, we carry out various experiments on our dataset as well as DP images captured under real-world environments. We use evaluation metrics in a public benchmark suite¹ and affine invariant metrics [15] for the evaluation of estimated depth in Table 1 and Table 3. To measure the quality of normal map in Table 2, we use the metric following the DiLiGenT benchmark [42].

Depth Benchmark. We compare our method with recent DP-based depth estimation approaches [15, 36] as well as widely used stereo matching networks [10, 27], a depth/normal network for stereo matching [29] and a state-of-the-art monocular depth estimation network [30], whose results are reported in Table 1 and in Figure 7. While other methods fail to handle defocus blur or struggle to find correspondences in human faces, our method predicts the most outstanding and stable results not only with a test set but also with unmet wild examples. In addition to our facial benchmark, we conduct an additional experiment on another real-world DP dataset [36] in Table 3 to validate the generalization of our network. For a fair comparison, we augment our network using synthetic DP dataset [3] and don’t apply any post-processing (*i.e.* bilateral or guided filter).

Surface Normal Benchmark. To the best of our knowledge, this is the first attempt to estimate both the surface normal and the defocus-disparity from

¹ <http://www.cvlabs.net/datasets/kitti/>

Metrics	Method					
	PSMNet [10]	StereoNet [27]	DPNet [15]	MDD [36]	NNet [29]	Ours
WMAE (↓)	0.102	0.111	0.132	0.107	0.103	0.085
WRMSE (↓)	0.154	0.214	0.192	0.168	0.143	0.133
$1-\rho$ (↓)	0.351	0.261	0.420	0.187	0.345	0.276

Table 3. Comparisons on the public dataset [36]. We provide quantitative comparison result of the methods in Table 1 on the public dataset [36].

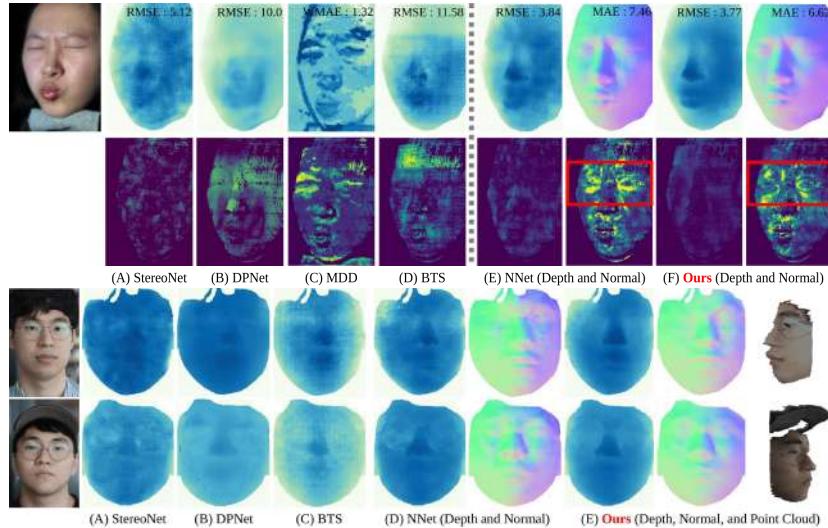


Fig. 7. Qualitative results. We show the qualitative results in the test set (**upper row**) and in the unmet real world (**lower row**). Compared to the other methods in Table 1. StereoDPNet clearly captures the surface and boundary depth of the face.

single DP images. Since the basic structure of ANM is derived from the recent depth and normal network [29] for multi-view stereo, we show the performance improvement of our ANM, compared to the baseline method [29] by adding each component in Table 2. We find that joint learning of disparity and surface normal leads to geometrically consistent and high-quality depth and surface normal, which has been demonstrated in previous works [38, 23].

4 Conclusion

We present a high-quality facial DP dataset incorporating 135,744 face images for 101 subjects with corresponding depth maps in metric scale and surface normal maps. Moreover, we introduce DP-oriented StereoDPNet for both depth and surface normal estimation. StereoDPNet successfully shows impressive results in the wild in Figure 8 by effectively handling the narrow baseline problem in DP.

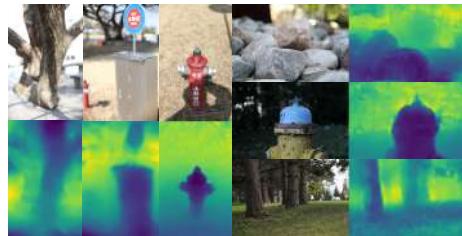


Fig. 8. Depth from single DP images in the wild. We show scene depth estimation results of StereoDPNet on outdoor photos, which are directly captured by us (**left col**) and in a public real-world DP dataset [2] for deblurring (**right col**).

Remarks. This paper is a re-publishing (summary presentation) of the paper which has been published in "ECCV 2022" by request of the IW-FCV2023 program committee to share the research results.

References

1. Aanæs, H., Jensen, R.R., Vogiatzis, G., Tola, E., Dahl, A.B.: Large-scale data for multiple-view stereopsis. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) pp. 1–16 (2016)
2. Abuolaim, A., Brown, M.S.: Defocus deblurring using dual-pixel data. In: Proceedings of the European conference on computer vision (ECCV). pp. 111–126. Springer (2020)
3. Abuolaim, A., Delbracio, M., Kelly, D., Brown, M.S., Milanfar, P.: Learning to reduce defocus blur by realistically modeling dual-pixel data. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 2289–2298 (2021)
4. Apple: Apple iphone 11 pro. <https://www.apple.com/iphone-11-pro/> (2019), accessed: 2019-09-20
5. ARCore: Augmented faces. <https://developers.google.com/ar/develop/java/augmented-faces> (2019), accessed: 2019-12-18
6. Bai, Z., Cui, Z., Rahim, J.A., Liu, X., Tan, P.: Deep facial non-rigid multi-view stereo. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5850–5860 (2020)
7. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3d faces. In: Proceedings of the 26th annual conference on Computer graphics and interactive techniques. pp. 187–194 (1999)
8. Chan, E.R., Lin, C.Z., Chan, M.A., Nagano, K., Pan, B., Mello, S.D., Gallo, O., Guibas, L., Tremblay, J., Khamis, S., Karras, T., Wetzstein, G.: Efficient geometry-aware 3D generative adversarial networks. In: CVPR (2022)
9. Chan, E.R., Monteiro, M., Kellnhofer, P., Wu, J., Wetzstein, G.: Pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5799–5809 (June 2021)
10. Chang, J.R., Chen, Y.S.: Pyramid stereo matching network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
11. Chen, C.H., Zhou, H., Ahonen, T.: Blur-aware disparity estimation from defocus stereo images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 855–863 (2015)
12. Chen, W., Mirdehghan, P., Fidler, S., Kutulakos, K.N.: Auto-tuning structured light by optical stochastic gradient descent. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
13. Feng, Y., Wu, F., Shao, X., Wang, Y., Zhou, X.: Joint 3d face reconstruction and dense alignment with position map regression network. In: Proceedings of the European conference on computer vision (ECCV) (2018)
14. Galaxy: Samsung galaxy s10. <https://www.samsung.com/us/mobile/galaxy-s10/> (2019), accessed: 2019-03-08
15. Garg, R., Wadhwa, N., Ansari, S., Barron, J.T.: Learning single camera depth estimation using dual-pixels. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019)

16. Google: Google photos: One year, 200 million users, and a whole lot of selfies. <https://blog.google/products/photos/google-photos-one-year-200-million/> (2016), accessed: 2016-05-27
17. Google: More controls and transparency for your selfies. <https://blog.google/outreach-initiatives/digital-wellbeing/more-controls-selfie-filters/> (2020), accessed: 2020-10-01
18. Guo, J., Zhu, X., Yang, Y., Yang, F., Lei, Z., Li, S.Z.: Towards fast, accurate and stable 3d dense face alignment. In: Proceedings of the European conference on computer vision (ECCV). pp. 152–168. Springer (2020)
19. Ha, H., Oh, T.H., Kweon, I.S.: A multi-view structured-light system for highly accurate 3d modeling. In: International Conference on 3D Vision (3DV) (2015)
20. Ha, H., Park, J., Kweon, I.S.: Dense depth and albedo from a single-shot structured light. In: International Conference on 3D Vision (3DV). pp. 127–134 (2015)
21. Han, Y., Lee, J.Y., So Kweon, I.: High quality shape from a single rgb-d image under uncalibrated natural illumination. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2013)
22. Hu, P., Ramanan, D.: Finding tiny faces. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 951–959 (2017)
23. Im, S., Ha, H., Choe, G., Jeon, H.G., Joo, K., Kweon, I.S.: High quality structure from small motion for rolling shutter cameras. In: Proceedings of International Conference on Computer Vision (ICCV) (2015)
24. Jensen, R., Dahl, A., Vogiatzis, G., Tola, E., Aanæs, H.: Large scale multi-view stereopsis evaluation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2014)
25. Jeon, H.G., Park, J., Choe, G., Park, J., Bok, Y., Tai, Y.W., So Kweon, I.: Accurate depth map estimation from a lenslet light field camera. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
26. Keselman, L., Iselin Woodfill, J., Grunnet-Jepsen, A., Bhowmik, A.: Intel realsense stereoscopic depth cameras. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (2017)
27. Khamis, S., Fanello, S., Rhemann, C., Kowdle, A., Valentin, J., Izadi, S.: Stereonet: Guided hierarchical refinement for real-time edge-aware depth prediction. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 573–590 (2018)
28. Kinect2: Kinect for windows sdk 2.0. <https://developer.microsoft.com/en-us/windows/kinect/> (2014), accessed: 2014-10-21
29. Kusupati, U., Cheng, S., Chen, R., Su, H.: Normal assisted stereo depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
30. Lee, J.H., Han, M.K., Ko, D.W., Suh, I.H.: From big to small: Multi-scale local planar guidance for monocular depth estimation. arXiv preprint arXiv:1907.10326 (2019)
31. Long, X., Lin, C., Liu, L., Li, W., Theobalt, C., Yang, R., Wang, W.: Adaptive surface normal constraint for depth estimation. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
32. Long, X., Liu, L., Theobalt, C., Wang, W.: Occlusion-aware depth estimation with adaptive normal constraints. In: Proceedings of the European conference on computer vision (ECCV). pp. 640–657. Springer (2020)

33. Luo, H., Nagano, K., Kung, H.W., Xu, Q., Wang, Z., Wei, L., Hu, L., Li, H.: Normalized avatar synthesis using stylegan and perceptual refinement. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11662–11672 (2021)
34. Nehab, D., Rusinkiewicz, S., Davis, J., Ramamoorthi, R.: Efficiently combining positions and normals for precise 3d geometry. ACM Transactions on Graphics (ToG) **24**(3), 536–543 (2005)
35. Pan, L., Chowdhury, S., Hartley, R., Liu, M., Zhang, H., Li, H.: Dual pixel exploration: Simultaneous depth estimation and image restoration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4340–4349 (June 2021)
36. Punnappurath, A., Abuolaim, A., Affifi, M., Brown, M.S.: Modeling defocus-disparity in dual-pixel sensors. In: 2020 IEEE International Conference on Computational Photography (ICCP) (2020)
37. Qi, X., Liao, R., Liu, Z., Urtasun, R., Jia, J.: Geonet: Geometric neural network for joint depth and surface normal estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 283–291 (2018)
38. Qiu, J., Cui, Z., Zhang, Y., Zhang, X., Liu, S., Zeng, B., Pollefeys, M.: Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
39. Richardson, E., Sela, M., Or-El, R., Kimmel, R.: Learning detailed face reconstruction from a single image. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1259–1268 (2017)
40. Scharstein, D., Szeliski, R.: High-accuracy stereo depth maps using structured light. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). vol. 1 (2003)
41. Shang, J., Shen, T., Li, S., Zhou, L., Zhen, M., Fang, T., Quan, L.: Self-supervised monocular 3d face reconstruction by occlusion-aware multi-view geometry consistency. In: Proceedings of the European conference on computer vision (ECCV). pp. 53–70. Springer (2020)
42. Shi, B., Wu, Z., Mo, Z., Duan, D., Yeung, S.K., Tan, P.: A benchmark dataset and evaluation for non-lambertian and uncalibrated photometric stereo. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
43. Silberman, N., Fergus, R.: Indoor scene segmentation using a structured light sensor. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) - Workshop on 3D Representation and Recognition (2011)
44. Tran, L., Liu, X.: Nonlinear 3d face morphable model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7346–7355 (2018)
45. Wadhwa, N., Garg, R., Jacobs, D.E., Feldman, B.E., Kanazawa, N., Carroll, R., Movshovitz-Attias, Y., Barron, J.T., Pritch, Y., Levoy, M.: Synthetic depth-of-field with a single-camera mobile phone. ACM Transactions on Graphics (ToG) **37**(4), 1–13 (2018)
46. Wu, S., Rupprecht, C., Vedaldi, A.: Unsupervised learning of probably symmetric deformable 3d objects from images in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)

47. Wu, S., Rupprecht, C., Vedaldi, A.: Unsupervised learning of probably symmetric deformable 3d objects from images in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1–10 (2020)
48. Wu, X., Zhou, J., Liu, J., Ni, F., Fan, H.: Single-shot face anti-spoofing for dual pixel camera. *IEEE Transactions on Information Forensics and Security* **16**, 1440–1451 (2020)
49. Xin, S., Wadhwa, N., Xue, T., Barron, J.T., Srinivasan, P.P., Chen, J., Gkioulekas, I., Garg, R.: Defocus map estimation and deblurring from a single dual-pixel image. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
50. Yang, F., Wang, J., Shechtman, E., Bourdev, L., Metaxas, D.: Expression flow for 3d-aware face component transfer. In: ACM SIGGRAPH 2011 papers, pp. 1–10 (2011)
51. Ying, X., Wang, L., Wang, Y., Sheng, W., An, W., Guo, Y.: Deformable 3d convolution for video super-resolution. *IEEE Signal Processing Letters* **27**, 1500–1504 (2020)
52. Yu, Z., Qin, Y., Li, X., Zhao, C., Lei, Z., Zhao, G.: Deep learning for face anti-spoofing: A survey. arXiv preprint arXiv:2106.14948 (2021)
53. Zhang, Y., Wadhwa, N., Orts-Escalano, S., Häne, C., Fanello, S., Garg, R.: Du 2 net: Learning depth estimation from dual-cameras and dual-pixels. In: Proceedings of the European conference on computer vision (ECCV). pp. 582–598. Springer (2020)
54. Zhou, H., Hadap, S., Sunkavalli, K., Jacobs, D.W.: Deep single-image portrait relighting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019)

Generative Bias for Robust Visual Question Answering*

Jae Won Cho¹, Dong-Jin Kim², Hyeonggon Ryu¹, and In So Kweon¹

¹ KAIST, Daejeon, South Korea

² Hanyang University, Seoul, South Korea

chojw@kaist.ac.kr & djdkim@hanyang.ac.kr & gonhy.ryu@kaist.ac.kr &
iskweon77@kaist.ac.kr

Abstract. The task of Visual Question Answering (VQA) is known to be plagued by the issue of VQA models exploiting biases within the dataset to make its final prediction. Various previous ensemble based debiasing methods have been proposed where an additional model is purposefully trained to be biased in order to aid in training a robust target model. However, these methods compute the bias for a model simply from the label statistics of the training data or from single modal branches. In this work, in order to better learn the bias a target VQA model suffers from, we propose a generative method to train the bias model *directly from the target model*, called GenB. In particular, GenB employs a generative network to learn the bias in the target model through a combination of the adversarial objective and knowledge distillation. We then debias our target model with GenB as a bias model, and show through extensive experiments the effects of our method on various VQA bias datasets including VQA-CP2 and VQA-CP1.

Keywords: Visual Question Answering · Vision & language · Robustness.

1 Introduction

Visual Question Answering (VQA) [4] is a challenging task that requires a model to correctly understand and predict an answer given a input pair of image and question. Various studies have shown that VQA is prone to biases within the dataset and tend to rely heavily on language biases present in the dataset [1,8,17], where VQA models tend to predict similar answers only depending on the question regardless of the image. In response to this, recent works have developed various bias reduction techniques, and recent methods have exploited ensemble based debiasing methods [5,6,9,15] extensively.

Among ensemble based methods, additional models are introduced to concurrently learn biases that might exist within each modality or dataset. For example, in works such as [5,9], the Question-Answer (QA) model is utilized to

* This paper is a short version of CVPR'23 Submission and 28th Samsung HumanTech Bronze award winner

determine the language prior biases that exist when a model is asked to give an answer based solely off of the question. This QA model is then utilized to train a robust “target” model, which is used for inference. The key purpose of an ensemble “bias” model is to capture the biases that are formed with its given inputs (*i.e.*, language prior biases from the QA model). In doing so, if this model is able to represent the bias well, this bias model can be used to teach the target model to avoid such biased answers. In other words, the better the bias model can learn the biases, the better the target model can avoid such biases.

Existing ensemble based methods either use pre-computed label statistics of training data (GGE-D [9] and LMH [6]), or single modal branches that compute the answer from either the question or image [5,6,9,12]. However, we conjecture that there is a limit to the bias representation that can be obtained from such methods, as the model’s representative capacity is limited due to its input. In addition, pre-computed label statistics represents only part of the bias [9]. As shown in Fig. 1, given a question type, the pre-computed label statistics (or known dataset bias) are noticeably different to the predictions of a model trained with the question or with the image and question. This discrepancy signifies that there is a part of the bias that we cannot fully model simply with the previous methods. Therefore, we propose a novel stochastic bias model that learns the bias *directly from the target model*.

More specifically, to directly learn the bias distribution of the *target model*, we model the bias model as a Generative Adversarial Network (GAN) [7] to stochastically mimic the target model’s answer distribution given the same question input by introducing a random noise vector. As seen through literature, most biases are held within the question [1], so we use questions as the main bias modality. In addition, we utilize knowledge distillation [10] on top of adversarial training to force the bias model to be as close as possible to the target model, so that the target model learns from harder negative supervision from the bias model. Finally, with our generative bias model, we then use our modified debiasing loss function to train our target model. Our final bias model is able to train the target model that outperforms previous uni-modal and multi-modal ensemble based debiasing methods by a large margin. To the best of our knowledge, we are the first to train the bias model by directly leveraging the behavior of the target model using a generative model for the task of VQA.

To show the efficacy and robustness of our method, we perform extensive experiments on commonly used robustness testing VQA datasets and various

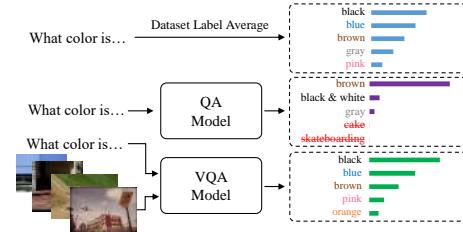


Fig. 1. Given a Question Type (“What color is...”), we show all of the averaged answers within the training dataset. The answer computed from the entire training dataset is the known dataset bias as in [6,9]. We see that the averaged model predictions of the Question-Answer Model and Visual-Question-Answer Model are significantly different.

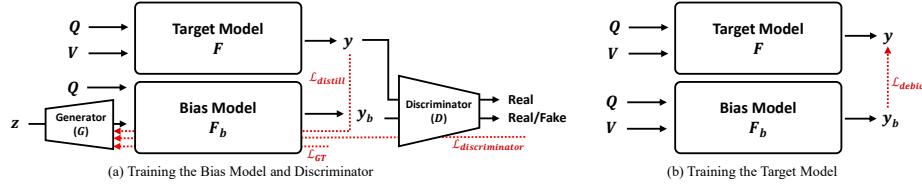


Fig. 2. (a) shows how we train our Bias Model and Discriminator. The Bias Model is trained with 3 different losses including the ground truth BCE (Eq. (1)), knowledge distillation (Eq. (3)), and adversarial Eq. (2)) losses. (b) shows how our Target Model is trained with the Bias model with debiasing loss functions (refer existing works). Note, steps (a) and (b) happen concurrently. Note that we only use the Target Model during inference.

different VQA architectures. Our method show the state-of-the-art results on all settings without the use of external human annotations and dataset reshuffling methods.

Our contributions are as follows: (1) We propose a novel bias model for ensemble based debiasing for VQA by directly leveraging the target model that we name *GenB*. (2) In order to effectively train GenB, we employ a Generative Adversarial Network and knowledge distillation loss to capture both the dataset distribution bias and the bias from the target model. (3) We achieve state-of-the-art performance on VQA-CP2 and VQA-CP1 using the simple UpDn baseline without extra annotations or dataset reshuffling.

2 Methodology

In this section, we explain VQA briefly and describe in detail our method GenB, how we train and debias with it, and a short analogous discussion.

2.1 Visual Question Answering Baseline

With an image and question as a pair of inputs, a VQA model learns to correctly predict an answer from the whole answer set \mathcal{A} . A typical VQA model $F(\cdot, \cdot)$ takes both a visual representation $\mathbf{v} \in \mathbb{R}^{n \times d_v}$ (a set of feature vectors computed from a Convolutional Neural Network given an image where n is the number of objects in the image and d_v being the vector dimension) and a question representation $\mathbf{q} \in \mathbb{R}^{d_q}$ (a single vector computed from a Glove [14] word embedding followed by a Recurrent Neural Network given a question) as input. Then, an attention module followed by a multi-layer perceptron classifier $F : \mathbb{R}^{n \times d_v} \times \mathbb{R}^{d_q} \rightarrow \mathbb{R}^{|\mathcal{A}|}$ which generates an answer logit vector $\mathbf{y} \in \mathbb{R}^{|\mathcal{A}|}$ (*i.e.*, $\mathbf{y} = F(\mathbf{V}, \mathbf{Q})$). Then, after applying a sigmoid function $\sigma(\cdot)$, our goal is to make an answer probability prediction $\sigma(\mathbf{y}) \in [0, 1]^{|\mathcal{A}|}$ close to the ground truth answer probability $\mathbf{y}_{gt} \in [0, 1]^{|\mathcal{A}|}$. In this work, we adopt one of the popular state-of-the-art architectures UpDn [3] widely used in VQA research.

2.2 Ensembling with Bias Models

In this work, our scope is bias mitigation through ensembling bias model similar to previous works [5,6,9]. In ensemble based methods, there exist a “bias” model that generates $\mathbf{y}_b \in \mathbb{R}^{|\mathcal{A}|}$ which we define as $F_b(\cdot, \cdot)$ and a “target” model, defined as $F(\cdot, \cdot)$. Note that, we discard $F_b(\cdot, \cdot)$ during testing and only use $F(\cdot, \cdot)$. As previously mentioned, the goal of the existing bias models is to overfit to the bias as much as possible. Then, given the overfitted bias model, the target model is trained with a debiasing loss function [5,6,9] to improve the robustness of the target model. Ultimately, the target model learns to predict an unbiased answer by avoiding the biased answer from the bias model. The bias model $F_b(\cdot, \cdot)$ can either be the same or different from the original $F(\cdot, \cdot)$ and there could be multiple models as well [12]. Although previous works try to leverage the bias from the individual modalities [5,6,9,12], we propose that this limits the ability of the model to represent biases. Hence, in order to represent the biases *similar to the target model*, we set the architecture of $F_b(\cdot, \cdot)$ to be the same as $F(\cdot, \cdot)$ and we use the UpDn [3] model.

2.3 Generative Bias

As mentioned in the Sec. 1, as our goal is to train a bias model that can generate stochastic bias representations, we use a random noise vector in conjunction with a given modality to learn both the dataset bias and the bias that the target model could exhibit. As the question is known to be prone to bias, we keep the question modality and use it as the input to our bias model $F_b(\cdot, \cdot)$. But instead of using the image features, we introduce a random noise vector $\mathbf{z} \in \mathbb{R}^{n \times 128}$ in addition to a generator network $G : \mathbb{R}^{n \times 128} \rightarrow \mathbb{R}^{n \times d_v}$ to generate the corresponding input to the bias model $F_b(\cdot, \cdot)$. Formally, given a random Gaussian noise vector $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$, a generator network $G(\cdot)$ synthesizes a vector that has the same dimension as the image feature representation, *i.e.*, $\hat{\mathbf{v}} = G(\mathbf{z}) \in \mathbb{R}^{n \times d_v}$. Ultimately, our model takes in the question \mathbf{q} and $G(\mathbf{z})$ as its input and generates the bias logit \mathbf{y}_b in the form $F_b(G(\mathbf{z}), \mathbf{q}) = \mathbf{y}_b$. Note, this can be done on another modality, (*i.e.*, $F_b(G(\mathbf{z}), \mathbf{v}) = \mathbf{y}_b$), but we found this is unhelpful. For simplicity, we consider generator and bias model as one network and rewrite $F_b(G(\mathbf{z}), \mathbf{q})$ in the form $F_{b,G}(\mathbf{z}, \mathbf{q})$ and call our “Generative Bias” method **GenB**.

2.4 Training the Bias Model

In order for our bias model GenB to learn the biases given the question, we use the traditional VQA loss, the Binary Cross Entropy Loss:

$$\mathcal{L}_{GT}(F_{b,G}) = \mathcal{L}_{BCE}(\sigma(F_{b,G}(\mathbf{z}, \mathbf{q})), \mathbf{y}_{gt}). \quad (1)$$

However, unlike existing works, we want the bias model to also capture *the biases in the target model*. Hence, in order to mimic the bias of the target model

as a random distribution of the answer, we propose adversarial training [7] to train our bias model. In particular, we introduce a discriminator that tries to distinguish the answers from the target model and the bias model as “real” and “fake” answers, respectively. The discriminator is formulated as $D(F(\mathbf{v}, \mathbf{q}))$ and $D(F_{b,G}(\mathbf{z}, \mathbf{q}))$ or rewritten as $D(\mathbf{y})$ and $D(\mathbf{y}_b)$. The objective of our generative adversarial network with generator $F_{b,G}(\cdot, \cdot)$ and $D(\cdot)$ can be expressed as:

$$\begin{aligned} & \min_{F_{b,G}} \max_D \mathcal{L}_{GAN}(F_{b,G}, D), \text{ where} \\ & \mathcal{L}_{GAN}(F_{b,G}, D) = \mathbb{E}_{\mathbf{v}, \mathbf{q}} \left[\log \left(D(F(\mathbf{v}, \mathbf{q})) \right) \right] + \mathbb{E}_{\mathbf{q}, \mathbf{z}} \left[\log \left(1 - D(F_{b,G}(\mathbf{z}, \mathbf{q})) \right) \right] \\ & = \mathbb{E}_{\mathbf{y}} \left[\log \left(D(\mathbf{y}) \right) \right] + \mathbb{E}_{\mathbf{y}_b} \left[\log \left(1 - D(\mathbf{y}_b) \right) \right]. \end{aligned} \quad (2)$$

The generator ($F_{b,G}$) tries to minimize the objective (\mathcal{L}_{GAN}) against an adversarial discriminator (D) that tries to maximize it. Through alternative training of D and $F_{b,G}$, the distribution of the answer vector from the bias model (\mathbf{y}_b) should be close to that from the target model (\mathbf{y}).

In addition, to further aid in the bias model’s ability to capture the intricate biases present in the target model, we add an additional knowledge distillation objective [10] that encourages the bias model to directly follow the behavior of the target model with only the \mathbf{q} given to it. We empirically find that it is beneficial to include a sample-wise distance based metric such as KL divergence. This method is similar to the approaches in the image to image translation task [11]. Then, the goal of the generator is not only to fool the discriminator but also to try to imitate the answer output of the target model in order to give the target model more challenging supervision in the form of *hard negative* sample synthesis. We add another objective to our adversarial training for $F_{b,G}(\cdot, \cdot)$ as:

$$\mathcal{L}_{distill}(F_{b,G}) = \mathbb{E}_{\mathbf{v}, \mathbf{q}, \mathbf{z}} \left[D_{KL}(F(\mathbf{v}, \mathbf{q}) \| F_{b,G}(\mathbf{z}, \mathbf{q})) \right]. \quad (3)$$

Ultimately, the final training loss for the bias model, or GenB, is as follows:

$$\begin{aligned} & \min_{F_{b,G}} \max_D \mathcal{L}_{GenB}(F_{b,G}, D), \text{ where} \\ & \mathcal{L}_{GenB}(F_{b,G}, D) = \mathcal{L}_{GAN}(F_{b,G}, D) + \lambda_1 \mathcal{L}_{distill}(F_{b,G}) + \lambda_2 \mathcal{L}_{GT}(F_{b,G}), \end{aligned} \quad (4)$$

where λ_1 and λ_2 are the loss weight hyper-parameters.

2.5 Debiasing the Target Model

Given a generated biased answer \mathbf{y}_b , there are several debiasing loss functions that we can use such as [5,6,9]. The GGE [9] loss is one of the best performing losses without the use of label distribution. The GGE loss takes the bias predictions/distributions and generates a gradient in the opposite direction to train

Table 1. Experimental results of our method on the VQA-CP2 test set and VQA-CP1 test set. **Best** and **second best** results are styled in this manner within the column. Among the compared baselines, our method GenB shows the best performance by a noticeable margin.

Method	Base	VQA-CP2 test				VQA-CP1 test			
		All	Yes/No	Num	Other	All	Yes/No	Num	Other
SAN [16]	-	24.96	38.35	11.14	21.74	32.50	36.86	12.47	36.22
GVQA [2]	-	31.30	57.99	13.68	22.14	39.23	64.72	11.87	24.86
S-MRL [5]	-	38.46	42.85	12.81	43.20	36.38	42.72	12.59	40.35
UpDn [3]	-	39.94	42.46	11.93	45.09	36.38	42.72	12.14	40.35
<i>Methods based on ensemble models</i>									
AReg [15]	UpDn	41.17	65.49	15.48	35.48	43.43	74.16	12.44	25.32
RUBi [5]	UpDn	44.23	67.05	17.48	39.61	50.90	80.83	13.84	36.02
LMH [6]	UpDn	52.45	69.81	44.46	45.54	55.27	76.47	26.66	45.68
CF-VQA(SUM) [12]	UpDn	53.55	91.15	13.03	44.97	57.03	89.02	17.08	41.27
CF-VQA(SUM) [12]	S-MRL	55.05	90.61	21.50	45.61	57.39	88.46	14.80	43.61
CF-VQA(SUM) [12] + IntroD [13]	S-MRL	55.17	90.79	17.92	46.73	-	-	-	-
GGE [9]	UpDn	57.32	87.04	27.75	49.59	-	-	-	-
GenB (Ours)	UpDn	59.15	88.03	40.05	49.25	62.74	86.18	43.85	47.03

the target model. With this starting point, we modify this equation with the ensemble of the biased model in this work as follows:

$$\mathcal{L}_{target}(F) = \mathcal{L}_{BCE}(\mathbf{y}, \mathbf{y}_{DL}), \quad (5)$$

where the i -th element of the pseudo-label \mathbf{y}_{DL} is defined as follows:

$$\mathbf{y}_{DL}^i = \min(1, 2 \cdot \mathbf{y}_{gt}^i \cdot \sigma(-2 \cdot \mathbf{y}_{gt}^i \cdot \mathbf{y}_b^i)), \quad (6)$$

where \mathbf{y}_{gt}^i and \mathbf{y}_b^i are the i -th element of the ground truth and the output of the biased model, respectively. The key point of difference is that unlike [9] that suppresses the output of the biased model with the sigmoid function, we use \mathbf{y}_b without using the sigmoid function. In this case, as the value of \mathbf{y}_{DL} can exceed 1, we additionally clip the value so that the value of \mathbf{y}_{DL} is bounded in [0, 1]. We empirically find these simple modifications on the loss function significantly improves the performance. We conjecture the unsuppressed biased output \mathbf{y}_b allows our target model to better consider the *intensity* of the bias, leading to a more robust target model. In addition, when we train the target model, we do not use the noise inputs as in $F_{b,G}(\mathbf{z}, \mathbf{q})$, rather we use the real images as such $F_b(\mathbf{v}, \mathbf{q})$, and use this output to train our target model. When the bias model is trained, it is trained with a noise vector to hallucinate the possible biases when only given the question, then, to fully utilize the biases that the bias model captures, we give it the real images.

3 Experiments

Dataset and evaluation metric. We conduct our experiments within the VQA datasets that are commonly used for diagnosing bias in VQA models. In particular, we test on the the VQA-CP2 and VQA-CP1 datasets [2]. For evaluation on all datasets, we take the standard VQA evaluation metric [4].

Baseline architecture. We adopt a popular VQA baseline architecture UpDn [3] as both our ensemble bias model F_b and our target model F . During training,

Table 2. Loss ablation for GenB. All inferences scores are based on the target model except the first row. Although the DSC and Distill losses independently do not show large improvement, our final model with all losses show a large margin of improvement.

Training Loss	Bias Model	VQA-CP2 test			
		All	Yes/No	Num	Other
BCE	UpDn	39.94	42.46	11.93	45.09
BCE	GenB	56.98	88.82	19.39	49.86
BCE + DSC	GenB	56.54	89.06	21.29	49.79
BCE + Distill	GenB	57.06	88.91	23.24	49.65
BCE + DSC + Distill	GenB	59.15	88.03	40.05	49.25

we train both the bias model and target model together, then we use the target model only for inference.

Results on VQA-CP2 and VQA-CP1. We compare GenB in relation to the recent state-of-the-art ensembling methods that focus on bias reduction as shown in Table 1. For VQA-CP2, we first list the *baseline architectures* and then compare only with in this paper due to lack of space. The *ensemble based methods* listed are, (AReg [15], RUBi [5], LMH [6], CF-VQA [12], and GGE [9]).

In Table 1, our method achieves state-of-the-art performance on VQA-CP2, surpassing the second best (GGE [9]) by 1.83%. The performance of our model on all three categories (“Yes/No,” “Num,” “Other”) are within the top-3 consistently for the same backbone architecture. Our method also performs highly favorably in the “Other” metric. We also show how our method performs on the VQA-CP1 dataset. Note that not all of the baselines are listed as we only list the scores that are made available in the respective papers. Our method also shows the state-of-the-art results on this dataset with a significant performance improvement over the second best among the methods compared, CF-VQA(SUM) [12] and our method improves the overall performance by 5.60% while also having the best performance in both “Num” and “Other” category, by 3.28% and 2.41% performance improvements, respectively.

3.1 Ablation Studies

Our method (GenB) includes several different components as shown from Sec. 2.3 to Sec. 2.5. To understand the effects of each component, we run an ablation study on the VQA-CP2 dataset. For all experiments, the results are of the target model and as the purpose of the bias model is to capture bias instead of correctly predicting the answers, we do not consider the predictions of the bias model. For all our ablation tables, we also add the UpDn baseline in the first row, the model in which our target model and bias model is based for comparison. To further understand whether GenB can be applied to other networks, we further include an ablation study of GenB on other VQA architectures.

4 Conclusion

In this paper, we started with this intuition “the better the bias model, the better we can debias the target model. Then how can we best model the bias?”

In response, we present simple, effective, and novel generative bias model that we call GenB. We use this generative model to learn the bias that may be inhibited by both the distribution and target model with the aid of generative networks, adversarial training, and knowledge distillation. In addition, in conjunction with our modified loss function, our novel bias model is able to debias our target model, and our target model achieves state-of-the-art performance on various bias diagnosing datasets and we believe that this work can be extended to other multi-modal and uni-modal research in understanding and mitigating bias.

References

1. Agrawal, A., Batra, D., Parikh, D.: Analyzing the behavior of visual question answering models. In: EMNLP (2016) [1](#), [2](#)
2. Agrawal, A., Batra, D., Parikh, D., Kembhavi, A.: Don't just assume; look and answer: Overcoming priors for visual question answering. In: CVPR (2018) [6](#)
3. Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L.: Bottom-up and top-down attention for image captioning and visual question answering. In: CVPR (2018) [3](#), [4](#), [6](#)
4. Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., Parikh, D.: Vqa: Visual question answering. In: ICCV (2015) [1](#), [6](#)
5. Cadene, R., Dancette, C., Ben-younes, H., Cord, M., Parikh, D.: Rubi: Reducing unimodal biases in visual question answering. In: NeurIPS (2019) [1](#), [2](#), [4](#), [5](#), [6](#), [7](#)
6. Clark, C., Yatskar, M., Zettlemoyer, L.: Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In: EMNLP (2019) [1](#), [2](#), [4](#), [5](#), [6](#), [7](#)
7. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NeurIPS (2014) [2](#), [5](#)
8. Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D.: Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In: CVPR (2017) [1](#)
9. Han, X., Wang, S., Su, C., Huang, Q., Tian, Q.: Greedy gradient ensemble for robust visual question answering. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1584–1593 (2021) [1](#), [2](#), [4](#), [5](#), [6](#), [7](#)
10. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015) [2](#), [5](#)
11. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: CVPR (2017) [5](#)
12. Niu, Y., Tang, K., Zhang, H., Lu, Z., Hua, X.S., Wen, J.R.: Counterfactual vqa: A cause-effect look at language bias. In: CVPR (2021) [2](#), [4](#), [6](#), [7](#)
13. Niu, Y., Zhang, H.: Introspective distillation for robust question answering. Advances in Neural Information Processing Systems **34**, 16292–16304 (2021) [6](#)
14. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: EMNLP (2014) [3](#)
15. Ramakrishnan, S., Agrawal, A., Lee, S.: Overcoming language priors in visual question answering with adversarial regularization. In: NeurIPS (2018) [1](#), [6](#), [7](#)
16. Yang, Z., He, X., Gao, J., Deng, L., Smola, A.: Stacked attention networks for image question answering. In: CVPR (2016) [6](#)
17. Zhang, P., Goyal, Y., Summers-Stay, D., Batra, D., Parikh, D.: Yin and yang: Balancing and answering binary visual questions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5014–5022 (2016) [1](#)

DDConv: Dilated Depthwise Convolution with YOLOv5 for Drone Imagery*

Jehwan Choi¹, Minseung Kim², Donggue Kim², and Kanghyun Jo¹

¹ Department of Electrical, Electronic and Computer Engineering,
University of Ulsan, Ulsan, South Korea

² School of Electrical Engineering
University of Ulsan, Ulsan, South Korea
jhchoi@islab.ulsan.ac.kr, kmsoiio@naver.com,
kdk6859@gmail.com, ac_ejo@islab.ulsan.ac.kr
<https://islab.ulsan.ac.kr/>

Abstract. Unmanned Aerial Vehicles (UAVs) with Convolutional Neural Network (CNN)-based artificial intelligence technologies have recently received high attention for various applications. In this paper, our focus is on object detection network research for real-time drone systems. Thus, we propose the DDConv block, considering the unique characteristics of drones, such as a wide shooting range, objects of various types and scales, and high resolution. The DDConv block analyzes images using dilated convolution and depth-wise convolution, and replaces the C3 module of the YOLOv5 backbone. The experimental results showed that the number of parameters and the GFLOPS value decreased by about 20%. The object detection time was recorded at 6.5 ms per image, which is almost twice as fast as the original network. Although accuracy slightly decreased, the detection results still found most of the objects well. In the future, we plan to apply this network for traffic analysis and surveillance systems.

Keywords: Object detection · drone dataset · dilated convolution · depth-wise convolution.

1 Introduction

Recently, drones combined with computer vision-based artificial intelligence technology have been utilized in various fields, such as autonomous flight, unmanned delivery, and missing person searches. The most crucial part of these technologies is object recognition, as the camera mounted on the drone must be capable of analyzing its surroundings and detecting and avoiding obstacles in its path. Therefore, many researchers are focusing on real-time object detection technology in drone images. The modification of Convolutional Neural Networks (CNNs), the

* This results was supported by "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(MOE)(2021RIS-003).

development of new algorithms, and modification methods for real-time object detection in drone images have been introduced in previous studies [1–3]. In addition, multi-object detection methods [4–7] and strategies for detecting small objects [8–11] are also attracting attention from many researchers. In this paper, we conduct a study focused on a real-time object detection network for drone images, which is the first step towards drone traffic analysis and surveillance systems. In addition, multi-object detection methods and small object detection strategies are also hot topics for many researchers. In this paper, we conduct a study focused on a drone image real-time object detection network which is the first step for drone traffic analysis and surveillance system.

First, we introduce the data analysis performed to create a suitable network for real-time object detection in drone images. The drone captures Bird's-Eye View (BEV) videos from a high altitude, which presents three challenges, as shown in Figure 1. Firstly, the types of objects captured are very diverse as a wide area is filmed, as depicted in Figure 1(a) with six types of objects and a large number of objects. Secondly, the shape of objects is irregular even within the same area, as the drone is moving while filming, as shown in Figure 1(b). Thirdly, the diversity of object forms is a distinct challenge, as objects may change their form due to differences in altitude, angle, and field of view, as seen in Figure 1(c).

In order to address the problems mentioned, this paper applies a convolution technique that can effectively analyze drone images in an object detection network. Firstly, using a large filter to calculate an image of a large area increases the probability of high-accuracy results but also significantly increases the computational load. Therefore, we apply a dilated convolution technique that provides a similar effect to using a large filter while maintaining a lower computational load. In addition, we apply a depth-wise convolution technique that has a similar effect to normal convolution but with a reduced computational load to ensure real-time object detection. This reduction in computation leads to a reduction in object detection time. Secondly, the number of detection heads is increased from three to four based on the scale of the objects in the drone image. As shown in Figure 2, objects in drone images can be divided into four scales, so the detection head must be configured accordingly to detect objects of various scales.

2 Related work

2.1 CNN-based object detection

Object detection is a computer technology that relates to computer vision and video processing to identify the presence of objects. It often uses the method of pre-extracting object features and detecting them within an image. In computer vision, there is a CNN technique for object detection. This method can preserve the spatial information of an image and identify features through convolution. YOLOv5 [12] is a popular real-time object detection method among the YOLO series, known for its fast computational speed. The YOLOv5 series includes versions such as nano, small, medium, large, and x-large. These versions differ in

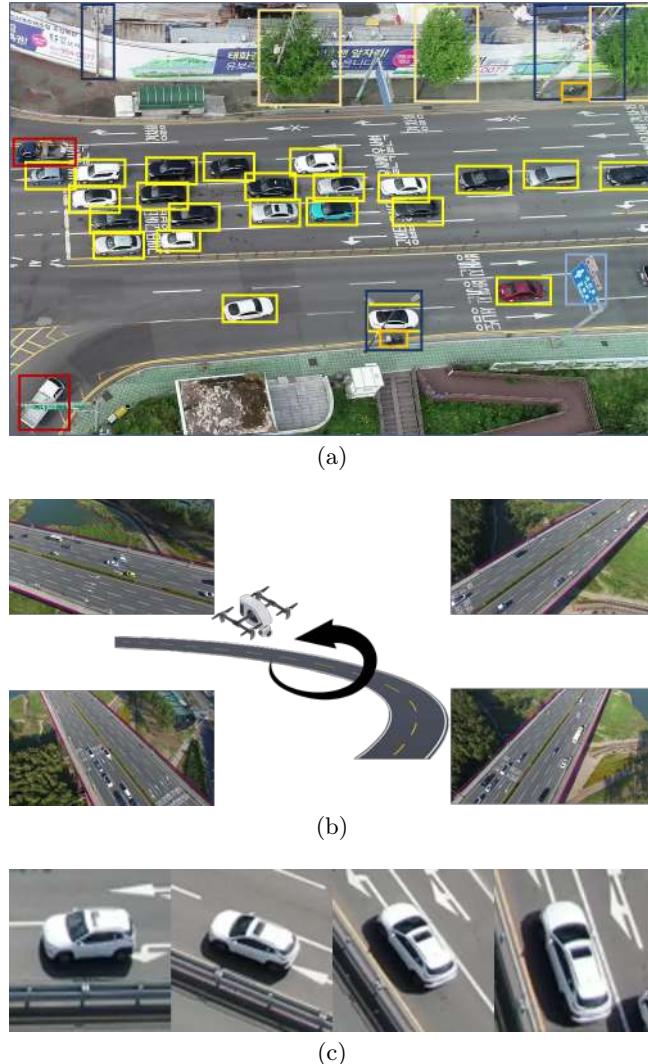


Fig. 1: Challenged of drone image object detection, (a) Existence of many types and number of objects in one image, (b) Difficulty in analyzing same-area information due to the irregular movement path of the drone, (c) the diversity according to the taking altitude, angle, and field of view in the same object



Fig. 2: Illustration of object scale comparison in drone images.

terms of accuracy and computational speed, achieved by adjusting the number of convolutions in the backbone. Performance can also be influenced by changing the convolution method, such as Transpose convolution, Separable convolution, Dilated convolution, Depth-wise Separable convolution, etc. Dilated convolution was selected as a method for efficiently learning wide-pixel photos. Dilated convolution is the main idea in the DilatedNet [13] that increases the reception file by adding zero padding inside the convolution filter, allowing the same number of input pixels while accommodating a wide range of inputs. This paper [14] also improves performance by making the CFEM(Context Feature Enhancement Module) a multi-path dilated convolutional layer.

2.2 High speed CNN method

There are two key evaluation indicators for CNN. The first is accuracy, and the second is speed. The topic of this paper is real-time object detection in drone images, and it focuses on speed because it is a priority to detect many objects quickly. The Depth-wise Separable Convolution used as a method for improving CNN speed performance. The paper MobileNets [15] presents the Depth-wise Separable Convolution method. This paper is a representative paper that speeds up CNN by changing the calculation method without focusing on reducing the

amount of calculation by increasing the filter size as 5x5 and 7x7. Therefore, Our paper applied the Depth-wise Separable Convolution method to speed up the CNN computation.

2.3 Object detection using drone image

When using drone dataset, the background occupies a large portion because it was shoted at a high altitude, and the size of objects to be detected is small. As a way to increase the accuracy of small object detection, there is a method of constructing a dataset using OBB(Object Bounding Box) in the paper [16], as shown in Fig1. In other ways, it's easier to find large objects in small images, and it's easier to find small objects in large images. For example, YOLOv1 [17] is fast but has a low ability to detect small things, and YOLOv3 [18] has increased its ability to catch small objects by performing prediction on three scales. This paper got the idea from this method and used the dilated convolution to YOLOv5's backbone to obtain a wide range of values, which would help detect small objects. In fact, it worked in a paper [19] that used dilated convolution to detect small objects.

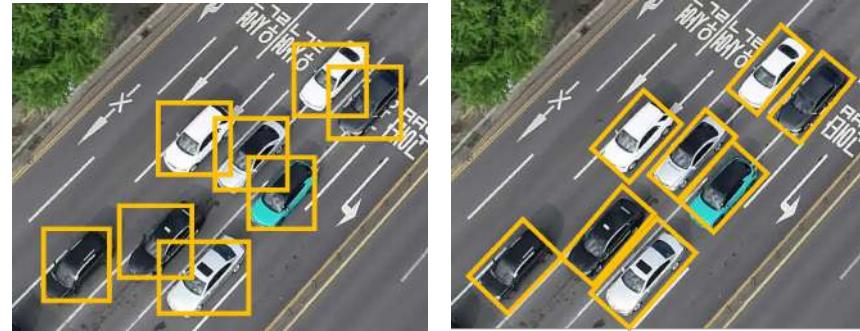


Fig. 3: Visualization of original annotation method(left) and proposed annotation method of DOTA(right)

3 Proposed method

The base-line network of proposed method in this paper is YOLOv5 [12] model. We propose the Dilated Depth-wise block for more efficient calculation and to get result faster instead of C3 module.

3.1 Detection strategy

As mentioned, there are three challenges in object detection tasks with drone imagery, which are all related to high resolution. High resolution means a large

image size and the ability to observe a wide area at once. However, this also increases the amount of computation and slows down learning and result derivation due to the wide area of convolution operations. In this paper, we aim to speed up learning and result derivation by reducing computation without sacrificing performance. Another characteristic of high resolution is the presence of various types and a large number of objects in the image. To address this, the detection head at the end of the network is increased to detect more objects in the high-resolution drone image.

Therefore, the detection strategy in this paper focuses on two aspects. Firstly, the proposed network aims to speed up computation without sacrificing detection accuracy in high-resolution images. This is achieved by using dilated convolution and depth-wise convolution. Dilated convolution maintains the same number of pixels as regular convolution but has a wider receptive field, and depth-wise convolution reduces the number of calculation parameters by performing calculations on each channel. Secondly, the detection head is increased from three scales to four scales to improve object detection in drone images. Drone images contain various objects of different sizes, such as small objects like people, medium-sized objects like cars and trucks, large objects like trees and streetlamps, and extra-large objects like apartments and buildings, as shown in Figure 2. Thus, a detection head with at least four scales is necessary to increase the probability of detecting multiple objects of different sizes.

3.2 Proposed module

The proposed network in this paper focuses on speeding up the object detection process in high resolution drone images. It achieves this by reducing the amount of computation without reducing the accuracy of detection. This is achieved by using a combination of dilated convolution and depth-wise convolution. The dilated convolution increases the receptive field while the depth-wise convolution reduces the number of computation parameters. The network uses the Dilated Depth-wise block instead of the c3 module used in the YOLOv5 backbone, which only uses 1x1 convolutions without considering the receptive field. To increase the probability of detecting various sized objects in the drone images, the detection head is increased from three scales to four scales.

The flow of the Dilated Depth-wise block is as follows. When the feature map is input, the feature map channel is divided into 4 parts without any calculation process. This is because there are other methods such as point-wise convolution, but the use of convolution layers directly increases the amount of computation. The feature maps divided into 4 parts pass into different calculation methods like (1) depth-wise convolution with size 1x1, (2) depth-wise convolution with size 3x3, (3) depth-wise convolution with a dilation ratio of 2, (4) depth-wise convolution with a dilation ratio of 3. The sum of the number of channels of the feature map generated through a total of 4 operations (1 to 4) is equal to the number of channels in the output feature map. The four feature maps generated are concatenated and passed to a 1x1 convolutional layer. The reason why the concatenated feature map is subjected to 1x1 convolution operation is

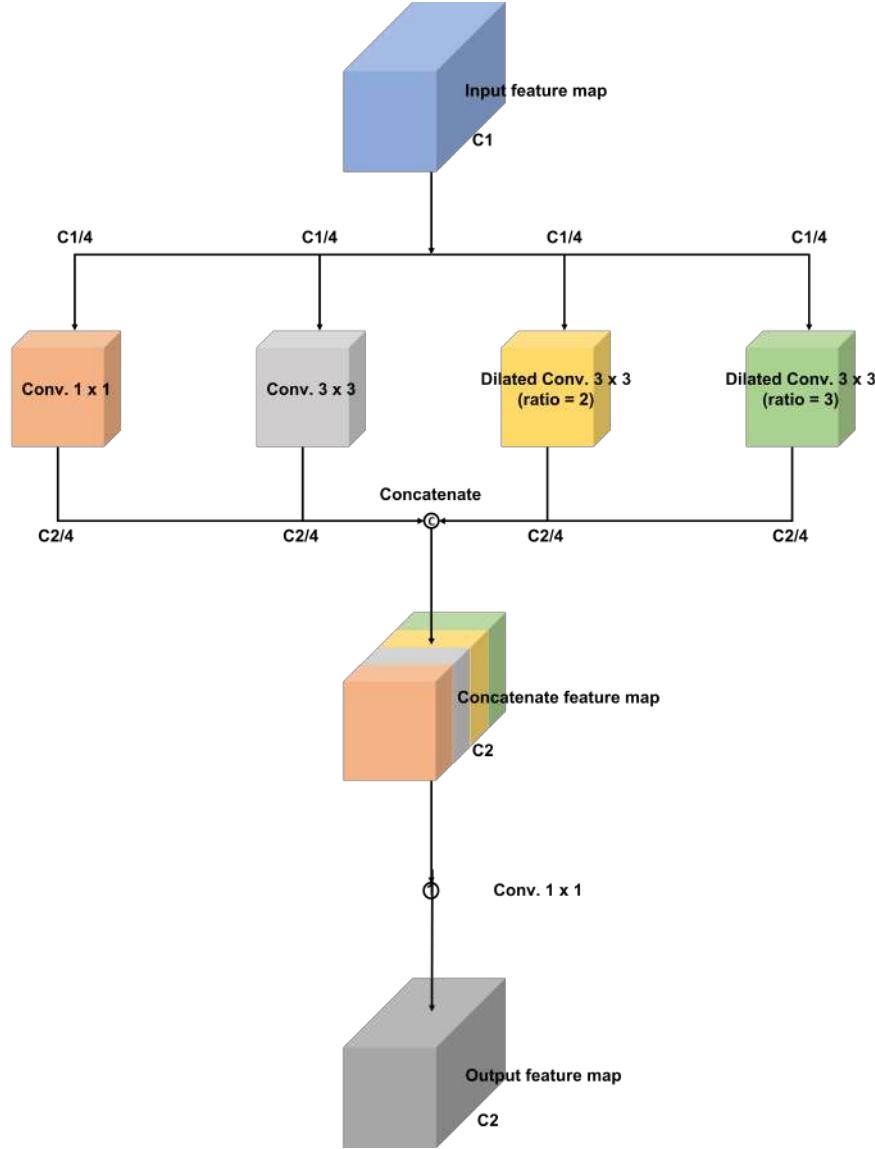


Fig. 4: Illustration of the proposed Dilated Depth-wise module.

that information in a narrow area and information in a wide area exist without sharing each other, so that information is shared using 1x1 convolution with the least amount of computation to obtain higher accuracy.

Also, the number of detection heads increased from 3 to 4. As mentioned in the detection strategy, object scales in drone images can be divided into four scales

(small, medium, large, and extra-large). Therefore, the number of scales of the detection head was modified according to the number of object scales.

4 Experiment

4.1 Dataset

The data used in the experiment are the autonomous drone dataset [20] built by University of Ulsan and the VisDrone dataset [21] built by Tianjin University. The challenge and proposed method of drone object detection presented in this paper are from the analysis of the autonomous drone dataset. After network modification, the experiments of same conditions were applied to the VisDrone dataset to prove that proposed network is not overfitted at autonomous drone dataset built by University of Ulsan.

The autonomous drone dataset provides videos, images, and JSON files taken at various altitudes and angles in tourist areas, city areas, and forest areas. Among them, tourist areas and city areas data were mainly used to build a similar environment for future work as we mentioned above such as traffic analysis and surveillance systems. The information of data used in this paper is shown in Table 1 and Figure 5.

Table 1: The information of data used in the experiment.

Category	Region_Place	Altitude	Angle	The number of image
City	Ulsan_Taehwa-bridge	70m	60°	2,343
City	Ulsan_Samho-bridge	60m	45°	2,291
City	Daegu_Geumho-district	60m	45°	1,854
Tourist	Daegu_Hwawon-amusement-park	80m	45°	1,768

4.2 Evaluation metrics

To evaluate the performance of the proposed network, accuracy and speed were used as evaluation criteria. In the case of accuracy of the network, two indicators are measured: mAP 0.5 and mAP 0.5 to 0.95. mAP (mean average precision) represents the average of the area under the PR curve for each class and is a metric for analyzing object detection accuracy through performance evaluation of precision and recall. The number after mAP means IoU (Intersection over Union) value. It means the value measured when the IoU score is 0.5 and measured when IoU score is gradually increasing the value from 0.5 to 0.95. In the case of network speed, parameters, GFLOPS (GPU Floating point Operations Per Second), and calculation speed per image were used as indicators for validation.

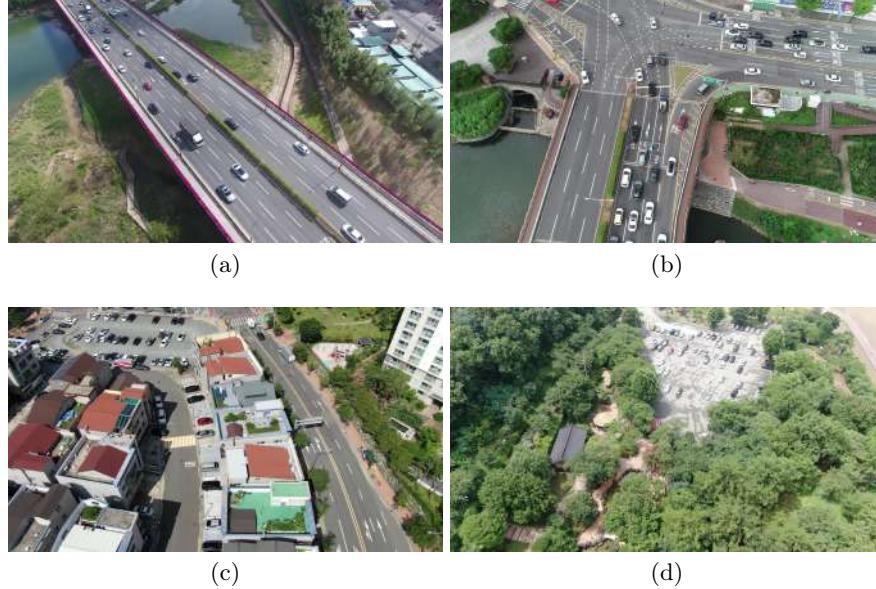


Fig. 5: Example images for each region of the autonomous drone dataset, (a) Ulsan_Samho-bridge, (b) Ulsan_Taehwa-bridge, (c) Daegu_Geumho-district, (d) Daegu_Hwawon-amusement-park.

4.3 Implementation setup

All experiments were conducted in the same environment, and the configuration environment was Intel Core i9-9960X, NVIDIA RTX 2080 Ti x 4EA, 125.5 GB memory. The training process was 100 epochs in all experiments, and all hyper-parameters such as batch size and learning rate, and depth multiple were set the same.

4.4 Result

The experimental results are detailed in Tables 2 and 3. The result applied to the autonomous drone dataset is shown in Table 2, and Table 3 is the result applied to the VisDrone dataset. Both results reduced the number of parameters and GFLOPS by about 20%. But, the object detection accuracy was slightly decreased. In the case of autonomous drone dataset, mAP50 decreased by 2.3% and mAP50-95 by 5.5%. Although the accuracy was decreased, as you can see in Figure 6, it can be seen that most objects in the image are well detected. In the case of VisDrone dataset, mAP50 decreased by 0.6%, and mAP50-95 increased by 0.23%. The biggest feature is object detection time. The original network recorded a total of 12.7 ms for pre-processing, inference, and NMS per image. However, the proposed network completed the same process in 6.5 ms per image.

Table 2: The result comparison between DDConv and C3 module using autonomous drone dataset.

Autonomous drone	DDConv		C3		Performance	
mAP(%)	50	50-95	50	50-95	50	50-95
all	60.3	38.8	62.6	44.3	2.3↓	5.5↓
tree	92.8	67.4	94.1	73.1	1.3↓	5.7↓
person	3.6	0.5	0.02	0	3.58↑	0.5↑
house	89.2	70.4	93.1	78.2	3.9↓	7.8↓
apartment	90.4	62	94.5	69.1	4.1↓	7.1↓
traffic sign	63.7	27.2	75.5	36.1	11.8↓	8.9↓
traffic light	15.7	4.3	0	0	15.7↑	4.3↑
streetlamp	78.8	41.1	84.2	51.7	5.4↓	10.6↓
car	92.9	62.3	94	68.3	1.1↓	6.0↓
bus	66.5	50.2	73.5	59.9	7.0↓	9.7↓
truck	69.6	41.4	75.9	51.5	6.3↓	10.1↓
Parameters	1,437,924		1,783,519		19.38%↓	
GFLOPS	3.3		4.2		21.43%↓	

Table 3: The result comparison between DDConv and C3 module using VisDrone dataset.

VisDrone	DDConv		C3		Performance	
mAP(%)	50	50-95	50	50-95	50	50-95
all	18.9	9.46	19.5	9.23	0.6↓	0.23↑
pedestrian	21.24	7.37	26.2	9.53	4.96↓	2.16↓
people	17.2	5.54	22.9	7.31	5.7↓	1.77↓
bicycle	2.34	0.74	2.22	0.76	0.12↑	0.02↓
car	54.6	32.4	56.6	33.7	2.0↓	1.3↓
van	18.8	12.0	14.1	8.77	4.7↑	3.23↑
truck	14.2	7.75	13.0	6.69	1.2↑	1.06↑
tricycle	9.21	4.42	8.98	4.28	0.23↑	0.14↑
awning-tricycle	4.91	2.85	3.7	2.06	1.21↑	0.79↑
bus	24.7	14.1	20.3	9.85	4.4↑	9.85↑
motor	21.8	7.43	27.5	9.37	5.7↓	1.94↓
Parameters	1,424,004		1,772,695		19.67%↓	
GFLOPS	3.3		4.2		21.43%↓	

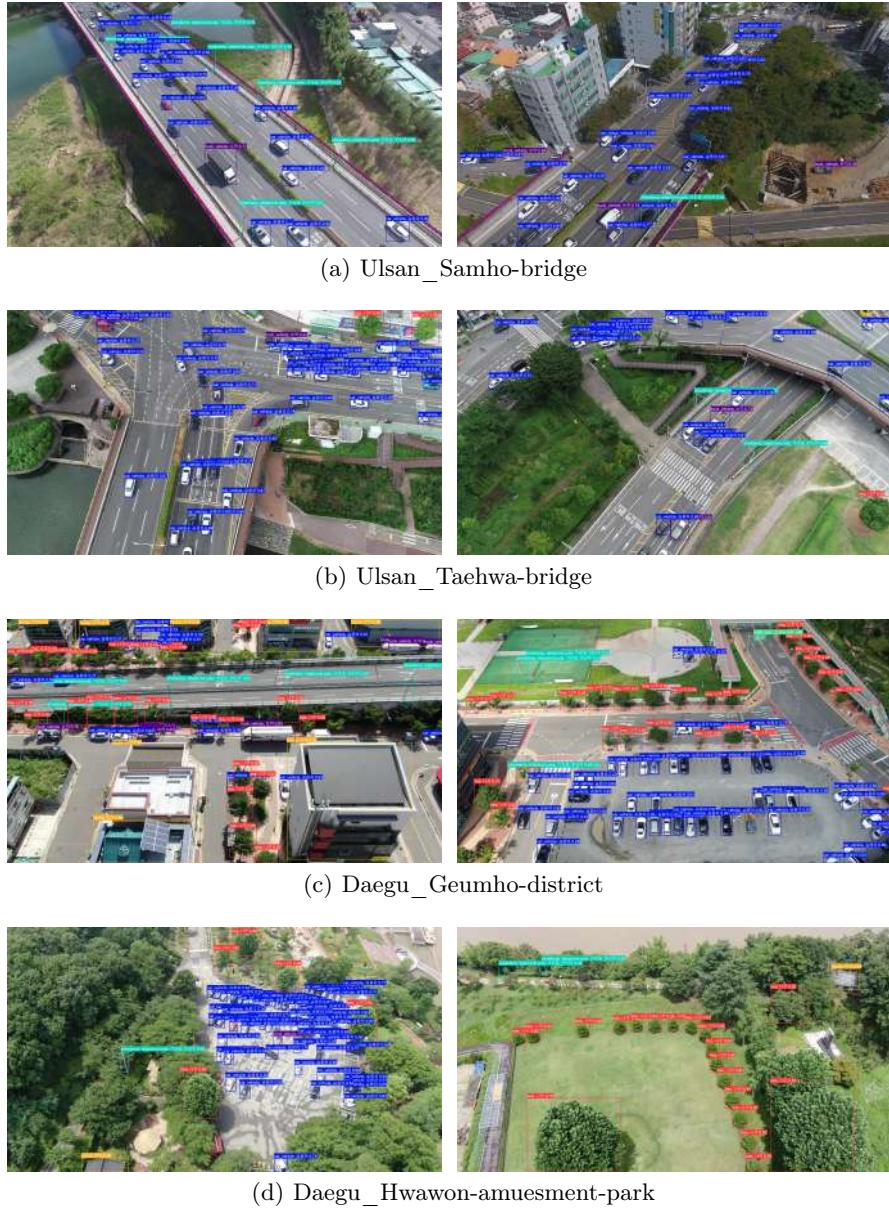


Fig. 6: Visualization of detection results on autonomous drone dataset using YOLOv5 with DDConv block.

5 Conclusion

This paper focused on the object detection work that is the basis of the technologies used in drone-based artificial intelligence systems. So, We proposed a DDConv to reduce the amount of computation of an object detection network for drone systems in which real-time is important. The DDConv includes dilated convolution and depth-wise convolution together to analyze a large area efficiently and to reduce the amount of computation for real-time systems. In addition, a detection head was added at the end of the network to find objects of more diverse scales. As a result of the experiment on the autonomous drone dataset, mAP50 decreased by 2.3% and mAP50-95 by 5.5%. In the case of the VisDrone dataset, mAP50 decreased by 0.6%, and mAP50-95 increased by 0.23%. But, both of two experiments decreased parameters and GFLOPS by about 20%. The object detection speed is almost twice as fast as the original network. The proposed network spends only 6.5 ms per image for inference. Although the accuracy is slightly lower than the original network, the majority of objects are detected as shown in Figure 6. Therefore, the proposed network in this paper is suitable for a drone-based real-time object detection systems.

6 Acknowledgements

This results was supported by "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(MOE)(2021RIS-003)

References

1. C. Chen, H. Min, Y. Peng, Y. Yang, and Z. Wang, "An intelligent real-time object detection system on drones," *Applied Sciences*, vol. 12, no. 20, 2022. [Online]. Available: <https://www.mdpi.com/2076-3417/12/20/10227>
2. J. Lee, J. Wang, D. Crandall, S. Šabanović, and G. Fox, "Real-time, cloud-based object detection for unmanned aerial vehicles," in *2017 First IEEE International Conference on Robotic Computing (IRC)*, 2017, pp. 36–43.
3. J. Choi and K. Jo, "Lightweight bird eye view detection network with bridge block based on yolov5," in *2022 International Workshop on Intelligent Systems (IWIS)*, 2022, pp. 1–4.
4. Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "FairMOT: On the fairness of detection and re-identification in multiple object tracking," *International Journal of Computer Vision*, vol. 129, no. 11, pp. 3069–3087, sep 2021. [Online]. Available: <https://doi.org/10.1007/s11263-021-01513-4>
5. S. Liu, X. Li, H. Lu, and Y. He, "Multi-object tracking meets moving uav," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 8866–8875.
6. W. Huang, X. Zhou, M. Dong, and H. Xu, "Multiple objects tracking in the uav system based on hierarchical deep high-resolution network," *Multimedia Tools Appl.*, vol. 80, no. 9, p. 13911–13929, apr 2021. [Online]. Available: <https://doi.org/10.1007/s11042-020-10427-1>

7. Y. Lin, M. Wang, W. Chen, W. Gao, L. Li, and Y. Liu, "Multiple object tracking of drone videos by a temporal-association network with separated-tasks structure," *Remote Sensing*, vol. 14, no. 16, 2022. [Online]. Available: <https://www.mdpi.com/2072-4292/14/16/3862>
8. X. Wu, W. Li, D. Hong, R. Tao, and Q. Du, "Deep learning for unmanned aerial vehicle-based object detection and tracking: A survey," *IEEE Geoscience and Remote Sensing Magazine*, vol. 10, no. 1, pp. 91–124, mar 2022. [Online]. Available: <https://doi.org/10.1109%2Fmgrs.2021.3115137>
9. X. Zhang, E. Izquierdo, and K. Chandramouli, "Dense and small object detection in uav vision based on cascade network," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 118–126.
10. H. Zhou, A. Ma, Y. Niu, and Z. Ma, "Small-object detection for uav-based images using a distance metric method," *Drones*, vol. 6, no. 10, 2022. [Online]. Available: <https://www.mdpi.com/2504-446X/6/10/308>
11. M. Liu, X. Wang, A. Zhou, X. Fu, Y. Ma, and C. Piao, "Uav-yolo: Small object detection on unmanned aerial vehicle perspective," *Sensors*, vol. 20, no. 8, 2020. [Online]. Available: <https://www.mdpi.com/1424-8220/20/8/2238>
12. G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, NanoCode012, Y. Kwon, K. Michael, TaoXie, J. Fang, imyhxy, Lorna, Yifu), C. Wong, A. V, D. Montes, Z. Wang, C. Fati, J. Nadar, Laughing, UnglvKitDe, V. Sonck, tkianai, yxNONG, P. Skalski, A. Hogan, D. Nair, M. Strobel, and M. Jain, "ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation," Nov. 2022. [Online]. Available: <https://doi.org/10.5281/zenodo.7347926>
13. F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015. [Online]. Available: <https://arxiv.org/abs/1511.07122>
14. T.-Y. Zhang, J. Li, J. Chai, Z.-Q. Zhao, and W.-D. Tian, "Improved yolov5 network with attention and context for small object detection," in *Intelligent Computing Methodologies*, D.-S. Huang, K.-H. Jo, J. Jing, P. Premaratne, V. Bevilacqua, and A. Hussain, Eds. Cham: Springer International Publishing, 2022, pp. 341–352.
15. A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilennets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.
16. G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang, "Dota: A large-scale dataset for object detection in aerial images," 2017. [Online]. Available: <https://arxiv.org/abs/1711.10398>
17. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2015. [Online]. Available: <https://arxiv.org/abs/1506.02640>
18. J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 2018. [Online]. Available: <https://arxiv.org/abs/1804.02767>
19. R. Hamaguchi, A. Fujita, K. Nemoto, T. Imaizumi, and S. Hikosaka, "Effective use of dilated convolutions for segmenting small object instances in remote sensing imagery," 2017. [Online]. Available: <https://arxiv.org/abs/1709.00179>
20. K. Jo. (2020) Autonomous drone dataset. [Online]. Available: <https://aihub.or.kr/aihubdata/data/view.do?currMenu=115topMenu=100>
21. P. Zhu, L. Wen, D. Du, X. Bian, H. Fan, Q. Hu, and H. Ling, "Detection and tracking meet drones challenge," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.

DASO: Distribution-Aware Semantics-Oriented Pseudo-label for Imbalanced Semi-Supervised Learning

Youngtaek Oh¹, Dong-Jin Kim², and In So Kweon¹

¹ Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea
 {youngtaek.oh, iskweon77}@kaist.ac.kr

² Hanyang University, Seoul, Republic of Korea
 dijnjusa@gmail.com

Abstract. Semi-supervised learning (SSL) is far from real-world application due to severely biased pseudo-labels caused by (1) class imbalance and (2) class distribution mismatch between labeled and unlabeled data. This paper addresses such a relatively under-explored problem. First, we propose a general pseudo-labeling framework that class-adaptively blends the semantic pseudo-label from a similarity-based classifier to the linear one from the linear classifier, after making the observation that both types of pseudo-labels have complementary properties in terms of bias. We further introduce a novel semantic alignment loss to establish balanced feature representation to reduce the biased predictions. We term the whole framework as **Distribution-Aware Semantics-Oriented (DASO) Pseudo-label**. We conduct extensive experiments in a wide range of imbalanced benchmarks and demonstrate that DASO reliably improves SSL learners especially when both (1) class imbalance and (2) distribution mismatch dominate.

Keywords: Semi-supervised Learning, Long-tailed Learning, Distribution Mismatch

1 Introduction

Semi-supervised learning (SSL) [4] has shown to be promising for leveraging unlabeled data to reduce the data cost [3,2,19]. The common approach is to produce *pseudo-labels* for unlabeled data based on model's predictions and utilize them for regularizing model training [13,17,19]. Although adopted in a variety of tasks, these algorithms often assume class-balanced data, while many real-world datasets exhibit *long-tailed* distributions [9]. With class-imbalanced data, pseudo-labels become severely biased to the majority classes due to confirmation bias [1]. Such pseudo-labels can further bias the model during training.

In this work, we present a new imbalanced SSL method for alleviating the bias in pseudo-labels, while discarding the common assumption that the class distribution of unlabeled data is the same with the label distribution. To this end, as shown in Fig. 1, we observe that semantic pseudo-labels [11] obtained from a

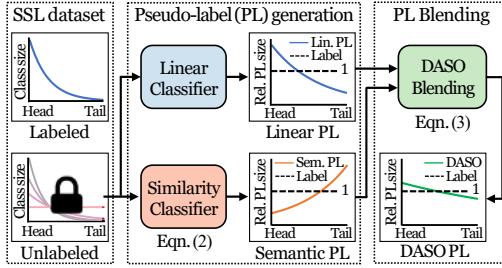


Fig. 1: DASO reduces the overall bias in pseudo-labels (PL) from unlabeled data by blending two complementary PLs from different classifiers.

similarity-based classifier [18] are biased towards minority classes as opposed to linear classifier-based pseudo-labels [17,19] being biased towards head classes. We draw the key inspiration from those complementary properties of two different types of pseudo-labels to develop a new pseudo-labeling scheme.

In this regard, we introduce a generic imbalanced SSL framework termed Distribution-Aware Semantics-Oriented (DASO) Pseudo-label. We propose to blend the linear and semantic pseudo-labels in different proportions for each class to reduce the overall bias. As such, without resorting to any class priors for the unlabeled data, DASO can reliably bring performance gain.

We further propose a simple yet effective semantic alignment loss to establish balanced feature representation. We consistently assign two different views of an unlabeled sample in *feature space* to the same prototype. These enhanced feature representations not only help linear classifier produce less biased predictions, but can also be reused for semantic pseudo-labels from similarity-based classifier.

The efficacy of DASO is extensively justified with the imbalanced versions of benchmarks: CIFAR-10/100 [15] and STL-10 [5]. We even test DASO with Semi-Aves [20], closely related to real-world scenarios. As such, DASO consistently benefits under various distributions of unlabeled data and degrees of imbalance, demonstrating to be a truly generic framework.

2 Proposed Method

2.1 Preliminaries

Problem setup. We consider K -class semi-supervised learning that leverages both labeled data $\mathcal{X} = \{(x_n, y_n)\}_{n=1}^N$ and unlabeled data $\mathcal{U} = \{u_m\}_{m=1}^M$ to train a model f . Note that the model $f = f_\phi^{\text{cls}} \circ f_\theta^{\text{enc}}$ consists of a feature encoder f_θ^{enc} followed by a linear classifier f_ϕ^{cls} , where θ and ϕ are the set of parameters of f_θ^{enc} and f_ϕ^{cls} . The input image x is paired with the label y to learn \mathcal{L}_{cls} (*e.g.*, cross-entropy) from the prediction $f(x)$. For the unlabeled data, a pseudo-label $\hat{p} \in \mathbb{R}^K$ is assigned to learn the unsupervised loss $\mathcal{L}_u = \Phi_u(\hat{p}, f(u))$, where Φ_u can be implemented via entropy [10] or consistency regularization [16,21], depending on the SSL learner. For FixMatch [19] as an example, the pseudo-label $\hat{p} = \text{OneHot}\left(\text{argmax}_k p_k^{(w)}\right)$ with $p^{(w)} = f(\mathcal{A}_w(u))$ provides the target for

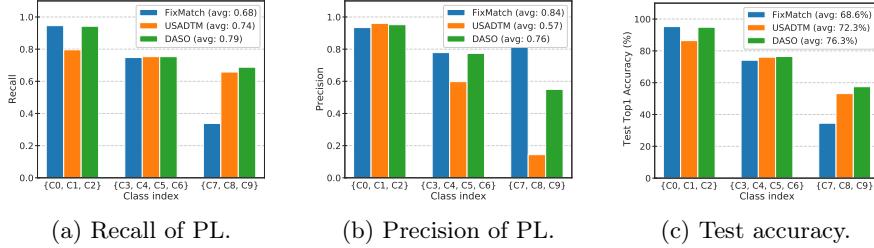


Fig. 2: Analysis on recall and precision of pseudo-labels (PL) and the corresponding test accuracy. Although USAADM [11] improves the recall of minority classes, the precision of those classes is significantly reduced. In contrast, DASO improves the recall of minority classes while sustaining the precision.

the prediction $p^{(s)} = f(\mathcal{A}_s(u))$ with some confident ones to the cross-entropy loss \mathcal{H} , where \mathcal{A}_w and \mathcal{A}_s correspond to weak augmentation (*e.g.*, random flip and crop) and advanced augmentation (*e.g.*, RandAugment [6]), respectively.

Imbalanced semi-supervised learning. Let us denote N_k and M_k as the number of labeled and unlabeled examples respectively in class k . The degree of imbalance for each data is characterized by the imbalance ratio, γ_l or γ_u , where we assume $\gamma_l = \frac{\max_k N_k}{\min_k N_k} \gg 1$. γ_u is specified in the same way using the actual labels without access during training. As note, class distribution of \mathcal{U} (*e.g.*, γ_u) can significantly diverge from \mathcal{X} in practice, and such varying distributions greatly affect the performances. In this regard, our goal is to debias the pseudo-labels with class-imbalanced data, while maintaining the performances of SSL algorithms with various, but still *unknown* class distribution of unlabeled data.

Trade-offs between linear and semantic pseudo-label. As shown in Fig. 2, we compare FixMatch [19] and USAADM [11] using linear and semantic pseudo-label respectively. From Figs. 2a and 2b, FixMatch achieves high recall in majority classes while low recall but high precision in the minorities, suggesting that actual minority class examples are biased towards head classes. In contrast, for USAADM, the actual majorities are biased towards minority classes. This is because the precision of tail classes has decreased significantly in Fig. 2b, while the recall has increased in sacrifice of the recall from head classes in Fig. 2a.

2.2 DASO Pseudo-label Framework

Framework overview. Without loss of generality, we consider DASO built on top of FixMatch [19] for convenience. First, the linear and semantic pseudo-label, \hat{p} and $q^{(w)}$ are produced with a feature $z^{(w)} = f_\theta^{\text{enc}}(\mathcal{A}_w(u))$ from the linear and similarity-based classifier, respectively. Then the final pseudo-label \hat{p}' is obtained from the distribution-aware blending process using \hat{p} and $q^{(w)}$, and it provides the target to $\mathcal{L}_u = \Phi_u(\hat{p}', p)$ instead of linear pseudo-label in the existing SSL learner. In case of FixMatch, the prediction of u corresponds to $p = p^{(s)} = f(\mathcal{A}_s(u))$. For the semantic alignment loss, the semantic pseudo-label $q^{(w)}$ provides the target for $q^{(s)}$ to the cross-entropy, where $q^{(s)}$ is the result of

the similarity-based classifier with $z^{(s)} = f_{\theta}^{\text{enc}}(\mathcal{A}_s(u))$. Note that we denote $q^{(w)}$ as \hat{q} for simplicity, unless confusion arises.

Balanced prototype generation. To execute a similarity-based classifier for obtaining the semantic pseudo-label, we first build a set of class prototypes $\mathbf{C} = \{c_k\}_{k=1}^K$ from \mathcal{X} , similar to [11]. In detail, we build a dictionary of memory queue $\mathbf{Q} = \{Q_k\}_{k=1}^K$ where each key corresponds to the class and Q_k denotes a memory queue for class k with the fixed size $|Q_k|$. The class prototype c_k for every class k is efficiently calculated by averaging the feature points in the queue Q_k , where we update Q_k for all k at every step by pushing new features from labeled data in the batch and discarding the most old ones when Q_k is full.

Linear and semantic pseudo-label generation. We obtain linear pseudo-label \hat{p} as: $\hat{p} = \sigma(f_{\phi}^{\text{cls}}(z^{(w)}))$. Semantic pseudo-label \hat{q} is obtained from the similarity classifier that measures the per-class similarity of a feature point z of either $z^{(w)}$ or $z^{(s)}$ to the prototypes \mathbf{C} : $q = \sigma(\text{sim}(z, \mathbf{C}) / T_{\text{proto}})$, where $\text{sim}(\cdot, \cdot)$ corresponds to cosine similarity, and T_{proto} is a temperature hyper-parameter. Note that \hat{p} is biased towards head classes while \hat{q} is the vice versa.

Distribution-aware blending. To obtain unbiased pseudo-label \hat{p}' , the semantic pseudo-label \hat{q} should be exploited *differently* across the class. To this end, we increase the exposure of the component of \hat{q} when \hat{p} is more biased to the head classes. Formally, we blend them with a set of distribution-aware weights $v = \{v_k\}_{k=1}^K$ to reduce the bias that might occur when using either \hat{p} or \hat{q} :

$$\hat{p}' = (1 - v_{k'})\hat{p} + v_{k'}\hat{q}, \quad (1)$$

where $v_k = \frac{1}{\max_k \hat{m}_k^{1/T_{\text{dist}}}} \left(\hat{m}_k^{1/T_{\text{dist}}} \right)$ and k' is the class prediction from \hat{p} . Note that \hat{m} is the normalized class distribution of the current pseudo-labels and T_{dist} is a hyper-parameter that intercedes the optimal trade-offs between \hat{p} and \hat{q} . Overall, in terms of the linear pseudo-label, the minority pseudo-labels will remain as minority, while pseudo-labels predicted as majority will be likely to recover the original classes thanks to large $v_{k'}$. This makes DASO flexible to various distributions of \mathcal{U} without resorting to any distribution.

Semantic alignment loss. To establish balanced feature representations, we propose new semantic alignment loss. In high-level, we align each unlabeled sample u to the most similar prototype used in the similarity classifier, by imposing *consistent assignment* for two augmented views $\mathcal{A}_w(u)$ and $\mathcal{A}_s(u)$ to the same c_k in feature space. Note \hat{q} provides the target for $q^{(s)}$ with cross-entropy \mathcal{H} :

$$\mathcal{L}_{\text{align}} = \mathcal{H}(\hat{q}, q^{(s)}). \quad (2)$$

Finally, the enhanced representation can implicitly guide the classifier f_{ϕ}^{cls} to produce less biased predictions in general.

Total objective. DASO can easily couple with other SSL algorithms with the modified pseudo-label, where the final DASO objective is as below:

$$\mathcal{L}_{\text{DASO}} = \mathcal{L}_{\text{cls}} + \lambda_u \mathcal{L}_u + \lambda_{\text{align}} \mathcal{L}_{\text{align}}, \quad (3)$$

where \mathcal{L}_{cls} and \mathcal{L}_u come from the base SSL learner, and $\mathcal{L}_{\text{align}}$ is newly introduced from DASO. Note that \mathcal{L}_u takes the blended pseudo-label in Eq. (1).

Algorithm	CIFAR10-LT				CIFAR100-LT			
	$\gamma = \gamma_l = \gamma_u = 100$	$\gamma = \gamma_l = \gamma_u = 150$	$\gamma = \gamma_l = \gamma_u = 10$	$\gamma = \gamma_l = \gamma_u = 20$	$N_1 = 500$	$N_1 = 1500$	$N_1 = 500$	$N_1 = 1500$
FixMatch [19]	67.8 ± 1.13	77.5 ± 1.32	62.9 ± 0.36	72.4 ± 1.03	45.2 ± 0.55	56.5 ± 0.06	40.0 ± 0.96	50.7 ± 0.25
w/ DARP [14]	74.5 ± 0.78	77.8 ± 0.63	67.2 ± 0.32	73.6 ± 0.73	49.4 ± 0.20	58.1 ± 0.44	43.4 ± 0.87	52.2 ± 0.66
w/ CReST+ [22]	76.3 ± 0.86	78.1 ± 0.42	67.5 ± 0.45	73.7 ± 0.34	44.5 ± 0.94	57.4 ± 0.18	40.1 ± 1.28	52.1 ± 0.21
w/ DASO (Ours)	76.0 ± 0.37	79.1 ± 0.75	70.1 ± 1.81	75.1 ± 0.77	49.8 ± 0.24	59.2 ± 0.35	43.6 ± 0.09	52.9 ± 0.42

Table 1: Comparison of accuracy (%) on CIFAR10/100-LT under $\gamma_l = \gamma_u$ setup.

Algorithm	CIFAR10-LT ($\gamma_l \neq \gamma_u$)				STL10-LT ($\gamma_u = N/A$)			
	$\gamma_u = 1$ (uniform)	$\gamma_u = 1/100$ (reversed)	$\gamma_l = 10$	$\gamma_l = 20$	$N_1 = 500$	$N_1 = 1500$	$N_1 = 500$	$N_1 = 1500$
FixMatch [19]	73.0 ± 3.81	81.5 ± 1.15	62.5 ± 0.94	71.8 ± 1.70	56.1 ± 2.32	72.4 ± 0.71	47.6 ± 4.87	64.0 ± 2.27
w/ DARP [14]	82.5 ± 0.75	84.6 ± 0.34	70.1 ± 0.22	80.0 ± 0.93	66.9 ± 1.66	75.6 ± 0.45	59.9 ± 2.17	72.3 ± 0.60
w/ CReST [22]	83.2 ± 1.67	87.1 ± 0.28	70.7 ± 2.02	80.8 ± 0.39	61.7 ± 2.51	71.6 ± 1.17	57.1 ± 3.67	68.6 ± 0.88
w/ CReST+ [22]	82.2 ± 1.53	86.4 ± 0.42	62.9 ± 1.39	72.9 ± 2.00	61.2 ± 1.27	71.5 ± 0.96	56.0 ± 3.19	68.5 ± 1.88
w/ DASO (Ours)	86.6 ± 0.84	88.8 ± 0.59	71.0 ± 0.95	80.3 ± 0.65	70.0 ± 1.19	78.4 ± 0.80	65.7 ± 1.78	75.3 ± 0.44

Table 2: Comparison of accuracy (%) for imbalanced SSL methods on CIFAR10-LT and STL10-LT under $\gamma_l \neq \gamma_u$ setup.

3 Experiments

3.1 Experimental Setup

Datasets. We conduct SSL experiments with various scenarios where the class distribution of unlabeled data can deviate from that of labeled data. We adopt CIFAR-10/100 [15] and STL-10 [5] typically adopted in SSL literature [19]. We make the imbalanced versions by exponentially decreasing the amount of samples per class following [7,14]. We also consider Semi-Aves [20], which is the large-scale collection of bird species with natural long-tailed distribution.

Baseline methods. We mainly adopt *FixMatch* [19] as baseline and consider *DARP* [14] and *CReST* [22] for comparison.

Training and evaluation. We train Wide ResNet-28-2 [23] on CIFAR10/100-LT and STL10-LT . For Semi-Aves, we fine-tune ResNet-34 [12] pre-trained on ImageNet [8]. For evaluation, we measure the top-1 accuracy.

3.2 Results on CIFAR10/100-LT and STL10-LT.

In case of $\gamma_l = \gamma_u$. We compare imbalanced SSL methods: DARP [14] and CReST+ [22] with the proposed DASO on FixMatch. Remarkably, DASO shows comparable or even better results in most setups with significant gains compared to baseline FixMatch, although DARP and CReST+ even push the predictions of unlabeled data to the label distribution using the assumption $\gamma_l = \gamma_u$ (*i.e.*, distribution alignment [2]). This verifies the efficacy of DASO for debiasing, even without resorting to the label distribution.

In case of $\gamma_l \neq \gamma_u$. For CIFAR10-LT, we consider two extreme cases for the class distribution of unlabeled data: uniform ($\gamma_u = 1$) and flipped long-tail ($\gamma_u = 1/100$) with respect to the labeled data. For STL10-LT, since we cannot control

Benchmark	Semi-Aves							
	$\mathcal{U} = \mathcal{U}_{in}$			$\mathcal{U} = \mathcal{U}_{in} + \mathcal{U}_{out}$				
Method	Last	Top1	Med20	Top1	Last	Top1	Med20	Top1
FixMatch [19]	53.8 ± 0.17	53.8 ± 0.13	45.7 ± 0.89	46.1 ± 0.50				
w/ DARP [14]	52.3 ± 0.48	52.1 ± 0.48	46.3 ± 0.70	46.4 ± 0.61				
w/ CReST [22]	52.1 ± 0.36	52.2 ± 0.27	43.6 ± 0.69	43.6 ± 0.68				
w/ CReST+ [22]	53.9 ± 0.38	53.8 ± 0.38	45.1 ± 1.09	45.2 ± 1.00				
w/ DASO (Ours)	54.5 ± 0.08	54.6 ± 0.12	47.9 ± 0.41	47.9 ± 0.38				

Table 3: Accuracy on Semi-Aves [20]. DASO shows the best among imbalanced SSL methods. DASO also performs well in presence of large \mathcal{U}_{out} .

the size and imbalance of unlabeled data due to unknown labels, we instead set $\gamma_l \in \{10, 20\}$ with the whole fixed unlabeled data. Table 2 summarizes the results of imbalanced SSL methods under the setups.

Surprisingly, DASO outperforms other baselines by significant margins in most cases. Though DARP [14] estimates the distribution of unlabeled data in advance as prior, the estimation accuracy decreases as using less labels for training. Under $\gamma_l \neq \gamma_u$, we evaluate both CReST and CReST+ [22]. Clearly, resorting to the label distributions as the prior for unlabeled data in CReST+ rather harms the accuracy since the assumption of $\gamma_l = \gamma_u$ is violated. The accuracy loss becomes more severe under $\gamma_u = 1/100$.

By virtue of debiased pseudo-labels from DASO, the abundant minority-class unlabeled samples are correctly used. Consequently, the results confirm that conditioning on a certain distribution for unlabeled data (*e.g.*, $\gamma_u = \gamma_l$) is undesirable in imbalanced SSL, and DASO greatly reduces the bias in presence of distribution mismatch, even without access to the distribution.

3.3 Results on Large-Scale Semi-Aves

We test DASO on a realistic Semi-Aves benchmark [20]. Both labeled data (\mathcal{X}) and unlabeled data (\mathcal{U}) show long-tailed distributions, while \mathcal{U} contains large *open-set* examples (\mathcal{U}_{out}) that do not belong to any of the classes in \mathcal{X} . The results are shown in Table 3. We report both cases: $\mathcal{U} = \mathcal{U}_{in}$ and $\mathcal{U} = \mathcal{U}_{in} + \mathcal{U}_{out}$, where \mathcal{U}_{in} contains examples that share the class of \mathcal{X} .

In case of $\mathcal{U} = \mathcal{U}_{in}$. As it has the distribution gap between \mathcal{X} and \mathcal{U} , DARP [14] and CReST [22] show only a slight gain or even unsatisfactory performances compared to FixMatch [19]. In contrary, DASO shows the best performance among the baselines with favorable improvements upon FixMatch.

In case of $\mathcal{U} = \mathcal{U}_{in} + \mathcal{U}_{out}$. Since \mathcal{U} contains large amount of *open-set* class examples, performance drop is observed consistently across all baselines. Among them, DASO shows the best performance with favorable gain. DARP [14] is slightly helpful for optimization. Concerning CReST and CReST+ [22], they rather performs poorly than FixMatch due to noisy predictions from \mathcal{U}_{out} . As such, DASO has superiority in the challenging but practical scenario of long-tailed distributions, even in presence of large amount of open-set examples.

	\mathcal{L}_{align}	C10	STL10
FixMatch	✗	68.25	55.53
DASO	✗	70.98	61.64
FixMatch	✓	73.15	58.51
DASO	✓	75.97	70.21

Table 4: Ablation study on blending and the semantic alignment loss.

	C10	STL10
$v_k = 0$	73.15	58.51
$v_k = 1$	72.35	62.60
$v_k = 0.5$	72.96	64.21
DASO	75.97	70.21

Table 5: Ablation study on pseudo-label blending strategy.

3.4 Ablation Study

Component analysis. Table 4 studies the two major components of DASO: distribution-aware pseudo-label blending and the semantic alignment loss. Both blending mechanism and $\mathcal{L}_{\text{align}}$ provides significant gain over FixMatch. For example, the blending and $\mathcal{L}_{\text{align}}$ achieve about 6% and 3% absolute gain, respectively, and combining both shows 15.7% gain in total on STL10.

Effect of pseudo-label blending. Table 5 studies the different way of pseudo-label blending on DASO with *constant* weights. Due to the bias in the pseudo-labels, using either linear ($v_k = 0$) or semantic ($v_k = 1$) pseudo-label leads to a marginal gain. In addition, blending them with the same ratio ($v_k = 0.5$) shows the lower performance compared to our final DASO, which demonstrates that distribution-aware class-adaptive blending is crucial for imbalanced SSL.

4 Conclusion

We proposed a novel distribution-aware semantics-oriented (DASO) pseudo-label for imbalanced semi-supervised learning. DASO adaptively blends the linear and semantic pseudo-labels within each class to mitigate the overall bias across the class. Moreover, we introduced semantic alignment loss. From extensive experiments, we showed the efficacy of DASO on challenging and realistic setups, especially when class imbalance and class distribution mismatch dominate.

Remarks This paper is a re-publishing (summary presentation) of the paper which has been published in *2022 IEEE CVF Computer Vision and Pattern Recognition Conference (CVPR)* by request of the IW-FCV2023 program committee to share the research results.

References

- Eric Arazo, Diego Ortego, Paul Albert, Noel E O’Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *International Joint Conference on Neural Networks (IJCNN)*, 2020. [1](#)
- David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *International Conference on Learning Representations (ICLR)*, 2020. [1](#), [5](#)
- David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2019. [1](#)
- Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning. *IEEE Transactions on Neural Networks*, 2009. [1](#)
- Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011. [2](#), [5](#)
- Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Advances in Neural Information Processing Systems (NIPS)*, 2020. [3](#)

7. Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. [5](#)
8. Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. [5](#)
9. Qi Dong, Shaogang Gong, and Xiatian Zhu. Imbalanced deep learning by minority class incremental rectification. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018. [1](#)
10. Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems (NIPS)*, 2005. [2](#)
11. Tao Han, Junyu Gao, Yuan Yuan, and Qi Wang. Unsupervised semantic aggregation and deformable template matching for semi-supervised learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2020. [1](#), [3](#), [4](#)
12. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. [5](#)
13. Dong-Jin Kim, Jinsoo Choi, Tae-Hyun Oh, and In So Kweon. Image captioning with very scarce supervised data: Adversarial semi-supervised learning approach. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019. [1](#)
14. Jaehyung Kim, Youngbum Hur, Sejun Park, Eunho Yang, Sung Ju Hwang, and Jinwoo Shin. Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2020. [5](#), [6](#)
15. Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Technical report*, 2009. [2](#), [5](#)
16. Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2016. [2](#)
17. Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, 2013. [1](#), [2](#)
18. Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2017. [2](#)
19. Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems (NIPS)*, 2020. [1](#), [2](#), [3](#), [5](#), [6](#)
20. Jong-Chyi Su and Subhransu Maji. The semi-supervised inaturalist-aves challenge at fgvc7 workshop, 2021. [2](#), [5](#), [6](#)
21. Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems (NIPS)*, 2017. [2](#)
22. Chen Wei, Kihyuk Sohn, Clayton Mellina, Alan Yuille, and Fan Yang. Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. [5](#), [6](#)
23. Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference (BMVC)*, 2016. [5](#)

Improvement of Robustness to Noise for Medical Image Segmentation by using Self-Supervised Learning Approach*

Yuta Konishi^[0000-0003-0565-564X] and Takio Kurita^[0000-0003-3982-6750]

Hiroshima University,
1-7-1 Kagamiyama, Higashi Hiroshima, 739-8521, Japan

Abstract. It is crucial to make the trained model robust to the distortions such as pixel noises in medical image segmentation. Recently it has been pointed out that self-supervised learning (SSL) methods such as SimCLR, VICReg, and Barlow Twins are closely related to spectral methods such as Laplacian Eigenmaps, Multidimensional Scaling, etc. This means that SSL can construct features invariant to the perturbations introduced by data augmentations. Since invariant feature extraction is also fundamental in medical image segmentation, in this paper, we proposed introducing SSL loss as a regularizer in U-Net for medical image segmentation. Pixel noise is applied to the training samples, and invariant features to such distortions are extracted in the hidden layer of U-Net. The effectiveness of the proposed approach is experimentally confirmed using the subset of Sunnybrook Cardiac Data (SCD) and Abdominal organs segmentation dataset by Combined (CT-MR) Healthy Abdominal Organ Segmentation (CHAOS) Challenge.

Keywords: Invariant feature extraction · Self-supervised learning · Medical image segmentation · U-Net · SimCLR.

1 Introduction

Medical image segmentation is used to identify the pixels of organs or lesions from medical images such as CT or MRI images and is regarded as one of the most important tasks in medical image analysis [12]. Deep learning is now recognized as one of the best approaches for medical image segmentation [30]. Many network architectures, such as the fully convolutional neural network (FCN) [17] or U-Net [24], have been used to segment medical images.

U-Net is one of the most well-known architectures for medical image segmentation. The encoder-decoder architecture is utilized, and skip connections between different stages of the network are introduced, as shown in Fig.1. Many researchers applied the U-Net base model for medical image segmentation[6, 8].

Invariant feature extraction is one of the central topics in machine learning and pattern recognition, and it is also important in deep learning. The standard

* Supported by KEKEN 21K12049.

approach to making robust to unnecessary variations is to train a deep learning model by using a large number of training samples that include all possible variations. There are some researches in which invariant features are extracted by using deep learning. For example, pose-invariant features are extracted using Convolutional Neural Networks (CNN) for pose-invariant face recognition [1]. Metric learning has also often been used for invariant feature extraction [13, 16]. Ueda et al. proposed an invariant feature extraction method using Gradient Reversal Layer (GRL) [27].

Self-Supervised Learning (SSL) is one of the most promising methods to learn data representations that generalize across downstream tasks [3]. Labels for the training samples are not required, but the knowledge of what makes some samples semantically close to others is trained. Usually, semantic similarity is constructed by augmenting the training samples through data augmentations.

One of the basic SSL methods is SimCLR (a simple framework for contrastive learning of visual representations) [5]. SimCLR learns representations by maximizing agreement between differently augmented views of the same sample via a contrastive loss in the latent space. Recently Balestrieri et al. [3] demonstrated that SSL methods such as SimCLR [5], VICReg [4], and Barlow Twins [29] are closely related with the spectral methods such as Laplacian Eigenmaps, Multidimensional Scaling, etc. This means that SSL is extracting features (embeddings) that are invariant to the perturbations introduced by data augmentations.

The invariant feature extraction is also fundamental in supervised learning. Ramyaa et al. proposed to combine Barlow Twins loss with the standard cross entropy loss for the supervised learning with CNN [23].

In this paper, we propose to use SSL loss as a regularizer in U-Net-based medical image segmentation. Pixel noise is applied to the training samples as the distortions to the medical images, and the invariant features to such distortions are extracted by introducing SSL loss in the hidden layers of the U-Net. To show the effectiveness of the proposed approach, we have performed experiments using the subset of Sunnybrook Cardiac Data (SCD) [22] and Abdominal organs segmentation dataset by Combined (CT-MR) Healthy Abdominal Organ Segmentation (CHAOS) Challenge[14].

The contributions of this paper are summarized as follows:

- (1) SSL loss in the hidden layers of the U-Net is introduced to make the trained model for medical image segmentation robust to the pixel noises.
- (2) The effectiveness of the proposed approach is experimentally confirmed using the subset of Sunnybrook Cardiac Data (SCD) [22] and Abdominal organs segmentation dataset by Combined (CT-MR) Healthy Abdominal Organ Segmentation (CHAOS) Challenge[14].

2 Related Work

2.1 Medical Image Segmentation

Image segmentation is a computer vision technique that divides a region in an image into several objects. It has been applied in a wide range of fields,

such as medical image analysis, scene understanding, robotic perception, video surveillance, augmented reality, image compression, automatic driving, and so on [19].

Image segmentation plays a crucial role in many medical image analyses in which the pixels of organs or lesions are identified from medical images such as CT or MRI images [21, 12]. Deep learning is now recognized as one of the best approaches for medical image segmentation [30].

Many network structures have been used for medical image segmentation. One of the basic deep learning models is Convolutional Neural Networks (CNNs). A CNN consists of a stack of layers such as convolution, pooling, and fully connected layers [26]. In the fully convolutional neural network (FCN) proposed by Long et al. [17], the fully convolutional layer is used at the last layer instead of the fully-connected layers in the standard CNN. With this replacement, the network makes a dense pixel-wise prediction easy.

One of the most well-known structures for medical image segmentation is U-Net, proposed by Ronneberger et al. [24]. By introducing deconvolution, the encoder-decoder architecture is realized in U-Net. Also, U-Net introduces skip connections between different stages of the network. These connections bypass the information between the layers of equal resolution in the encoding path to the decoding path. This is the most important property of U-Net. Many researchers applied the U-Net base model for medical image segmentation [6, 8].

Deep learning-based models have achieved good segmentation accuracy. However, to train the network, a large number of annotated training samples are required [17]. Collecting such huge training samples is often very tough and expensive in medical image analysis. The most common approach to increase the size of the training samples is data augmentation in which a set of perturbations are applied to the images in the training samples [18, 7, 20].

Another solution to this problem is transfer learning. Transfer learning employs the knowledge learned in a different source domain to a target task [25]. Transfer learning has been proven to have better performance when the tasks of the source and target network are more similar.

It is common in medical images that the anatomy of interest only occupies a very small portion of the image. Namely, most pixels belong to the background area, while these small organs (anomalies) are more important for medical diagnosis. Training a network with such data often leads to the trained network being biased toward the background. A popular solution to this issue is sample re-weighting, where a higher weight is applied to the foreground patches. Dice loss is often used for automatic re-weighting [15, 24, 31, 28].

Another approach is introducing the prior knowledge into the loss function as a regularizer. For example, Euler characteristics (EC) from topology are used to calculate the number of isolated objects on segmented vessel regions in the fundus image. It is used as the regularizer for training [10]. It is also useful to utilize information on the neighboring pixel relationship. Hakim et al. [11] proposed introducing a regularization term defined based on the differences of neighboring

pixels. The regularization term can be represented as Graph Laplacian computed from the output of the network and the ground-truth image.

2.2 Self-Supervised Learning and Invariant Feature Extraction

Recently it has been shown that Self-Supervised Learning (SSL) can extract features with the same level as supervised learning with large training samples [3]. SSL can build representations of data without labels and give significant advances in various applications such as natural language processing, speech processing, and computer vision [2].

In SSL for computer vision applications, the distortions or perturbations are added to the original image. The features extracted from the distorted images are trained so that they are close to each other. This is achieved by maximizing the similarity of representations obtained with different distortions using a variant of Siamese networks [9]. Thus SSL can learn invariant representations (embeddings) to the added distortions of the input image.

SimCLR SimCLR (a simple framework for contrastive learning of visual representations) is one of the basic methods for contrastive self-supervised learning [5]. SimCLR learns representations by maximizing agreement between differently augmented views of the same sample via a contrastive loss in the latent space.

Let $\{\mathbf{x}_k | k = 1, \dots, N\}$ be the training samples in a mini-batch. At first, for each training sample in the mini-batch, a stochastic data augmentation is applied to randomly generated two views of the same sample, denoted $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{x}}_j$, which are considered as a positive pair. Then we obtain $2N$ pairs of the augmented samples derived from the samples in the mini-batch. These augmented samples include a positive pair $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{x}}_j$, which are generated from the same training sample \mathbf{x}_i . The pairs of the augmented samples are fed into the neural network encoder to get the hidden representation $\mathbf{h}_i = f(\tilde{\mathbf{x}}_i)$. The contrastive loss is applied after the hidden representation is mapped by a small neural network projection head as $\mathbf{z}_i = g(\mathbf{h}_i)$.

The loss function for a positive pair of examples (i, j) is defined as

$$l_{i,j} = -\log \frac{\exp(sim(\mathbf{z}_i, \mathbf{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(sim(\mathbf{z}_i, \mathbf{z}_k)/\tau)} \quad (1)$$

where $sim(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u}^T \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}$ is the cosine similarity between two vectors \mathbf{u} and \mathbf{v} and $\mathbb{1}_{[k \neq i]}$ is an indicator function evaluating to 1 if $k \neq i$. τ denotes a temperature parameter that controls the scale. This loss function can be used to learn to keep positive pairs in a mini-batch close together and other pairs apart.

Other SSL methods For SSL, it is important to prevent a collapse in which the encoders produce constant or non-informative representations. Bardes et al.

proposed VICReg (Variance-Invariance-Covariance Regularization), which explicitly avoids the collapse problem with two regularization terms [4]. One term maintains the variance of each embedding dimension above a threshold, and the other decorrelates each pair of variables.

Another method is Barlow Twins, which applies H. Barlow's redundancy-reduction principle [29]. The objective function of Barlow Twins measures the cross-correlation matrix between the embeddings of two identical networks fed with distorted versions of a batch of samples and tries to make this matrix close to the identity matrix. This makes the embedding vectors of distorted versions similar while minimizing the redundancy between the components of these vectors. It is reported that Barlow Twins is competitive with state-of-the-art methods for SSL.

Balestrieri et al. [3] demonstrated that SSL methods such as SimCLR, VICReg, and Barlow Twins are closely related to spectral methods such as Laplacian Eigenmaps, Multidimensional Scaling, etc. This shows that invariant feature extraction is fundamental in SSL.

Since it is obvious that the invariant feature extraction is also important in supervised learning, Barlow Twins loss is combined with the standard cross-entropy loss as a regularizer in the supervised learning with CNN [23]. This paper proposes to use SSL loss as a regularizer in U-Net for medical image segmentation.

3 Proposed Method

3.1 Overview of the network architecture

As discussed in 2.2, SSL can learn representations that are invariant to the distortions applied to images. Taking advantage of this property, we designed a mechanism to promote learning that is robust to pixel noises in the medical image segmentation tasks.

A branch of the linear layer is connected to the middle layer of the segmentation model (U-Net) as shown in Figure 1, and SSL is performed with the output vectors of the branch. Each part of U-Net is named as shown in Figure 1, and a branch for SSL is connected to an arbitrary location.

3.2 Training flow

We follow the learning method used in SimCLR's paired data learning. Gaussian noise are applied to the original images, and they are paired with the original images.

The original and the distorted images are fed to the mainstream (U-Net), and the outputs of the U-Net are used to compute the segmentation loss function for each image. The SSL branch of the sub-stream outputs the feature vectors of the original and the distorted images, and these vectors are used to compute the SSL loss function. This allows us to capture representations that are invariant

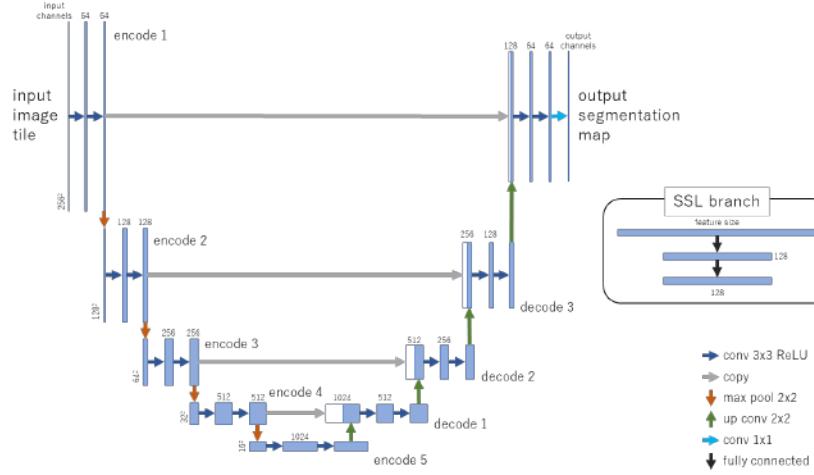


Fig. 1. Overview of U-Net and the SSL branch. The SSL branch is connected to the location encode1, encode2, encode3, encode4, encode5, decode1, decode2 or decode3.

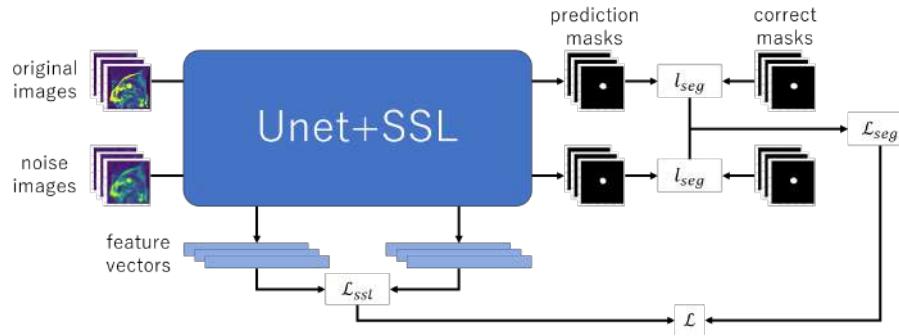


Fig. 2. Training flow of the proposed method. The best accuracy is obtained when SSL branch is connected to encode2 or decode2.

to this distortion (noise) during training for the segmentation task and to make the trained model robust to such distortions. The overview of the training flow of the proposed method is shown in Figure 2.

In the proposed learning flow, Segmentation learning and SSL are performed simultaneously. This means that the loss functions must be computed and fused. In this study, the loss function is defined as the weighted sum of the loss functions of segmentation and SSL as

$$\mathcal{L} = \lambda \mathcal{L}_{seg} + (1 - \lambda) \mathcal{L}_{ssl} \quad (2)$$

where \mathcal{L}_{seg} and \mathcal{L}_{ssl} are the loss functions of segmentation and SSL and λ is a hyper-parameter to control the ratio of the two loss functions. Since the segmentation loss function is computed for each of the original and the distorted images, the segmentation loss \mathcal{L}_{seg} is defined by their respective averages as

$$\mathcal{L}_{seg} = \frac{l_{seg}(Y_{original}) + l_{seg}(Y_{noise})}{2} \quad (3)$$

where l_{seg} is the loss function of segmentation and $Y_{original}$ and Y_{noise} are the outputs of U-Net for the original and the distorted images. In this study, Cross-entropy Loss is used for segmentation loss and InfoNCE Loss (eq (1))

$$l_{i,j} = -\log \frac{\exp(sim(\mathbf{z}_i, \mathbf{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(sim(\mathbf{z}_i, \mathbf{z}_k)/\tau)}$$

is used for SSL loss.

4 Experiments

4.1 Datasets

To evaluate the effectiveness of the proposed approach, we have performed experiments using two datasets. They are the subset of Sunnybrook Cardiac Data (SCD) [22] and Abdominal organs segmentation dataset by Combined (CT-MR) Healthy Abdominal Organ Segmentation (CHAOS) Challenge[14]. Images of the datasets are resized (bilinear) to 256×256 pixels.

Subset of SCD. The SCD also called the 2009 Cardiac MR Left Ventricular Segmentation Challenge data, consists of 45 cine MRI images of various patients and conditions. The SCD subset used in this study consists of gray-scale cardiac MRI images (short-axis images) and expert-masked data of the left ventricular region. The masked data is a binary image with 1 inside the region of the left ventricle and 0 in other regions. The training data set consists of 234 image pairs, and the validation data set consists of 26 image pairs. They do not overlap each other.

Abdominal organs segmentation dataset by CHAOS challenge. The CHAOS Challenge is aimed at segmenting organs (liver, kidneys, spleen) from abdominal CT and MRI data. CT and MRI are provided in DICOM image data, each with masked images of abdominal organs. The CT dataset is data acquired for the pre-evaluation of living liver transplant donors and is intended for the segmentation of the liver. The MRI data set consists of data from two different sequences (T1-DUAL and T2-SPIR) and is intended for the segmentation of the four abdominal organs (liver, right and left kidneys, and spleen). The MRI T2-SPIR data set was used in this experiment. As mentioned earlier, this data set is DICOM image data, so it was converted to JPEG image data for easier handling. Of the total MRI images, 531 were used as training data and 92 as validation data. The classes to be classified are the four abdominal organs (liver, right and left kidneys, and spleen) as described above.

4.2 Experimental details

Distortions The distortion used in this study is Gaussian noise. Gaussian noise is statistical noise that has the same probability density function as the Gaussian distribution. The noise image was generated by adding 0.3 times the Gaussian noise (standard normal distribution) to the original image.

Learning parameters The batch size was set to 9, and Adam was used as the optimizer. For the subset of SCD, the number of epochs was set to 100, and the learning rate was set to 0.001, which was multiplied by 0.5 every 25 epochs. The weight decay was set to 0.001. For the Abdominal organs segmentation dataset, the number of epochs was set to 250, and the learning rate was set to 0.0001, which was multiplied by 0.5 every 40 epochs. The weight decay was set to 0.01.

Evaluation Multi-class IoU and pixel-wise accuracy, which are common metrics for segmentation tasks, were used for evaluation. After training the model, prediction using the trained model is performed on the original images and the distorted images, and each is evaluated.

4.3 Ablation study using SCD dataset

To confirm the usefulness of our proposed method, we conducted a preliminary experiment using the subset of SCD. The U-Net was trained with only the original images of the subset of SCD. This is denoted as baseline1. Also, the U-Net was trained with the samples in which 50% of the samples are replaced with the distorted samples. This is denoted as baseline2. Then the performance of the proposed method is compared with these baselines.

In these experiments, the parameter λ in the loss function was set to 0.8.

Optimum Location of SSL branch To find the best location of the SSL branch, we connected the SSL branch in different layers in the U-Net and evaluated the test accuracy of the trained models for the original test samples and the distorted images of the test samples. In this experiment, SSL branch was connected to 8 locations of U-Net encode1-5, decode1-3, and each of them was trained to compare their performance. The results of the experiment are shown in Table 1.

Table 1. Comparison of accuracy for the subset of SCD. To find the optimal location of the SSL branch of the proposed method, the accuracy was evaluated by changing the location of the SSL branch in the U-Net.

model	multi-class IoU (%)		pixel-wise accuracy (%)	
	original images	distorted images	original images	distorted images
baseline 1	94.63	49.12	99.81	98.23
baseline 2	94.28	93.53	99.80	99.77
encode1	94.54	93.07	99.81	99.75
encode2	95.32	94.06	99.83	99.79
encode3	94.47	90.44	99.80	99.66
encode4	94.16	92.50	99.78	99.72
encode5	94.87	92.48	99.82	99.72
decode1	94.28	92.83	99.79	99.75
decode2	95.56	93.92	99.84	99.78
decode3	94.71	93.05	99.81	99.75

From Table 1, it is noticed that the accuracy of the proposed method is better than the baselines, especially for the distorted test images. This means that the proposed approach can make the trained model robust to distortion such as pixel noise.

The best accuracy was achieved at decode2 for the original images and encode2 for the distorted images. The results suggest that visually relevant features, such as the contours of objects in the image, are more effective in making the learned model robust to variations such as pixel noise. In contrast, the extraction of class information is more important for the original images. Thus the reason why these results are obtained is probably that the distortions (pixel noises) used in this experiment are local and the deeper layers are probably effective for more global distortions.

In subsequent experiments, the SSL branch is connected to encode2, which had the best accuracy for the distorted images, and decode2, which had the best accuracy on the decoder side.

Optimum value of λ . Next, we performed experiments to find the best value of the parameter λ which controls the valance between the segmentation loss and SSL loss. The accuracy for the original test samples and the distorted samples was evaluated by changing the parameter λ from 0.1 to 0.9 in 0.1 increments.

Table 2. Comparison of accuracy by λ (%)

λ	multi-class IoU		pixel-wise accuracy	
	original images	distorted images	original images	distorted images
0.1	93.92	90.31	99.78	99.66
0.2	94.39	91.57	99.80	99.70
0.3	93.67	85.92	99.78	99.51
0.4	94.17	93.29	99.78	99.76
0.5	94.34	92.63	99.79	99.73
0.6	94.73	92.54	99.80	99.74
0.7	95.08	93.18	99.82	99.76
0.8	95.32	94.06	99.83	99.79
0.9	94.93	93.84	99.82	99.78

Table 2 shows the accuracy obtained for each parameter λ . The best accuracy is achieved for the distorted images when the value of λ is 0.8. It is also noticed that the proposed method is robust to the small changes of this parameter λ .

In the following experiments, the value of λ is set to 0.8.

4.4 Experiments with Abdominal organs segmentation dataset

We have also conducted experiments on the Abdominal organs segmentation dataset, which consists of color images and the task is multi-class segmentation. The results are summarized in Table 3.

Table 3. Comparison of accuracy with Abdominal organs segmentation dataset.

model	multi-class IoU (%)		pixel-wise accuracy (%)	
	original images	distorted images	original images	distorted images
baseline 1	87.45	19.11	99.43	95.52
baseline 2	85.47	81.63	99.23	98.86
encode2	83.92	79.24	99.10	98.89
decode2	83.89	82.64	99.00	98.91

From Table 3, it can be confirmed that for distorted images, the best accuracy for both multi-class IoU and per-pixel accuracy is obtained when the SSL branch is connected to decode2. This suggests that in the case of multi-class segmentation, the accuracy can be improved by acquiring features from the U-Net decoder side using the SSL branch.

Figure 3 shows the segmentation results for the distorted training image of the Abdominal organs segmentation dataset. It can be seen that the proposed method is able to segment organ contours more clearly than baseline2. The results show that the proposed method is robust to distortions such as pixel noise by introducing SSL loss in the hidden layer of U-Net.

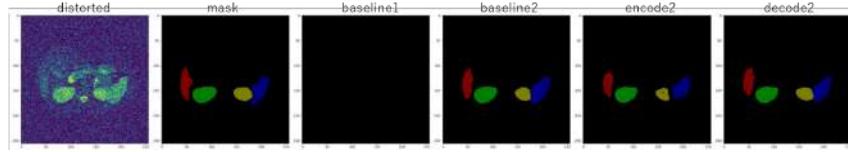


Fig. 3. Comparison of the segmentation results of the baselines and the proposed methods for the distorted testing images. (red: liver, green: right kidney, yellow: left kidney, blue: spleen)

5 Conclusion

We proposed a learning method for U-Net with SSL to make the trained model robust against image distortions such as pixel noise. The proposed method (U-Net with SSL) can construct the segmentation model by extracting features that are invariant to distortions in the paired data. The effectiveness of the proposed approach was experimentally confirmed by using the subset of Sunnybrook Cardiac Data (SCD) and Abdominal organs segmentation dataset.

In this paper, we used only pixel noise as image distortion. We think the approach proposed in this paper can apply to the other types of image distortions. Experiments for such distortions will be our future works.

References

1. Ahmed, S.B., Ali, S.F., Ahmad, J., Adnan, M., Fraz, M.M.: On the frontiers of pose invariant face recognition: a review. *Artificial Intelligence Review* **53**(4), 2571–2634 (2020)
2. Baevski, A., Hsu, W.N., Xu, Q., Babu, A., Gu, J., Auli, M.: Data2vec: A general framework for self-supervised learning in speech, vision and language. arXiv preprint arXiv:2202.03555 (2022)
3. Balestrieri, R., LeCun, Y.: Contrastive and non-contrastive self-supervised learning recover global and local spectral embedding methods. arXiv preprint arXiv:2205.11508 (2022)
4. Bardes, A., Ponce, J., LeCun, Y.: Vicreg: Variance-invariance-covariance regularization for self-supervised learning. arXiv preprint arXiv:2105.04906 (2021)
5. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020)
6. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3d u-net: learning dense volumetric segmentation from sparse annotation. In: International conference on medical image computing and computer-assisted intervention. pp. 424–432. Springer (2016)
7. Golan, R., Jacob, C., Denzinger, J.: Lung nodule detection in ct images using deep convolutional neural networks. In: 2016 international joint conference on neural networks (IJCNN). pp. 243–250. IEEE (2016)
8. Gordienko, Y., Gang, P., Hui, J., Zeng, W., Kochura, Y., Alienin, O., Rokovy, O., Stirenko, S.: Deep learning with lung segmentation and bone shadow exclusion

- techniques for chest x-ray analysis of lung cancer. In: International conference on computer science, engineering and education applications. pp. 638–647. Springer (2018)
9. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06). vol. 2, pp. 1735–1742. IEEE (2006)
 10. Hakim, L., Kavitha, M.S., Yudistira, N., Kurita, T.: Regularizer based on euler characteristic for retinal blood vessel segmentation. *Pattern Recognition Letters* **149**, 83–90 (2021)
 11. Hakim, L., Zheng, H., Kurita, T.: Improvement for single image super-resolution and image segmentation by graph laplacian regularizer based on differences of neighboring pixels. Manuscript submitted for publication (2021)
 12. Hesamian, M.H., Jia, W., He, X., Kennedy, P.: Deep learning techniques for medical image segmentation: achievements and challenges. *Journal of digital imaging* **32**(4), 582–596 (2019)
 13. Hu, J., Lu, J., Tan, Y.P.: Discriminative deep metric learning for face verification in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1875–1882 (2014)
 14. Kavur, A.E., Gezer, N.S., Barış, M., Aslan, S., Conze, P.H., Groza, V., Pham, D.D., Chatterjee, S., Ernst, P., Özkan, S., Baydar, B., Lachinov, D., Han, S., Pauli, J., Isensee, F., Perkonigg, M., Sathish, R., Rajan, R., Sheet, D., Dovletov, G., Speck, O., Nürnberg, A., Maier-Hein, K.H., Bozdağı Akar, G., Ünal, G., Dicle, O., Selver, M.A.: CHAOS Challenge - combined (CT-MR) healthy abdominal organ segmentation. *Medical Image Analysis* **69**, 101950 (Apr 2021). <https://doi.org/https://doi.org/10.1016/j.media.2020.101950>, <http://www.sciencedirect.com/science/article/pii/S1361841520303145>
 15. Kleesiek, J., Urban, G., Hubert, A., Schwarz, D., Maier-Hein, K., Bendszus, M., Biller, A.: Deep mri brain extraction: A 3d convolutional neural network for skull stripping. *NeuroImage* **129**, 460–469 (2016)
 16. Liu, Y., Gong, X., Chen, J., Chen, S., Yang, Y.: Rotation-invariant siamese network for low-altitude remote-sensing image registration. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **13**, 5746–5758 (2020)
 17. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015)
 18. Milletari, F., Ahmadi, S.A., Kroll, C., Plate, A., Rozanski, V., Maiostre, J., Levin, J., Dietrich, O., Ertl-Wagner, B., Bötzel, K., et al.: Hough-cnn: deep learning for segmentation of deep brain regions in mri and ultrasound. *Computer Vision and Image Understanding* **164**, 92–102 (2017)
 19. Minaee, S., Boykov, Y.Y., Porikli, F., Plaza, A.J., Kehtarnavaz, N., Terzopoulos, D.: Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* pp. 1–1 (2021). <https://doi.org/10.1109/TPAMI.2021.3059968>
 20. Perez, L., Wang, J.: The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621* (2017)
 21. Pham, D.L., Xu, C., Prince, J.L.: Current methods in medical image segmentation. *Annual Review of Biomedical Engineering* **2**(1), 315–337 (2000). <https://doi.org/10.1146/annurev.bioeng.2.1.315>, <https://doi.org/10.1146/annurev.bioeng.2.1.315>, pMID: 11701515

22. Radau, P., Lu, Y., Connelly, K., Paul, G., Dick, A., Wright, G.: Evaluation framework for algorithms segmenting short axis cardiac mri. The MIDAS Journal-Cardiac MR Left Ventricle Segmentation Challenge (07 2009). <https://doi.org/10.54294/g80ruo>
23. Ramyaa, M., Jonathan, M., Kurita, T.: Supervised learning for convolutional neural network with barlow twins. In: ICANN2022 (submitted) (2022)
24. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. pp. 234–241. Springer International Publishing, Cham (2015)
25. Shie, C.K., Chuang, C.H., Chou, C.N., Wu, M.H., Chang, E.Y.: Transfer representation learning for medical image analysis. In: 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). pp. 711–714 (2015). <https://doi.org/10.1109/EMBC.2015.7318461>
26. Tajbakhsh, N., Jeyaseelan, L., Li, Q., Chiang, J.N., Wu, Z., Ding, X.: Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation. *Medical Image Analysis* **63**, 101693 (2020). <https://doi.org/https://doi.org/10.1016/j.media.2020.101693>, <https://www.sciencedirect.com/science/article/pii/S136184152030058X>
27. Ueda, M., Kanda, K., Miyao, J., Miyamoto, S., Nakano, Y., Kurita, T.: Invariant feature extraction for cnn classifier by using gradient reversal layer. In: 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC). pp. 851–856. IEEE (2021)
28. Yudistira, N., Kavitha, M., Itabashi, T., Iwane, A.H., Kurita, T.: Prediction of sequential organelles localization under imbalance using a balanced deep u-net. *Scientific reports* **10**(1), 1–11 (2020)
29. Zbontar, J., Jing, L., Misra, I., LeCun, Y., Deny, S.: Barlow twins: Self-supervised learning via redundancy reduction. In: International Conference on Machine Learning. pp. 12310–12320. PMLR (2021)
30. Zhou, T., Ruan, S., Canu, S.: A review: Deep learning for medical image segmentation using multi-modality fusion. *Array* **3**, 100004 (2019)
31. Zhou, Y., Xie, L., Shen, W., Wang, Y., Fishman, E.K., Yuille, A.L.: A fixed-point model for pancreas segmentation in abdominal ct scans. In: International conference on medical image computing and computer-assisted intervention. pp. 693–701. Springer (2017)

Bidirectional Domain Mixup for Domain Adaptive Semantic Segmentation*

Minseok Seo², Yuhyun Kim¹ and Dong-Geol Choi¹

¹ Hanbat National University, Daejeon, South Korea

yuhyun.dev@gmail.com, dgchoi@hanbat.ac.kr

² SI Analytics, Daejeon, South Korea

minseok.seo@si-analytics.ai

Abstract. Mixup provides interpolated training samples and allows the model to obtain smoother decision boundaries for better generalization. The idea can be naturally applied to the domain adaptation task, where we can mix the source and target samples to obtain domain-mixed samples for better adaptation. However, the extension of the idea from classification to segmentation (i.e., structured output) is nontrivial. In this paper, we propose a new data mixing method, bidirectional domain mixup (BDM). In specific, we achieve domain mixup in two-step: cut and paste. Given the warm-up model trained from any adaptation techniques, we forward the source and target samples and perform a simple threshold-based cutout of the unconfident regions (**cut**). After then, we fill-in the dropped regions with the other domain region patches (**paste**). We coupled our proposal with various state-of-the-art adaptation models and observe significant improvement consistently.

Keywords: Semantic Segmentation · Unsupervised Learning · Domain Adaptation

1 Introduction

To reduce the annotation budget in semantic segmentation that require pixel level annotation, there have been many domain adaptation (DA) approaches using relatively inexpensive source (*e.g.* simulator-based) data [13, 14] and unlabeled target (*e.g.* real) data. However, deep neural networks show poor generalization performance in real data because they are sensitive to domain misalignments such as layout [9], texture [20], structure [16], and class distribution [27]. To deal with it, many approaches have been proposed, including adversarial training [16], entropy minimization [17], and self-training [27, 12, 23].

Among them, cross-domain data mixing based approaches [2, 25, 15, 6] recently show state-of-the-art performances. Early works [2] are largely inspired by a popular data augmentation method, CutMix [22], and borrow some rectangular patches from one domain to fill-in the random hole of other domain of image.

* This paper is the short version of AAAI'23 and is NEVER considered an official publication.

Many variants improve data mixing strategy with mixing the region of randomly sampled classes [15], heuristics on relationship between classes [25], and image-level soft mixup [6].

Motivated by the progress, we further delve into domain mixing approaches and propose **Bidirectional Domain Mixup (BDM)** framework. Beyond the previous unidirectional sample mixing [2, 25, 15, 6], the framework mix the samples in both direction, mix the source patches on the target sample (*i.e.* source-to-target) and vice versa (*i.e.* target-to-source). Specifically, we mainly adopt two core steps of data mixing approach: 1) **Cut**: how can we identify uninformative patches and 2) **Paste**: which patches from other domains bring better supervision signals.

First, we promote to learn domain transferable and generalized features by cutting out the source-specific and nosily predicted region for source and target data, respectively. To supplement scarce supervisory signal due to the cut process, we design the paste step to fulfill the three key functionalities: 1) As semantic segmentation network heavily rely on the context [3, 1, 24, 21], it is important to **maintain intrinsic spatial structure** of images. Thus we leverage spatial continuity to pick a patch that will be pasted on given the hole region. 2) Previous data mixing approaches [25, 15] usually paste the randomly selected classes with its correlated classes. This design choice exacerbates the class imbalanced problem [7], resulting in low performance in sample-scarce class. Instead, we induce **class-balanced learning** by giving the high probability to paste the patches with rare classes. 3) Lastly, proposed mixing method **prevent the noisy learning** by avoiding to paste low-confident patches.

We combine these findings to achieve a new state-of-the-art in the standard benchmarks of domain adaptive semantic segmentation, GTA5 → Cityscapes and SYNTHIA → Cityscapes setting. In addition, BMD consistently provides significant performance improvements when build upon the various representative UDA approaches, including adversarial training [16], entropy minimization [17], and self-training [12, 23].

2 Method

To bridge source and target domains that come from different distributions, we simulate intermediate domains by generating domain mixed samples. In the next section, we first describe the overall pipeline to train a network with domain-mixed samples. Next, we introduce how these samples are generated in detail.

2.1 The Bidirectional Domain Mixup Framework

Given a labeled source dataset $\mathcal{D}^S = \{(x_i^S, y_i^S)\}_{i=1}^{N_S}$, and a unlabeled target dataset $\mathcal{D}^T = \{x_i^T\}_{i=1}^{N_T}$, our goal is to transfer the knowledge learned from source domain to unlabeled target domain. To do so, we utilize domain-mixed samples and propose a new data mixing method, bidirectional domain mixup (BDM), to generate them. As illustrated in Fig. 1, this framework comprises the following four major components.

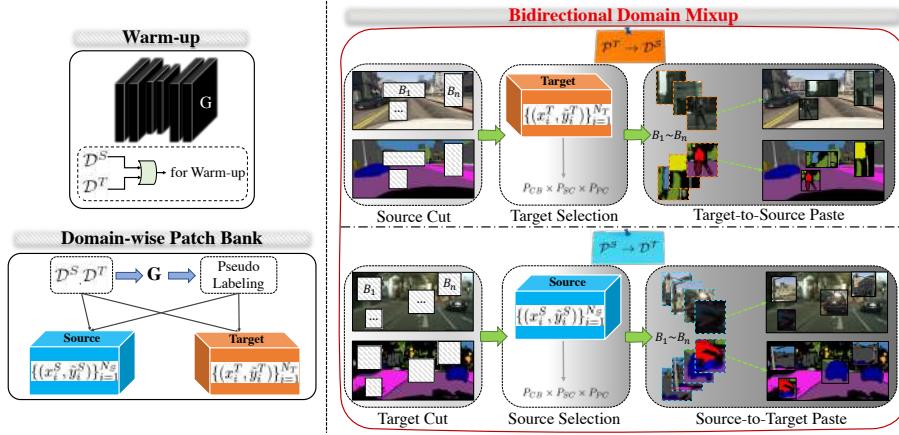


Fig. 1. Overview of BDM framework. Before training a network with the proposed BDM framework, we first warm up the model with any previous UDA method. Given the model $G(\cdot)$, we generate pseudo labels for the source and target domain and store the images and corresponding pseudo labels in the domain-wise patch bank. The mixup process is conducted in a bidirectional way, target-to-source $P_T \rightarrow P_S$ and source-to-target $P_S \rightarrow P_T$. These samples are guided models to learn domain generalized features.

- We first apply a previous domain adaptation method as a *warm-up*. Our framework allows any previous domain adaptation method. To show the generality, we choose the representative methods including adversarial training [16], entropy minimization [17], and self-training [12, 23].
- *Domain-wise patch banks*, B^S and B^T , are constructed to store the samples of each domains. We divide images and corresponding pseudo labels into non-overlapping patches and the resulting pairs are stored in domain-wise patch banks. For the both domain, we adopt simple strategy [20] to generate pseudo labels, $\{\tilde{y}_i^S\}_{i=1}^{N_S}$ and $\{\tilde{y}_i^T\}_{i=1}^{N_T}$.
- During the training, a minibatch of source and target images, x^S and x^T , are sampled. *Bidirectional domain mixup(BDM)* generate domain-mixed samples, x_{mix}^S and x_{mix}^T , via mixture of images of one domain with patches from other domain. Thus, this cross-domain mixing is in bidirectional way, source-to-target and target-to-source. Specifically, some rectangular regions(*i.e.* patches) in target samples are cut and source patches retrieved from the source patch bank are pasted (*i.e.* source-to-target direction), and vice versa. We also apply same operation on labels, resulting in domain-mixed labels, y_{mix}^S and y_{mix}^T .
- Given the mixed images and labels, a segmentation network is trained with standard cross-entropy losses L_{cross} . The final loss is formulated as follows: $L_{final} = L_{cross}(x_{mix}^S, y_{mix}^S) + L_{cross}(x_{mix}^T, y_{mix}^T)$.

2.2 Cut

Cut is a process that masks out contiguous sections (*i.e.* multiple rectangular regions) of input and corresponding labels. We introduce widely used random patch cutout [5, 22, 15] and proposed confidence based cutout.

Confidence based cutout. Random region cutout is a simple but strong baseline that is adopted in various tasks such as UDA [15], Semi-DA [2] and SSL [19]. However, the random region cut the regions regardless of whether it is informative. Instead, we target to cut where provide noisy supervision in terms of learning the generalized features. To this end, we see that it is important to discard the regions with low confident predictions for the following reasons: 1) for the source domain, it remove non-transferable and source specific regions, 2) it prevent to learn from noisy pseudo labels of target data.

Given the source images x_i^S and their pseudo labels \tilde{y}_i^S , we calculates the ratio of the uncertain region over the randomly generated region $\mathbf{B} = (r_x, r_y, r_w, r_h)$. If the ratio of the uncertain region is above the cutout threshold γ , the region is cut. The proposed cutout is summarized as follows:

$$\hat{x}_i^S, \hat{y}_i^S = \begin{cases} \text{Cutout}(x_i^S, \tilde{y}_i^S), & \text{if } \mathcal{H}(\tilde{y}_i^S, \mathbf{B}) > \gamma, \\ x_i^S, \tilde{y}_i^S & \text{otherwise,} \end{cases} \quad (1)$$

where \mathcal{H} denote a function that computes the ratio of the uncertain region over region \mathbf{B} . And, the threshold γ is set to 0.2.

2.3 Paste

To generate the domain-mixed samples, we sample the patch from the other domain and paste it back to the region that are cutout. This enables the resulting samples to include both source and target patches.

We additionally consider three important factors during the patch sampling. First, we introduce **class-balanced** sampling. As random sampling tend to bias the mixup samples mainly toward the frequent classes, we offset this undesirable effect using class-balanced sampling. Since the long-tailed category distribution tends to be similar for different domains, we compute the class distribution of the source domain [23, 10] to provide more chance for patches that include rare classes. Second, we consider **spatial continuity** of the natural scene. The random sampling can produce mixed samples that are geometrically unnatural, and thus causes severe train and test time inconsistency. Instead, based on the fact that the vehicle egocentric view has strong fixed spatial priors [3, 26], we compute spatial priors for each semantic class [26] in the source domain and use this statistics during sampling. Finally, we take **pseudo label confidence** into account.

Patch Generation. In the source and target datasets, classes such as *train* and *bike* have a small number of samples, so to consider the class online, it is

necessary to find a sample including a 6 rare class from all samples. Since this method is memory inefficient, we cut the patch containing each class and save it.

We generate a patch by cutting the pseudo labeled datasets $\{(x_i^S, \tilde{y}_i^S)\}_{i=1}^{N_S}$ and $\{(x_i^T, \tilde{y}_i^T)\}_{i=1}^{N_T}$ at equal intervals by the number of horizontal \mathbf{W} and vertical \mathbf{H} . Therefore, one image and its pseudo label are divided into $\mathbf{W} \times \mathbf{H}$ patches. After that, the patches extracted from each location are grouped into one patch sequence. Next, the patch containing the class $\{C_i\}_{i=1}^K$, \mathbf{K} is the number of classes, from the pseudo label of the patch sequence representing each spatial location is stored in the child patch sequence, and it is possible to duplicate it. Therefore, patches considering spatial location and class existence are grouped into a total of $\mathbf{W} \times \mathbf{H} \times \mathbf{K}$ patch sequence. Finally, at the patch sequence of $\mathbf{W} \times \mathbf{H} \times \mathbf{K}$, the normalized confidence calculated by the method in Eq. 4 is sorted in ascending order, and \mathbf{R} patch sequences are generated at regular intervals. The number of finally generated patch sequence is $\mathbf{W} \times \mathbf{H} \times \mathbf{K} \times \mathbf{R}$

Patch Selection

Class-balanced Patch Sampling. In addition to Zipfian distribution of object categories [11], the difference in the intrinsic size of each object makes pixel-level class imbalances more severe in semantic segmentation. However, the random patch sampling for paste make bias toward the frequent classes, as it naturally follow the probability of existence.

To alleviate the class imbalance problem in paste process, we propose patch-level oversampling. Given the total number of pixels for each classes in the source labels $\{\bar{N}^i\}_{i=1}^K$, the probability P_{CB} that each class patch is selected can be formulated as follows:

$$\begin{aligned}\hat{P}_{CB} &= \left\{ \left(-\log\left(\frac{\bar{N}^i}{\sum_{i=0}^{\mathbf{K}} \bar{N}^i}\right) \right)^{\alpha} \right\}_{i=1}^K, \\ P_{CB} &= \left\{ \left(\frac{\hat{P}_{CB}^i}{\sum_{i=0}^{\mathbf{K}} \hat{P}_{CB}^i} \right) \right\}_{i=1}^K,\end{aligned}\tag{2}$$

where \mathbf{K} is the number of classes, α is the sharpening coefficient. We set α to 2 in all experiments.

Sampling with Spatial Continuity. The spatial layout between the source (synthetic) and target dataset share large similarities. Therefore, we propose spatial continuity based paste that considers spatial relationship instead of pasting patches at random positions. The probability of selecting each location $\{\text{Patches}_i\}_{i=1}^{\mathbf{W} \times \mathbf{H}}$ of the patches generated through Patch Generation section to be mixed with the patch locations cutout through Cut section is calculated as follows:

$$\begin{aligned}\hat{SC} &= \operatorname{argmax}\{\text{SC}_i(o_w, o_h)_{i=1}^K\}, \\ P_{SC} &= \{\hat{SC}(\text{Patches}_i(o_{\hat{w}^i}, o_{\hat{h}^i}))\}_{i=1}^{\mathbf{W} \times \mathbf{H}},\end{aligned}\tag{3}$$

where $\{\text{SC}_i\}_{i=1}^K$ is the source domain class-wise spatial prior kernel map generated by CBST-SP [26]. where o_h, o_w is the center coordinates of the cutout patch, o_h^i, o_w^i is the center coordinates of the patch at the i-th location. Note that, we normalized the sum of the set P_{SC} to 1.

Sampling with Normalized Confidence. Opposite to confidence based cutout, in the past, we give high probability to the patches that include confident pixels. To faithfully measure the confidence level of a patch, we take the difficulty of each class into account and design the normalized confidence of a patch. We first calculate the average confidence of classes using the set of pseudo labels and use it to represent the difficulty of each class. Then, the confidence of patches in the patch bank is normalized at pixel-level by subtracting the difficulty score according to predicted classes (Norm). Intuitively, it measures the *relative* confidence level. The resulting normalized confidence maps are spatially averaged (Average), and patches are sorted in ascending order according to it (Sort).

$$\begin{aligned}\hat{B}^T &= \{\text{Average}(\text{Norm}(B_i^T))\}_{i=1}^{N_T}, \\ \hat{B}^S &= \{\text{Average}(\text{Norm}(B_i^S))\}_{i=1}^{N_S}, \\ \bar{B}^T &= \text{Sort}(\hat{B}^T), \\ \bar{B}^S &= \text{Sort}(\hat{B}^S)\end{aligned}\tag{4}$$

Finally, the patch is divided into three batches: the low, the middle, and the high confidence group. The probability P_{PC} that the patch in each group is selected is $\{0.1, 0.3, 0.6\}$.

Probability of selection of each patch sequence. We select the patch jointly considering class balance, spatial continuity, and pseudo label confidence for BDM. Since each probability is independent, the probability that each patch sequence is selected is $P_{CB} \times P_{SC} \times P_{PC}$.

3 Experiments

In this section, we present experimental results to validate the proposed BDM for domain adaptive semantic segmentation.

We first describe experimental configurations in detail. After that, we validate our BDM on two public benchmark datasets. Note that the Intersection-over-Union (IoU) metric is used for all the experiments.

3.1 Experimental Settings

Dataset. We evaluate our proposed Bidirectional Domain Mixup on two popular domain adaptive semantic segmentation benchmarks(SYNTHIA → Cityscapes, and GTA5 → Cityscapes). Cityscapes [4] is a real-world urban scene dataset consisting of a training set with 2,975 images, a validation set with 500 images

Table 1. Comparison with state-of-the-art models on GTA5 → Cityscapes. We highlight the mIoU of tail classes (*i.e.* mIoU-tail) along with per-class IoU and overall mIoU. Our results are averaged over five runs.

Method	mIoU	mIoU-tail	Head Classes												Tail Classes						
			road	sidewalk	building	wall	fence	pole	vegetation	terrain	sky	person	car	truck	bus	light	sign	rider	train	motorcycle	bike
Source Only	36.6	24.0	75.8	16.8	77.2	12.5	21.0	25.5	81.3	24.6	70.3	53.8	49.9	17.2	25.9	30.1	20.1	26.4	6.5	25.3	36.0
Adaptseg [16]	41.4	25.0	86.5	25.9	79.8	22.1	20.0	23.6	81.8	25.9	75.9	57.3	76.3	29.8	32.1	33.1	21.8	26.2	7.2	29.5	32.5
ADVENT [17]	45.5	25.7	89.4	33.1	81.0	26.6	26.8	27.2	83.9	36.7	78.8	58.7	84.8	38.5	44.5	33.5	24.7	30.5	1.7	31.6	32.4
CCM [9]	49.9	26.9	93.5	57.6	84.6	39.3	24.1	25.2	85.0	40.6	86.5	58.7	85.8	49.6	56.4	35.0	17.3	28.7	5.4	31.9	43.2
IAST [12]	51.5	34.0	93.8	57.8	85.1	39.5	26.7	26.2	84.9	32.9	88.0	62.6	87.3	39.2	49.6	43.1	34.7	29.0	23.2	34.7	39.6
DACS [15]	52.1	32.6	89.9	39.6	87.8	30.7	39.5	38.5	87.9	43.9	88.7	67.2	84.4	45.7	50.1	46.4	52.7	35.7	0.0	27.2	33.9
DSP [6]	55.0	36.9	92.4	48.0	87.4	33.4	35.1	36.4	87.7	43.2	89.8	66.6	89.9	57.3	56.1	41.6	46.0	32.1	0.0	44.1	57.8
CAMix [25]	55.2	37.9	93.3	58.2	86.5	36.8	31.5	36.4	87.2	44.6	88.1	65.0	89.7	46.9	56.8	35.0	43.5	24.7	27.5	41.1	56.0
CorDA [18]	56.6	38.5	94.7	63.1	87.6	30.7	40.6	40.2	87.6	47.0	89.7	66.7	90.2	48.9	57.5	47.8	51.6	35.9	0.0	39.7	56.0
ProDA [23]	57.5	42.0	87.8	56.0	79.7	46.3	44.8	45.6	88.6	45.2	82.1	70.7	88.8	45.5	59.4	53.5	53.5	39.2	1.0	48.9	56.4
DAP [8]	59.8	44.6	94.5	63.1	89.1	29.8	47.5	50.4	89.5	50.2	87.0	73.6	91.3	50.2	52.9	56.7	58.7	38.6	0.0	50.2	63.5
Adaptseg [16] + Ours	57.4 (+16.0)	44.4	89.3	50.0	88.4	45.6	45.4	41.1	78.0	35.6	82.3	69.2	87.5	55.7	57.8	49.9	60.2	45.6	8.1	45.5	57.2
ADVENT [17] + Ours	57.6 (+12.1)	38.9	91.3	51.8	86.7	49.9	49.2	53.3	85.8	47.9	85.7	62.3	87.8	55.5	54.4	43.1	43.3	45.9	4.4	46.3	50.4
IAST [12] + Ours	61.0 (+9.5)	46.5	92.1	59.6	89.9	52.9	55.7	49.2	89.3	46.7	86.3	59.1	88.3	54.8	55.9	44.6	45.8	42.0	39.2	50.3	57.3
ProDA [23] + Ours	63.9 (+6.4)	47.8	89.2	60.1	83.8	61.5	63.6	66.7	90.4	51.1	83.5	72.6	88.0	51.2	65.3	58.2	59.3	47.8	1.0	60.1	60.9
Target Only	64.5	53.0	96.2	75.5	87.7	38.0	39.6	43.4	88.2	52.4	89.5	69.7	91.4	66.2	69.7	46.6	62.8	49.5	45.0	49.0	65.1

and a testing set with 1,525 images. We use the unlabeled training dataset as $\{D_i^T\}_{i=1}^{2,975}$ and evaluate our Bidirectional Domain Mixup with 500 images from the validation set. SYNTHIA [14] is a synthetic urban scene dataset. We pick SYNTHIA-RAND-CITYSCAPES subset as the source domain, which shares 16 semantic classes with Cityscapes. In total, 9,400 images from SYNTHIA dataset are used as source domain training data $\{D_i^S\}_{i=1}^{9,400}$ for the task. GTA5 [13] dataset is another synthetic dataset sharing 19 semantic classes with Cityscapes. 24,966 urban scene images are collected from a physically-based rendered video game Grand Theft Auto V (GTAV) and are used as source training data $\{D_i^S\}_{i=1}^{24,966}$. We view 6 and 5 with relatively few training samples as tail-classes for each source domain(GTA5, SYNTHIA), respectively.

3.2 Comparison with State-of-the art

In this section, we compare our proposed method with the top-performing UDA approach.

Table 1 shows the comparisons on GTA5 → Cityscapes setting. DACS, which classmix at random locations without considering the class distribution of the source dataset, significantly improved performance in classes with a large number of samples, but significantly decreased in tail classes such as *bike* and *train*. CAMix, a classmix method considering the contextual relationship, solved the problem of performance degradation of tail classes through consistency loss and dynamic threshold. However, considering only the contextual relationship, the performance decreased in *wall*, *light*, and *rider* classes, which have significantly different contextual relationship between GTA5 and Cityscapes.

On the other hand, our BDM jointly consider class balance, spatial continuity, and pseudo label confidence and achieve state-of-the-art with an mIoU score of

Table 2. Comparison with state-of-the-art models on SYNTHIA → Cityscapes. Our results are averaged over five runs.

Method	mIoU	mIoU-tail	Head Classes									Tail Classes				
			road	sidewalk	building	vegetation	sky	person	car	bus	light	sign	rider	motorcycle	bike	
Source Only	40.3	20.8	64.3	21.3	73.1	63.1	67.6	42.2	73.1	15.3	7.0	27.7	19.9	10.5	38.9	
Adaptseg [16]	45.8	18.5	79.5	37.1	78.2	78.0	80.3	53.7	67.1	29.4	9.3	10.6	19.2	21.8	31.6	
ADVENT [9]	48.0	16.9	85.6	42.2	79.7	80.4	84.1	57.9	73.3	36.4	5.4	8.1	23.8	14.2	33.0	
CCM [9]	52.9	29.0	79.6	36.4	80.6	81.8	77.4	56.8	80.7	45.2	22.4	14.9	25.9	29.9	52.0	
IAST [12]	57.0	35.2	81.9	41.5	83.3	83.4	85.0	65.5	86.5	38.2	30.9	28.8	30.8	33.1	52.7	
DACS [15]	54.8	32.1	80.5	25.1	81.9	83.6	90.7	67.6	82.9	38.9	22.6	23.9	38.3	28.4	47.5	
DSP [6]	59.9	35.9	86.4	42.0	82.0	87.2	88.5	64.1	83.8	65.4	31.6	33.2	31.9	28.8	54.0	
CAMix [25]	59.7	33.8	91.8	54.9	83.6	83.8	87.1	65.0	85.5	55.1	23.0	29.0	26.4	36.8	54.1	
CorDA [18]	62.8	41.5	93.3	61.6	85.3	84.9	90.4	69.7	85.6	38.4	36.6	42.8	41.8	32.6	53.9	
ProDA [23]	62.0	40.3	87.8	45.7	84.6	88.1	84.4	74.2	88.2	51.1	54.6	37.0	24.3	40.5	45.6	
DAP [8]	64.3	48.7	84.2	46.5	82.5	89.3	87.5	75.7	91.7	73.5	53.6	45.7	34.6	49.4	60.5	
Adaptseg [*] [16] + Ours	50.6(+4.8)	24.1	84.5	44.3	79.5	84.2	83.3	60.1	69.3	33.2	19.6	18.7	23.6	24.0	34.7	
ADVENT [16] + Ours	53.8(+5.8)	27.1	85.4	47.7	82.9	86.5	85.7	64.5	72.0	39.3	25.2	22.2	25.6	26.1	36.4	
IAST [12] + Ours	62.9(+5.9)	42.6	88.2	50.2	88.5	85.6	89.7	70.2	88.3	44.8	42.3	33.5	39.0	39.4	59.0	
ProDA [23] + Ours	66.8(+4.8)	47.7	91.0	55.8	86.9	85.8	85.7	84.1	86.0	55.2	58.3	44.7	40.3	45.0	50.6	
Target Only	72.3	54.6	96.2	75.5	87.7	88.2	89.5	69.7	91.4	69.7	46.6	62.8	49.5	49.0	65.1	

* Note that pre-trained weights are not provided, so use them after reproduction.

63.9% when ProDA was selected as the warm-up model. Despite the great overall scores, it showed a low IoU in the *train* class. The rationale behind this is the severely poor performance of the chosen warm-up model in that class. Instead, when the warm-up model is switched to IAST, we achieved much improved scores (23.2% → 39.2%) IoU in the *train* class.

Last but not least, our method shows consistent performance improvement with four different warm-up models, showing the generality of our framework.

Table 2 shows the comparisons of SYNTHIA → Cityscapes adaptation. Again, our BDM achieved state-of-the-art with an mIoU score of 66.8% when ProDA was selected as the warm-up model. These experimental results indicate that BDM is valid not only in the GTA5 dataset, where the scene layout between the source and target dataset is highly similar but also in the SYNTHIA dataset which includes images with different viewpoints (e.g. top-down view).

4 Conclusions

In this paper, we proposed Bidirectional Domain Mixup (BDM), a cutmix method that cut the low confidence region and selects a patch to paste according to class-balance(CB), spatial continuity(SC), and pseudo label confidence(PC) in the corresponding region. Our proposed BDM achieves state-of-the-art in GTA5 to cityscapes benchmark and SYNTHIA to cityscapes benchmark with a large gap.

References

1. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence* **40**(4), 834–848 (2017)
2. Chen, S., Jia, X., He, J., Shi, Y., Liu, J.: Semi-supervised domain adaptation based on dual-level domain mixing for semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11018–11027 (2021)
3. Choi, S., Kim, J.T., Choo, J.: Cars can't fly up in the sky: Improving urban-scene segmentation via height-driven attention networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9373–9383 (2020)
4. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3213–3223 (2016)
5. DeVries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552 (2017)
6. Gao, L., Zhang, J., Zhang, L., Tao, D.: Dsp: Dual soft-paste for unsupervised domain adaptive semantic segmentation. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 2825–2833 (2021)
7. Gupta, A., Dollar, P., Girshick, R.: Lvvis: A dataset for large vocabulary instance segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5356–5364 (2019)
8. Huo, X., Xie, L., Hu, H., Zhou, W., Li, H., Tian, Q.: Domain-agnostic prior for transfer semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7075–7085 (2022)
9. Li, G., Kang, G., Liu, W., Wei, Y., Yang, Y.: Content-consistent matching for domain adaptive semantic segmentation. In: European conference on computer vision. pp. 440–456. Springer (2020)
10. Li, R., Li, S., He, C., Zhang, Y., Jia, X., Zhang, L.: Class-balanced pixel-level self-labeling for domain adaptive semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11593–11603 (2022)
11. Manning, C., Schütze, H.: Foundations of statistical natural language processing. MIT press (1999)
12. Mei, K., Zhu, C., Zou, J., Zhang, S.: Instance adaptive self-training for unsupervised domain adaptation. In: European conference on computer vision. pp. 415–430. Springer (2020)
13. Richter, S.R., Vineet, V., Roth, S., Koltun, V.: Playing for data: Ground truth from computer games. In: European conference on computer vision. pp. 102–118. Springer (2016)
14. Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M.: The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3234–3243 (2016)
15. Tranheden, W., Olsson, V., Pinto, J., Svensson, L.: Dacs: Domain adaptation via cross-domain mixed sampling. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1379–1389 (2021)

16. Tsai, Y.H., Hung, W.C., Schulter, S., Sohn, K., Yang, M.H., Chandraker, M.: Learning to adapt structured output space for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7472–7481 (2018)
17. Vu, T.H., Jain, H., Bucher, M., Cord, M., Pérez, P.: Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2517–2526 (2019)
18. Wang, Q., Dai, D., Hoyer, L., Van Gool, L., Fink, O.: Domain adaptive semantic segmentation with self-supervised depth estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2021)
19. Wang, Y., Wang, H., Shen, Y., Fei, J., Li, W., Jin, G., Wu, L., Zhao, R., Le, X.: Semi-supervised semantic segmentation using unreliable pseudo-labels. arXiv preprint arXiv:2203.03884 (2022)
20. Yang, Y., Soatto, S.: Fda: Fourier domain adaptation for semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4085–4095 (2020)
21. Yi, J.S.K., Seo, M., Park, J., Choi, D.G.: Using self-supervised pretext tasks for active learning. In: Proc. ECCV (2022)
22. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6023–6032 (2019)
23. Zhang, P., Zhang, B., Zhang, T., Chen, D., Wang, Y., Wen, F.: Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12414–12424 (2021)
24. Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H., et al.: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6881–6890 (2021)
25. Zhou, Q., Feng, Z., Gu, Q., Pang, J., Cheng, G., Lu, X., Shi, J., Ma, L.: Context-aware mixup for domain adaptive semantic segmentation. IEEE Transactions on Circuits and Systems for Video Technology (2022)
26. Zou, Y., Yu, Z., Kumar, B., Wang, J.: Domain adaptation for semantic segmentation via class-balanced self-training. arXiv preprint arXiv:1810.07911 (2018)
27. Zou, Y., Yu, Z., Kumar, B.V., Wang, J.: Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: Proceedings of the European Conference on Computer Vision (ECCV) (September 2018)

LabOR: Labeling Only if Required for Domain Adaptive Semantic Segmentation*

Inkyu Shin Dong-Jin Kim Jae Won Cho Sanghyun Woo Kwanyong Park In
So Kweon

KAIST, Daejeon, South Korea and Hanyang University, Seoul, South Korea
clsrbbg33@kaist.ac.kr,
djkim@hanyang.ac.kr,{chojw,shwoo93,pkyong7,iskweon77}@kaist.ac.kr

Abstract. Unsupervised Domain Adaptation (UDA) for semantic segmentation has been actively studied to mitigate the domain gap between label-rich source data and unlabeled target data. Despite these efforts, UDA still has a long way to go to reach the fully supervised performance. To this end, we propose a **Labeling Only if Required** strategy, **LabOR**, where we introduce a human-in-the-loop approach to adaptively give scarce labels to points that a UDA model is uncertain about. In order to find the uncertain points, we generate an inconsistency mask using the proposed adaptive pixel selector and we label these segment-based regions to achieve near supervised performance with only a small fraction (about 2.2%) ground truth points, which we call “Segment based Pixel-Labeling (SPL).” To further reduce the efforts of the human annotator, we also propose “Point based Pixel-Labeling (PPL),” which finds the most representative points for labeling within the generated inconsistency mask. This reduces efforts from 2.2% segment label → 40 points label while minimizing performance degradation. Through extensive experimentation, we show the advantages of this new framework for domain adaptive semantic segmentation while minimizing human labor costs.

Keywords: Active Domain Adaptation · Semantic Segmentation · Human-in-the-loop

1 Introduction

Semantic segmentation enables understanding of image scenes at the pixel level, and is critical for various real-world applications such as autonomous driving [18] or simulated learning for robots [6]. Unfortunately, the pixel level understanding task in deep learning requires tremendous labeling efforts in both time and cost. Therefore, unsupervised domain adaptation (UDA) [7] addresses this problem by utilizing and transferring the knowledge of label-rich data (source data) to unlabeled data (target data), which can reduce the labeling cost dramatically [17]. According to the adaptation methodology, UDA can be largely divided into **Adversarial learning based** [12,15] DA and **Self-training based** [14] DA. While the former focuses on minimizing task-specific loss for source domain and domain adversarial loss, the self-training strategy retrains the model with generated target-specific pseudo labels. Among them, IAST [14] achieves state-of-

* This paper is the short version of ICCV’21

I. Shin et al.

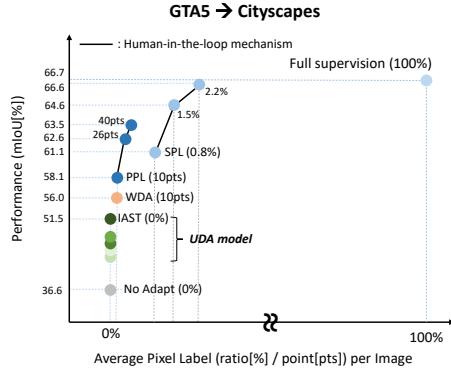


Fig. 1. Average Pixel Label per image vs. Performance. Our novel human-in-the-loop framework, **LabOR** (PPL and SPL) significantly outperforms not only previous UDA state-of-the-art models (e.g., IAST [14]) but also DA model with few labels (e.g., WDA [16]). Note that our PPL requires negligible number of label to achieve such performance improvements (25 labeled points per image), and our SPL shows the performance comparable with fully supervised learning (0.1% mIoU gap). Detailed performance can be found in Table. 1.

the-art performance in UDA by effectively mixing adversarial based and self-training based strategies.

Despite the relentless efforts in developing UDA models, the performance limitations are clear as it still lags far behind the fully supervision model. As visualized in Fig. 1, the recent UDA methods remain at around ($\sim 50\%$ mIoU) which is far below the performance of full supervision ($\sim 65\%$ mIoU) on GTA5 [18] \rightarrow Cityscapes [5].

Motivated by the limitation of UDA, we present a new perspective of domain adaptation by utilizing a minute portion of pixel-level labels in an adaptive human-in-the-loop manner. We name this framework **Labling Only if Required** (LabOR), which is described in Fig. 2. Unlike conventional self-training based UDA that retrains the target network with the pseudo labels generated from the model predictions, we utilize the model predictions to find uncertain regions that require human annotations and train these regions with ground truth labels in a supervised manner. In particular, we find regions where the two different classifiers mismatch in predictions. In order to effectively find the mismatched regions, we introduce additional optimization step to maximize the discrepancy between the two classifiers like [4,20]. Therefore, by comparing the respective predictions from the two classifiers on a pixel level, we create a mismatched area that we call the *inconsistency mask* which can be regarded uncertain pixels. We call this framework the “Adaptive Pixel Selector” which guides a human annotator to label on proposed pixels. This results in the use of a very small number of pixel-level labels to maximize performance. Depending on how we label the proposed areas, we propose two different labeling strategies, namely “Segment based Pixel-Labeling (SPL)” and “Point based Pixel-Labeling (PPL).” While SPL labels every pixels on the inconsistency mask in a segment-like manner, PPL places its focus more on the labeling effort efficiency by finding the representative *points* within a proposed segment. We empirically

LabOR: Labeling Only if Required for Domain Adaptive Semantic Segmentation

show that the two proposed “Pixel-Labeling” options not only help a model achieve near supervised performance but also reduces human labeling costs dramatically.

2 Related Work

Domain Adaptation with Few Labels. Despite extensive studies in UDA, the performance of UDA is known to be much lower than that of supervised learning [19]. In order to mitigate this limitation, various works have tried to leverage ground truth labels for the target dataset. For example, semi-supervised domain adaptation, which utilize randomly selected image-level labels per class as the labeled training target examples, has been recently studied for image classification [19], semantic segmentation [27], and image captioning [3,8]. However, these naive semi-supervised learning approaches do not consider which target images should be labeled given a fixed budget size. Similar to semi-supervised domain adaptation, some works have used active learning [21] to give labels to a small portion of the dataset [24]. These works leverage a model to find data points that would increase the performance of the model the most. Furthermore, in order to reduce the labeling effort per image for target images in domain adaptation, a method to leverage weak labels, several points per image, has also been studied [16].

In contrast, our work differentiates itself by allowing the model to automatically pinpoint to the human annotator which points to label on a pixel-level that would have the best potential performance increase instead of randomly picking labels which can possibly be already easy for the model to predict. In addition, unlike the semi-supervised model which has random annotations prior to training, we allow the model to let the annotator know which points in an image are best to increase performance. Although at first glance our method may seem similar to active learning in the human-in-the-loop aspect, our work is the first to propose a method on the *pixel-level* instead of image-level. Overall, our pixel-level sampling approach is not only efficient, but also orthogonal to the existing active, weak label, or semi-supervised domain adaptation frameworks.

3 Proposed Method

In this section, we introduce our method from inconsistency mask generation to adaptive pixel labeling.

3.1 Problem Definition: Domain Adaptation

Let us denote $g_\phi(\cdot)$ as the network backbone with the parameter ϕ that generates features from an input \mathbf{x} . Then, with the classification layer including softmax activation $f_\theta(\cdot)$ with the parameter θ , a class prediction (probability) is computed ($\hat{\mathbf{Y}} = p(\mathbf{Y}|\mathbf{x}; \theta, \phi) = f_\theta \circ g_\phi(\mathbf{x}) \in \mathbb{R}^{W \times H \times K}$, where W and H are width and height of the segmentation map, and K is the total number of classes). The combined network $f_\theta \circ g_\phi(\cdot)$ can be implemented with typical semantic segmentation generators [1,2]. A typical semantic segmentation model is trained with cross-entropy loss $CE(\cdot, \cdot)$ with the ground truth label $\mathbf{Y} \in \mathbb{R}^{W \times H}$. Furthermore, let us denote $\mathcal{S} = \{(\mathbf{x}_s, \mathbf{Y}_s)\}_{s=1}^S$ as

I. Shin et al.

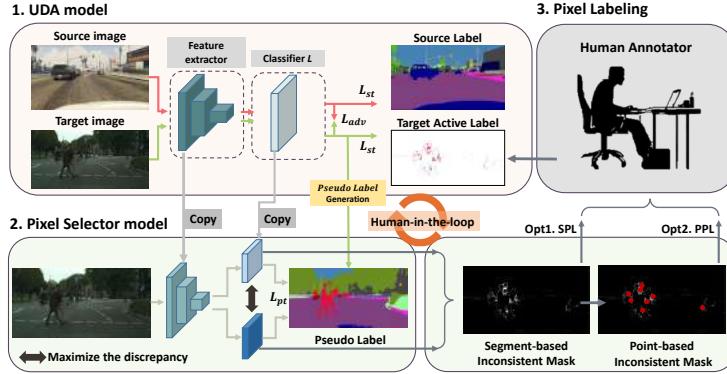


Fig. 2. The overview of the proposed adaptive pixel-basis labeling, LabOR. This framework is made up of two models: UDA model and Pixel selector model. The UDA model initially trained from conventional adversarial learning forwards target image to generate pseudo label. Different from normal self-training training scheme [14] that utilizes the generated label to retrain the model directly, we instead train a pixel selector model to brings out inconsistent mask where human annotator is guided to label. In this process, we use pseudo label training loss, L_{pt} which contains pseudo label cross entropy loss and classifiers' discrepancy loss. With those human labels, we return to the original UDA model for training that uses L_{st} .

the labeled images from the source dataset and $\mathcal{T} = \{\mathbf{x}_t\}_{t=1}^T$ as the unlabeled images from the target dataset. Unsupervised Domain Adaptation (UDA) tries to leverage both the abundant labeled source dataset and the small number of unlabeled target dataset to train a deep neural network.

Recent unsupervised domain adaptive semantic segmentation use self-training methods [14,32] and have shown state-of-the-art performances and are optimized as follows: In practice, the model alternates between generating pseudo-labels $\tilde{\mathbf{Y}}_t(\mathbf{x}_t) \in \mathbb{R}^{W \times H}$ for an image \mathbf{x}_t based on the model prediction $p(\mathbf{Y}|\mathbf{x}; \theta, \phi)$ and retraining the model on the target dataset with the generated pseudo labels. The goal of self-training based domain adaptation [14,32] is to devise an effective loss function and a way to generate pseudo labels. Specifically, CRST [32] propose class-balanced pseudo label generation strategy and confident region KLD minimization to prevent overfitting on pseudo labels. IAST [14] tackles the class-balanced pseudo label generation which ignores the individual attributes of instance to design an instance adaptive selector. Moreover, IAST adds an entropy minimization approach on unlabeled pixels. Self-training based domain adaptation far underperforms a fully supervised model. This can be attributed to two reasons. First, cutting out unconfident pixels and re-training with the thresholded labels is not intuitive as the model forced to be trained with only the pixels that model itself is confident in. Second, existing pseudo label generation commonly originates from specific manually set hyperparameters, causing incorrect pseudo labels which degrades the performance. To address this issue, we propose a new perspective of self-training based domain adaptation with a human-in-the-loop approach by using a human annotator to label a small number of informative *pixels*. As the human annotator annotates the pixels where the model is uncertain, the labeled pixels ultimately act as a guide for the model.

LabOR: Labeling Only if Required for Domain Adaptive Semantic Segmentation

We call this method **Labeling Only if Required (LabOR)**. In order to minimize the efforts of the human annotator, we must answer the key question “*what is an informative pixel to label?*” In other words, our goal is to find the pixels where the model is uncertain. To this end, we propose to select the pixels that show the highest *classifier discrepancy* motivated by the classifier discrepancy based domain adaptation method, MCDDA [20].

3.2 Generating Inconsistency Mask

Fig. 2 illustrates an overview of our proposed method. First, we pre-train a model with the labeled source dataset \mathcal{S} by minimizing supervised cross-entropy loss:

$$\mathcal{L}_s(\theta, \phi) = \mathbb{E}_{(\mathbf{x}_s, \mathbf{y}_s) \in \mathcal{S}} [\text{CE}(\mathbf{Y}_s, p(\mathbf{Y}|\mathbf{x}_s; \theta, \phi))]. \quad (1)$$

Following this, in order to improve the effectiveness of self-training, we utilize warm-up with adversarial training [14] before moving on to self-training.

$$\mathcal{L}_{adv}(\theta, \phi) = \mathbb{E}_{\mathbf{x}_s \in \mathcal{S}, \mathbf{x}_t \in \mathcal{T}} [\text{Adv}(p(\mathbf{x}_s; \theta, \phi), p(\mathbf{x}_t; \theta, \phi))]. \quad (2)$$

Then we copy the parameters of the backbone and the classifier (twice for classifier) (i.e., $\theta'_1 \leftarrow \theta, \theta'_2 \leftarrow \theta, \phi' \leftarrow \phi$) to create our Adaptive Pixel Selector model ($f_{\theta'_1}, f_{\theta'_2}, g_{\phi'}$). This model is only used for the purposes of pixel selection and has no effect on the performance. Using this newly created model, we optimize the model with the two auxiliary classifiers and increase the discrepancy in relation to each other. After this, we propose to find the pixels where the two classifiers have different output class predictions. Using the different output class predictions, we create a mask consisting of pixels that are inconsistent $M(\mathbf{x}_t; \phi', \theta'_1, \theta'_2) \in \mathbb{R}^{W \times H}$, and we call this the *inconsistency mask*. The mask generation would be formulated as follows:

$$M(\mathbf{x}_t) = [\arg \max_K f_{\theta'_1} \circ g_{\phi'}(\mathbf{x}_t) \neq \arg \max_K f_{\theta'_2} \circ g_{\phi'}(\mathbf{x}_t)]. \quad (3)$$

For simplicity, we abuse the notation $M(\mathbf{x}_t; \phi', \theta'_1, \theta'_2)$ as $M(\mathbf{x}_t)$. We conjecture that if the two classifiers trained on the same dataset generate different predictions for the same region, then it means the model prediction shows a high variance in that input region. Therefore we conclude that this *inconsistency mask* represents the pixels the model is the most unsure about. In other words, we hypothesize that by giving ground truth labels for these pixels to guide the model, the model would more easily bridge the gap between the domains and improve the generalizability of the model. The detailed method on giving ground truth labels will be described in the next subsection.

Given $\phi', \theta'_1, \theta'_2$, we first apply the self-training loss function with the pseudo labels (one-hot vector labels generated from $\hat{\mathbf{Y}}_t = p(\mathbf{Y}|\mathbf{x}_t; \theta, \phi)$), which has been utilized in various tasks [9,10,14,23,32]:

$$\begin{aligned} \mathcal{L}_{self}(\phi', \theta'_1, \theta'_2) \\ = \mathbb{E}_{\mathbf{x}_t \in \mathcal{T}} [\text{CE}(\arg \max_K \hat{\mathbf{Y}}_t, p(\mathbf{Y}|\mathbf{x}_t; \theta'_1, \phi')) \\ + \text{CE}(\arg \max_K \hat{\mathbf{Y}}_t, p(\mathbf{Y}|\mathbf{x}_t; \theta'_2, \phi'))]. \end{aligned} \quad (4)$$

I. Shin et al.

Then, in order to optimize the two auxiliary classifiers to increase the discrepancy in relation to each other, we introduce an additional training stage to optimize the auxiliary classifiers to increase the distance between the classifiers' outputs. In addition, we also minimize the classifier discrepancy with respect to the backbone feature extractor $g_{\phi'}$, which results in a similar formulation to the classifier discrepancy maximization in MCDDA [20]:

$$\begin{aligned} & \min_{\phi'} \max_{\theta'_1, \theta'_2} \mathcal{L}_{\text{dis}}(\phi', \theta'_1, \theta'_2) \\ &= \min_{\phi'} \max_{\theta'_1, \theta'_2} \mathbb{E}_{\mathbf{x}_t \in \mathcal{T}} \left[\|f_{\theta'_1} \circ g_{\phi'}(\mathbf{x}_t) - f_{\theta'_2} \circ g_{\phi'}(\mathbf{x}_t)\|_1 \right]. \end{aligned} \quad (5)$$

Note that the goal of classifier discrepancy maximization in MCDDA is to create tighter decision boundaries in order to align the latent feature distributions between the source and the target domains. In contrast, we maximize the classifier discrepancy for the sole purposes of generating a more representative inconsistency mask so that the human annotator can give ground truth labels to pixels that truly require labels. After optimizing the auxiliary classifiers $(\theta'_1, \theta'_2, \phi')$, we utilize the different outputs from these classifiers and compare them in a pixel-to-pixel manner using (3) to obtain $M(\mathbf{x}_t)$. After the human annotator gives ground truth labels to the uncertain pixels based on $M(\mathbf{x}_t)$, the model (f_{θ}, g_{ϕ}) is then trained with the target dataset \mathcal{T} with the given ground truth labeled pixels $\tilde{\mathbf{Y}}_t(\mathbf{x}_t)$:

$$\mathcal{L}_t(\theta, \phi) = \mathbb{E}_{\mathbf{x}_t \in \mathcal{T}} [\text{CE}(\tilde{\mathbf{Y}}_t(\mathbf{x}_t), p(\mathbf{Y}|\mathbf{x}_t; \theta, \phi))]. \quad (6)$$

Then the process starting from copying $(\theta'_1 \leftarrow \theta, \theta'_2 \leftarrow \theta, \phi' \leftarrow \phi)$, optimizing $\mathcal{L}_{\text{self}}(\phi', \theta'_1, \theta'_2)$ and $\mathcal{L}_{\text{dis}}(\phi', \theta'_1, \theta'_2)$, to inconsistency generation $M(\mathbf{x}_t)$ is repeated. We repeat the process 3 times as we empirically found that the number of uncertain pixels and the model performance converges after 3 stages.

3.3 Adaptive Pixel Labeling

Given an inconsistency mask $M(\mathbf{x}_t)$, the question arises as how to give labels to the pixels. With this in mind, we propose two different methods for giving ground truth annotations with different focuses and strengths.

Segment based Pixel-Labeling (SPL). As the inconsistency mask shows all pixels that the model is uncertain about, we consider giving ground truth annotations for all the pixels selected. We call this method the Segment based Pixel-Labeling (SPL). In SPL, no further calculations are needed after the inconsistency mask has been generated, and after the pixels are annotated, the model $p(\mathbf{Y}|\mathbf{x}; \theta, \phi)$ is further trained. Empirically, we find that the inconsistency mask for each stage averages in percent of pixel of total pixels per image at 1% and totals to 2.2% at the final stage as some uncertain pixels are overlapped. The performance of SPL achieves near supervised learning, and it far exceeds the performance of our next method, which is more focused on drastically reducing human annotation labor.

Point based Pixel-Labeling (PPL). We also propose another Pixel-Labeling method that sets its focus on minimizing human annotation costs; we call this method the Point

LabOR: Labeling Only if Required for Domain Adaptive Semantic Segmentation

based Pixel-Labeling (PPL). Although PPL receives an inconsistency mask like SPL, we propose to label only the most *representative* pixels in the inconsistency mask instead of labeling all the pixels. Among the most representative pixels, we deliberately choose to maximize diversity by selecting all unique classes present in the inconsistency mask.

Given a set of uncertain pixels (inconsistency mask $M(\mathbf{x})$) and a model's output probability prediction for all the pixels $\hat{\mathbf{Y}} = \{\hat{\mathbf{Y}}_{i,j} \in \mathbb{R}^K | i \in [1, W], j \in [1, H]\}$, we first cluster the pixels that the model $p(\mathbf{Y}|\mathbf{x}; \theta, \phi)$ predicts to be the same class. We define the set of uncertain pixels \mathcal{D}^k for class k as follows:

$$\mathcal{D}^k = \{(i, j) \in M(\mathbf{x}) | k = \arg \max_K \hat{\mathbf{Y}}_{i,j}\}. \quad (7)$$

Then we compute the class prototype vector μ^k for each class k as the mean vectors of \mathcal{D}^k :

$$\mu^k = \frac{1}{|\mathcal{D}^k|} \sum_{(i,j) \in \mathcal{D}^k} \hat{\mathbf{Y}}_{i,j} \in \mathbb{R}^K. \quad (8)$$

Finally, we select the points that has the most similar probability vector for each prototype vector to construct the set of selected points P :

$$P(\mathbf{x}) = \left\{ \arg \min_{(i,j) \in \mathcal{D}^k} d(\mu^k, \hat{\mathbf{Y}}_{i,j}) \right\}_{k=1}^K. \quad (9)$$

We use cosine distance for a distance measure $d(\cdot, \cdot)$. Note that as \mathcal{D}^k can be a null set for some classes, $0 \leq |P(\mathbf{x}_t)| \leq K$, if the model fails to predict a certain class. At each stage, on average, the model generates 12 clusters, and cumulatively we average on giving 40 ground truth labels per target image \mathbf{x}_t in an image of size 640×1280 . This calculates to a $\approx 0.0049\%$ of the image being given ground truth labels. In comparison to SPL, which averages ≈ 18022 pixels $\rightarrow 2.2\%$ of entire image, we further reduce the human labeling costs by 0.2%. Due to the drastically reduced amount of ground truth annotations, PPL naturally under-performs in relation to SPL. Nevertheless, we empirically show that the performance gain of PPL over other UDA or weakly supervised DA methods is still significant.

4 Experiments

In this section, we conduct extensive experiments to analyze our methods both quantitatively and qualitatively.

4.1 Dataset

We evaluate our model on the most common adaptation benchmark of GTA5 [18] to Cityscapes [5]. Following the standard protocols from previous works [14,13], we adapt the model to the Cityscapes training set and evaluate the performance on the validation set.

I. Shin et al.

Method	GTA5 → Cityscapes																			
	Road	SW	Build	Wall	Fence	Pole	TL	TS	Veg.	Terrain	Sky	PR	Rider	Car	Truck	Bus	Train	Motor	Bike	mIoU
No Adapt	75.8	16.8	77.2	12.5	21.0	25.5	30.1	20.1	81.3	24.6	70.3	53.8	26.4	49.9	17.2	25.9	6.5	25.3	36.0	36.6
AdaptSegNet [25]	86.5	36.0	79.9	23.4	23.3	35.2	14.8	14.8	83.4	33.3	75.6	58.5	27.6	73.7	32.5	35.4	3.9	30.1	28.1	42.4
ADVENT [26]	89.9	36.5	81.2	29.2	25.2	28.5	32.3	22.4	83.9	34.0	77.1	57.4	27.9	83.7	29.4	39.1	1.5	28.4	23.3	43.8
SIMDA [28]	90.6	44.7	84.8	34.3	28.7	31.6	35.0	37.6	84.7	43.3	85.3	57.0	31.5	83.8	42.6	48.5	1.9	30.4	39.0	49.2
LTIR [11]	92.9	55.0	85.3	34.2	31.1	34.9	40.7	34.0	85.2	40.1	87.1	61.0	31.1	82.5	32.3	42.9	0.3	36.4	46.1	50.2
PCEDA [29]	91.0	49.1	85.6	37.2	29.7	33.7	38.1	39.2	85.4	35.4	85.1	61.1	32.8	84.1	45.6	46.9	0.0	34.2	44.5	50.5
FDA [30]	92.5	53.3	82.4	26.5	27.6	36.4	40.6	38.9	82.3	39.8	78.0	62.6	34.4	84.9	34.1	53.1	16.9	27.7	46.4	50.5
CBST [31]	91.8	53.5	80.5	32.7	21.0	34.0	28.9	20.4	83.9	34.2	80.9	53.1	24.0	82.7	30.3	35.9	16.0	25.9	42.8	45.9
CRST(MRKLID) [32]	91.0	55.4	80.0	33.7	21.4	37.3	32.9	24.5	85.0	34.1	80.8	57.7	24.6	84.1	27.8	30.1	26.9	26.0	42.3	47.1
TPLD [22]	94.2	60.5	82.8	36.6	16.6	39.3	29.0	25.5	85.6	44.9	84.4	60.6	27.4	84.1	37.0	47.0	31.2	36.1	50.3	51.2
IAST [14]	93.8	57.8	85.1	39.5	26.7	26.2	43.1	34.7	84.9	32.9	88.0	62.6	29.0	87.3	39.2	49.6	23.2	34.7	39.6	51.5
WDA [16] (Point)	94.0	62.7	86.3	36.5	32.8	38.4	44.9	51.0	86.1	43.4	87.7	66.4	36.5	87.9	44.1	58.8	23.2	35.6	55.9	56.4
Ours (PPL: Point)	96.1	71.8	88.8	47.0	46.5	42.2	53.1	60.6	89.4	55.1	91.4	70.8	44.7	90.6	56.7	47.9	39.1	47.3	62.7	63.5
Ours (SPL: Segment)	96.6	77.0	89.6	47.8	50.7	48.0	56.6	63.5	89.5	57.8	91.6	72.0	47.3	91.7	62.1	61.9	48.9	47.9	65.3	66.6
Supervised	96.9	77.1	89.8	45.6	49.9	47.4	55.8	64.1	90.0	58.2	92.8	71.9	46.9	91.4	60.3	65.8	54.3	44.6	64.7	66.7

Table 1. Experimental results on GTA5 → Cityscapes. While our PPL method already surpass previous UDA state-of-the-art models (e.g., IAST [14]) and DA model with few labels(e.g., WDA [16]) by only leveraging (around 40 labeled points per image), our SPL method shows the performance comparable with fully supervised learning (only 0.1% mIoU gap).

4.2 Experimental Results on GTA5 → Cityscapes

We show our quantitative results of both of our methods PPL and SPL compared to other state-of-the-art UDA methods [13,25,26] in Table. 1. Although out of our scope, we compare our method to Weak-label DA (WDA) [16] to show the competitiveness of our approach. To truly understand the capabilities of our approach, we also include the result of the fully supervised model. Table. 1 shows that our LabOR SPL outperforms all state-of-the-art UDA or WDA approaches in all cases by a large margin. Even when compared to the fully supervised method, SPL is only down by 0.1 mIoU in comparison. We believe this is a remarkable finding that can potentially be explored to hopefully surpass the performance of fully supervised methods. Even though our LabOR PPL only utilized point level supervision for the target dataset, PPL also shows significant performance gains over previous state-of-the-art UDA or WDA methods.

5 Conclusion

In this work, we tackle performance discrepancy of Unsupervised Domain Adaptation and proposed a new framework for domain adaptive semantic segmentation in a human-in-the-loop manner while generating the most informative pixel points that we call **Labeling Only if Required, LabOR**. Based on a self-training platform, we build our method to select the most *informative* pixels and introduce two pixel selection methods that we call “Segment based Pixel-Labeling” and “Point based Pixel-Labeling.”

[Remarks]

This paper is a re-publishing (summary presentation) of the paper which has been published in ICCV’21 by request of the IW-FCV2023 program committee to share the research results.

LabOR: Labeling Only if Required for Domain Adaptive Semantic Segmentation

References

1. Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.
2. Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.
3. Tseng-Hung Chen, Yuan-Hong Liao, Ching-Yao Chuang, Wan-Ting Hsu, Jianlong Fu, and Min Sun. Show, adapt and tell: Adversarial training of cross-domain image captioner. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, 2017.
4. Jae Won Cho, Dong-Jin Kim, Yunjae Jung, and In So Kweon. Mcdal: Maximum classifier discrepancy for active learning. *arXiv preprint arXiv:2107.11049*, 2021.
5. 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.
6. Florian Golemo, Adrien Ali Taiga, Aaron Courville, and Pierre-Yves Oudeyer. Sim-to-real transfer with neural-augmented robot simulation. In Aude Billard, Anca Dragan, Jan Peters, and Jun Morimoto, editors, *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pages 817–828. PMLR, 29–31 Oct 2018.
7. Raguraman Gopalan, Ruohan Li, and Rama Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *2011 international conference on computer vision*, pages 999–1006. IEEE, 2011.
8. Dong-Jin Kim, Jinsoo Choi, Tae-Hyun Oh, and In So Kweon. Image captioning with very scarce supervised data: Adversarial semi-supervised learning approach. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.
9. Dong-Jin Kim, Jinsoo Choi, Tae-Hyun Oh, Youngjin Yoon, and In So Kweon. Disjoint multi-task learning between heterogeneous human-centric tasks. In *Proc. of Winter Conference on Applications of Computer Vision (WACV)*, 2018.
10. Dong-Jin Kim, Xiao Sun, Jinsoo Choi, Stephen Lin, and In So Kweon. Detecting human-object interactions with action co-occurrence priors. In *Proc. of European Conf. on Computer Vision (ECCV)*, 2020.
11. Myeongjin Kim and Hyeran Byun. Learning texture invariant representation for domain adaptation of semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12975–12984, 2020.
12. Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6936–6945, 2019.
13. Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2019.
14. Ke Mei, Chuang Zhu, Jiaqi Zou, and Shanghang Zhang. Instance adaptive self-training for unsupervised domain adaptation. *arXiv preprint arXiv:2008.12197*, 2020.
15. Kwanyong Park, Sanghyun Woo, Inkyu Shin, and In-So Kweon. Discover, hallucinate, and adapt: Open compound domain adaptation for semantic segmentation. In *Proc. of Neural Information Processing Systems (NeurIPS)*, 2020.

I. Shin et al.

16. Sujoy Paul, Yi-Hsuan Tsai, Samuel Schulter, Amit K. Roy-Chowdhury, and Manmohan Chandraker. Domain adaptive semantic segmentation using weak labels. In *European Conference on Computer Vision (ECCV)*, 2020.
17. Stephan R Richter, Zeeshan Hayder, and Vladlen Koltun. Playing for benchmarks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2213–2222, 2017.
18. Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Proc. of European Conf. on Computer Vision (ECCV)*, volume 9906 of *LNCS*, pages 102–118. Springer International Publishing, 2016.
19. Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy, 2019.
20. Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3723–3732, 2018.
21. Burr Settles. Active learning. *Synthesis lectures on artificial intelligence and machine learning*, 6(1):1–114, 2012.
22. Inkyu Shin, Sanghyun Woo, Fei Pan, and In So Kweon. Two-phase pseudo label densification for self-training based domain adaptation. In *European Conference on Computer Vision*, pages 532–548. Springer, 2020.
23. Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Proc. of Neural Information Processing Systems (NeurIPS)*, 2020.
24. Jong-Chyi Su, Yi-Hsuan Tsai, Kihyuk Sohn, Buyu Liu, Subhransu Maji, and Manmohan Chandraker. Active adversarial domain adaptation, 2020.
25. Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 7472–7481, 2018.
26. Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2517–2526, 2019.
27. Zhonghao Wang, Yunchao Wei, Rogerio Feris, Jinjun Xiong, Wen-Mei Hwu, Thomas S. Huang, and Humphrey Shi. Alleviating semantic-level shift: A semi-supervised domain adaptation method for semantic segmentation, 2020.
28. Zhonghao Wang, Mo Yu, Yunchao Wei, Rogerio Feris, Jinjun Xiong, Wen mei Hwu, Thomas S. Huang, and Humphrey Shi. Differential treatment for stuff and things: A simple unsupervised domain adaptation method for semantic segmentation, 2020.
29. Yanchao Yang, Dong Lao, Ganesh Sundaramoorthi, and Stefano Soatto. Phase consistent ecological domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9011–9020, 2020.
30. Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4085–4095, 2020.
31. Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proc. of European Conf. on Computer Vision (ECCV)*, pages 289–305, 2018.
32. Yang Zou, Zhiding Yu, Xiaofeng Liu, B.V.K. Vijaya Kumar, and Jinsong Wang. Confidence regularized self-training. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

Attribute Auxiliary Clustering for Person Re-identification

Ge Cao and Kanghyun Jo*

Department of Electrical, Electronic and Computer Engineering, University of Ulsan,
Ulsan, 44610, Korea
 {caoge,acejo}@ulsan.ac.kr

Abstract. The main objective of the person re-identification task is to retrieve the specific identity under multiple non-overlapping camera scenarios. Though unsupervised person re-ID has already achieved great performance and even surpasses some classic supervised re-ID methods, the existing methods pay much attention to training the neural networks with the memory-based idea which ignore the quality of the generated pseudo label. The quality of the clustering process does not only depend on the intra-cluster similarity but also on the number of clusters. In this paper, our approach employs an attribute auxiliary clustering method for person re-ID task. The proposed method could divide the generated cluster by the leveraged attribute label. Employed the attribute auxiliary clustering, the task changed from unsupervised case to weakly supervised case. The method is compared with state-of-the-art and analyzes the effectiveness caused by the variation of the cluster number. The proposed approach achieves great performance on the public Market-1501 datasets.

Keywords: weakly supervised person re-identification · attribute auxiliary clustering · cluster number variation

1 Introduction

The main objective of the person re-identification task is to retrieve the specific identity under multiple non-overlapping camera scenarios [1]. With the increasing requirements for video surveillance and the urge for lower label annotating costs, unsupervised person re-ID got more attention in the past few years. For dealing with the unsupervised person re-ID task, purely unsupervised re-ID [11], [20], [12], [16], [3] and the unsupervised domain adaptation are the widely applied method [2], [12], [22], [23].

In this paper, we focus on the purely unsupervised person re-ID task. The state-of-the-art methods [3] extracted feature embedding through neural network [13] and then employed the clustering algorithms, DBSCAN [4] commonly to generate the pseudo label for training samples. With the generated pseudo label, we can train as a supervised case. Finally, a contrastive loss [27] is employed for training. Though the existing method has already achieved great performance, it still didn't reach the upper bound of the baseline, where the upper bound means

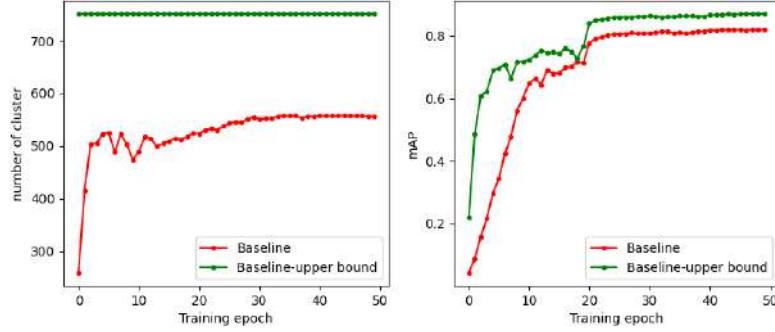


Fig. 1. The left and right subfigure shows the comparison of the number of clusters and performance between baseline and its upper bound (supervised), respectively

the performance when the clustering process gains the ideal results (Assuming the clustering results are completely correct). In Fig. 1, we show the comparison of the number of clusters and performance between the baseline ClusterContrast [3] and its upper bound (supervised). The left subfigure shows that even after the training finished, the clustering process could only divide the training samples into around 500 clusters which is much less than the ideal number of clusters. Correspondingly, when we can get the ideal clustering results the upper bound obviously surpass the baseline by a large margin.



Fig. 2. Parts of clustering results selected from the final epoch's clustering results, where the samples of the same row are selected from the same cluster. Among them, the samples in the blue box and green box are captured from different identities.

The result demonstrates that the unsupervised method has not achieved the ideal performance and the key reason lies in the low quality of clustering quality. Fig. 2 displays parts of clustering results selected from the final epoch’s clustering results, where the samples of the same row are selected from the same cluster. Among them, the samples in the blue box and green box are captured from different identities. The results show that the clustering could not recognize the highly similar vision features. But for human beings, we can easily find that the first four samples of the first row are captured from a male but the last four samples of the first row are captured from a female and in the same condition as the second row. In this paper, we generate the clustering results both in feature space and attribute space. The attributes of the identity annotate the sample at the semantic level. Our contribution could be summarized in three-fold:

- We leverage the attribute label and propose the attribute auxiliary clustering (AAC) method to explore the attribute auxiliary weakly supervised person re-ID task.
- The analysis of performance caused by cluster number variation is indicated in this paper.
- We comprehensively evaluate and compare the performance of AAC with state-of-the-art, which surpasses other weakly supervised person re-ID works.

2 Related Work

2.1 Unsupervised Person Re-ID Works

Despite the classic algorithm computing without deep learning, unsupervised person re-ID can be categorized into two situations. With the annotated label in the source domain, unsupervised domain adaptation (UDA) [2], [12], [22], [23] methods are the first category. Among them, ECN [22] firstly applied the memory bank idea [19] to store the features and update with the training process. SpCL [12] proposed a novel self-paced contrastive learning framework that gradually creates a more reliable cluster, which to refine the memory dictionary features. The second category is purely unsupervised person re-ID (USL) [11], [20], [12], [16], [3] which only focuses on the target dataset and does not leverage any labeled data. MMCL [11] employed the memory bank in the USL field and calculated the pseudo label with similarity. CAP [16] applied the cluster method DBSCAN to generate the pseudo label and construct the memory bank at cluster-level and proxy-level (detailed in camera id). ClusterContrast [3] summarized the mainstream contrastive learning-based USL method and mainly focused on controlling the cluster size for consistency in the training process. The proposed AAC is based on the ClusterContrast framework, and due to the leveraging of the attribute label, AAC is exploring the re-ID field under the weakly supervised case.

2.2 Attribute Auxiliary Person Re-ID

Thanks to the work and attribute annotation by Lin et al. [25], researchers are easier to train and learn the identity embedding with auxiliary attributes. GPS

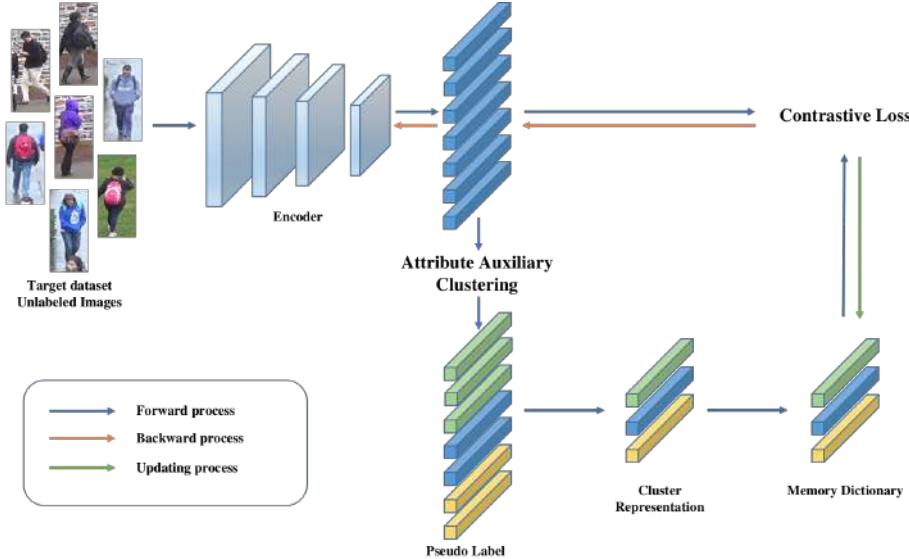


Fig. 3. The overview pipeline of the proposed method. The proposed attribute auxiliary clustering (AAC) method is applied for generating the pseudo label for the training samples. The ClusterNCE loss which is introduced in Eq. 1 is applied for the contrastive loss.

[24] constructs the relationship graph for identity attribute and human body part, which could represent the unique signature of the identity. The graph-based signature can also be employed in unsupervised cases [18]. TJ-AIDL [17] simultaneously trains with attribute level and feature level to transfer attribute and identity label information to the target domain. The proposed AAC re-allocated the clustering results rather than applying the attribute for training in the attribute-semantic space.

3 Methodology

The pipeline for purely unsupervised person re-ID is described in Section 3.1, which includes the re-ID problem formulation and training strategy followed [3]. The proposed attribute auxiliary clustering method is demonstrated in Section 3.2.

3.1 USL Person Re-ID pipeline

For the training process, given target dataset $X = \{x_1, x_2, \dots, x_{N_t}\}$, we can extract discriminative feature embeddings $F = \{f_1, f_2, \dots, f_{N_t}\}$ by the encoder network [13], where N_t denotes the number of training samples. The follow-up series of works employed for USL training is shown in Fig. 3. For the testing



Fig. 4. Parts of clustering results selected from the final epoch's clustering results, where the samples of the same row are selected from the same cluster. Among them, the samples in the blue box and red box are captured from different identities.

process, given query sample q and gallery samples $G = \{g_1, g_2, \dots, g_{N_g}\}$, get the feature embedding f_q and $\{f_{g_1}, f_{g_2}, \dots, f_{g_{N_g}}\}$ from the trained encoder network, then calculate the similarity between f_q and f_g , and finally rank the list.

In the training process, after extracting feature embedding from the encoder network, we employ the classic clustering algorithm DBSCAN [4] for generating the pseudo labels for training samples, which are denoted as $\{y_1, y_2, \dots, y_{N_t}\}$. This work applies the ClusterNCE loss followed ClusterContrast [3] as the contrastive loss:

$$L = -\log \frac{\exp(f_q \cdot \phi_+ / \tau)}{\sum_{k=0}^K \exp(f_q \cdot \phi_k / \tau)} \quad (1)$$

where ϕ_+ is the positive cluster representation vector of q , and ϕ_k is negative unique representation vector of the k -th cluster. The cluster representation ϕ_k is initialized by Eq. 2:

$$\phi_k = \frac{1}{|N_k|} \sum_{f_i \in N_k} f_i \quad (2)$$

where N_k is the set of samples in the k -th cluster, and it is verified as the encoder network trains in the process. During the training process, we select K samples in P clusters and construct the training minibatch. The cluster representation vectors are updated by:

$$\phi_k \leftarrow m\phi_k + (1-m)q \quad (3)$$

where m is the momentum updating rate. And the above process followed [3] is framed as the baseline in this paper.

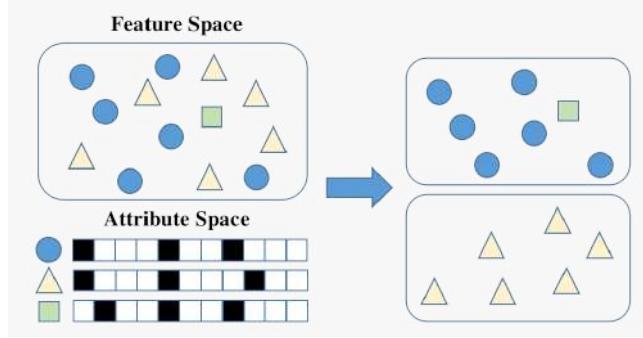


Fig. 5. Illustration of the proposed attribute auxiliary clustering (AAC) method. The samples in different shapes denote the sample captured from different identities. The samples are classified as the same cluster initially and re-clustered into the different clusters by applying AAC.

3.2 Attribute Auxiliary Clustering

Though baseline [3] has already gained state-of-the-art performance in the USL field, it still has plenty of space for improvement as shown in Fig. 1.

The upper bound of baseline: The upper bound means that we get the completely correct clustering results in every epoch. Obviously, the existing technology cannot reach that case, so the upper bound would happen when we directly apply the ground truth of the training label. For the baseline-upper-bound, we divide the training samples into N_t clusters directly applying the GT, so it is under the supervised case. The results are shown in Fig. 1 and Table. 1, which surpasses the baseline by a large margin. Due to the great potential for improving the clustering quality, we leverage the attribute label $A = \{a_1, a_2, \dots, a_{N_t}\}$ for fine-tuning the clustering results. Operating with the clustering results of the final epoch of the baseline, there are mainly two cases. The first case is shown in Fig. 2, where a cluster contains samples captured from two or more identities and the samples from each identity could be separately clustered. And another case is shown in Fig. 4, where a cluster contains many clusters but the samples from some identities are just one or two, which cannot be individually clustered.

The process of AAC is shown in Fig. 5, where the samples in different shapes denote the sample captured from different identities. The samples are classified as the same cluster initially and re-clustered into the different clusters by applying AAC.

In the training process, given the training samples $\{x_1, x_2, \dots, x_{N_t}\}$ and corresponding attribute label $\{a_1, a_2, \dots, a_{N_t}\}$, we extract the feature embedding $\{f_{g_1}, f_{g_2}, \dots, f_{g_{N_t}}\}$ from the encoder network. Then DBSCAN [4] is employed for generating the pseudo labels $\{y_1, y_2, \dots, y_{N_t}\}$. The samples which have the same pseudo labels y_i are classified as the same cluster and some clusters contain different attribute labels a_i . For the samples which are in the same cluster and the

number of the samples with the same attribute label more than a threshold δ , we document their attribute label as t_1, t_2, \dots, t_K , where K means the number of different attribute label in one cluster. The signal δ is set for avoiding generating some bad clusters with only a few samples which would be unbalanced distributed. So in the AAC algorithm, we will ignore the second case above.

We use the $y_i \rightarrow y'_i$ to denote the process that the training sample x_i should be re-clustered with new pseudo labels y'_i :

$$\begin{aligned} & y_i \rightarrow y'_i \\ & s.t. \sum_{i=0, i \neq k}^{N_t} a_i \bigoplus t_k = 0 \end{aligned} \quad (4)$$

The discussions of the threshold δ and the start epoch for applying the AAC method are introduced in the ablation study.

4 Experiments

4.1 Datasets and Implementation

Datasets *Market-1501* [14] is a widely used public person re-ID dataset, which captured 12,936 samples with 751 identities in the training set, 3,368 and 15,913 samples captured from 750 identities for query and gallery set. The attribute label is provided by [25], which has 27 attributes for each training sample.

Implementation The ResNet50 [13] pre-trained on ImageNet [21] is employed for the encoder network. Followed [3], the feature embedding is 2048- d extracted by a global average pooling, batch normalization, and the L2-normalization layer.

The input of the samples is resized to 128×256 and processed by random horizontal flipping, padding, random cropping, and random erasing. The batch size is equal to 256 (16 samples from each identity). Adam is applied for the optimizer with 5e-4 of the weight decay. The initial learning rate is 5.5e-4 and reduced to ten times smaller every 20 epochs in a total of 60 epochs.

The maximum distance is set to 0.6 and the minimal number of clusters is set to 4 for the DBSCAN setting. The threshold δ is set to 5. and from the first epoch, we start to apply the AAC method.

4.2 Comparison with State-of-the-arts

We compare the proposed method with stat-of-the-arts. The method with attributes weakly supervised is few so we compare it with some SOTA USL papers. For Table. 1, the 'Setting' means the training case they applied and 'Auxiliary' means whether any auxiliary information is leveraged. And the mAp, rank-1 score, rank-5 score, and rank-10 score of the proposed AAC method surpasses the baseline [3] by 3.9%, 2.0%, 0.4%, and 0.6%, respectively.

Table 1. Comparison results with the state-of-the-arts on Market-1501 [14] dataset. In the table, AAC denotes the proposed attribute auxiliary clustering algorithm by this paper, GT denotes the ground truth, and the signal † denotes the results are tested under the same implementation with the proposed idea. The best results are bold in this table. Additionally, the upper bound of the baseline is shown as the maximum limit of unsupervised work.

Method	Reference	Setting	Auxiliary	Market1501				
				mAP	rank-1	rank-5	rank-10	
LOMO [5]	CVPR15	USL	None	8.0	27.2	41.6	49.1	
BOW [6]	ICCV15	USL	None	14.8	35.8	52.4	60.3	
UDML [7]	CVPR16	USL	None	12.4	34.5	52.6	59.6	
DECAMEL [8]	TPAMI18	USL	None	32.4	60.2	76.0	81.1	
TJ-AIDL [17]	CVPR18	Weakly	Attribute	26.5	58.2	74.8	81.1	
DBC [10]	BMVC19			41.3	69.2	83.0	87.8	
BUC [9]	AAAI19	USL	None	38.3	66.2	79.6	84.5	
MMCL [11]	CVPR20	USL	None	45.5	80.3	89.4	92.3	
SpCL [12]	NeurIPS20	USL	None	73.1	88.1	95.1	97.0	
GCL [20]	CVPR21	USL	None	66.8	87.3	93.5	95.5	
CAP [16]	AAAI21	USL	Camera ID	79.2	91.4	96.3	97.7	
A2G [15]	Access21	Weakly		71.6	87.4	95.2	97.2	
ClusterContrast† [3]	ACCV22			None	82.1	92.3	96.9	97.6
AAC	This paper	Weakly	Attribute	86.0	94.3	97.9	98.5	
Baseline-upper	This paper	Supervised	GT	87.2	95.0	98.3	99.1	

Fig. 6 shows the performance comparison and cluster number comparison among the baseline [3] (USL), AAC (weakly supervised), and baseline upper bound (supervised). The right subfigure shows that the performance of the proposed AAC surpasses the baseline a lot and is already close to the supervised upper bound. For the left subfigure which shows the cluster number variation with the training epoch, the cluster number increases very rapidly after applying AAC in some of the first epochs. It is caused by the poor clustering quality, as shown in Fig. 7, the clustering results of some of the first epochs are very bad for training, most clusters contain many samples captured from many identities. So when we apply the AAC idea with a small threshold δ , the cluster number would increase a lot and then decrease to the stable situation with the training.

4.3 Ablation Studies.

In this section, we introduce the extended experiments about the starting epoch for applying the AAC method and the effectiveness contributed by a changed number of clusters. As shown in Fig. 7, the samples are mostly clustered into wrong pseudo labels, so it is necessary for exploring whether the AAC should be applied in the initial epoch. Table. 2 demonstrates the performance when applying AAC in different start epochs with threshold $\delta=10$, and Fig. 8 shows the

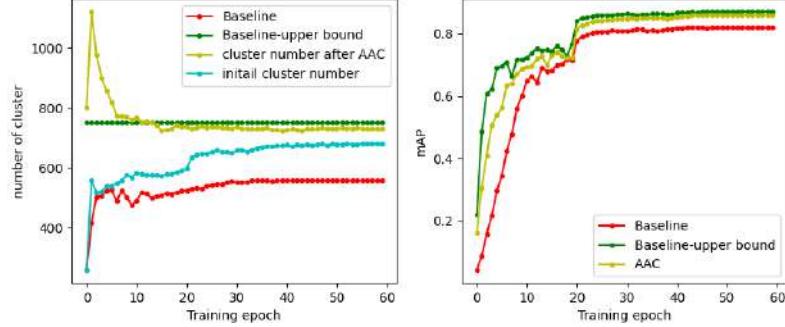


Fig. 6. The left subfigure shows the comparison of the number of clusters among the baseline, baseline upper bound (supervised), initial cluster number (before applying AAC), and cluster number after AAC (after applying AAC). The right subfigure shows the comparison of the mAP performance among the baseline, baseline upper bound, and the proposed AAC. The value of AAC is tested under the best performance.

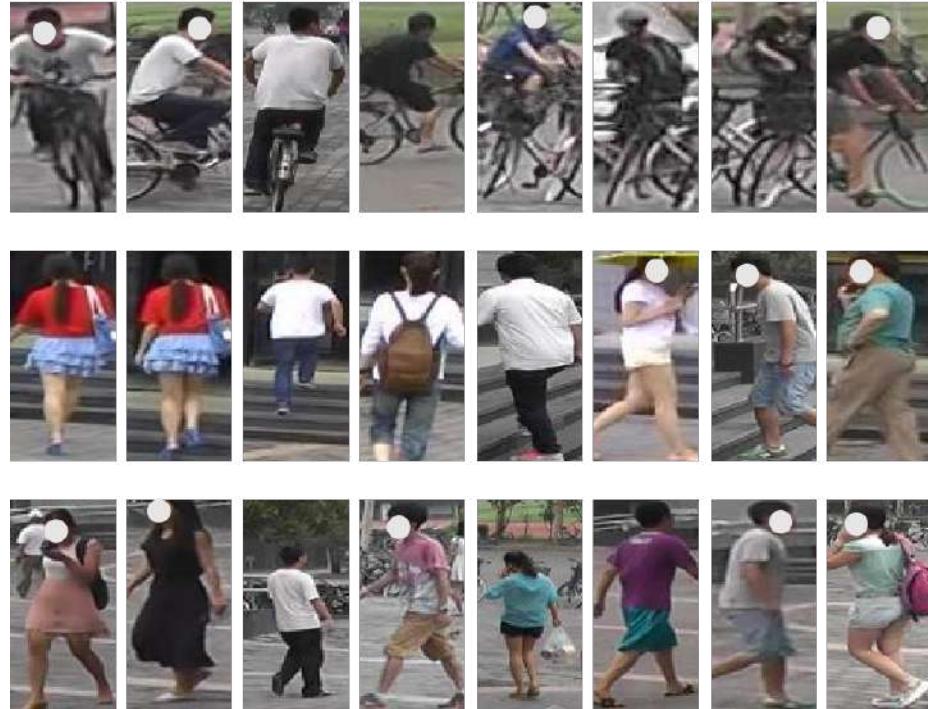


Fig. 7. Parts of clustering results selected from the first epoch's clustering results, where the samples of the same row are selected from the same cluster. Among them, the samples are mostly clustered into the wrong pseudo-label.

Table 2. Retrieval accuracy with different epochs for starting applying the proposed AAC method ($\delta=10$ in this experiment). The best performance is bold.

Start Epoch	Market-1501			
	mAP	rank-1	rank-5	rank-10
0	84.9	93.6	97.3	98.2
5	84.6	93.6	97.4	98.3
10	84.2	93.8	97.5	98.2
15	83.2	92.9	96.9	97.8
20	83.0	92.6	96.7	97.7
25	82.7	92.7	96.7	97.6
30	82.6	92.4	96.6	97.7
35	82.2	92.3	96.5	97.5
40	82.0	92.3	96.5	97.6

performance when applying AAC with the different epochs. The results indicate that applying AAC during the training process achieves the best performance.

About the ablation study for the threshold δ , we test the performance from 4-10 for the Market-1501 dataset. The results do not have a linear pattern and we achieve the best performance with $\delta=5$.

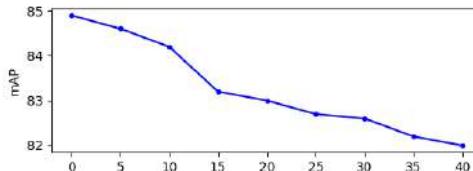


Fig. 8. The performance when applying AAC in different epochs.

Discussions: The effectiveness of applying AAC from the first epoch is best because of the low inter-class variations which caused a low initial cluster number in the Market-1501 dataset (the cluster numbers of some of the first epochs are much smaller than the total identity number).

5 Conclusions

This paper proposes the attribute auxiliary clustering method for weakly supervised person re-identification work. It re-allocates the pseudo label for training samples and effectively improves the performance and convergence speed compared with the baseline. The experiments show that the proposed idea achieves state-of-the-art.

Table 3. Retrieval accuracy with different epochs for starting applying the proposed AAC method. The best performance is bold.

δ	Market-1501			
	mAP	rank-1	rank-5	rank-10
3	85.8	94.3	97.8	98.8
4	85.9	94.3	97.9	98.5
5	86.0	94.2	97.6	98.4
6	85.5	94.1	97.6	98.5
7	85.4	93.7	97.3	98.3
8	85.4	93.6	97.3	98.1
9	85.7	93.8	97.8	98.7
10	83.0	92.6	96.7	97.7

Acknowledgements

This result was supported by "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(MOE)(2021RIS-003).

References

1. Ye, M., Shen, J., Lin, G., Xiang, T., Shao, L., and Hoi, S. C. H., "Deep Learning for Person Re-identification: A Survey and Outlook", *< i>arXiv e-prints</i>*, 2020.
2. Liangchen Song, Cheng Wang, Lefei Zhang, Bo Du, Qian Zhang, Chang Huang, and Xinggang Wang. Unsupervised domain adaptive re-identification: Theory and practice. PR, 2020. 1, 2
3. Dai, Z., Wang, G., Yuan, W., Liu, X., Zhu, S., and Tan, P., "Cluster Contrast for Unsupervised Person Re-Identification", *< i>arXiv e-prints</i>*, 2021.
4. Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X.; et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In Kdd.
5. S. Liao, Y. Hu, X. Zhu, and S. Z. Li, "Person Re-identification by Local Maximal Occurrence Representation and Metric Learning," *arXiv e-prints*, p. arXiv:1406.4216, Jun. 2014.
6. L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," 12 2015, pp. 1116–1124
7. P. Peng, T. Xiang, Y. Wang, M. Pontil, S. Gong, T. Huang, and Y. Tian, "Unsupervised cross-dataset transfer learning for person reidentification," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1306–1315.
8. H. Yu, A. Wu, and W. Zheng, "Unsupervised person re-identification by deep asymmetric metric embedding," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 4, pp. 956–973, 2020.
9. Y. Lin, X. Dong, L. Zheng, Y. Yan, and Y. Yang, "A bottom-up clustering approach to unsupervised person re-identification," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 8738–8745, 07 2019.
10. G. Ding, S. H. Khan, and Z. Tang, "Dispersion based clustering for unsupervised person re-identification," in BMVC, 2019.

11. D. Wang and S. Zhang, "Unsupervised Person Re-identification via Multi-label Classification," arXiv e-prints, p. arXiv:2004.09228, Apr. 2020.
12. Yixiao Ge, Feng Zhu, Dapeng Chen, Rui Zhao, and Hongsheng Li. Self-paced contrastive learning with hybrid memory for domain adaptive object re-id. In NeurIPS, 2020. 1, 2, 3, 7, 8, 10, 11
13. He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In CVPR.
14. Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In ICCV, 2015. 2, 5, 6, 7, 8.
15. G. Tang, X. Gao, Z. Chen and H. Zhong, "Graph Neural Network Based Attribute Auxiliary Structured Grouping for Person Re-Identification," in IEEE Access, doi: 10.1109/ACCESS.2021.3069915.
16. Menglin Wang, Baisheng Lai, Jianqiang Huang, Xiaojin Gong, and Xian-Sheng Hua. Camera-aware proxies for unsupervised person re-identification. In AAAI, 2021. 2, 3, 4, 7, 8, 11.
17. Jingya Wang, Xiatian Zhu, Shaogang Gong, and Wei Li. Transferable joint attribute-identity deep learning for unsupervised person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2275–2284, 2018.
18. G. Cao, Q. Tang and K. Jo, "Graph-based Attribute-aware Unsupervised Person Re-identification with Contrastive learning," 2022 International Workshop on Intelligent Systems (IWIS), Ulsan, Korea, Republic of, 2022, pp. 1-6, doi: 10.1109/IWIS56333.2022.9920894.
19. Wu, Z., Xiong, Y., Yu, S., and Lin, D., "Unsupervised Feature Learning via Non-Parametric Instance-level Discrimination", <i>arXiv e-prints</i>, 2018.
20. Chen, H., Wang, Y., Lagadec, B., Dantcheva, A., and Bremond, F., "Joint Generative and Contrastive Learning for Unsupervised Person Re-identification", <i>arXiv e-prints</i>, 2020.
21. J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.
22. Z. Zhong, L. Zheng, Z. Luo, S. Li, and Y. Yang, "Invariance Matters: Exemplar Memory for Domain Adaptive Person Re-identification," arXiv e-prints, p. arXiv:1904.01990, Apr. 2019.
23. Y. Fu, Y. Wei, G. Wang, Y. Zhou, H. Shi, and T. Huang, "Self-similarity Grouping: A Simple Unsupervised Cross Domain Adaptation Approach for Person Re-identification," arXiv e-prints, p. arXiv:1811.10144, Nov. 2018.
24. B. X. Nguyen, B. D. Nguyen, T. Do, E. Tjiputra, Q. D. Tran and A. Nguyen, "Graph-based Person Signature for Person Re-Identifications," 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2021, pp. 3487-3496, doi: 10.1109/CVPRW53098.2021.00388.
25. Yutian Lin, Liang Zheng, Zhedong Zheng, Yu Wu, Zhilan Hu, Chenggang Yan, and Yi Yang. Improving person re-identification by attribute and identity learning. Pattern Recognition, 95:151–161, 2019.
26. H.-X. Yu, W.-S. Zheng, A. Wu, X. Guo, S. Gong, and J.-H. Lai, "Unsupervised Person Re-identification by Soft Multilabel Learning," arXiv e-prints, p. arXiv:1903.06325, Mar. 2019.
27. Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748, 2018. 1, 2

UDA-COPE: Unsupervised Domain Adaptation for Category-level Object Pose Estimation *

Taeyeop Lee, Byeong-Uk Lee, Inkyu Shin, Jaesung Choe,
Ukcheol Shin, In So Kweon, and Kuk-Jin Yoon

KAIST

Abstract. Learning to estimate object pose often requires ground-truth (GT) labels, such as CAD model and absolute-scale object pose, which is expensive and laborious to obtain in the real world. To tackle this problem, we propose an unsupervised domain adaptation (UDA) for category-level object pose estimation, called **UDA-COPE**. Inspired by recent multi-modal UDA techniques, the proposed method exploits a teacher-student self-supervised learning scheme to train a pose estimation network without using target domain pose labels. We also introduce a bidirectional filtering method between the predicted normalized object coordinate space (NOCS) map and observed point cloud to not only make our teacher network more robust to the target domain but also to provide more reliable pseudo labels for the student network training. Our results demonstrate the effectiveness of our proposed method both quantitatively and qualitatively. Notably, without leveraging target-domain GT labels, our proposed method achieved comparable or sometimes superior performance to existing methods that depend on the GT labels.

Keywords: Object Pose Estimation · Unsupervised Domain Adaptation · Augmented Reality (AR) · Virtual Reality (VR) · Robotics

1 Introduction

Object pose estimation is one of the crucial tasks used in various robotics and computer vision applications for robot manipulation [33, 31, 28, 7] and augmented reality (AR) [23, 19, 20]. Using sensor data such as images or point clouds, this task aims to estimate the poses of target objects including 3D orientation, 3D location, and size information.

Previous 6D object pose estimation methods follow the instance-level pose estimation schemes [26, 32, 21, 22, 28, 11, 10] that rely on given 3D CAD model information (*e.g.*, keypoints, geometry) and the size of known objects. However, these methods typically have difficulty estimating the pose of unknown objects since they do not yet have 3D CAD models as priors. In contrast to the instance-level scheme, category-level object pose estimation [29, 25, 4, 17, 30, 5] approaches are more efficient in that a single network can infer multiple classes at once.

* This paper is the short version of CVPR’22 official publication.

In particular, Wang *et al.* [29] introduced a pioneering representation called Normalized Object Coordinate Space (NOCS), to align different object instances within one category in a shared 3D orientation. By estimating per-category NOCS maps, it is able to estimate the 6D pose of unseen objects without prior 3D CAD models. Its strengths have led to the use of NOCS representation in the following studies [25, 4, 17, 30, 5].

However, current object pose estimation research mostly relies on supervised learning, which requires expensive GT labels such as 3D object CAD models and absolute object pose. These labels are not only difficult to obtain in the real world but are also unreliable due to the human-annotation. Because of this difficulty, most of the training depends on synthetic datasets [24, 13, 26] and is usually not feasible in real-world applications due to domain gaps.

To cope with the real-world data scarcity problem, we take a look at unsupervised domain adaptation (UDA) methods [12, 16, 36]. UDA approaches often consider two types of datasets, the source domain (*i.e.* synthetic dataset) and the target domain (*i.e.* real-world dataset) dataset. The main goal of the UDA methods is to successfully make deep learning networks robust to the target domain using only the GT labels of the source domain. Various techniques exist, such as pseudo label generation [12, 16], teacher and student networks with momentum updates [1, 34], adversarial learning [3, 14, 2], and etc.

In this paper, we propose an Unsupervised Domain Adaptation for Category-level Object Pose Estimation (UDA-COPE). The proposed method effectively transfers task knowledge from a synthetic domain to a real domain by exploiting a multi-modal self-supervised learning scheme using pseudo labels. Our UDA-COPE concentrates on how to make high-quality pseudo-labels that are efficiently targeted for the category-level pose estimation task. To this end, we designed *bidirectional point filtering* to remove noisy and inaccurate points based on pose optimization. Moreover, our framework achieved better performance than the previous supervised methods [29, 25, 4, 30].

2 Proposed Method

Given an RGB image I , point cloud P , and segmentation labels S , our architecture aims to regress the 6D pose and size $s \in \mathbb{R}^3$ of objects. The 6D pose is defined as the rigid transformation of $[R|t]$: rotation $R \in SO(3)$, and translation $t \in \mathbb{R}^3$. Following previous studies [25, 4, 30, 5, 17], the segmentation labels S are used to crop the RGB images and point clouds. We leverage the NOCS representation to align different object instances within one category in a shared orientation. The categorical object pose $[R|t]$ and size s are estimated by Umeyama algorithm [27] with RANSAC [8], which optimizes $[R|t]$ and s by minimizing the distances between point cloud P and an estimated NOCS map N .

We first illustrate our network architecture (Sec. 2.1). Then, we introduce training methods of supervised learning using synthetic dataset (Sec. 2.2) and unsupervised domain adaption using real-world dataset (Sec. 2.3).

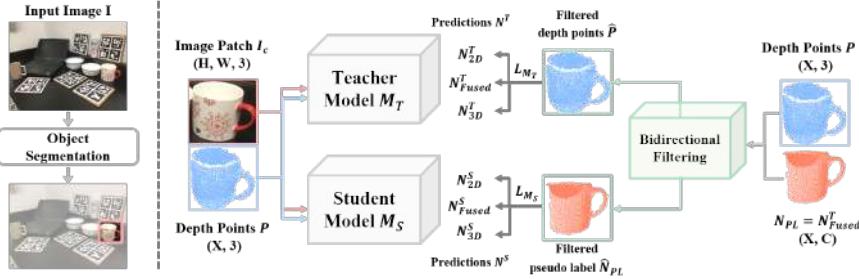


Fig. 1. Overview of unsupervised domain adaptation for category-level object pose estimation (UDA-COPE). UDA-COPE utilizes pseudo label-based teacher/student training scheme. Our proposed bidirectional point filtering method removes the noisy pseudo labels and gives reliable guidance to the student network. At the same time, filtered depth points gives additional self-supervision to the teacher network so that it can be robust to the domain gap between the synthetic and real dataset.

2.1 Network Architecture

Recent category-level object pose estimation methods [25, 4, 17] take an RGB-D input to extract the 2D/3D features. We designed separate 2D/3D branches to extract features from both modalities. We use PSPNet [35] with ResNet34 [9] for 2D feature extraction and the Mink16UNet34 [6] for 3D feature extraction. At this time, the 2D feature is extracted by sampling features that are validly matched with point cloud P from the feature volume. Finally, we have a fused branch that combines each feature from both branches. Every branch estimates a NOCS map (N) with a separate NOCS header, which consists of three multi-layer perceptrons (MLP) layers. We designate the NOCS map estimation of each branch as N_{2D} , N_{3D} , N_{Fused} , according to respective feature property.

2.2 Pre-Training with Synthetic Data

Inspired by pseudo label (PL) based methods [12, 16], our method consists of a teacher and a student model. Fig. 1 shows the overview of our teacher and student model. The initial prediction of teacher model M_T becomes a pseudo label N_{PL} for a student model, and student model M_S learns from the pseudo label as a GT. Our teacher and student model have the same structure as was described in Sec. 2.1.

We first train our teacher model in a supervised manner using the labeled synthetic dataset. For the NOCS map prediction using the GT information, we utilize cross-entropy loss, as in, $H(N_{gt}, N^T)$, where the supervision is given to all predictions from three branches. Additionally, to make our teacher network more robust, we apply the 2D image and 3D points augmentation and use consistency loss L_C so that each modality can output consistent results. Total loss for the

teacher network on the synthetic dataset is formulated as follows:

$$\begin{aligned} L_{MT} &= \lambda_N H(N_{gt}, N^T) + \lambda_C L_C, \\ L_C &= H(N^T, N_{Aug}^T) \end{aligned} \quad (1)$$

where N_{Aug}^T is the NOCS map prediction from the augmented input, and λ_N and λ_C are weighting parameters. Notation for the modality of the predictions is discarded for better readability.

2.3 Pose-Aware Unsupervised Domain Adaptation

After training from the synthetic dataset, the most straightforward yet naive approach is to train the student network using the prediction of the teacher network. However, using the initial prediction from the teacher model as a pseudo-label can be risky. The risk is due to the lack of robustness of the teacher model itself, or more importantly, because of the insufficient knowledge that the teacher model holds with respect to real-world scenarios, due to the domain gap between the simulated and real worlds. Techniques such as data augmentation and momentum updates might help the feasibility but are still restricted. Therefore, we need additional guidance for the teacher model to estimate high-quality predictions, and more reliable pseudo labels for our student model to learn from.

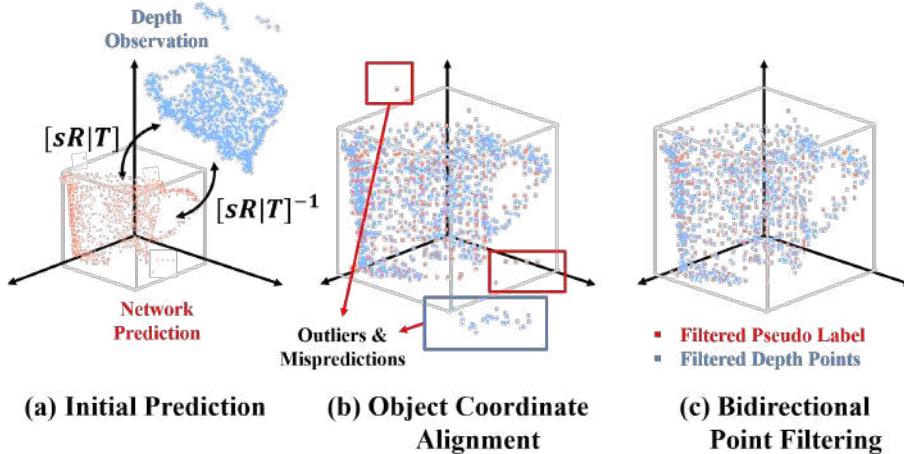


Fig. 2. Overview of bidirectional point filtering method. Given pseudo labels and depth points (a), we estimate the pose and size using the Umeyama [27] algorithm and RANSAC [8], and align the depth points to normalized object coordinate (b). The pseudo label (red) and aligned depth points (blue) have noisy and inaccurate points. After our bidirectional point filtering, the noisy points are removed to give more reliable supervision for both teacher and student (c).

Bidirectional Point Filtering To solve these problems, we propose the bidirectional point filtering method which simultaneously removes the noise of the pseudo labels for the student and filters noisy depth points P for a teacher network. Fig. 2 shows an overview of the proposed bidirectional filtering method. Our bidirectional filtering method uses the P and N_{PL} as input and initially estimates the pose $[R|t]$ and size s using the Umeyama algorithm [27] with RANSAC [8]. Then it aligns the depth points P to the NOCS coordinate by applying the inverse of the estimated pose, as in multiplying the matrix $[sR|t]^{-1}$. We denote aligned depth points as P' . And then we calculate the point-wise 3D distance d between the aligned depth points P' and pseudo label N_{PL} to filter out noisy points from both sides using ρ as the threshold. Finally, we get the refined pseudo label \hat{N}_{PL} and filtered aligned depth \hat{P} . Our bidirectional point filtering can be expressed as:

$$\begin{aligned} d(n) &= \|P'(n) - N_{PL}(n)\| \quad \text{where } \forall n \in [1|P'], \\ \hat{N}_{PL} &= \{N_{PL}(n) : d(n) < \rho\}, \\ \hat{P} &= \{P'(n) : d(n) < \rho\}, \end{aligned} \tag{2}$$

Fig. 2 shows that our bidirectional filtering method removes outliers of pseudo label N_{PL} and depth P , and results in refined pseudo label \hat{N}_{PL} and filtered depth points \hat{P} .

Self-Supervised Learning After the bidirectional filtering, we jointly train the teacher network and student network using the filtered pseudo labels \hat{N}_{PL} and filtered aligned depth points \hat{P} . Noted that we only use the filtered points \hat{P} for the teacher training, which may be a smaller subset of an original P . We use cross-entropy loss to train a student model using clean pseudo labels \hat{N}_{PL} . The student model loss is defined as:

$$L_{MS} = -\frac{1}{|\hat{N}_{PL}|} \sum_{n=1}^{|\hat{N}_{PL}|} H(\hat{N}_{PL}(n), N^S(n)), \tag{3}$$

where N^S is the predictions of our student network. At the same time, the teacher learns real data knowledge from observation. We use cross-entropy loss by utilizing geometric consistency between our filtered aligned depth \hat{P} and estimated pseudo labels N^T . The teacher model loss is defined as:

$$L_{MT} = -\frac{1}{|\hat{P}|} \sum_{n=1}^{|\hat{P}|} H(\hat{P}(n), N^T(n)). \tag{4}$$

We train our teacher model with a small learning rate for stable teacher network training. For both teacher and student models, we compute the loss for all estimations, N_{2D} , N_{3D} , N_{Fused} , which shows better result than applying the loss to only N_{Fused} . We denote all estimation losses as all modality (AM) loss.

3 Experiments

3.1 Comparison with State-of-the-art

We compared our methods with state-of-the-art methods that were trained on different datasets and labels: 1) labeled synthetic dataset, 2) labeled synthetic and real datasets, 3) labeled synthetic and unlabeled real datasets. All methods were evaluated on the REAL275 dataset. Note that only the approaches with the ability to perform multi-class category-level pose estimation using a single network were considered. RGB, Depth and RGB-D denotes the modality of the network input, and most of the RGB based approaches utilize depth information in the pose optimization or refinement process.

Method	Input	Syn	Real	Real	mAP (\uparrow)					
			w/ Label	w/o Label	3D ₅₀	3D ₇₅	5°2cm	5°5cm	10°2cm	10°5cm
CPS++ [18]	RGB	✓			72.6	-	-	25.8	-	-
Metric Scale [15]	RGB	✓			68.1	32.9	2.2	5.3	10.0	24.7
NOCS [29]	RGB	✓			36.7	3.4	-	3.4	-	20.4
SPD [25]	RGB-D	✓			71.0	43.1	11.4	12.0	33.5	37.8
NOCS [29]	RGB	✓	✓		78.0	30.1	7.2	10.0	13.8	25.2
SPD [25]	RGB	✓	✓		75.2	46.5	15.7	18.8	33.7	47.4
SPD [25]	RGB-D	✓	✓		77.4	53.5	19.5	21.6	43.5	54.0
CASS [4]	RGB-D	✓	✓		77.7	-	-	23.5	-	58.0
CR-Net [30]	RGB-D	✓	✓		79.3	55.9	27.8	34.3	47.2	60.8
DualPoseNet [17]	RGB-D	✓	✓		79.8	62.2	29.3	35.9	50.0	66.8
SGPA [5]	RGB-D	✓	✓		80.1	61.9	35.9	39.6	61.3	70.7
CPS++ [18]	RGB	✓		✓	72.8	-	-	25.2	-	-
Ours	RGB	✓		✓	82.0	59.0	24.4	27.0	49.3	54.8
Ours	D	✓		✓	79.6	57.8	21.2	29.1	48.7	65.9
Ours	RGB-D	✓		✓	82.6	62.5	30.4	34.8	56.9	66.0

Table 1. Quantitative comparison with state-of-the art methods on the REAL275 dataset. Empty entries either could not be evaluated or were not reported in the original paper.

Method	Syn	Real	mAP (\uparrow)							
		w/o Label	3D ₂₅	3D ₅₀	3D ₇₅	5°2cm	5°5cm	10°2cm	10°5cm	10°10cm
CPS++ (RGB)	✓		84.5	72.6	-	-	25.8	-	-	55.4
CPS++ (RGB)	✓	✓	84.6 (+0.1)	72.8 (+0.2)	-	-	25.2 (-0.6)	-	-	58.6 (+3.2)
Ours (RGB)	✓		83.3	79.9	49.7	15.4	18.3	37.6	46.7	48.9
Ours (RGB)	✓	✓	83.8 (+0.5)	82.0 (+2.1)	59.0 (+9.3)	24.4 (+9.0)	27.0 (+8.7)	49.3 (+11.7)	54.8 (+8.1)	56.9 (+8.0)
Ours (RGB-D)	✓	✓	84.0 (+0.7)	82.6 (+2.7)	62.5 (+12.8)	30.4 (+15.0)	34.8 (+16.5)	56.9 (+19.3)	66.0 (+19.3)	68.3 (+19.4)

Table 2. Quantitative comparison of unsupervised pose estimation approaches on the REAL275 dataset. Empty entries are either not able to be evaluated or not reported in the original paper. Performance margins are calculated compared to the synthetic-only results.

Supervised Pose Estimation methods. Table 1 summarizes the results of the state-of-the-art category-level object pose estimation methods. Obviously, supervised training with the real data annotation significantly improved the overall performance, as are revealed by comparing the results of NOCS [29] and SPD [25] on different training dataset conditions. However, our unsupervised

method showed results superior to NOCS [29], SPD [25], CASS [4], and CR-Net [30]. Compared to two of the strongest previous approaches, SGPA [5] and DualPoseNet [17], ours still showed comparable performance. This indicates that our proposed filtered pseudo label based UDA-COPE is robust when estimating object pose in unseen real-world instances.

Unsupervised Pose Estimation methods. Table 2 summarizes the results of CPS++ and our method on source only, and source with unlabeled target training conditions. CPS++ [18] provides self-supervision by computing the consistency between the observed depth map and the rendered depth. The rendered depth is obtained by projecting an estimated 3D shape with the predicted pose. The results from row 1 and row 2 in Table 2 show that for CPS++, using unlabeled real data marginally improved performance, and sometimes even worsened it, as in $5^\circ, 5\text{cm}$ metric. We believe that their self-supervision is unreliable because of ambiguous 3D shape reconstruction using only a single-view RGB image.

Comparing row 3 and row 4, it can be seen that our proposed method shows improved results for every metrics, with some metrics showing notable margins such as an 8.7 mAP (48%) increase in $5^\circ, 5\text{cm}$. Also, in the last row, our RGB-D result had better performance than the single modality based outputs. Therefore, we claim that our proposed algorithm is more effective by utilizing a pseudo label based learning scheme with modality and pose-aware self-supervision. The effectiveness of each components will be ablated thoroughly in the following sections.

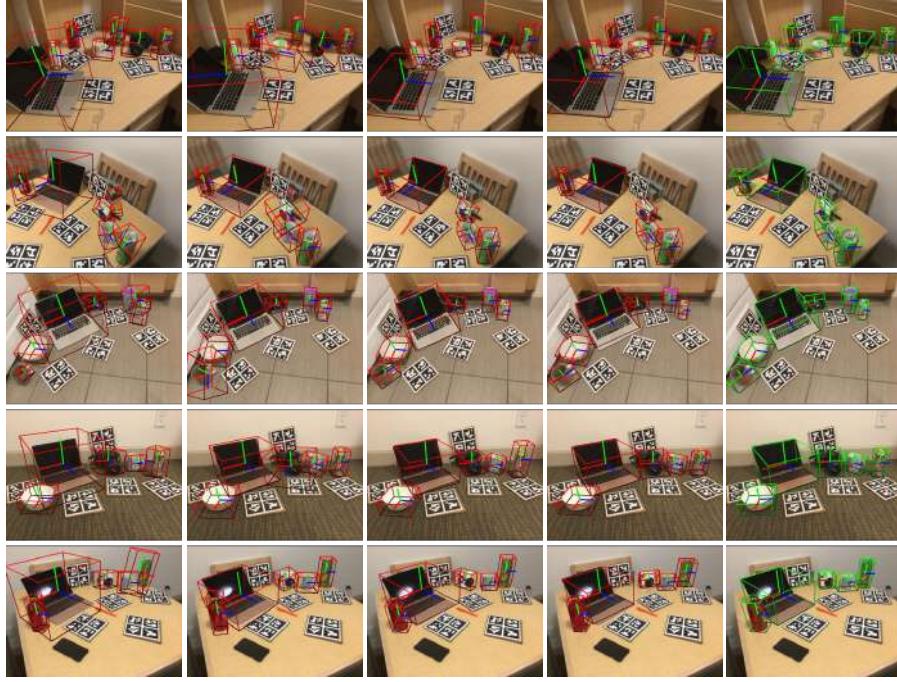
3.2 Qualitative Results

Fig. 3 shows qualitative results on the REAL275 dataset. We compare our results with some of the supervised methods, NOCS [29], SPD [25] and DualPoseNet [17]. Our method estimated pose and sizes more accurately than NOCS and SPD, especially on cameras and laptops. Compared to the state-of-the-art approach, DualPoseNet, ours exhibited comparable predictions, although it was not trained with the GT labels of the real dataset.

Fig. 4 visualizes some examples of the real training set with GT labels and our pseudo labels. 6D poses were obtained and visualized using the Umeyama algorithm [27], using GT NOCS map and our pseudo label NOCS map.

4 Conclusions

We propose UDA-COPE, unsupervised domain adaptation for category-level object pose estimation which addresses the real-world lack-of-label problem using multi-modality (RGB-D). Specifically, we designed a bidirectional point filtering method to filter noisy pseudo labels, and observed depth points, where the filtered depth points improve the robustness of the teacher network, and the filtered pseudo label helps efficient student network training. Both provide for better domain adaptation with real-world pose estimation. Experiments showed that our proposed pipeline and pose-aware point filtering results were comparable to or sometimes better than the performance of fully supervised approaches.



(a) NOCS [29] (b) SPD [25] (c) Dual [17] (d) Ours (e) GT

Fig. 3. Qualitative comparison on the REAL275 dataset. Compared to two of the strongest supervised approaches, SGPA [5] and DualPoseNet [17], ours still showed comparable performance even in unsupervised settings. This indicates that our proposed filtered pseudo label-based UDA-COPE is robust when estimating object pose in unseen real-world instances.

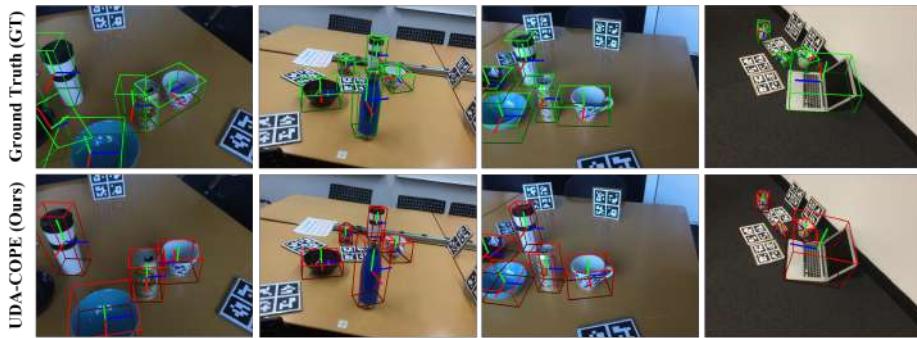


Fig. 4. Noisy GT label examples of the Real training dataset. Human-annotated GT pose labels on the real dataset (top row) are sometimes more inaccurate than our predicted pseudo labels (bottom row). The real data annotations were mainly performed automatically using aruco markers. For some of the failure cases, additional ICP or manual human annotations were needed. Therefore, frames with inaccurate labels exist, which might disrupt supervised training.

Remark

This paper is a re-publishing (summary presentation) of the paper, which has been published in CVPR 2022 by request of the IW-FCV2023 program committee to share the research results.

References

1. Araslanov, N., Roth, S.: Self-supervised augmentation consistency for adapting semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 15384–15394 (2021)
2. Bousmalis, K., Irpan, A., Wohlhart, P., Bai, Y., Kelcey, M., Kalakrishnan, M., Downs, L., Ibarz, J., Pastor, P., Konolige, K., et al.: Using simulation and domain adaptation to improve efficiency of deep robotic grasping. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 4243–4250 (2018)
3. Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., Krishnan, D.: Unsupervised pixel-level domain adaptation with generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3722–3731 (2017)
4. Chen, D., Li, J., Wang, Z., Xu, K.: Learning canonical shape space for category-level 6d object pose and size estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11973–11982 (2020)
5. Chen, K., Dou, Q.: Sgpa: Structure-guided prior adaptation for category-level 6d object pose estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 2773–2782 (2021)
6. Choy, C., Gwak, J., Savarese, S.: 4d spatio-temporal convnets: Minkowski convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3075–3084 (2019)
7. Du, G., Wang, K., Lian, S., Zhao, K.: Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review. *Artificial Intelligence Review* **54**(3), 1677–1734 (2021)
8. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* **24**(6), 381–395 (1981)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016)
10. He, Y., Huang, H., Fan, H., Chen, Q., Sun, J.: Ffb6d: A full flow bidirectional fusion network for 6d pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3003–3013 (2021)
11. He, Y., Sun, W., Huang, H., Liu, J., Fan, H., Sun, J.: Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11632–11641 (2020)
12. Jaritz, M., Vu, T.H., Charette, R.d., Wirbel, E., Pérez, P.: xmuda: Cross-modal unsupervised domain adaptation for 3d semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 12605–12614 (2020)

13. Kehl, W., Manhardt, F., Tombari, F., Ilic, S., Navab, N.: Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 1521–1529 (2017)
14. Lee, H.Y., Tseng, H.Y., Huang, J.B., Singh, M., Yang, M.H.: Diverse image-to-image translation via disentangled representations. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 35–51 (2018)
15. Lee, T., Lee, B.U., Kim, M., Kweon, I.S.: Category-level metric scale object shape and pose estimation. IEEE Robotics and Automation Letters (RA-L) **6**(4), 8575–8582 (2021)
16. Li, Y., Yuan, L., Vasconcelos, N.: Bidirectional learning for domain adaptation of semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6936–6945 (2019)
17. Lin, J., Wei, Z., Li, Z., Xu, S., Jia, K., Li, Y.: Dualposenet: Category-level 6d object pose and size estimation using dual pose network with refined learning of pose consistency. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 3560–3569 (2021)
18. Manhardt, F., Wang, G., Busam, B., Nickel, M., Meier, S., Minciullo, L., Ji, X., Navab, N.: Cps++: Improving class-level 6d pose and shape estimation from monocular images with self-supervised learning. arXiv preprint arXiv:2003.05848 (2020)
19. Marchand, E., Uchiyama, H., Spindler, F.: Pose estimation for augmented reality: a hands-on survey. IEEE Transactions on Visualization and Computer Graphics (TVCG) **22**(12), 2633–2651 (2015)
20. Marder-Eppstein, E.: Project tango. In: ACM SIGGRAPH 2016 Real-Time Live!, pp. 25–25 (2016)
21. Park, K., Patten, T., Vincze, M.: Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 7668–7677 (2019)
22. Peng, S., Liu, Y., Huang, Q., Zhou, X., Bao, H.: Pvnet: Pixel-wise voting network for 6dof pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4561–4570 (2019)
23. Runz, M., Buffier, M., Agapito, L.: Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects. In: IEEE International Symposium on Mixed and Augmented Reality (ISMAR). pp. 10–20 (2018)
24. Sundermeyer, M., Marton, Z.C., Durner, M., Brucker, M., Triebel, R.: Implicit 3d orientation learning for 6d object detection from rgb images. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 699–715 (2018)
25. Tian, M., Ang Jr, M.H., Lee, G.H.: Shape prior deformation for categorical 6d object pose and size estimation. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)
26. Tremblay, J., To, T., Sundaralingam, B., Xiang, Y., Fox, D., Birchfield, S.: Deep object pose estimation for semantic robotic grasping of household objects. In: Conference on Robot Learning (CoRL) (2018), <https://arxiv.org/abs/1809.10790>
27. Umeyama, S.: Least-squares estimation of transformation parameters between two point patterns. IEEE Transactions on Pattern Analysis & Machine Intelligence (TPAMI) **13**(04), 376–380 (1991)
28. Wang, C., Xu, D., Zhu, Y., Martín-Martín, R., Lu, C., Fei-Fei, L., Savarese, S.: Densefusion: 6d object pose estimation by iterative dense fusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3343–3352 (2019)

29. Wang, H., Sridhar, S., Huang, J., Valentin, J., Song, S., Guibas, L.J.: Normalized object coordinate space for category-level 6d object pose and size estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2642–2651 (2019)
30. Wang, J., Chen, K., Dou, Q.: Category-level 6d object pose estimation via cascaded relation and recurrent reconstruction networks. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2021)
31. Wong, J.M., Kee, V., Le, T., Wagner, S., Mariottini, G.L., Schneider, A., Hamilton, L., Chipalkatty, R., Hebert, M., Johnson, D.M., et al.: Segicp: Integrated deep semantic segmentation and pose estimation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 5784–5789 (2017)
32. Xiang, Y., Schmidt, T., Narayanan, V., Fox, D.: Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. Robotics: Science and Systems (RSS) (2018)
33. Zeng, A., Yu, K.T., Song, S., Suo, D., Walker, E., Rodriguez, A., Xiao, J.: Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 1386–1383 (2017)
34. Zhang, P., Zhang, B., Zhang, T., Chen, D., Wang, Y., Wen, F.: Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 12414–12424 (2021)
35. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2881–2890 (2017)
36. Zou, Y., Yu, Z., Liu, X., Kumar, B., Wang, J.: Confidence regularized self-training. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 5982–5991 (2019)

Dynamic Circular Convolution for Image Classification

Xuan-Thuy Vo, Duy-Linh Nguyen, Adri Priadana, and Kang-Hyun Jo

Department of Electrical, Electronic and Computer Engineering,
University of Ulsan, Ulsan (44610), South Korea

Email: xthuy@islab.ulsan.ac.kr;
{ndlinh301,priadana3202}@mail.ulsan.ac.kr; acejo@ulsan.ac.kr

Abstract. In recent years, Vision Transformer (ViT) has achieved an outstanding landmark in disentangling diverse information of visual inputs, superseding traditional Convolutional Neural Networks (CNNs). Although CNNs have strong inductive biases such as translation equivariance and relative positions, they require deep layers to model long-range dependencies in input data. This strategy results in high model complexity. Compared to CNNs, ViT can extract global features even in earlier layers through token-to-token interactions without considering geometric location of pixels. Therefore, ViT models are data-efficient and data-hungry, in another work, learning data-dependent and producing high performances on large-scale datasets. Nonetheless, ViT has quadratic complexity with the length of the input token because of the natural dot product between query and key matrices. Different from ViTs-and-CNNs-based models, this paper proposes a Dynamic Circular Convolution Network (DCCNet) that learns token-to-token interactions in Fourier domain, relaxing model complexity to $O(N \log N)$ instead of $O(N^2)$ in ViTs, and global Fourier filters are treated dependently and dynamically rather than independent and static weights in conventional operators. The token features, dynamic filters in spatial domain are transformed to frequency domain via Fast Fourier Transform (FFT). Dynamic circular convolution, in lieu of matrix multiplication in Fourier domain, between Fourier features and transformed filters are performed in a separable way along channel dimension. The output of circular convolution is revered back to spatial domain by Inverse Fast Fourier Transform (IFFT). Extensive experiments are conducted and evaluated on large-scaled dataset ImageNet1k and small dataset CIFAR100. On ImageNet1k, the proposed model achieves 75.4% top-1 accuracy and 92.6% top-5 accuracy with the budget 7.5M parameters under similar setting with ViT-based models, surpassing ViT and its variants. When fine-tuning the model on smaller dataset, DCCNet still works well and gets the state-of-the-art performances. Both evaluating the model on large and small datasets verifies the effectiveness and generalization capabilities of the proposed method.

Keywords: Vision Transformer · Dynamic Global Weights · Fourier Transform · Image Classification

1 Introduction

In the view of understanding involved visual data, the model compresses high dimension of image data to lower spaces and keeps informative features through processing layer-by-layer of the model. The way the model compresses and extracts the features relies on what the image encompasses. As we interpret datas, one point in the image contains two components: content (intensity values) $c \in \mathbb{R}^3$ and geometric information $w \in \mathbb{R}^2$. The image is interpreted as $I \in \mathbb{R}^{5 \times N}$, where $N = H \times W$ is number of pixels in the image. With the formulation of convolution, CNNs aggregate information of local windows to the center of the local windows in the sliding manner and also capture the relative position w_{i-j} inside local window. General speaking, CNN models [8, 13, 20] can extract helpful features that the image contains and result in translation equivariance and locality. Otherwise, Transformer invented by [32] views a sentence as a sequence of words (tokens) and compute word-to-word relationship and dynamically aggregate these features by global multi-head self-attention blocks for machine translation. With the success of Transformer in both general modeling capabilities and scalable models, ViT [5] tries to adapt self-attention operation in computer vision. Each image is separated into a sequence of patches (tokens), and the model learns an affinity matrix of token-to-token similarity. The ViT only considers content-to-content relationships from the input images or input features and can fail to capture positional information. The lack of geometric w_{i-j} results in weak inductive biases. The model needs a lot of data to compensate for the absence of w_{i-j} .

In terms of model complexity, the convolution operation is more efficient than the self-attention block. To extract global features, CNN-based models stack a series of convolution layers with residual connections that create a large computational cost. At the heart of Transformer, self-attention operation requires quadratic complexity with the lengths of input tokens and the model is not acceptable to adapt self-attention operation at earlier layers. Especially for down-stream tasks, these networks perform predictions on the input features with high resolution. With the bottleneck computation of ViT, many methods try to reduce the cost $O(N^2)$ to $O(N)$ [22], sub-sample the query, key, and value matrices [33, 34], and compute attention in local windows mimicking convolution [18, 19]. Another line of research is to enhance the weak inductive biases of the transformer. The affinity matrix is supplemented with positional information such as absolute positional embedding [32], relative positional embedding [2, 4, 19, 23]. Other works [14, 15, 21, 22] attempt to combine the strengths of convolution and self-attention operations to build hybrid networks. They inherit the strong inductive biases of CNNs and the strong modeling of ViTs, and deliver better performance than pure CNNs and ViTs.

On the research trend of Transformer, this paper develops a new operator, dubbed Dynamic Circular Convolution (DCC), which can extract and aggregate global features by performing the circular convolution between reweighted global Fourier kernels and Fourier transformed features. The reweighting coefficients are generated conditioned on the input features and are dynamically adopted

according to the content of the input. The DCC layers are used to replace self-attention blocks in ViT, called DCCNet. Our proposed DCCNet brings four benefits: (1) Global features are extracted in one layer; (2) the content and geometric information of the input image are utilized when computing circular convolution; (3) the generated weights are input-dependent instead of input-independent in conventional convolution; and (4) the complexity is $O(N(\log N))$ rather than $O(N^2)$ in Transformer.

To verify the effectiveness of the proposed method, we conduct the experiments on the large dataset Imagenet1k, and small dataset CIFAR100. As a result, the DCCNet surpasses the baseline ViT and its variant by a clear margin under the same setting and budget (7.5M parameters and 1.2 GFLOPs).

2 Related Works

In this section, we briefly review some related works about Convolutional Neural Networks, Vision Transformer and its variant, and Fourier transform in computer vision.

CNNs: In 2012, with the development of parallel hardware computation, AlexNet [13] successfully train the convolution networks on large datasets and open new directions in Computer Vision. VGG [28] enlarge the network depth by stacking a sequence of plain 3×3 convolutions. Even though VGGNet achieves the large improvement on large-scale ImageNet dataset, the model causes vanishing gradient problem when the depth beyond 19 layers. ResNet [8] proposes residual blocks that can eliminate vanishing gradient and number of layers are stacked up to 1000 layers. From that event, many works are introduced to improve the baseline ResNet such as dense connection [11], deformable convolution [3], depthwise seperable convolution [10, 27], and multiple branches [30].

ViT: Recently, ViT [5] integrated the original Transformer [32] developed for natural language processing and established new state-of-the-art performances on image classification and downstream tasks. Because ViT has a simple structure and uniform representation, there are a lot of works that improve ViT model in both learning and cost. PVT [33] builds a multi-scale vision transformer network that gradually decreases spatial dimensions across stages. On each stage of PVT, key and value matrices are down-sampled to smaller token sizes. Instead of computing attention from all tokens, Swin [19] models local attention in pre-defined windows and also constructs hierarchical networks inspired by CNNs-based models. With these insightful properties, Swin outperforms the strong baseline ResNet [8] and sets new records in detection, segmentation and tracking performance. MobileViTv2 [22] proposes a separable self-attention operation that reduces the cost of original self-attention from $O(N)^2$ to $O(N)$.

Compensation for weak inductive biases of self-attention operation, methods [4, 18, 19, 23] integrates relative positional information to attention maps. CPE [2] introduces a conditional positional encoding based on local relative neighborhood

of 3×3 depthwise convolution. Rather than marrying convolution operations to Transformer models, MobileViT [21] embeds Transformer blocks to stage 3, 4, 5 of MobileNetv2 [27]. Similar paradigm, NextViT [14] designs a hybrid network for embedded devices based on integration of the group convolution blocks in earlier stages and original self-attention blocks in later stages. EfficientFormer [15] adapt the idea of PoolFormer [38] and MetaFormer [39] and neural architecture search for constrained devices.

Based on the intuitive designs of Transformer and its variants, HorNet [24] extends matrix multiplication of self-attention operation to high-order interactions based on depth-wise separable convolution and recursive gates. FocalNet [37] uses multi-scale depth-wise separable convolutions and gated aggregation at each convolution to output multiple modulations.

Fourier Transform: FFC [1] proposes fast Fourier convolution and independently applies convolution and ReLU activation functions on the real and imaginary input features. Lama [29] adapt FFC operation to image inpainting. GFNet [25] learns global features in Fourier domain based on circular convolution and ViT models. AFNO [6] separates complex tensors into real and imaginary parts and utilizes the MLP module to mix these two parts. In this paper, we extend the circular convolution in GFNet to be dynamic and efficient. In GFNet, complex features and global filters are multiplied independently on each channel. Therefore, there is a way the model can efficiently learn the feature on both the spatial and channel axes. Moreover, in our core operator, both real and imaginary parts of the complex tensors are learned together instead of separation in the AFNO network.

3 Methodology

In this section, we leverage the overall network of the ViT [5] into our DCCNet in subsection 3.1 and analysis the proposed dynamic circular convolution block in subsection 3.2.

3.1 Overall Architecture

The DCCNet follows the single-scale architecture of the original ViT [5], shown in Fig. 1. Given input image with dimension $I \in \mathbb{R}^{3 \times H \times W}$, Patch Embedding splits and flattens the image I into a sequence of tokens with size $d_P \times N$, where N is the number of the tokens¹, H and W are height and width of image. Specifically, we use patch sizes of $P \times P$ and strides with patch window value over the image to produce the total tokens $N = \frac{H \cdot W}{P^2}$ and $d_N = 3 * P^2$. Followed by non-overlap processing of the ViT implementation, 16×16 convolution with stride 16 is used in Patch Embedding as patch generation, corresponding to each token with size 16×16 .

¹ Consistent term with original Transformer [32], also called number of patches

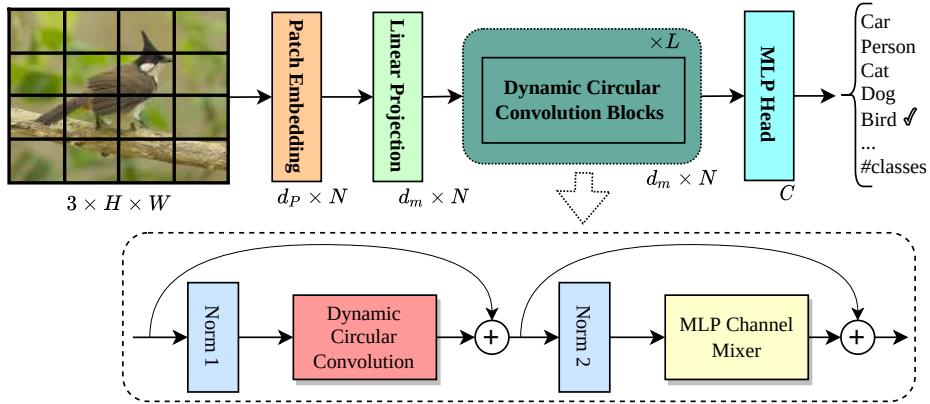


Fig. 1. The overall architecture of the DCCNet. N indicates the number of tokens with channel dimension d_m and L is the number of stacked DCC blocks. d_P , d_m are channel dimension after patch embedding, and channel dimension of the model. C is the number of predefined classes.

Linear Projection module projects a sequence of tokens with channel dimension d_P to a sequence of tokens with d_m . We use Linear layer to perform this process. The Dynamic Circular Convolution (DCC) block learns the token-token interaction that results in long-range dependencies between tokens. The DCC block includes two processes: (1) spatial mixings are performed by Dynamic Circular Convolution, and (2) MLP Channel Mixer mix token information along channel dimension. Between two processes, residual connections are used according to [38, 39] and each token is normalized by Layer Normalization before forwarding to each module. The detailed analysis of the DCC block is described in subsection 3.2. MLP Mixer contains two linear layers with expansion rate r . During training, based on [5, 25], we set $r = 4$ for all blocks.

Finally, Global Average Pooling (GAP) in MLP Head flattens a set of tokens to 1D dimension d_m and one linear layer projects flatten token d_m to number of classes C .

3.2 Dynamic Circular Convolution

The pipeline of the DCC operation is described in Fig. 2. Given the input feature $X \in \mathbb{R}^{d_m \times N}$, we reshape and permute the input X to 2D dimension $d_m \times H_P \times W_P$. Hence, the order of pixels in the input feature is still preserved. The permuted input features are processed in three steps: (1) 2D FFT (Fast Fourier Transform) transforms X in spatial domain to frequency domain by Fast Fourier Transform [1]; circular convolution is performed between transformed tensor and dynamic kernels to model global features; and 2D IFFT (Inverse Fast Fourier Transform) reserves dynamic and global tensor back to spatial domain.

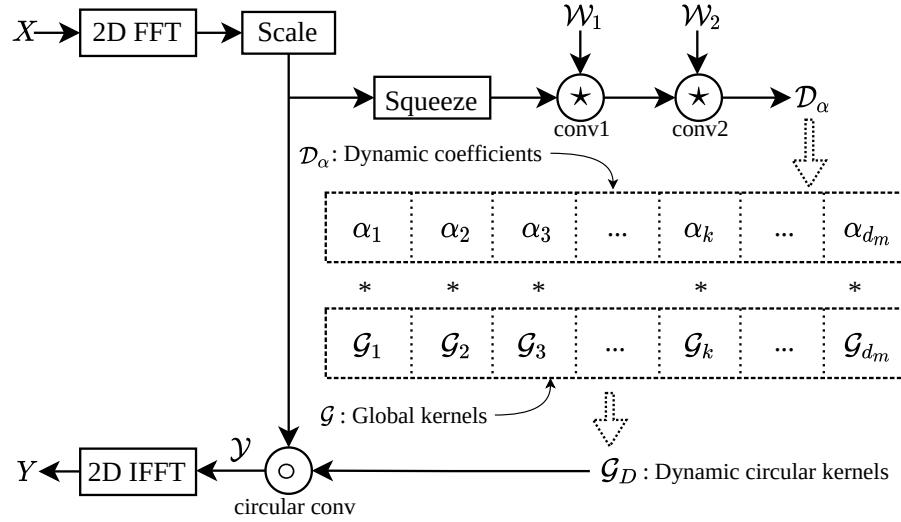


Fig. 2. The detailed architecture of the Dynamic Circular Convolution (DCC). 2D FFT and 2D IFFT denote Fast Fourier Transform and Inverse Fast Fourier Transform. \mathcal{W}_1 , \mathcal{W}_2 are learnable parameters in frequency domain. Squeeze denotes mean computation along spatial dimension.

Given the permuted input with dimension $d_m \times H_P \times W_P$, complex tensor is generated by 2D FFT as follows,

$$\mathcal{X}[:, u, v] = \mathcal{F}(X) = \sum_{m=0}^{H_P-1} \sum_{n=0}^{W_P-1} X[:, m, n] e^{-j2\pi(\frac{um}{H_P} + \frac{vn}{W_P})}, \quad (1)$$

where $\mathcal{X}[:,]$ is used to get the index of the channel. u, v are the coordinate of each output complex values $\mathcal{X} \in \mathbb{C}^{d_m \times H_P \times W_P}$ and m, n are the coordinate of each input real values $X \in \mathbb{R}^{d_m \times H_P \times W_P}$. $H_P = \frac{H}{P}$, $W_P = \frac{W}{P}$ are the height and width of the permuted sequence of tokens. Conventionally, angular frequencies along height and width dimensions are computed as:

$$\omega_h = 2\pi f_h = 2\pi \frac{u}{H_P}, \quad (2)$$

$$\omega_w = 2\pi f_w = 2\pi \frac{v}{W_P}. \quad (3)$$

In equation 1, there is a one-to-one mapping from the real domain to the frequency domain. It means that we convert the non-periodic signal to a periodic signal based on the theorem of the Fourier transform and fully preserve all the information of the input. The image can be decomposed into a function of *sine* waves.

One of the insightful property of Fourier transform is that there has a conjugate symmetry of the complex tensor \mathcal{X} and leveraging such property can reduce the

model complexity without losing information [1]. This view can be represented as:

$$\mathcal{X}[:, u, v] = \mathcal{X}^*[:, H_P - u, W_P - v]. \quad (4)$$

Therefore, the model complexity is $O(H_P W_P \log(H_P W_P))$ with respect to the length of the input tokens. While ViT-based models have quadratic complexity with the length of the input tokens, we enjoy much lower computational costs. A half of complex tensor $\mathcal{X}_s = \mathcal{X}[:, :, 0 : W_P/2 + 1]$ need to be computed and restored. It can relax memory intensive and still extract global features. Inside the equation 1, since the *sum* operation is used, the *scale* step is proposed to normalize all the accumulated values. During implementation, *scale* is conducted by average operation.

The model learns global features through self-attention operations or large kernel sizes. In this paper, we employ matrix multiplication between complex tensors \mathcal{X} and global kernels. These global kernels are the same size as the scaled input $\mathcal{X}_s \in \mathbb{C}^{d_m \times H_P \times W_P/2+1}$. Hence, matrix multiplication between them in the spatial domain is called circular convolution in the frequency domain. In GFNet [25], they treat global kernels independently and statically because circular convolution is separable. It leads to a way that can mix the information of the input tensor along the channel dimension. Inspired by weight generation of self-attention operation [32], we define dynamic coefficients $\mathcal{D}_\alpha \in \mathbb{C}^{d_m \times 1 \times 1}$ as follows:

$$\mathcal{D}_\alpha = \{\alpha_1, \dots, \alpha_{d_m}\} = \mathcal{W}_2 \star (\mathcal{W}_1 \star f(\mathcal{X}_s)), \quad (5)$$

where \star is convolution operation. $f(\cdot)$ indicates squeeze function that converts 2D input \mathcal{X}_s to 1D vector. $\mathcal{W}_1 \in \mathbb{C}^{d_m \times \frac{d_m}{r}}$ and $\mathcal{W}_2 \in \mathbb{C}^{\frac{d_m}{r} \times d_m}$ are linear transformations in the frequency domain, mixing information of the squeezed complex tensor. Then, the dynamic coefficients \mathcal{D}_α is used to redistribute the static global kernel $\mathcal{G} \in \mathbb{C}^{d_m \times H \times (W/2+1)}$ via element-wise matrix multiplication,

$$\mathcal{G}_D = \{\alpha_i * \mathcal{G}_i | \alpha_i \in \mathbb{C}; \mathcal{G}_i \in \mathbb{C}^{H \times (W/2+1)}\} \in \mathbb{C}^{d_m \times H \times (W/2+1)}, \quad (6)$$

The dynamic circular kernel \mathcal{G}_D is convoluted with the scaled input \mathcal{X}_s to output global receptive field,

$$\mathcal{Y} = \mathcal{X}_s \circ \mathcal{G}_D, \quad (7)$$

where $\mathcal{Y} \in \mathbb{C}^{d_m \times H \times (W/2+1)}$ is the output of the circular convolution and \circ denotes circular convolution.

Finally, we reserve the Fourier feature back to the spatial domain using the 2D Inverse Fast Fourier Transform (IFFT) and this operation is addressed as follows:

$$Y[:, m, n] = \mathcal{F}^{-1}(\mathcal{Y}) = \frac{1}{N} \sum_u^{H-1} \sum_v^{W-1} \mathcal{Y}[:, u, v] e^{j2\pi(\frac{um}{H} + \frac{vn}{W})}, \quad (8)$$

where N is the number of tokens used for normalization.

Table 1. Comparison with state-of-the-art models on ImageNet validation set

Method	Top-1 Acc (%)	Top-5 Acc (%)	#params	GFLOPs
T2T-ViT-7 [40]	71.7	-	4.3M	1.2
DeiT-Ti [31]	72.2	91.1	5.7M	1.3
gMLP-Ti [17]	72.3	-	7.0M	1.3
PiT-Ti [9]	73.0	-	4.9M	0.71
TNT-Ti [7]	73.9	91.9	6.1M	1.4
GFNet-Ti [25]	74.6	92.2	7.5M	1.3
LocalViT-T [16]	74.8	92.6	5.9M	1.3
ViTAE [36]	75.3	92.7	4.8M	1.5
DCCNet (our)	75.5	92.7	7.7M	1.2

4 Experiments and Results

4.1 Experiments

Datasets: The proposed DCCNet is trained and evaluated on the large-scale dataset ImageNet1k [26], and the small dataset CIFAR100 [12]. For ImageNet1k, this dataset includes 1.2M training images and 50k validation images with 1000 categories. CIFAR10 contains 50k training and 10k testing images from 10 classes. Like CIFAR10, CIFAR100 contains 50k training and 10k testing images with 100 classes.

Experimental Setup: All implementations are conducted using the Pytorch framework, and the codebase is *Timm* [35] for fair comparisons with other methods. We follow the setting of methods [5, 25]. The model is trained for 300 epochs on two Tesla V100 GPUs. The batch size is 512 images per GPU, and the input images are resized to 224×224 . The basic learning rate is $5 \times e^{-4}$ and learning schedule is cosine with warmup epochs of 5. The optimizer is AdamW with momentum 0.9 and weight decay 0.05. The DCCNet does not use EMA model and strong data augmentation like [19, 20].

4.2 Results

ImageNet dataset: Table 1 shows the main results evaluated on ImageNet validation set between DCCNet and other methods. As a result, DCCNet achieves 75.5% Top-1 accuracy and 92.2% Top-5 accuracy, which surpasses state-of-the-art ViT-based models around 7M parameters and 1.2 GFLOPs, such as 71.7% Top-1 in T2T [40], 72.2% in DeiT [31], 72.3% Top-1 in gMLP [17], 73.0% Top-1 in PiT [9], 73.9% Top-1 in TNT [7], 74.6% Top-1 in GFNet, 74.8% Top-1 in LocalViT [16], and 75.3% Top-1 in ViTAE [36]. This comparison verifies the effectiveness of the proposed DCCNet.

CIFAR 100: Table 2 describes the comparison between the DCCNet and other methods on the small dataset CIFAR100. With largely smaller parameters and GFLOPs than other methods, the DCCNet gets 84.1% Top-1 accuracy under budget 7.5M parameters and 1.2 GFLOPs.

Table 2. Results on small dataset CIFAR 100

Method	Top-1 Acc	#params	#GFLOPs
DeiT-T [31]	67.59	5.3M	0.4
PVT-T [33]	69.62	15.8	0.6
Swin-T [19]	78.07	27.5M	1.4
DCCNet (ours)	84.10	7.5	1.2

Ablation study: We investigate the effect of reduction ratio $r \in \{8, 16, 32\}$ in linear matrices of the DCC block on the model performance and cost illustrated in Table 3. When changing the reduction r , the Top-1 performances are similar. For a trade-off between accuracy and cost, we select $r = 16$ for all experiments.

Table 3. Ablation study on the reduction ratio r

Reduction r	Top-1 Acc (%)	Top-5 (Acc%)	#params	GFLOPs
8	75.4	92.7	7.9	1.3
16	75.5	92.7	7.7	1.2
32	75.2	92.4	7.6	1.2

Amplitude and Phase Spectrum: We visualize the amplitude and phase spectrum on Fig. 3. As we can see, the detailed patterns in the amplitude spectrum are clear, and its spectrum has the symmetric property demonstrated in [1].

5 Conclusion

This paper presents a feature extractor based on Fast Fourier Transform and dynamic weight generations, called DCCNet. All spatial operations, especially for circular convolution, are performed in the frequency domain through the FFT. Leveraging the conjugate symmetry of FFT can result in better performance and efficient model complexity. Instead of static weight in conventional circular convolution, this work dynamically produces complex weight matrices of circular convolution conditioned on the input features. And this operator also mixes the information of a complex weight tensor along the channel dimension. This channel mixing can complement circular convolution that is separable and input-independent. Experiments are conducted on both large and small datasets, and the DCCNet achieves better performance than other methods. It verifies the effectiveness of the proposed methods and its generalization capability.

Acknowledgement

This result was supported by “Region Innovation Strategy (RIS)” through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE)(2021RIS-003).

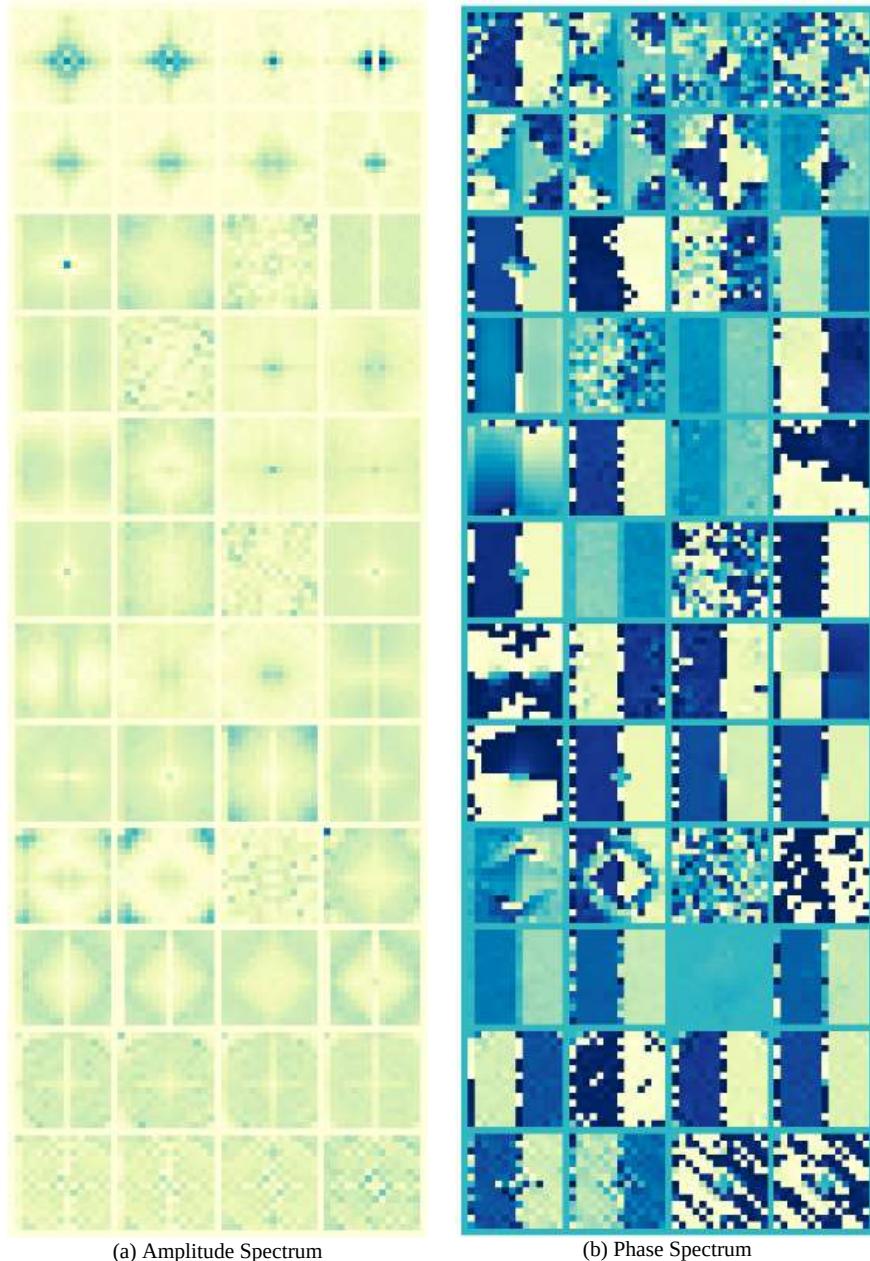


Fig. 3. The amplitude spectrum (a) and phase spectrum (b) of the dynamic circular convolution.

References

1. Chi, L., Jiang, B., Mu, Y.: Fast fourier convolution. *Advances in Neural Information Processing Systems* **33**, 4479–4488 (2020)
2. Chu, X., Tian, Z., Zhang, B., Wang, X., Wei, X., Xia, H., Shen, C.: Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882* (2021)
3. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: *Proceedings of the IEEE international conference on computer vision*. pp. 764–773 (2017)
4. Dai, Z., Liu, H., Le, Q.V., Tan, M.: Coatnet: Marrying convolution and attention for all data sizes. *Advances in Neural Information Processing Systems* **34**, 3965–3977 (2021)
5. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: *International Conference on Learning Representations* (2021), <https://openreview.net/forum?id=YicbFdNTTy>
6. Guibas, J., Mardani, M., Li, Z., Tao, A., Anandkumar, A., Catanzaro, B.: Efficient token mixing for transformers via adaptive fourier neural operators. In: *International Conference on Learning Representations* (2021)
7. Han, K., Xiao, A., Wu, E., Guo, J., Xu, C., Wang, Y.: Transformer in transformer. *Advances in Neural Information Processing Systems* **34**, 15908–15919 (2021)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 770–778 (2016)
9. Heo, B., Yun, S., Han, D., Chun, S., Choe, J., Oh, S.J.: Rethinking spatial dimensions of vision transformers. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 11936–11945 (2021)
10. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017)
11. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 4700–4708 (2017)
12. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
13. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. *Communications of the ACM* **60**(6), 84–90 (2017)
14. Li, J., Xia, X., Li, W., Li, H., Wang, X., Xiao, X., Wang, R., Zheng, M., Pan, X.: Next-vit: Next generation vision transformer for efficient deployment in realistic industrial scenarios. *arXiv preprint arXiv:2207.05501* (2022)
15. Li, Y., Yuan, G., Wen, Y., Hu, E., Evangelidis, G., Tulyakov, S., Wang, Y., Ren, J.: Efficientformer: Vision transformers at mobilenet speed. In: Oh, A.H., Agarwal, A., Belgrave, D., Cho, K. (eds.) *Advances in Neural Information Processing Systems* (2022), <https://openreview.net/forum?id=NXHXoYMLIG>
16. Li, Y., Zhang, K., Cao, J., Timofte, R., Van Gool, L.: Localvit: Bringing locality to vision transformers. *arXiv preprint arXiv:2104.05707* (2021)
17. Liu, H., Dai, Z., So, D., Le, Q.V.: Pay attention to mlps. *Advances in Neural Information Processing Systems* **34**, 9204–9215 (2021)

18. Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., et al.: Swin transformer v2: Scaling up capacity and resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12009–12019 (2022)
19. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10012–10022 (2021)
20. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11976–11986 (2022)
21. Mehta, S., Rastegari, M.: Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer. In: International Conference on Learning Representations (2022), <https://openreview.net/forum?id=vh-0sUt8H1G>
22. Mehta, S., Rastegari, M.: Separable self-attention for mobile vision transformers. arXiv preprint arXiv:2206.02680 (2022)
23. Min, J., Zhao, Y., Luo, C., Cho, M.: Peripheral vision transformer. In: Oh, A.H., Agarwal, A., Belgrave, D., Cho, K. (eds.) Advances in Neural Information Processing Systems (2022), <https://openreview.net/forum?id=nE8IJLT7nW->
24. Rao, Y., Zhao, W., Tang, Y., Zhou, J., Lim, S.L., Lu, J.: Hornet: Efficient high-order spatial interactions with recursive gated convolutions. Advances in Neural Information Processing Systems (NeurIPS) (2022)
25. Rao, Y., Zhao, W., Zhu, Z., Lu, J., Zhou, J.: Global filter networks for image classification. Advances in Neural Information Processing Systems **34**, 980–993 (2021)
26. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**(3), 211–252 (2015)
27. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4510–4520 (2018)
28. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
29. Suvorov, R., Logacheva, E., Mashikhin, A., Remizova, A., Ashukha, A., Silvestrov, A., Kong, N., Goka, H., Park, K., Lempitsky, V.: Resolution-robust large mask inpainting with fourier convolutions. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2149–2159 (2022)
30. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826 (2016)
31. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: International Conference on Machine Learning. pp. 10347–10357. PMLR (2021)
32. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
33. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 568–578 (2021)

34. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pvt v2: Improved baselines with pyramid vision transformer. Computational Visual Media **8**(3), 415–424 (2022)
35. Wightman, R.: Pytorch image models. <https://github.com/rwightman/pytorch-image-models> (2019). <https://doi.org/10.5281/zenodo.4414861>
36. Xu, Y., Zhang, Q., Zhang, J., Tao, D.: Vitae: Vision transformer advanced by exploring intrinsic inductive bias. Advances in Neural Information Processing Systems **34**, 28522–28535 (2021)
37. Yang, J., Li, C., Dai, X., Gao, J.: Focal modulation networks. In: Oh, A.H., Agarwal, A., Belgrave, D., Cho, K. (eds.) Advances in Neural Information Processing Systems (2022), <https://openreview.net/forum?id=ePhEbo0391>
38. Yu, W., Luo, M., Zhou, P., Si, C., Zhou, Y., Wang, X., Feng, J., Yan, S.: Metaformer is actually what you need for vision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10819–10829 (2022)
39. Yu, W., Si, C., Zhou, P., Luo, M., Zhou, Y., Feng, J., Yan, S., Wang, X.: Metaformer baselines for vision. arXiv preprint arXiv:2210.13452 (2022)
40. Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z.H., Tay, F.E., Feng, J., Yan, S.: Tokens-to-token vit: Training vision transformers from scratch on imagenet. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 558–567 (2021)

Task-specific Scene Structure Representations*

Seunghyun Shin, Jisu Shin, and Hae-Gon Jeon

AI Graduate School, GIST, South Korea
 {seunghyuns98, jsshin98}@gm.gist.ac.kr, haegonj@gist.ac.kr

Abstract. Understanding the informative structures of scenes is essential for low-level vision tasks. Unfortunately, it is difficult to obtain a concrete visual definition of the informative structures because influences of visual features are task-specific. In this paper, we propose a single general neural network architecture for extracting task-specific structure guidance for scenes. To do this, we unfold the traditional graph-partitioning problem into a learnable network, named *Scene Structure Guidance Network (SSGNet)*, to represent the task-specific informative structures. In addition, our SSGNet is light-weight ($\sim 55K$ parameters), and can be used as a plug-and-play module for off-the-shelf architectures. Our main contribution is to show that such a simple network can achieve state-of-the-art results for several low-level vision applications including joint upsampling and image denoising even on unseen datasets, compared to existing methods which use structural embedding frameworks.

Keywords: Low-level Vision · Structure Guidance · Unsupervised

1 Introduction

Methods for estimating scene structures have attracted wide research attention for the past several decades. Clearly, the goodness of scene structures depends on the target applications, and is defined by either training data or objective functions. Nevertheless, the question of how to effectively exploit structure guidance information remains unanswered.

In this paper, we propose a *Scene Structure Guidance Network (SSGNet)*, a single general neural network architecture for extracting task-specific structural features of scenes. Our SSGNet is lightweight in both size and computation, and is a plug-and-play module that can be applied to any baseline low-level vision architectures. The SSGNet computes a set of parameterized eigenvector maps, whose combination is selectively determined in favor of the target domain. To achieve this, we introduce two effective losses: (1) *Eigen loss*, motivated by the traditional graph partitioning problem [21], forms a basis set of scene structures based on weight graphs on an image grid. (2) *Spatial loss* enforces the sparsity of each eigenvector for diverse representations of scene structures. We note that, without any supervision, our SSGNet can successfully learn to generate

* This paper is the short version of AAAI'23 and is NEVER considered an official publication.

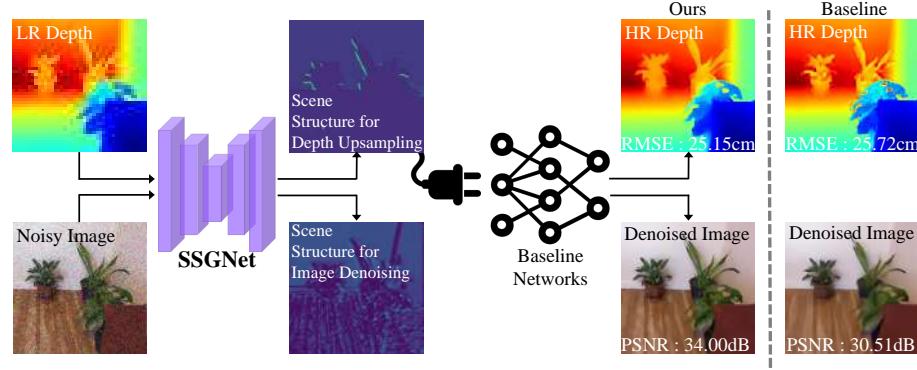


Fig. 1. Our SSGNet is a lightweight architecture and can be applied as a plug-and-play module to improve the performance of baseline networks for low-level vision tasks.

task-specific and informative structural information as shown in Figure 1. To demonstrate the wide applicability of our SSGNet, we conduct extensive experiments on several low-level vision applications, including joint upsampling and image denoising, and achieve state-of-the-art results, even in cross-dataset generalization.

2 Related Work

2.1 Low-level vision tasks

The goal of low-level vision tasks such as denoising, super-resolution, deblurring and inpainting is to recover a sharp latent image from an input image that has been degraded by the inherent limitations of the acquisition systems. In the past decade, there have been significant improvements in low-level vision tasks, and recently deep learning-based techniques have especially proven to be powerful systems.

With the help of inductive bias [4], convolutional neural networks (CNNs) with a pixel-wise photo consistency loss [16] are adopted. To mitigate the issue on inter-pixel consistency on CNNs, generative adversarial networks (GANs) [8]-based methods are proposed to produce visually pleasing results with perceptual losses [14] based on high-level semantic features. Nowadays, a vision transformer (ViT) [6] has been used to capture both local and global image information by leveraging the ability to model long-range context.

Such approaches have shown good progress with structural details. For regularization, adding robust penalties to objective functions [23] suppresses high-frequency components, and hence the results usually provide a smooth plausible reconstruction. However, those constraints often suffer from severe overfitting to noisy labels and are sensitive to hyperparameters, which leads to a lack of model generality.

2.2 Structural information

Extensive studies on low-level vision have verified the feasibility and necessity of the image prior including image edges and gradients. One of the representative works involves joint image filters which leverage a guidance image as a prior and transfer its structural details to a target image for edge-preserved smoothing [24, 11, 28].

Such structure information can be defined in practice, depending on the tasks. Both super-resolution [20, 22, 26, 7] and image denoising [17], which utilize a patch similarity, generate gradient maps to reconstruct high frequency details or suppress image noises. Works in [9, 13] infer object boundaries to refine initial predictions in visual perception tasks, including depth estimation/completion. Also, image inpainting [19, 27, 10, 2], filling in missing parts of corrupted scenes, adopt edge maps from traditional method like Canny edge detector [1] to hallucinate their own scene structures.

In spite of promising results from the state-of-the-art methods learning meaningful details for each task, they require a high modeling capacity with numerous parameters and ground-truth structure maps for training. In contrast, our SSGNet, a very small network generating scene structures without any supervision, has advantages for various low-level vision tasks, simply by embedding as an additional module.

3 Methodology

Motivation Spectral graph theory proves that the eigenvectors of the graph Laplacian yield minimum-energy graph partitions, and each smallest eigenvector partitions the graph into soft-segments based on its adjacent matrix.

In the image domain, a reference pixel and its similarity to neighboring pixels can be interpreted as a node and edges in a graph, respectively. In general, affinity is defined by appearance similarities (the absolute of intensity differences). With this motivation, images can be decomposed into soft image clusters from a pre-computed affinity matrix. In addition, scene configurations in images can be described as a set of eigenvectors whose smallest eigenvalues indicate connected components on the affinity matrix.

Scene Structure Guidance Network In this work, our goal is to train the proposed network, SSGNet, without any supervision because it is infeasible to define a unique objective function for a task-specific structure guidance. To accomplish this, we devise a learnable and parametric way of efficiently representing scene structures. The output of our SSGNet is associated with learnable weights that will be finetuned in accordance with an objective function of target applications. To optimize SSGNet in an unsupervised manner, we define a loss function \mathcal{L}_{ssg} which is a linear combination of two loss terms as follows:

Eigen Loss The main objective of SSGNet is to obtain a set of smallest eigenvectors \mathbf{Y} of the graph Laplacian \mathbf{L} . Since an image is segmented based on a constructed affinity matrix in spectral graph theory, the form of the matrix depends on the pixel-level similarity encoding. In this work, we adopt the sparse

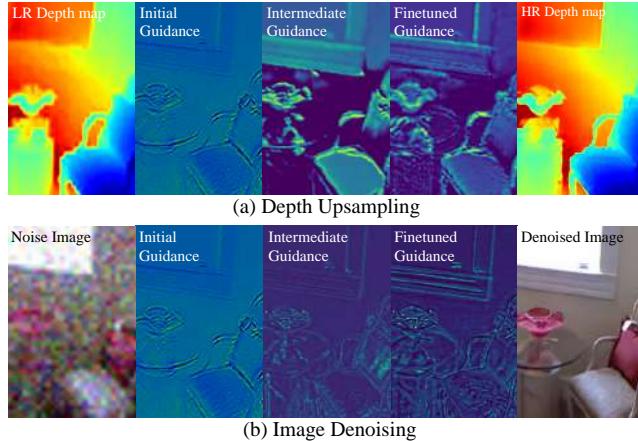


Fig. 2. Examples of task-specific scene structures: intial, intermediate and final results from SSGNet for (a) joint depth upsampling and (b) image denoising.

KNN-matting matrix [3], which collects nonlocal neighborhoods j of a pixel i by the k-nearest neighbor algorithm (KNN). Using the sparse KNN-matting matrix, we can take account of both spatial distance and color information with less computational cost than a traditional similarity matrix. The graph Laplacian \mathbf{L} is finally obtained by $\mathbf{L} = \mathbf{D} - \mathbf{W}$.

We can finally obtain a set of eigenvectors \mathbf{Y} by minimizing the quadratic form of \mathbf{L} , \mathcal{L}_{eigen} , as below:

$$\mathcal{L}_{eigen} = \sum_k \mathbf{Y}_k^T \mathbf{L} \mathbf{Y}_k. \quad (1)$$

Spatial Loss Our spatial loss $\mathcal{L}_{spatial}$ considers the sparsity of each eigenvector to enforce diverse representations of scene structure, defined as below:

$$\mathcal{L}_{spatial} = \sum_k (|\mathbf{Y}_k|^\gamma + |1 - \mathbf{Y}_k|^\gamma) - 1, \quad (2)$$

where $|\cdot|$ indicates an absolute value, and the hyperparameter γ is set to 0.9 in our implementation. With the $\mathcal{L}_{spatial}$ and the softmax operation together, it makes each pixel across the eigenvectors have different value due to the sparsity penalty, which produces diverse feature representations of image structures.

In total, the final loss function for SSGNet is defined as:

$$\mathcal{L}_{ssg} = \mathcal{L}_{eigen} + \lambda \mathcal{L}_{spatial} \quad (3)$$

where λ is the hyper-parameter, and is empirically set to 40.

Our SSGNet is pretrained on a single dataset and can be embedded in various baseline networks after passing through an additional single convolution layer which acts as an attention module. In favor of the target domain on each task,

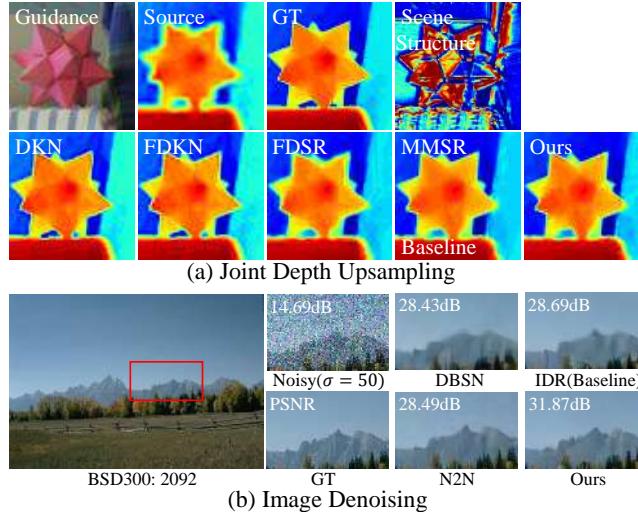


Fig. 3. Comparison results on (a) joint depth upsampling and (b) image denoising.

Dataset	Scale	Supervised				Self-Supervised	
		DKN[15]	FDKN[15]	FDSR[12]	MMSR[5]	Ours	
2014	$\times 4$	RMSE	2.878	2.593	3.217	<u>1.953</u>	1.819
		MAE	0.739	0.659	0.595	<u>0.573</u>	0.451
	$\times 8$	RMSE	3.642	3.510	3.606	<u>2.765</u>	2.714
		MAE	0.775	0.871	0.885	<u>0.785</u>	0.675

Table 1. Quantitative results on joint depth upsampling tasks. The best and the second best results are marked as **bold** and underlined, respectively. (unit:cm)

this layer produces adaptive structural information of input scenes by linearly combining the set of eigenvectors. In Figure 2, we visualize how the eigenvectors from SSGNet change differently at each iteration during finetuning on each task. We claim that it is possible for our SSGNet to capture informative and task-specific structures through gradient updates from backpropagation.

4 Experiments

We conduct a variety of experiments on low-level vision tasks, including self-supervised joint depth upsampling and unsupervised single image denoising, to demonstrate the effectiveness of our SSGNet. Prior to the evaluations, we train our SSGNet on a well-known NYUv2 dataset. With the pre-trained weight of SSGNet, we embed it to the baseline networks, which are existing CNN architectures, and finetune on each task.

As shown in Table 1, MMSR [5] with our SSGNet embedded achieves the best performance over the comparison methods. In addition, as shown in Table 2, IDR [29] with our SSGNet embedded achieves the best performance among all the competitive methods regardless of the noise levels. Figure 3 also shows some

Method	Kodak		BSD300		BSD68	
	$\sigma = 25$	$\sigma = 50$	$\sigma = 25$	$\sigma = 50$	$\sigma = 25$	$\sigma = 50$
DBSN[25]	PSNR	32.07	28.81	31.12	27.87	28.81
	SSIM	0.875	0.783	0.881	0.782	0.818
N2N[18]	PSNR	32.39	29.23	31.39	28.17	29.15
	SSIM	0.886	0.803	0.889	0.799	0.831
IDR[29]	PSNR	<u>32.36</u>	<u>29.27</u>	<u>31.48</u>	<u>28.25</u>	<u>29.20</u>
	SSIM	0.884	0.803	0.890	0.802	0.835
Ours	PSNR	32.39	29.34	31.52	28.33	29.25
	SSIM	0.885	0.806	0.891	0.805	0.835

Table 2. Quantitative results on single image denoising.

example results. We highlight that the result demonstrates the strong generalization capabilities of our SSGNet on unseen data again.

5 Conclusion

In this paper, we present a single general network for representing task-specific scene structures. We cast the problem of the acquisition of informative scene structures as a traditional graph partitioning problem on the image domain, and solve it using a lightweight CNN framework without any supervision, *Scene Structure Guidance Network (SSGNet)*. Our SSGNet computes coefficients of a set of eigenvectors, enabling to efficiently produce diverse feature representations of a scene with our proposed two loss terms, the eigen loss and the spatial loss. We show the promising performance gains for both the joint depth upsampling and image denoising, even with the good cross-dataset generalization capability. **Remark** This paper is a re-publishing of the paper which has been published in AAAI2023 by request of the IW-FCV2023 program committee to share the research results.

References

1. Canny, J.: A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* **8**(6), 679–698 (1986)
2. Cao, C., Fu, Y.: Learning a sketch tensor space for image inpainting of man-made scenes. In: Proceedings of International Conference on Computer Vision (ICCV) (2021)
3. Chen, Q., Li, D., Tang, C.K.: Knn matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* **35**(9), 2175–2188 (2013)
4. Cohen, N., Shashua, A.: Inductive bias of deep convolutional networks through pooling geometry. In: International Conference on Learning Representations (ICLR) (2017)
5. Dong, X., Yokoya, N., Wang, L., Uezato, T.: Learning mutual modulation for self-supervised cross-modal super-resolution. In: Proceedings of European Conference on Computer Vision (ECCV) (2022)
6. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (ICLR) (2021)
7. Fang, F., Li, J., Zeng, T.: Soft-edge assisted network for single image super-resolution. *IEEE Transactions on Image Processing (TIP)* **29**, 4656–4668 (2020)
8. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Proceedings of the Neural Information Processing Systems (NeurIPS) (2014)
9. Gu, S., Zuo, W., Guo, S., Chen, Y., Chen, C., Zhang, L.: Learning dynamic guidance for depth image enhancement. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
10. Guo, X., Yang, H., Huang, D.: Image inpainting via conditional texture and structure dual generation. In: Proceedings of International Conference on Computer Vision (ICCV) (2021)
11. He, K., Sun, J., Tang, X.: Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* **35**(6), 1397–1409 (2012)
12. He, L., Zhu, H., Li, F., Bai, H., Cong, R., Zhang, C., Lin, C., Liu, M., Zhao, Y.: Towards fast and accurate real-world depth super-resolution: Benchmark dataset and baseline. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
13. Jin, L., Xu, Y., Zheng, J., Zhang, J., Tang, R., Xu, S., Yu, J., Gao, S.: Geometric structure based and regularized depth estimation from 360 indoor imagery. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
14. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Proceedings of European Conference on Computer Vision (ECCV) (2016)
15. Kim, B., Ponce, J., Ham, B.: Deformable kernel networks for joint image filtering. *International Journal on Computer Vision (IJCV)* **129**(2), 579–600 (2021)
16. Li, Y., Huang, J.B., Ahuja, N., Yang, M.H.: Deep joint image filtering. In: Proceedings of European Conference on Computer Vision (ECCV) (2016)
17. Liu, Y., Anwar, S., Zheng, L., Tian, Q.: Gradnet image denoising. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW) (2020)

18. Moran, N., Schmidt, D., Zhong, Y., Coady, P.: Noisier2noise: Learning to denoise from unpaired noisy data. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
19. Nazeri, K., Ng, E., Joseph, T., Qureshi, F., Ebrahimi, M.: Edgeconnect: Structure guided image inpainting using edge prediction. In: Proceedings of International Conference on Computer Vision Workshop (ICCVW) (2019)
20. Pickup, L., Roberts, S.J., Zisserman, A.: A sampled texture prior for image super-resolution. In: Proceedings of the Neural Information Processing Systems (NeurIPS) (2003)
21. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* **22**(8), 888–905 (2000)
22. Sun, J., Xu, Z., Shum, H.Y.: Image super-resolution using gradient profile prior. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2008)
23. Tibshirani, R.: Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)* **58**(1), 267–288 (1996)
24. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Proceedings of International Conference on Computer Vision (ICCV) (1998)
25. Wu, X., Liu, M., Cao, Y., Ren, D., Zuo, W.: Unpaired learning of deep image denoising. In: Proceedings of European Conference on Computer Vision (ECCV) (2020)
26. Xie, J., Feris, R.S., Sun, M.T.: Edge-guided single depth image super resolution. *IEEE Transactions on Image Processing (TIP)* **25**(1), 428–438 (2015)
27. Yang, J., Qi, Z., Shi, Y.: Learning to incorporate structure knowledge for image inpainting. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI) (2020)
28. Zhang, Q., Shen, X., Xu, L., Jia, J.: Rolling guidance filter. In: Proceedings of European Conference on Computer Vision (ECCV) (2014)
29. Zhang, Y., Li, D., Law, K.L., Wang, X., Qin, H., Li, H.: Idr: Self-supervised image denoising via iterative data refinement. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2022)

Learning Depth from Focus in the Wild

Changyeon Won[✉] and Hae-Gon Jeon^{✉*}

Gwangju Institute of Science and Technology
 cywon1997@gist.ac.kr and haegonj@gist.ac.kr

Abstract. For better photography, most recent commercial cameras including smartphones have either adopted large-aperture lens to collect more light or used a burst mode to take multiple images within short times. These interesting features lead us to examine depth from focus/defocus. In this work, we present a convolutional neural network-based depth estimation from single focal stacks. Our method differs from relevant state-of-the-art works with three unique features. First, our method allows depth maps to be inferred in an end-to-end manner even with image alignment. Second, we propose a sharp region detection module to reduce blur ambiguities in subtle focus changes and weakly texture-less regions. Third, we design an effective downsampling module to ease flows of focal information in feature extractions. In addition, for the generalization of the proposed network, we develop a simulator to realistically reproduce the features of commercial cameras, such as changes in field of view, focal length and principal points. By effectively incorporating these three unique features, our network achieves the top rank in the DDFF 12-Scene benchmark on most metrics. We also demonstrate the effectiveness of the proposed method on various quantitative evaluations and real-world images taken from various off-the-shelf cameras compared with state-of-the-art methods. Our source code is publicly available at <https://github.com/wcy199705/DfFintheWild>.

Keywords: depth from focus, image alignment, sharp region detection and simulated focal stack dataset.

1 Introduction

As commercial demand for high-quality photographic applications increases, images have been increasingly utilized in scene depth computation. Most commercial cameras, including smartphone and DSLR cameras have two interesting configurations: large-aperture lens and a dual-pixel (DP) sensor. Both are reasonable choices to collect more light and to quickly sweep the focus through

* Corresponding author

This paper is the short version of ECCV'22 and is NEVER considered an official publication.



Fig. 1. Results of our true end-to-end DfF framework with comparisons to state-of-the-art methods.

multiple depths. Because of this, images appear to have a shallow depth of field (DoF) and are formed as focal stacks with corresponding meta-data such as focal length and principal points. One method to accomplish this is to use single dual-pixel (DP) images which have left and right sub-images with narrow baselines and limited DoFs. A straightforward way is to find correspondences between the left and right sub-images [3]. Despite an abundance of research, such methods are heavily dependent on the accurate retrieval of correspondences due to the inherent characteristics of DP images. Pixel disparities between the two sub-images result in blurred regions, and the amount of spatial shifts is proportional to the degree of blurrings. Another group of approaches solves this problem using different angles. The out-of-focus regions make it possible to use depth-from-defocus (DfD) techniques to estimate scene depths [11]. Since there is a strong physical relationship between scene depths and the amount of defocus blurs, the DfD methods account for it in data-driven manners by learning to directly regress depth values. However, there is a potential limitation to these works [11]. A classic issue, an aperture effect, makes an analysis of defocus blur in a local window difficult. In addition, some of them recover deblurred images from input, but image deblurring also belongs to a class of ill-posed inverse problems for which the uniqueness of the solution cannot be established [8]. These shortcomings motivate us to examine depth from focus (DfF) as an alternative.

In this work, we achieve a high-quality and well-generalized depth prediction from single focal stacks. Our contributions are threefold (see Fig.1): First, we compensate the change in image appearance due to magnification during the focus change, and the slight translations from principal point changes. Compared to recent CNN-based DfD/DfF works [4,10,14] which either assume that input sequential images are perfectly aligned or use hand-crafted feature-based alignment techniques, we design a learnable context-based image alignment, which works well in defocusing blurred images. Second, the proposed sharp region detection (SRD) module addresses blur ambiguities resulting from subtle defocus changes in weakly-textured regions. Third, we also propose an efficient down-sampling (EFD) module for the DfF framework. With this depth from focus network, we achieve state-of-the-art results over various public datasets as well as the top rank in the DDFB benchmark [4]. Ablation studies indicate that each of these technical contributions appreciably improves depth prediction accuracy.

2 Methodology

Our network is composed of two major components: One is an image alignment model for sequential defocused images. Another component is a focused feature representation, which encodes the depth information of scenes.

2.1 A Network for Defocus Image Alignment

Since camera field of views (FoVs) vary according to the focus distance, a zoom-like effect is induced during a focal sweep [5], called focal breathing. Because of the focal breathing, an image sharpness cannot be accurately measured on the same pixel coordinates across focal slices. Recent CNN-based approaches disregard the focal breathing because either all public synthetic datasets for DfF/DfD, whose scale is enough to generalize CNNs well, provide well-aligned focal stacks, or are generated by single RGB-D images. Because of this gap between real-world imagery and easy to use datasets, their generality is limited. Therefore, as a first step to implementing a comprehensive, all-in-one solution to DfF, we introduce a defocus image alignment network.

Field of view. Scene FoVs are calculated by work distances, focus distances, and the focal length of cameras. Since the work distances are fixed during a focal sweep, relative values of FoVs (Relative FoVs) are the same as the inverse distance between sensor and lens. We thus perform an initial alignment of a focal stack using these relative FoVs. We note that needed values to calculate relative FoVs are available by accessing the metadata information in cameras without any user calibration.

Nevertheless, the alignment step is not perfectly appropriate for focal stack images due to hardware limitations, as described in [5]. Most smartphone cameras control their focus distances by spring-installed voice coil motors (VCMs). The VCMs adjust the positions of the camera lens by applying voltages to a nearby electromagnet which induces spring movements. Since the elasticity of the spring can be changed by temperature and usage, there will be an error between real focus distances and values in the metadata. In addition, the principal point of cameras also changes during a focal sweep because the camera lens is not perfectly parallel to the image sensor, due to some manufacturing imperfections. Therefore, we propose an alignment network to adjust this mis-alignment and a useful simulator to ensure realistic focal stack acquisition.

Alignment network. As shown in Fig.2, our alignment network has 3-level encoder-decoder structures, similar to the previous optical flow network [7]. The encoder extracts multi-scale features, and multi-scale optical flow volumes are constructed by concatenating the features of a reference and a target focal slice. The decoder refines the multi-scale optical flow volumes in a coarse-to-fine manner using feature warping (F-warp). However, we cannot directly use the existing optical flow framework for alignment because defocus blur breaks the brightness constancy assumption [13]. To address this issue, we constrain the flow using three basis vectors with corresponding coefficients (α , β , γ) for each scene motion. To compute the coefficients instead of the direct estimation of the flow

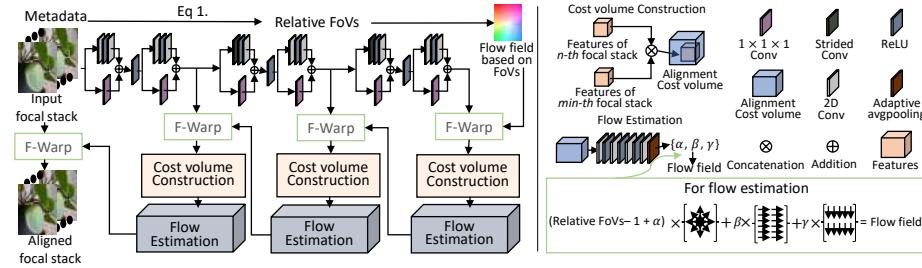


Fig. 2. An illustration of our alignment network. Given initially-aligned images with camera metadata, this network produces an aligned focal stack. In the flow estimation, we use three basis functions to model radial, horizontal and vertical motions of VCMs.

field, we add an adaptive average pooling layer to each layer of the decoder. The first basis vector accounts for an image crop which reduces errors in the FoVs. We elaborate the image crop as a flow that spreads out from the center. The remaining two vectors represent x - and y -axis translations, which compensate for errors in the principal point of the cameras. These parametric constraints of flow induce the network to train geometric features which are not damaged by defocus blur. We optimize this alignment network using a robust loss function L_{align} , proposed in [9], as follows:

$$L_{align} = \sum_{n=0}^N \rho(I_n(\Gamma + D(\Gamma)) - I_{min}(\Gamma)), \quad (1)$$

where $\rho(\cdot) = (|\cdot| + \varepsilon)^q$. q and ε are set to 0.4 and 0.01, respectively. I_n is a focal slice of a reference image, and I_{min} is the target focal slice. $D(\Gamma)$ is an output flow of the alignment network at a pixel position, Γ .

Simulator. Because public datasets do not describe changes in FoVs or hardware limitations in off-the-shelf cameras, we propose a useful simulator to render realistic sequential defocus images for training our alignment network. Given metadata of cameras used, our simulator renders focal stacks induced from blur scales based on the focus distance and the error ranges of the basis vector.

2.2 Focal Stack-oriented Feature Extraction

For high-quality depth prediction, we consider two requirements that must be imposed on our network. First: feature downsampling such as a convolution with strides and pooling layers is necessary to reduce the computations in low-level computer vision task. Second: Feature representations for DFF need to identify subtle distinctions in blur magnitudes between input images.

Sharp Region Detector. The initial feature of each focal slice is needed to communicate with other neighboring focal slices, to measure the focus of the pixel of interest. In Fig.3 (left), we extract features using a 2D ResNet block and add an attention score which is computed from them by 3D convolutions and a ReLU activation. The 3D convolution enables the detection of subtle defocus

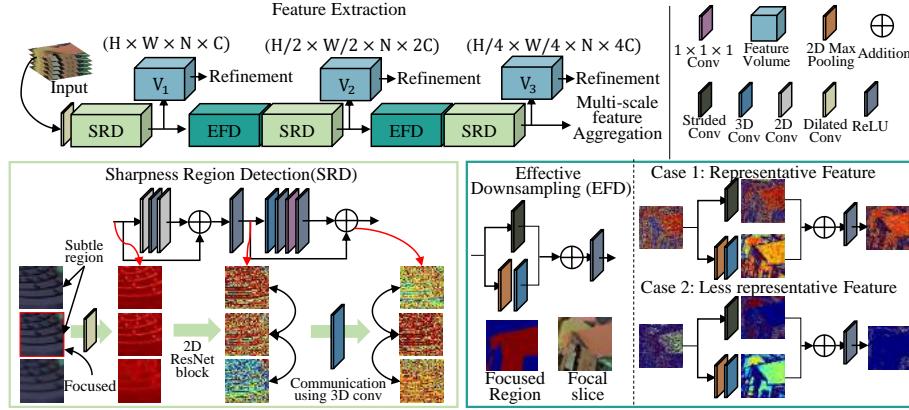


Fig. 3. An architecture of our feature extraction. If feature maps from neighbor focal slices have similar values, our SRD gives an attention score to the sharpest focal slice. Our EFD preserves informative defocus feature representation during downsampling.

Table 1. Quantitative evaluation on DDFF 12-Scene [4]. We directly refer to the results from [14]. Since the result of DefocusNet [10] is not uploaded in the official benchmark, we only bring the MSE value from [10]. **bold**: Best, Underline: Second best. Unit: pixel.

Method	MSE ↓	RMSE log ↓	AbsRel ↓	SqRel ↓	Bump ↓	$\delta = 1.25 \uparrow$	$\delta = 1.25^2 \uparrow$	$\delta = 1.25^3 \uparrow$
DDFF [4]	$9.7e^{-4}$	0.32	0.29	0.01	<u>0.6</u>	61.95	85.14	92.98
DefocusNet [10]	$9.1e^{-4}$	-	-	-	-	-	-	-
AiFDepthNet [14]	$8.6e^{-4}$	<u>0.29</u>	<u>0.25</u>	0.01	<u>0.6</u>	<u>68.33</u>	<u>87.40</u>	93.96
Ours	$5.7e^{-4}$	<u>0.21</u>	<u>0.17</u>	0.01	<u>0.6</u>	<u>77.96</u>	93.72	<u>97.94</u>

variations in weakly texture-less regions by communicating the features with neighbor focal slices.

EFFECTIVE DOWNSAMPLING. Unlike stereo matching networks that use convolutions with strides for downsampling features [12], the stride of a convolution causes a loss in spatial information because most of the focused regions may not be selected. The EFD module employs a 2D max-pooling as a downsampling operation and applies a 3D convolution to its output. Through our EFD module, our network can both take representative values of focused regions in a local window and communicate the focal feature with neighbor focal slices.

3 Evaluation

3.1 Comparisons to State-of-the-art Methods

We validate the robustness of the proposed network by showing experimental results on various public pre-aligned datasets.

DDFF 12-Scene [4]. DDFF 12-Scene dataset provides focal stack images and its ground truth depth maps captured by a light-field camera and a RGB-D sensor, respectively. The images have shallow DoFs and show texture-less regions. Our method shows the better performance than those of recent published

Table 2. Quantitative evaluation on DefocusNet dataset [10] (unit: meter), 4D Light Field dataset [6]) and Smartphone dataset [5] (unit: meter). For DefocusNet dataset and 4D Light Field dataset. For Smartphone dataset [5], we multiply confidence scores on metrics ('MAE' and 'MSE') which are respectively denoted as 'MAE*' and 'MSE*'.

Method	DefocusNet Dataset [10]			4D Light Field [6]			Smartphone [5]		
	MAE ↓	MSE ↓	AbsRel ↓	MSE ↓	RMSE ↓	Bump ↓	MAE* ↓	MSE* ↓	Secs ↓
DefocusNet [10]	0.0637	0.0175	0.1386	0.0593	0.2355	2.69	0.1650	0.0800	0.1598
AiFDepthNet [14]	0.0549	0.0127	0.1115	0.0472	0.2014	1.58	0.1568	0.0764	0.1387
Ours	0.0403	0.0087	0.0809	0.0230	0.1288	1.29	0.1394	0.0723	0.1269

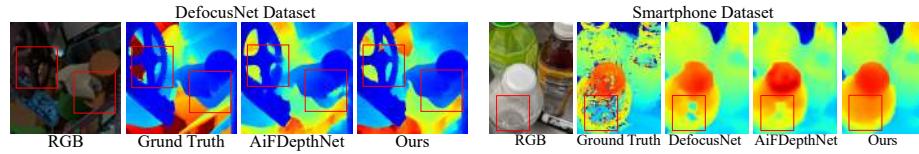


Fig. 4. Qualitative results on DefocusNet dataset and Smartphone dataset.

works in Tab.1 and achieves the top rank in almost evaluation metrics of the benchmark site.

DefocusNet Dataset [10]. This dataset is rendered in a virtual space and generated using Blender Cycles renderer [1]. Focal stack images consist of only five defocused images whose focus distances are randomly sampled in an inverse depth space. The quantitative results are shown in Tab.2. As shown in Fig.4, our method successfully reconstructs the smooth surface and the sharp depth discontinuity rather than previous methods.

4D Light Field Dataset [6]. This synthetic dataset has 10 focal slices with shallow DoFs for each focal stack. The number of focal stacks in training and test split is 20 and 4, respectively.

Smartphone [5]. This dataset shows real-world scenes captured from Pixel 3 smartphones. As expected, our network achieves the promising performance over the state-of-the-art methods, whose results are reported in Tab.2 and Fig.4.

Table 3. Ablation studies for SRD and EFD.

Module	MAE ↓	MSE ↓	RMSE log ↓	AbsRel ↓	SqRel ↓	$\delta = 1.25 \uparrow$	$\delta = 1.25^2 \uparrow$	$\delta = 1.25^3 \uparrow$
SRD → 2D ResNet block	0.0421	0.0095	0.1614	0.0842	0.0142	0.9082	0.9722	0.9873
SRD → 3D ResNet block	0.0409	0.0088	0.1576	0.0818	0.0128	0.9123	0.9725	0.9891
EFD → Maxpooling + 3D Conv	0.0421	0.0094	0.1622	0.0845	0.0143	0.9125	0.9712	0.9849
EFD → Avgpooling + 3D Conv	0.0422	0.0097	0.1628	0.0830	0.0141	0.9126	0.9718	0.9860
EFD → Strided Conv	0.0419	0.0091	0.1630	0.0842	0.0135	0.9144	0.9725	0.9867
EFD → 3D Pooling Layer	0.0414	0.0089	0.1594	0.0843	0.0132	0.9088	0.9747	0.9886
Ours	0.0403	0.0087	0.1534	0.0809	0.0130	0.9137	0.9761	0.9900

3.2 Ablation Study

We carry out extensive ablation studies to demonstrate the effectiveness of each module of the proposed network.

Alignment network. To evaluate our alignment network, we render focal stacks using our simulator which generates defocused images based on a camera

<https://competitions.codalab.org/competitions/17807#results>

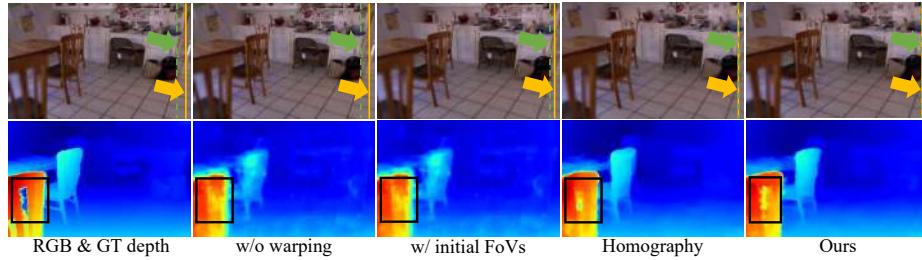


Fig. 5. Ablation study on our alignment network. The first row refer a target and reference focal slice whose FoVs have the smallest and the biggest values, respectively. The second row shows depth estimation results in accordance to the alignment methods. Homography denotes a classical homography method [2].

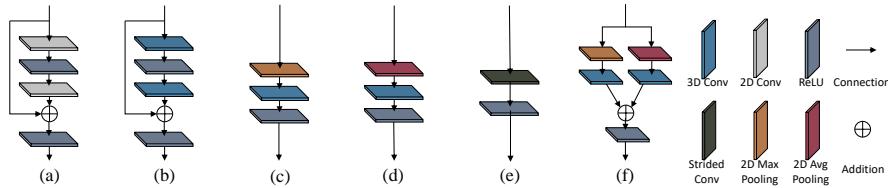


Fig. 6. Candidate modules of our SRD and EFD. (a) 2D ResNet block, (b) 3D ResNet block, (c) Max pooling + 3D Conv, (d) Average pooling + 3D Conv, (e) Strided Conv and (f) 3D pooling layer.

metadata. The qualitative results are reported in Fig.5. By using GPUs, our alignment network achieves much faster and competitive performance with the classic homography-based method.

SRD and EFD. We compare our modules with other feature extraction modules depicted in Fig.6. The quantitative result is reported in Tab.3.

When we replace our SRD module with either 3D ResNet block or 2D ResNet block only, there are performance drops, even with more learnable parameters for the 3D ResNet block. We also compare our EFD module with four replaceable modules: max-pooling+3D Conv, average pooling+3D Conv, Stride convolution and 3D pooling layer. As expected, our EFD module achieves the best performance because it allows better gradient flows preserving defocus property.

4 Conclusion

In this paper, we have presented a novel and true end-to-end Dff architecture. To do this, we first propose a trainable alignment network for sequential defocused images. We then introduce a novel feature extraction and an efficient downsampling module for robust Dff tasks. The proposed network achieves the best performance in the public Dff/Dfd benchmark and various evaluations.

Limitation. There are still rooms for improvements. A more sophisticated model for flow fields in the alignment network would enhance depth prediction results. More parameters can be useful for extreme rotations.

Remarks. This paper is a summary presentation of the paper which has been published in ECCV2022 by request of the IW-FCV2023 program committee to share the research results.

References

1. Community, B.O.: Blender—a 3d modelling and rendering package. Blender Foundation (2018)
2. Evangelidis, G.D., Psarakis, E.Z.: Parametric image alignment using enhanced correlation coefficient maximization. *IEEE transactions on pattern analysis and machine intelligence* **30**(10), 1858–1865 (2008)
3. Garg, R., Wadhwa, N., Ansari, S., Barron, J.T.: Learning single camera depth estimation using dual-pixels. In: Proceedings of International Conference on Computer Vision (ICCV) (2019)
4. Hazirbas, C., Soyer, S.G., Staab, M.C., Leal-Taixé, L., Cremers, D.: Deep depth from focus. In: Proceedings of Asian Conference on Computer Vision (ACCV) (2018)
5. Herrmann, C., Bowen, R.S., Wadhwa, N., Garg, R., He, Q., Barron, J.T., Zabih, R.: Learning to autofocus. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
6. Honauer, K., Johannsen, O., Kondermann, D., Goldluecke, B.: A dataset and evaluation methodology for depth estimation on 4d light fields. In: Proceedings of Asian Conference on Computer Vision (ACCV) (2016)
7. Hui, T.W., Tang, X., Loy, C.C.: Liteflownet: A lightweight convolutional neural network for optical flow estimation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
8. Levin, A., Fergus, R., Durand, F., Freeman, W.T.: Image and depth from a conventional camera with a coded aperture. *ACM transactions on graphics (TOG)* **26**(3), 70–es (2007)
9. Liu, P., King, I., Lyu, M.R., Xu, J.: Ddfow: Learning optical flow with unlabeled data distillation. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI) (2019)
10. Maximov, M., Galim, K., Leal-Taixé, L.: Focus on defocus: bridging the synthetic to real domain gap for depth estimation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
11. Pan, L., Chowdhury, S., Hartley, R., Liu, M., Zhang, H., Li, H.: Dual pixel exploration: Simultaneous depth estimation and image restoration. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
12. Shen, Z., Dai, Y., Rao, Z.: Cfnet: Cascade and fused cost volume for robust stereo matching. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
13. Suwanakorn, S., Hernandez, C., Seitz, S.M.: Depth from focus with your mobile phone. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
14. Wang, N.H., Wang, R., Liu, Y.L., Huang, Y.H., Chang, Y.L., Chen, C.P., Jou, K.: Bridging unsupervised and supervised depth from focus via all-in-focus supervision. In: Proceedings of International Conference on Computer Vision (ICCV) (2021)

Human Face Detector with Gender Identification by Split-based Inception Block and Regulated Attention Module

Adri Priadana¹, Muhamad Dwisnanto Putro², Duy-Linh Nguyen¹, Xuan-Thuy Vo¹, and Kang-Hyun Jo¹

¹ Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Ulsan, South Korea

² Department of Electrical Engineering, Universitas Sam Ratulangi, Manado, Indonesia

priadana3202@mail.ulsan.ac.kr, dwisnantoputro@unsrat.ac.id,
ndlinh301@mail.ulsan.ac.kr, xthuy@islab.ulsan.ac.kr, acejo@ulsan.ac.kr

Abstract. Smart digital advertising platforms have been widely arising. These platforms require a human face detector with gender identification to assist them in the determination of providing relevant advertisements. The detector is also prosecuted to identify the gender of a masked face in post-coronavirus situations and demanded to operate on a CPU device to lower system expenses. This work presents a lightweight Convolution Neural Network (CNN) architecture to build a gender identification integrated with face detection to respond to these issues. This work proposes a split-based inception block to efficiently extract features at various sizes by partially applying different convolution kernel sizes, levels, and regulated attention module to improve the quality of the feature map. It produces slight parameters that drive the architecture efficiency and can operate quickly in real-time. To validate the performance of the proposed architecture, UTKFace and Labeled Faces in the Wild (LFW) datasets, modified with an artificial mask, are utilized as training and validation datasets. This offered architecture is compared to different lightweight and deep architectures. Regarding the experiment results, the proposed architecture outperforms masked face gender identification on the two datasets. In addition, the proposed architecture, which integrates with face detection to become a human face detector with gender identification can run 135 frames per second in real-time on a CPU configuration.

Keywords: Human Face Detector · Face Gender Identification · Convolutional Neural Network (CNN) · Split-based Inception Block · Regulated Attention Module.

1 Introduction

The advancement of information technology has stimulated the rapid development of smart digital advertising, not only in online media but also in offline

media. It is proven because these platforms appear in many public places, such as airports, stations, and markets [4]. Practically, smart digital advertising platforms are handily personalized and customized. Therefore, it can display dynamic contents as determined by the provider. Nevertheless, the market demands effective mechanisms that make these platforms can provide targeted advertising [1]. This mechanisms will offer more advantages in the digital advertising strategy [20].

Providing targeted advertising can be accomplished by personalizing the audience facing the platform. The audience's gender, which is an essential attribute, can be used in segmenting the readers. These platforms can provide better appropriate advertising for each reader by recognizing their gender [15]. This scheme can be achieved with the reader's face detection and classification.

Nowadays, Convolutional Neural Network (CNN) has verified a bunch of victories in image-based detection and classification tasks. The common direction in designing CNN architectures is to develop deeper architectures to reach higher accuracy [7,25]. However, it tends to generate architecture with a large number of parameters. It makes the architecture inefficient to operate, especially on low-cost devices in real-time. In the case of advertising platform implementation, it requires a low-cost device, such as a CPU device, to minimize the implementation expense [3,13]. Hence, it requires an efficient face gender detector, which can be suitably operated on a CPU in real-time.

A new challenge arises after the spread of the COVID-19 virus extensively. It makes people required or used to wear masks on their faces when they are traveling. It makes part of the face area occluded, such as the mouth, which is one of the essential features for recognizing gender through the face. Therefore, it needs an efficient human face detector with gender identification ability that can detect and recognize the gender of a masked face. This work presents an efficient human face detector with gender identification by a few parameters that can efficiently detect and identify a masked face gender while maintaining its performance.

An efficient CPU-based human face detector with gender identification called GenderMask-CPU proposed a lightweight architecture with a split-based inception block and regulated attention module (SiramNet). The split-based inception block is offered to efficiently extract features at various sizes by partially applying different convolution kernel sizes and levels. The regulated attention module, which consist of the channel and spatial, are employed to enhance the feature map grade. It produces scant parameters and guides the detector to work efficiently and fast. In summary, the main contribution of this work is twofold, i.e.,

1. An efficient architecture with a split-based inception block and regulated attention module (SiramNet) is proposed, which generates slight parameters. The split-based inception block can efficiently extract multi-size feature areas of the feature maps. The attention module can maintain the essential features of the face area, which can increase the gender accuracy of the classification.

2. A fast human face detector with gender identification is introduced, which can operate in real-time on a CPU device efficiently and fast. The performance of the offered architecture is proven to compete with other deep and light CNN architectures on UTKFace [30] and Labeled Faces in the Wild (LFW) [10] datasets, modified with an artificial mask utilized from [2].

2 Related Work

In recent years, CNN architectures, designed for face gender recognition work, have progressed with impressive improvement, especially in performance. Various modified versions of CNNs have been developed to optimize face gender recognition. HyperFace-ResNet [21], a CNN architecture, was proposed to perform gender recognition from a face. The architecture develops and adjusts ResNet [5] architecture and reaches good performance on LFW datasets. In [4], a CNN architecture has been employed to recognize gender and implemented in the monitoring system. The architecture utilized MobilenetV2 [22] architecture and generated 3.5 million parameters.

Nowadays, efficient face gender detectors emerge specially designed for CPU devices to encounter market demand which can reduce implementation costs. MPCConvNet [17] based on the CNN architecture was developed and generated 659,650 parameters. The architecture proposed a multi-perspective convolution used to capture various feature regions of the object. The architecture reaches good performance on UTKFace and LFW datasets. SufiaNet [16] based on the CNN architecture was developed and only generated 226,574 parameters. SufiaNet [16] is a shallow architecture supported by a global attention module. The architecture gains sufficient performance on UTKFace and LFW datasets.

3 The Proposed Method

This work proposes a CNN architecture to recognize a gender of a masked face, as shown in Fig. 1. This architecture is structured as a backbone and classification module, generating 441,460 parameters.

3.1 The Backbone

CNN-based feature extraction has shown excellent performance. However, this extractor tends to generate enormous parameters [6]. Therefore, an efficient backbone module is proposed to develop a fast architecture, especially one that can run on the CPU in real time. This architecture employs three main convolution layers with same 3×3 kernel size, managed sequentially by three times the number of kernels growing, i.e. 16, 32, and 64. This mechanism seeks to acquire more information on the latter layers. Following each convolution layer, a batch normalization (BN) method and Leaky ReLU (Leaky Rectified Linear Unit) activation are applied to deal with the vanishing gradient. A dropout strategy puts

previous to the final convolution operation is also used to impede overfitting. Three max-pooling operations are assigned in this backbone to down-sample the feature maps. One layer of 4×4 and two layers of 2×2 sizes max-pooling with two strides are applied.

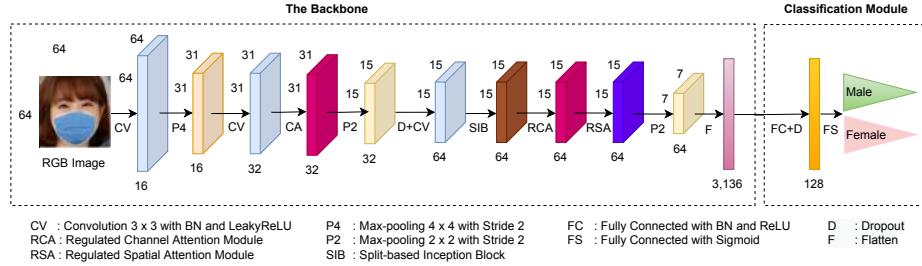


Fig. 1. The proposed architecture of the gender identification of masked faces contains a backbone with a split-based inception block and regulated attention module.

3.2 The Split-based Inception Block

To improve the feature extractor on the backbone module, this work proposes a split-based inception block and applies the block after the last convolution layer. Inspired by the inception block [26], this module employs four branches of convolution layer with different levels and kernel sizes, as shown in Fig. 2. They are convolution layers with 1×1 , 3×3 , two times 3×3 , and 5×5 kernel sizes. Unlike the original inception block that applies the convolution layer with the same number of a kernel as the input, this block divides the input feature map \mathbf{X} become four components $[\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \mathbf{X}_4]$. Then, it applies convolution operation with different levels and kernel sizes mentioned before, which is represented as follows:

$$\begin{aligned} SIB(\mathbf{X}) = \mathbf{X} + & (SELU(BN(C1(D(\mathbf{X}_1)))) \oplus SELU(BN(C3(D(\mathbf{X}_2)))) \\ & \oplus SELU(BN(C3(D(SELU(BN(C3(D(\mathbf{X}_3)))))))) \\ & \oplus SELU(BN(C5(D(\mathbf{X}_4)))), \end{aligned} \quad (1)$$

where $C1, C2, C3$ are convolution layers with 1×1 , 3×3 , and 5×5 kernel sizes, respectively. $SELU$ is Scaled Exponential Linear Units ($SELU$) activation [12], D is dropout operation, BN is batch normalization operation, and \oplus is the concatenate operation. This block will extract more information from different levels and area sizes efficiently. At the last stage, a residual mechanism [6] is applied to combine the concatenate operation result with the input feature map \mathbf{X} by an addition operation.

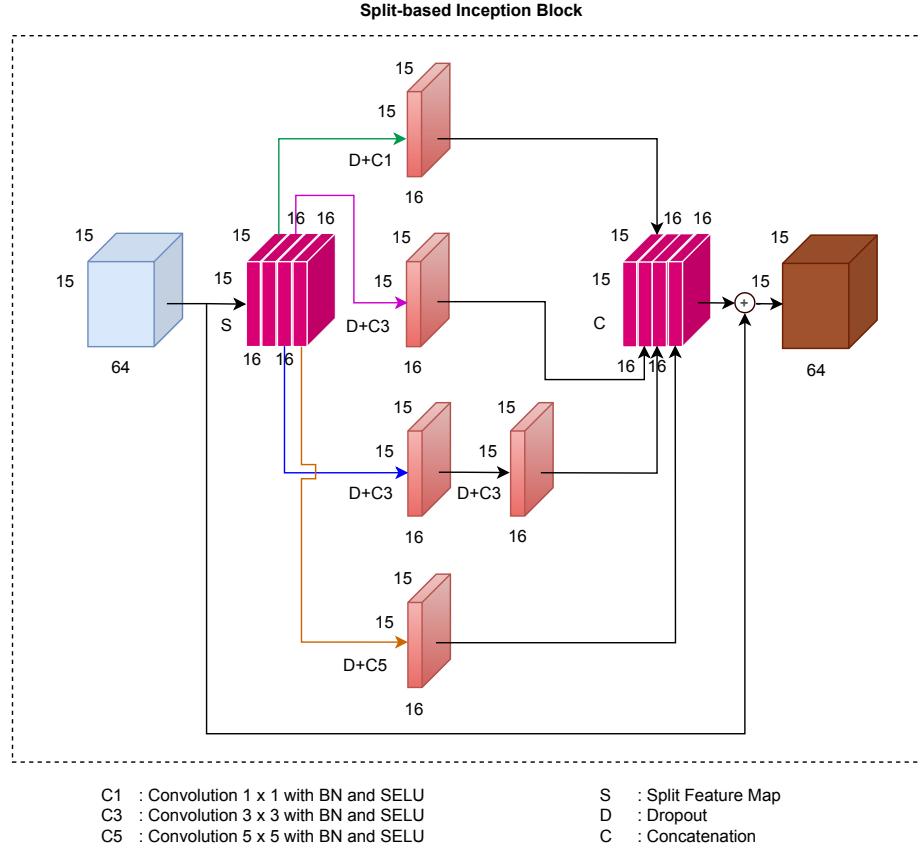


Fig. 2. The proposed Split-based Inception block.

3.3 The Regulated Attention Module (RAM)

A backbone with few parameters feebly discriminates interest features of the face. Therefore, the Regulated Attention module (RAM) is proposed and applied to improve essential facial features. This module consists of a regulated channel attention module (RCA) and a regulated spatial attention module (RSA). Inspired by the attention module in [9], RCA performs a global average-pooling operation to aggregate each feature map based on channel. However, we do not use fully connected layers but apply softmax activation directly after the pooling operation to calculate the probability of channel importance level. A softmax activation is used rather than the sigmoid activation because it can establish long-range channel dependency [29]. Imbued from the attention module [18], this architecture puts a 3×3 depthwise convolution layer before the pooling operations to allow the individual channel to expand learning efficiently, as shown in Fig. 3. Different from [18], this architecture only applies global average

pooling to squeeze the number of operations. Further, we proposed a 1×1 depthwise convolution layer located after softmax activation and before performing a channel-wise multiplication in the last step to regulate the attention weights individually represented as follows:

$$RCA(\mathbf{X}) = \mathbf{X} * DC1(\sigma(GA(DC3(\mathbf{X})))), \quad (2)$$

where $DC1$ and $DC3$ are 1×1 and 3×3 depthwise convolution layers, respectively. GA is a global average-pooling operation and σ is a softmax activation. \mathbf{X} is an input of the RCA.

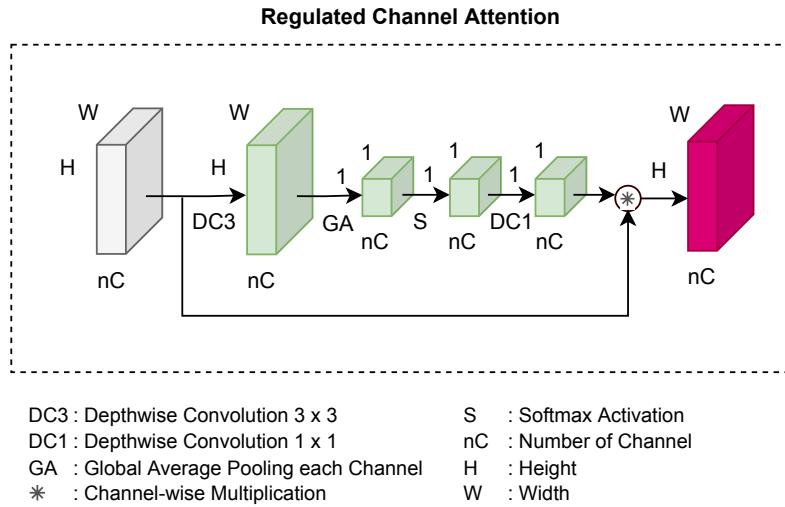


Fig. 3. The proposed Regulated Channel Attention module.

Motivated by [28], a global average-pooling operation is assigned on RSA to aggregate spatial features across the channel. This operation renders a feature vector and describes the feature overview of the corresponding channel. However, a softmax activation is voted than a sigmoid activation to calculate the spatial importance level. This activation can establish spatial dependency. A 1×1 depthwise convolution layer is also applied after softmax activation and before performing a spatial-wise multiplication to regulate the attention weights with a shared parameter, as shown in Fig. 4. It is represented as follows:

$$RSA(\mathbf{X}) = \mathbf{X} * DC1(\sigma(GA(\mathbf{X}))), \quad (3)$$

where $DC1$ is a 1×1 depthwise convolution layer and GA is a global average-pooling across the channel. σ is a softmax activation and \mathbf{X} is an input of RSA.

RCA is assigned following the second convolution layer and the split-based inception block. This module will enhance the grade of the intermediate and

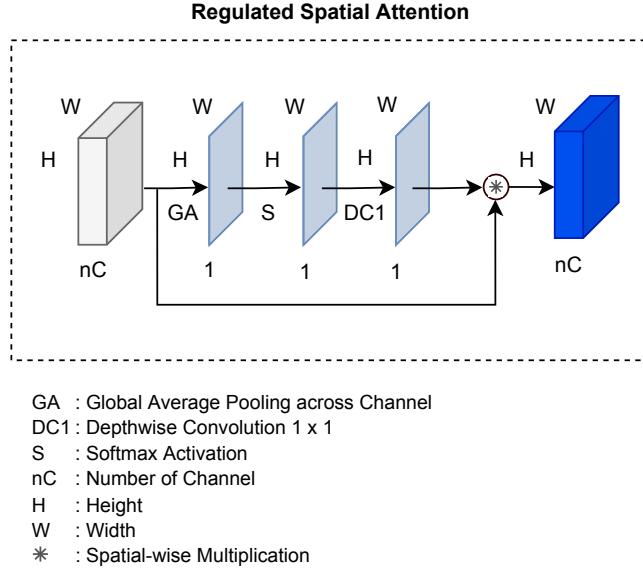


Fig. 4. The proposed Regulated Spatial Attention module.

latter features. On the other hand, RSA is only assigned following the last RCA, which drives the architecture to focus on the location of informative spatial features after it extracts the high-level features.

3.4 Classification Module

The backbone module is tasked to extract features from masked faces. Then, the results will be fed to the classification module employed to reckon the probability of each gender class. This operation leads to deciding whether the masked face is male or female. This classification module is composed of two dense layers with 128 and 2 units, respectively. A batch normalization and ReLU (Rectified Linear Unit) activation are applied after the first dense layer, and the Sigmoid activation is applied after the second dense layer. The Sigmoid activation will render the input into scenarios that could describe the prediction decision of whether the masked face is male or female. In order to discourage overfitting, it applies a dropout operation after the ReLU activation.

3.5 Face Detector

In this work, face detection is required for integrating with masked face gender recognition to build a masked face gender detector. It is employed to locate and get the region of the face or masked face referred to as a Region of Interest (RoI). An efficient face detection model with cheap operation is required to operate

brief in the real-time. Hence, a face detector named LWFCPU [19] is utilized. It employs only several convolutional layers that generate slight parameters. The ROI, which comes from the face detection operation, will become an input of the proposed gender recognition architecture. It will be resized and cropped to a particular size appropriate for the architecture input.

4 Experimental Settings

4.1 Dataset Pre-Processing

In this work, UTKFace and LFW datasets, which are labeled as females and males, are used for training and validation separately. Firstly, each facial image of these datasets is resized into 64×64 pixels appropriated with the input of the proposed gender recognition architecture. To generate masked face instances, we follow [2] to overlay one type of mask (Surgical) on UTKFace and LFW images, which produce 22,841 and 10,374 masked face images, respectively, and the examples are shown in Fig. 5. In this experiment, each dataset is split using a random permutation mechanism into 70% as a training set and 30% as a validation set. This mechanism will generate the unique order of the instances.



Fig. 5. The examples of the UTKFace and Labeled Faces in the Wild (LFW) datasets, modified with an artificial mask utilized from [2].

4.2 Implementation Details

The experiment is executed on the NVIDIA GTX 1080Ti 11GB to accelerate the training on the proposed architecture by using Tensorflow and Keras framework libraries. UTKFace and LFW datasets modified with an artificial mask referenced from [2] are used as training and validation to ratify the performance of the proposed architecture, which trains with three hundred epochs. The Adam optimizer is employed to optimize the weight on the Binary Cross-Entropy loss.

The datasets are trained by using 10^{-2} initial learning rate, which will reduce to 75% if the accuracy does not improve in every 20 epochs. Intel Core i7-9750H CPU@2.6 GHz with 20GB RAM is used to investigate the speed in frame per second (FPS) of the proposed architecture and the detector.

5 Results

5.1 Evaluation on Datasets

UTKFace. A face dataset labeled in gender, age, and ethnicity, is used for training and validation to ratify the performance of the proposed architecture. This dataset consists of 23,708 instances with various positions, expressions, resolutions, and lighting. This dataset also covers age variations ranging from 0 to 116. This dataset was modified with an artificial mask utilized from [2] and generated 22,841 masked face images. The proposed architecture, which only employs 441,460 parameters, gains 91.17% of validation accuracy. The proposed architecture outperforms deep CNN architectures [24,7,27], as sown in Table 1. Moreover, the proposed architecture reaches accuracy surpassing the three lightweight architectures, SqueezeNet [11], SufiaNet [16], and MPConvNet [17], which differed by 2.4, 1.16, and 0.98, respectively.

Table 1. Evaluation results on UTKFace dataset, modified with an artificial mask utilized from [2].

Architectures	Number of Parameters	Validation Accuracy
MobileNetV2 [23]	2,260,546	87.93
ResNet50V2 [7]	23,568,898	87.99
VGG13 [24] with BN	34,467,906	88.07
SqueezeNet [11] with BN	735,306	88.77
VGG16 [24] with BN	39,782,722	89.23
VGG11 [24] with BN	34,413,698	89.26
InceptionV3 [27]	21,806,882	89.64
SufiaNet [16]	226,574	90.01
MPConvNet [17]	659,650	90.19
SiramNet (ours)	441,460	91.17

LFW. A face dataset labeled in gender consists of 13,234 instances with unbalance proportion between males and females, about 77% and 23%. This dataset was also modified with an artificial mask utilized from [2] and generated 10,374 masked face images. The proposed architecture, which only employs 441,460 parameters, gains 95.64% of validation accuracy. The proposed architecture also outperforms deep CNN architectures [24,7,27], as sown in Table 2. Moreover, the

proposed architecture also reaches accuracy surpassing the three lightweight architectures, SqueezeNet [11], SufiaNet [16], and MPConvNet [17], which differed by 1.38, 0.38, and 0.27, respectively.

Table 2. Evaluation results on LFW dataset, modified with an artificial mask utilized from [2].

Architectures	Number of Parameters	Validation Accuracy
MobileNetV2 [23]	2,260,546	79.93
VGG13 [24] with BN	34,467,906	91.18
InceptionV3 [27]	21,806,882	92.58
ResNet50V2 [7]	23,568,898	92.96
VGG16 [24] with BN	39,782,722	93.35
VGG11 [24] with BN	34,413,698	93.88
SqueezeNet [11] with BN	735,306	93.99
SufiaNet [16]	226,574	95.13
MPConvNet [17]	659,650	95.37
SiramNet (ours)	441,460	95.64

5.2 Ablation Study

This work performs the ablation study to investigate how much the proposed split-based inception block and attention module will impact the validation accuracy result. This ablative study conducts by repealing the block or module and then calculating the validation accuracy on the UTKFace dataset. As can be seen in Table 3, utilizing the proposed split-based inception block and applying this block after the last convolution layer can increase the accuracy by 0.12%. The proposed RCA can escalate the accuracy by 0.38%. Moreover, the proposed RSA module can also escalate the accuracy by 0.2% by adding only two parameters.

Table 3. Ablation study on UTKFace dataset, modified with an artificial mask utilized from [2].

Group Split Inception Block	Regulated Channel Attention Module	Regulated Spatial Attention Module	Number of Parameters	Validation Accuracy
			426,338	90.47
✓			440,306	90.59
✓	✓		441,458	90.97
✓	✓	✓	441,460	91.17

5.3 Runtime Efficiency

The proposed architecture recognizes gender from a masked face using only 441,460 parameters. The architecture operates in real-time at 272.80 and 135.02 frames per second for gender identification and gender identification integrated with face detection [19], respectively. The proposed efficient architecture becomes the second fastest compared to other deep and lightweight architectures, as shown in Table 4. Even though SufiaNet [16] has become the fastest architecture, the validation accuracy is not better than our proposed architecture, with a difference of 1.16 and 0.38 on the UTKFace and LFW datasets, modified with an artificial mask utilized from [2], respectively. Fig. 6 shows the recognition result of the GenderMask-CPU, in which the green bounding box means a male face and the magenta bounding box means a female face. Although this detector is specially designed for faces with mask shown in Fig. 6 (a), it can also work on faces without mask shown in Fig. 6 (b).

Table 4. Runtime efficiency on an Intel Core i7- 9750H CPU.

Architectures	Number of Parameters	GFLOPs	Gender Recognition (FPS)	Gender Recognition integrated with Face Detection (FPS)
VGG16 [24] with BN	39,782,722	2.2900	43.28	37.15
VGG13 [24] with BN	34,467,906	1.6100	51.14	42.76
VGG11 [24] with BN	34,413,698	1.2700	55.49	45.71
ResNet50V2 [7]	23,568,898	0.5710	57.19	46.79
InceptionV3 [27]	21,806,882	0.4050	64.43	51.47
MobileNetV2 [23]	2,260,546	0.0501	118.71	81.20
SqueezeNet [11] with BN	735,306	0.0833	231.40	122.75
MPConvNet [17]	659,650	0.0670	269.86	132.04
SufiaNet [16]	226,574	0.0218	327.29	145.37
SiramNet (ours)	441,460	0.0293	272.80	135.02

5.4 Attention Modules Comparison

The proposed regulated attention module (RAM) is also compared with other common attention modules, as shown in Table 5. This module compares with Squeeze-and-Excitation (SE) [8], Bottleneck Attention Module (BAM) [14], and Convolutional Block Attention Module (CBAM) [28]. These attention modules are applied at the same place, i.e. after the second convolution layer and the split-based inception block on the UTKFace dataset, modified with an artificial mask utilized from [2], to perform a fair comparison. The validation accuracy of the proposed architecture with RAM is higher than the proposed architecture with BAM, SE, or CBAM, which differ by 0.48%, 0.39%, and 0.12%, respectively.

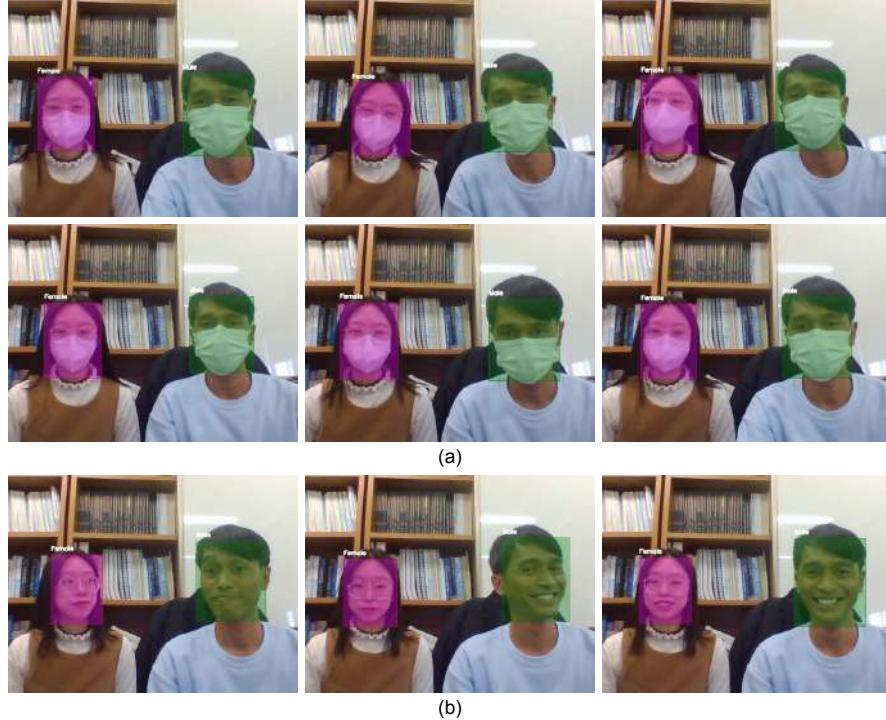


Fig. 6. The correct detection results of the GenderMask-CPU detector for masked (a) and non-masked (b) faces.

Table 5. Comparisons of Different Attention Modules applied on the Proposed Architecture on UTKFace dataset, modified with an artificial mask utilized from [2].

Attention Modules	Number of Parameters	GFLOPs	Validation Accuracy	Gender Recognition (FPS)	Gender Recognition integrated with Face Detection (FPS)
BAM [14]	449,736	0.0348	90.69	238.19	125.40
SE [8]	440,946	0.0284	90.78	307.44	145.18
CBAM [28]	441,084	0.0286	91.05	273.84	135.84
RAM (ours)	441,460	0.0293	91.17	272.80	135.02

6 Conclusion

An efficient CPU-based human face detector with gender identification called GenderMask-CPU is proposed and offers a lightweight architecture with a split-based inception block and regulated attention module. This lightweight architecture assigns a few convolution operations that make the architecture only

generates 441,460 parameters. This work offered a split-based inception block to efficiently extract features at various sizes by partially applying different convolution kernel sizes and levels. The regulated attention module is also proposed to improve the quality of the feature map. This architecture acquires competitive performance compared to other lightweight and deep CNN architectures on the UTKFace and Labeled Faces in the Wild (LFW) datasets, modified with an artificial mask utilized from [2]. Accordingly, when operating on a CPU device in real-time, GenderMask-CPU is capable of running at 135 frames per second while identifying the gender of masked faces. This detector outperforms other lightweight and deep competitors' architecture. In a forthcoming study, other mechanisms, such as Transformer, can be conducted to improve the identification accuracy. The augmentation strategy can also be explored to improve the dataset varieties that can increase the performance of masked face gender recognition.

Acknowledgment

This result was supported by "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(MOE)(2021RIS-003).

References

1. Alhalabi, M., Hussein, N., Khan, E., Habash, O., Yousaf, J., Ghazal, M.: Sustainable smart advertisement display using deep age and gender recognition. In: 2021 International Conference on Decision Aid Sciences and Application (DASA). pp. 33–37. IEEE (2021)
2. Anwar, A., Raychowdhury, A.: Masked face recognition for secure authentication. arXiv preprint arXiv:2008.11104 (2020)
3. Bandung, Y., Hendra, Y.F., Subekti, L.B.: Design and implementation of digital signage system based on raspberry pi 2 for e-tourism in indonesia. In: 2015 International Conference on Information Technology Systems and Innovation (ICITSI). pp. 1–6. IEEE (2015)
4. Greco, A., Saggese, A., Vento, M.: Digital signage by real-time gender recognition from face images. In: 2020 IEEE International Workshop on Metrology for Industry 4.0 & IoT. pp. 309–313. IEEE (2020)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778. IEEE (2016)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016). <https://doi.org/10.1109/CVPR.2016.90>
7. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: European conference on computer vision. pp. 630–645. Springer (2016)
8. Hu, J., Shen, L., Albanie, S., Sun, G., Wu, E.: Squeeze-and-excitation networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **42**(8), 2011–2023 (2019)

9. Hu, J., Shen, L., Albanie, S., Sun, G., Wu, E.: Squeeze-and-excitation networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **42**(8), 2011–2023 (2020). <https://doi.org/10.1109/TPAMI.2019.2913372>
10. Huang, G.B., Mattar, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In: Workshop on faces in 'Real-Life' Images: detection, alignment, and recognition (2008)
11. Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K.: SqueezeNet: Alexnet-level accuracy with 50x fewer parameters and$\frac{1}{10}$ 0.5 mb model size. arXiv preprint arXiv:1602.07360 (2016)
12. Klambauer, G., Unterthiner, T., Mayr, A., Hochreiter, S.: Self-normalizing neural networks. *Advances in neural information processing systems* **30** (2017)
13. Mishima, K., Sakurada, T., Hagiwara, Y.: Low-cost managed digital signage system with signage device using small-sized and low-cost information device. In: 2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC). pp. 573–575. IEEE (2017)
14. Park, J., Woo, S., Lee, J.Y., Kweon, I.S.: Bam: Bottleneck attention module. arXiv preprint arXiv:1807.06514 (2018)
15. Priadana, A., Maarif, M.R., Habibi, M.: Gender prediction for instagram user profiling using deep learning. In: 2020 International Conference on Decision Aid Sciences and Application (DASA). pp. 432–436. IEEE (2020)
16. Priadana, A., Putro, M.D., Jeong, C., Jo, K.H.: A fast real-time face gender detector on cpu using superficial network with attention modules. In: 2022 International Workshop on Intelligent Systems (IWIS). pp. 1–6 (2022). <https://doi.org/10.1109/IWIS56333.2022.9920714>
17. Priadana, A., Putro, M.D., Jo, K.H.: An efficient face gender detector on a cpu with multi-perspective convolution. In: 2022 13th Asian Control Conference (ASCC). pp. 453–458 (2022). <https://doi.org/10.23919/ASCC56756.2022.9828048>
18. Priadana, A., Putro, M.D., Vo, X.T., Jo, K.H.: An efficient face-based age group detector on a cpu using two perspective convolution with attention modules. In: 2022 International Conference on Multimedia Analysis and Pattern Recognition (MAPR). pp. 1–6. IEEE (2022)
19. Putro, M.D., Nguyen, D.L., Jo, K.H.: Lightweight convolutional neural network for real-time face detector on cpu supporting interaction of service robot. In: 2020 13th International Conference on Human System Interaction (HSI). pp. 94–99. IEEE (2020)
20. Putro, M.D., Priadana, A., Nguyen, D.L., Jo, K.H.: A faster real-time face detector support smart digital advertising on low-cost computing device. In: 2022 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM). pp. 171–178 (2022). <https://doi.org/10.1109/AIM52237.2022.9863289>
21. Ranjan, R., Patel, V.M., Chellappa, R.: Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE transactions on pattern analysis and machine intelligence* **41**(1), 121–135 (2017)
22. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4510–4520 (2018). <https://doi.org/10.1109/CVPR.2018.00474>
23. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4510–4520. IEEE (2018)

24. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
25. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1–9 (2015). <https://doi.org/10.1109/CVPR.2015.7298594>
26. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2818–2826 (2016). <https://doi.org/10.1109/CVPR.2016.308>
27. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2818–2826. IEEE (2016)
28. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: Cbam: Convolutional block attention module. In: Proceedings of the European conference on computer vision (ECCV). pp. 3–19 (2018)
29. Zhang, H., Zu, K., Lu, J., Zou, Y., Meng, D.: Epsanet: An efficient pyramid squeeze attention block on convolutional neural network. In: Proceedings of the Asian Conference on Computer Vision. pp. 1161–1177 (2022)
30. Zhang, Z., Song, Y., Qi, H.: Age progression/regression by conditional adversarial autoencoder. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5810–5818 (2017)

Novel Surveillance System for Suspicious Activities Analysis using Deep Learning

Aditya Kakde, Bhavana Kaushik, Deepika Koundal, Durgansh Sharma, and Neelu Jyothi Ahuja

Abstract Surveillance is about tracking suspicious actions rather than recognizing objects in the scene. Current surveillance systems employ classification and detection procedures, but fail to explain why they are used, given that the human eye is capable of seeing them as well. For surveillance to be effective, it must not only report an occurrence but also warn if there is a risk of an incident occurring. As a result, suspicious actions must be investigated by comparing current data to past data in the form of a picture. Human involvement cannot compare this past data with the most recent data from millions of data points since it would take too long. As a result, we suggest a system that uses an end-to-end system to analyse some questionable actions in real-time. To accurately detect any suspicious actions using sensor data, the proposed surveillance system employs Content-based Image Retrieval (CBIR) combined with a deep learning algorithm. Using real-time graphical analysis and feature extraction also allows for improved administration of results and data. The suggested system completed CBIR with deep learning and demonstrated its graphical analysis and feature extraction in real-time, demonstrating its uniqueness over previous systems. The suggested system includes a dashboard that can be used to analyse not only what happened at that specific time, but also what happened in the previous days and how they differed from what happened today. With this suspicious analysis method, one can not only determine whether or not

Aditya Kakde
University of Petroleum and Energy Studies, Dehradun, India e-mail: adityakakde100@gmail.com

Bhavana Kaushik
University of Petroleum and Energy Studies, Dehradun, India e-mail: bkaushik@ddn.upes.ac.in

Deepika Koundal
University of Petroleum and Energy Studies, Dehradun, India e-mail: Deepika Koundal@ddn.upes.ac.in

Durgansh Sharma
Christ (Deemed to be University), Ghaziabad, India e-mail: durgansh.sharma@christuniversity.in

Neelu Jyothi Ahuja
University of Petroleum and Energy Studies, Dehradun, India e-mail: neelu@ddn.upes.ac.in

an incident has occurred, but also receive a warning if there is a possibility of an incident.

Key words: Content Based Image Retrieval, Computer Vision, Convolution Neural Network (CNN), Real-time Feature Extraction, Graphical Analysis, Satellite Imagery, Surveillance System

1 Introduction

Remote sensing has a fascinating and rich history. It all began during World War I when aerial photography proved to be an effective instrument for exploration and observation. The pre-Hispanic Civilization of Nazca started distant observation long back. 'Earth observation' as a source of traditional indicators to create geoglyphs (known as Nazca Lines) that could only be seen from a certain height was used in ancient days. However, in today's world, this type of monitoring is still carried out in the form of satellite photographs that are taken from satellites to witness the earth's motion and suspicious events. It is always crucial to observe the ground movements for identifying suspicious activity for example construction of new buildings, storms, wildfires and rising sea levels. From farmers analysing their crops to urban architects properly charting roadways, satellite pictures have a wide range of uses. Increased sea levels, hurricanes, and wildfires all can be detected with the help of satellites that monitor the environment. Radar satellites are being used by geologists to anticipate volcanic eruptions and locate fault lines. Military satellites, which are generally used for surveillance and investigation by intelligence specialists, and satellites that are generally used for communication and entertainment are termed commercial satellites, GPS satellites are used for direction-finding applications, and scientific satellites are used for weather studies, planetary research, and assessing agricultural patterns. The current research proposes a technique for detecting suspicious behaviour using satellite pictures (that are openly available on Google). Surveillance is said to be perfect and efficient when it detects any anomalous or distrustful activities precisely. Current surveillance systems are generally operated by humans, and they involve incessant human attention to detect any unusual/suspicious activity. As humans are intervening, the efficiency of the system decreases with time due to the exhaustion and tiredness factor of humans. The above challenge can be resolved by the automation of the surveillance system. The purpose of the automated mechanism is to give a warning in the form of an alarm or any other method when a predefined abnormal action happens [1]. Satellite imagery is fetching more researchers and scholar community for various challenges and issues, and it has been used for provincial-level mapping, location planning, and defencelessness or destruction valuations in the latest events. In non-emergency situations, satellite imagery has been used to estimate population estimates [2]. Currently, some 5,300 satellites orbit Earth, which means that thousands of cameras are taking real-time photographs above you. Satellite photography pro-

vides a unique perspective for photographing the planet, which can aid scientists and others in identifying patterns and trends [3]. Furthermore, to make educated guesses about unusual and unusual behaviour. As part of the country's coastal security fortification, Indian Space Research Organisation (ISRO) satellite imageries will quickly monitor distrustful vessels and boats heading into the seas, according to the home ministry [4]. Using satellite photography to correctly detect changes to the Earth's surface may help with everything from climate change studies and farming to human migration patterns and nuclear non-proliferation. However, dynamically integrating photos from a variety of sources — for example, those that indicate surface changes (such as new building development) against those that show substance changes (such as water to sand) was unfeasible [5]. With a fresh suggested model capability, what's suspicious happening on the ground may now be easily spotted in terms of a new building and military movements for security purposes using satellite photos. The suggested system employs a deep Convolutional Neural Network to perform Content-Based Image Retrieval (CBIR). CBIR uses query pictures to search a huge database for images, and content denotes the shape, colour, texture, and other associated information of both the query and stored photos. To conduct Content-Based Image Retrieval, the proposed system uses a deep Convolutional Neural Network (CBIR). CBIR searches a large database for photographs using input images, where content refers to various characteristics of an image like shape, colour and texture connected with both the input and stored photos [6]. In [7], the characteristics of satellite photographs maintained in the record are mined using a VGG19 convolution neural network model. VGG19 was chosen above other extremely deep learning models due to the systems specification constraint. It's a 19-layer and 5-pooling layers CNN with both the convolutional and fully connected layers using ReLU activation functions, and the output layer using softmax activation functions to acquire critical characteristics to recognise alike images from kept images based on an input image. The query photographs' characteristics are compared to stored database picture features during the testing phase, and similarity is calculated. The similarity is used to find the most comparable photos.

2 Literature Review

It was proposed in [8] that a deep CNN be utilised with a 4-layer neural model using the ReLU function as activation was employed on the CIFAR-10 data, and it had a six times quicker training error rate. The unique architecture, which consisted of 5 convolutional layers with overlapping pooling and local response normalisation, was then evaluated on the ILSVRC-2010 data, and it obtained the lowest error percentage. And after that, it is put to the test on the ILSVRC-2012, where it attained the lowest error percentage of 67 and 40.9. In [7], the ILSVRC-2012 dataset was used to test a deep learning model. The grouping accuracy improved after increasing the layers of the deep learning model. When pictures were scaled to (256,512), the 19 layers of CNN succeeded with the lowest top-1 testing error of 23.7% and top-5

testing error of 6.8%. A review of CBIR was given in [6], which discussed the sorts of characteristics that are utilised to identify similarities. Among the characteristics are colour, texture, shape, spatial, low-level cues, region-based approaches, and extraction of features using a deep learning model. Using a combination of colour and form data, the deep learning approach was employed in [9] to locate the most comparable photographs to the query image. The images used in the experiment were sourced from the internet. CBIR used a transfer learning technique to recover brain tumours [10]. A unique paradigm was given to aid the radiologist in recognising tumour types in unclear instances. In [11], CBIR was employed as a transfer learning method for fetching trademarks as an image. In [12], the transfer learning approach was used with the CBIR mechanism on biological and remote sensing data, resulting in the conclusion that CNN is more dependable than typical CBIR systems. In [13], a comparative analysis of CNN and CNN-SVM was studied, and it was determined that CNN-SVM should be used. When CNN was combined with GPU, however, it yields better results as compared to CNN-SVM by 0.5 per cent and consumed a reduced amount of time. The article by [14] proposes a detection approach for a ship as an object based on spectral reflectance in challenging settings such as darkness, fog, and hazes using multi-spectral satellite pictures. A neural network dubbed LFNet (lightweight fusion network) was also developed to validate regions with ships by combining ship reflectance and picture colour information. In addition, the proposed research shows that they can detect adverse weather. Continuously moving satellite images and videos are utilised for a lot of monitoring and tracking. A work published in [15] suggests that micro vehicle recognition be done on satellite videos utilising a multi-morphological-cue-based discriminating algorithm to isolate the vehicle from the background noise. The effectiveness of maritime surveillance systems has improved in recent years, and one suggested system, which utilises a Haar-like classifier for boat identification, is provided in [16]. They also proposed an upgrade in [17], which was based on detection and integrated data from diverse sources to get improved results. The results, which include images shot in a variety of lighting conditions and with a variety of camera settings, highlight the technique's utility. In the [18], routine satellite remote sensing surveillance on oil spills using SAR photographs is carried out, which assists in determining the percentage of oil spillage by monitoring satellite data for the previous five years in the area of the Bohai Sea and north of the Yellow Sea. As observed in [19], one of the study projects in the field of surveillance focuses on marine traffic understanding utilising multi-sensor satellite data processing, which recommends the programmed identification of the vessel's movement-related characteristics as well as vessel velocity vector calculation. The gradient-based method improves the accuracy of the estimate of wake motion-related characteristics. In [20], issues such as concerns and difficulties, distrustful movement identification of humans, the methodology for suspicious activity identification, datasets, with assessment methods were discussed, as well as a comparison of numerous surveillance systems. In [21], to detect violent actions in real-time, a crowd detection system based on convolutional long short-term is applied. In [22], a new deep YOLOv3 detection technique with 106 convolutional layers and 2 dense connected layers has been developed. The surveillance

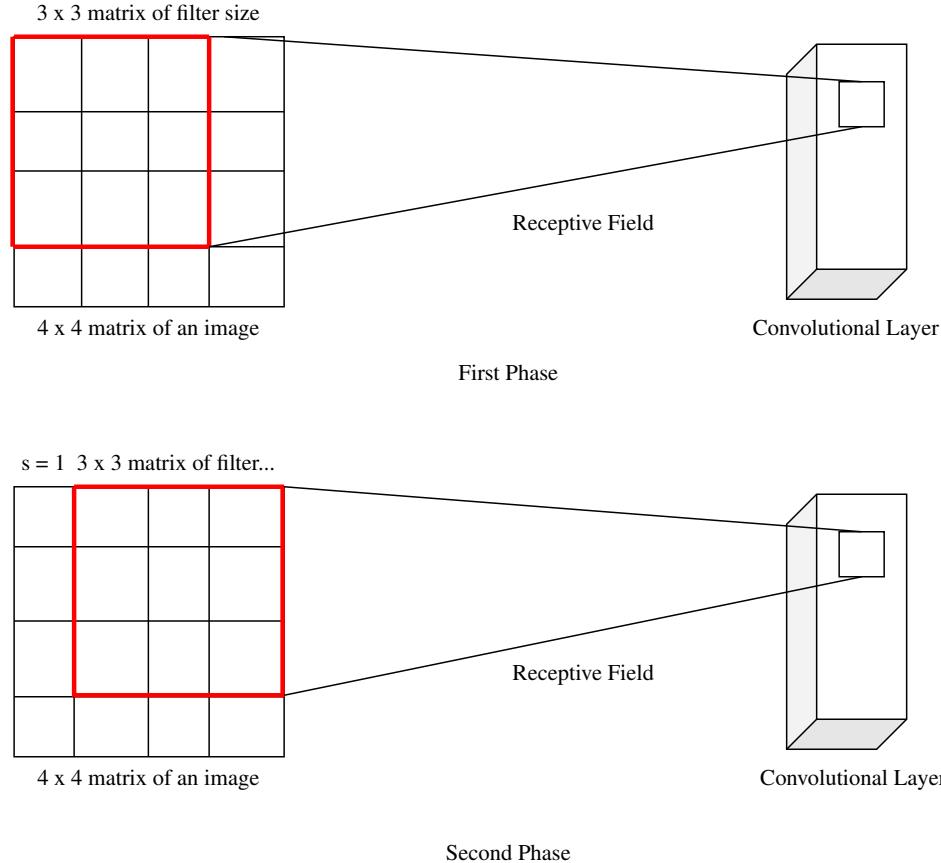
system is based on a detecting system and is utilised with a small drone. In [23], using Quantized SSD Mobilenet V2 and Tiny YOLOv3 models, a surveillance system based on the detection and categorization of military vehicles was presented, with the Tiny YOLOv3 model providing the best performance. In [24], for surveillance using UAVs, an ideal deep learning algorithm called Optimal UAV-based Layer Distribution (OULD) and OULD with Mobility Prediction (OULD-MP) is utilised to decrease latency during data classification. In [25], a monitoring and surveillance system is proposed that uses a deep learning model on aerial photos, with the best performance coming from YOLOv4 twice. A detection mechanism underpins the suggested model. However, none of them discusses real-time feature extraction or real-time graphical analysis of confidence ratings. The classification portion was also discussed, but not the analysis of historical data and how it might be compared to current data. Furthermore, no logic has been provided as to why a bounding box is required over an object that can be seen with one's own eyes. Some discussed detection in multi-spectral images, however, it was proposed as video surveillance, for which this study gave reasons in Section 3, why it isn't very trustworthy. The tabular analysis of the literature review can be seen in table 1.

3 Proposed Work

To conduct Content-Based Image Retrieval, the proposed system employs an algorithm with pictures as a query. The fact is emphasised that video is not essential for the identification of suspicious behaviour because video monitoring may produce challenges such as object tracking of multiple objects, shadow identification, fuzzy objects, clutter, and gatherings [20]. The convolutional layer examines the whole image and fetches the critical characteristics and features of an image. The characteristics of pictures in the database are trained using the VGG19 deep learning model. Throughout the validation phase, the details of the input pictures are compared with the characteristics of the testing images, and likeness is calculated between both sets of images. The similarity is used to find the most comparable photos. In addition, the prototype can extract characteristics in real-time. It uses the scheduler principle, which involves creating a folder in which a scheduler may be started. First, a time is assigned to a scheduler. When someone uploads a photo to that folder, the scheduler checks to see if it was uploaded, and if it was, the training will begin immediately. The size of the filter, strides, and padding values are other convolutional layer characteristics. The padding value is set to 1, which preserves the convolutional layers' and strides' spatial resolution as shown in Figure 1. Using the receptive field, the convolutional layer analyses the pictures part by part, forming a 3×3 matrix. Because $s = 1$, the value of stride in the given picture is 1. This indicates that another matrix will have a change of one column and one row which is described in Figure 2. The process of convolution involves the dot product of part of an image with the selected filters to give a single value after summing up. As a consequence, we receive the extracted feature. This method is known as convolu-

Table 1 Tabular analysis of the literature review

Author	Title	Dataset	Methods
A. Latif et. al [6]	Content-based Image Retrieval and Feature Extraction: A Comprehensive Review	Not Applicable	CBIR
K. Simonyan and A. Zisserman [7]	Very Deep Convolutional Network for ILSVRC Large Scale Image Recognition	ILSVRC	VGG
A. Krizhevsky et. al [8]	ImageNet Classification with Deep Convolution Neural Network	ILSVRC	AlexNet
R. Rajkumar and M.V. Sudhamani [9]	Content based Image Retrieval System using Combination of Color and Shape Features and Siamese Neural Network	Data from world wide web (www)	scrapped CBIR
Z. Swati et. al [10]	Content-Based Brain Tumor Retrieval for CE-MRI MR Images Using Transfer Learning	CE-MRI	VGG19-based novel feature extraction framework
S.Hasan et. al [11]	Trademark Image Retrieval using Transfer Learning	FlickerLogos-32 and Logos-32	TIR system using AlexNet
P. Sadeghi-Tehran [12]	Scalable Database Indexing and Fast Image Retrieval Based on Deep Learning and Hierarchically Nested Structure Applied to Remote Sensing and Plant Biology	MalayaKew and UCM	CBIR with FEGCN, BOVW and MFF
O. Mohamed et. al [13]	Content-Based Image Retrieval Using Convolutional CNN-SVM	ImageNet and Caltech256	
X.Xie et. al [14]	Ship Detection in Multispectral Satellite Images Under Complex Environment	Multi-spectral images of ships from 4 satellites	LFNet
W. Ao et. al [15]	Needles in a Haystack: Tracking City-Scale Moving Vehicles from Continuously Moving Satellite	Satellite videos	Novel algorithm based on local noise modelling
D. Bloisi et. al [16]	Automatic Maritime Surveillance with AIS Visual Target Detection		Haar-Cascade
D. Bloisi et. al [17]	Enhancing Automatic Maritime Surveillance Systems with Visual Information	EO and IR data	Visual detection, visual tracking, and data fusion
L.Bing et. al [18]	Spatial Distribution Characteristics of SAR images Oil Spills in the Bohai Sea Based on Satellite Remote Sensing and GIS		Framework based on remote sensing and geographical information system (GIS)
M. Reggiannini and L. Bedini [19]	Multi-Sensor Satellite Data Processing for Marine Traffic Understanding	Remote sensing data of ships	Tailored gradient estimator in the early processing stages
R. K. Tripathi et. al [20]	Suspicious human activity recognition: a review	Not Applicable	Surveillance
T.Saba [21]	Real time anomalies detection in crowd using convolutional memory network	Standard crowd dataset	Conv-LSTM
K. Madasamy et. al [22]	OSDDY: embedded system-based surveillance detection system with small drone using deep YOLO	Data of open field and marine environment from drones	YOLOv3
P. Gupta et. al [23]	Edge device based Military Vehicle Detection and Classification from UAV	Novel data of military vehicles	SSD v2 and Tiny YOLOv3
M. Jouhari et. al [24]	Distributed CNN Inference on Resource-Constrained UAVs for Surveillance Systems: Design and Optimization	Data collected from UAVs	LeNet and VGG166
H. Gupta and O. Verma [25]	Monitoring and surveillance of urban road traffic using low altitude drone images: a deep learning approach	AU-AIR	YOLOv4

**Fig. 1** Process of Feature Extraction

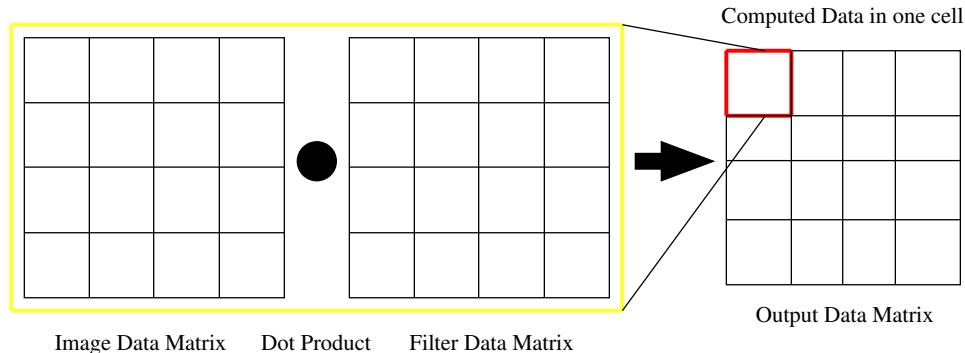
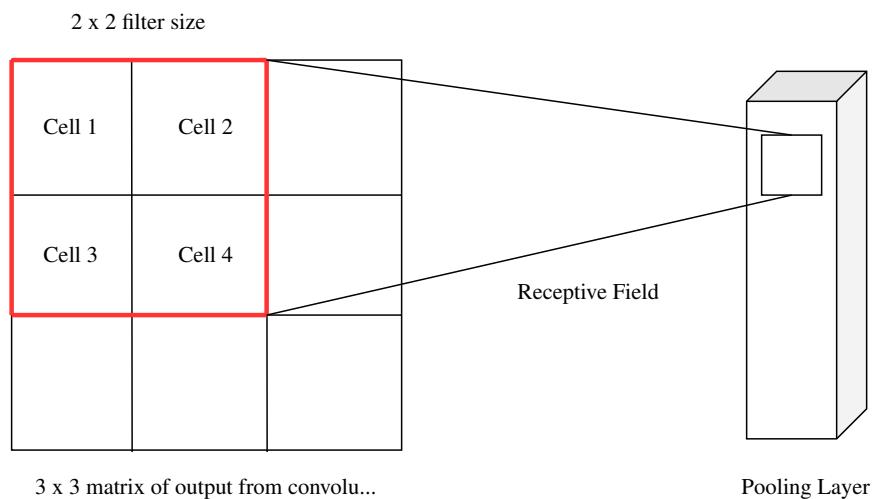
tion. The image's dimension is steadily reduced during the process, leaving just the most significant elements which can be seen through Eq. 1.

$$\frac{n_h + 2p - f}{s} + 1 \times \frac{n_w + 2p - f}{s} + 1 \quad (1)$$

where nh stands for the height of the image and nw is the width of an input image, padding is denoted by p, f stands for a dimension of the kernel, and s stands for stride. To reduce the input image's size pooling mechanism is used which is described in Figure 3.

In the pooling process, max pooling is applied where 4 cells get retrieved from the 2 x 2 matrix and assess the cell with the greatest value which can be seen in Eq. 2.

$$\max(Cell1, Cell2, Cell3, Cell4) \quad (2)$$

**Fig. 2** Process of Convolution**Fig. 3** Process of Pooling

In this phase, the image is likewise down-scaled to capture the most important parts. The following formula seen in Eq. 3 may be used to compute it:

$$\frac{n_h - f}{s} + 1 \times \frac{n_w - f}{s} + 1 \quad (3)$$

where n_h stands for the height of the image and n_w is the width of an input image, f is the dimension of the kernel and $s = \text{Strides}$ (should be a completely divisible integer) and it is shown in Figure 4.

The dot product, also known as convolution, will be used to produce the original picture data and filter data first, as illustrated in the flowchart above. The data is subsequently transferred to the pooling layer, which takes into consideration the largest number value from the resulting matrix. The pictures are not trained in the

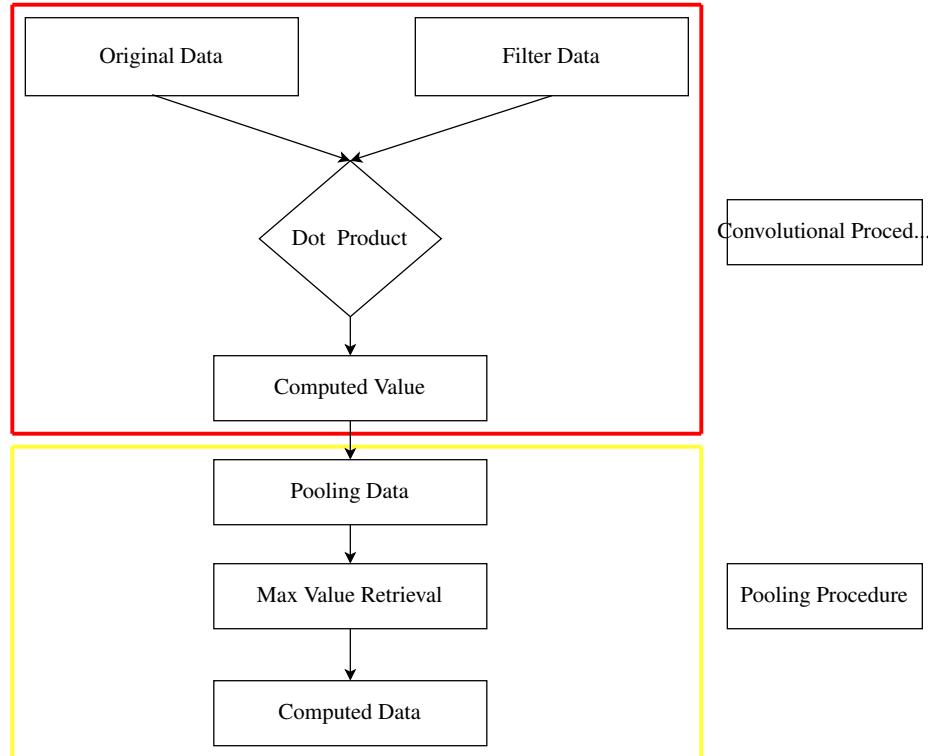


Fig. 4 Flowchart of the feature extraction process.

suggested method. When we train a neural network, we enable it to learn data, but in this situation, all we do is acquire the features of the photos so that we may compute their similarity. After using this approach for fetching the features on both database and query photographs, the dot product of their feature vectors will be determined which can be seen in Eq. 4.

$$\sum_{i=1}^n q_i \cdot \sum_{i=1}^n d_i \cdot T \quad (4)$$

where q is the feature set of the input picture, d is the feature set of the stored picture, and T is indicating the transposition of the feature set. Then, because we need to order the most comparable photos, a quick approach is utilized.

$$(N - 1) + \frac{2}{N} \sum_{k=0}^{N-1} Q_k \quad (5)$$

where $N = \text{array size}$ and $k \in 0, \dots, n-1$. To organise an array in ascending order as given in Eq. 5, we reverse this procedure to make a decreasing ordered list to obtain the most comparable picture.

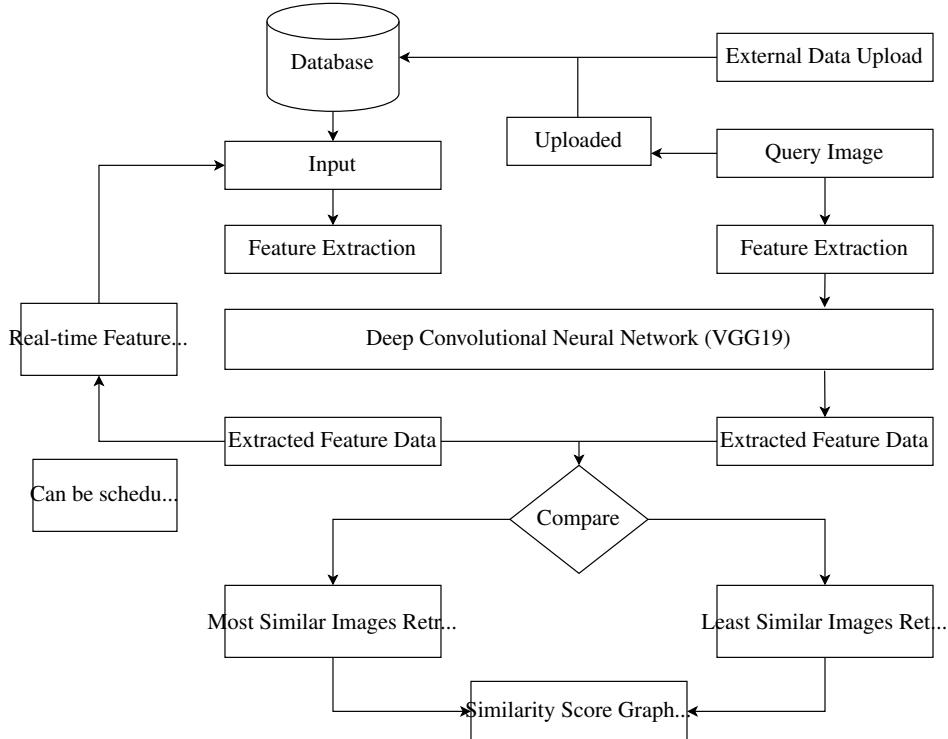
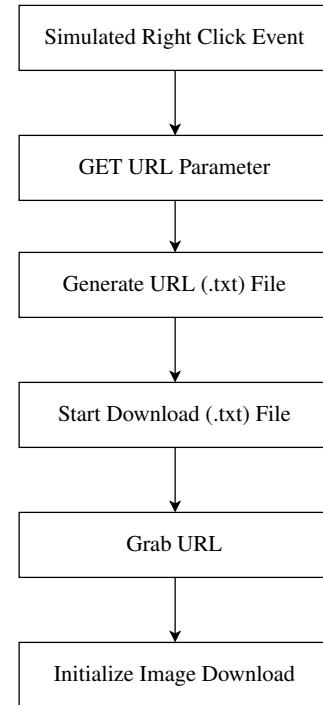


Fig. 5 Flowchart of proposed surveillance systems.

4 Experimental Setup

The uniqueness of the suggested monitoring mechanism, which is fully described in Fig. 5 and the experimental setup are discussed in this section. The information is initially uploaded to the dashboard and then stored in the record. Images are sent into the deep learning CNN model to extract features of all stored images. These characteristics include colour, texture, patterns, and so on, unlike typical CBIR, where only a single kind of feature is extracted for matching purposes. The feature vector is fetched with the same CNN model when we submit a query or a test picture to the database. The stored images and input data's likeness are then calculated. If a match is discovered, the most similar image is returned, and if a large number of photographs are required, the match is returned in decreasing order. In terms of real-time feature extraction, the training phase is planned daily at noon (though this may be altered), and it constantly scans the given photographs and initiates training at an infinite range.

Fig. 6 Steps of fetching the images from Google.



5 Results and Discussions

This section discusses about the dataset used, work output generated, and the analysis of the outcome.

5.1 Dataset

This module discusses the dataset as well as the desired outcome. The photographs are acquired from the internet because the collection of satellite photos of the site is not available. The approach is depicted in the flowchart in Fig. 6. This process simulates a right-click activity for receiving the URL of each picture from the context menu without navigating to another page. The second phase includes the Get URL method to get a URL parameter from a query string since Google keeps the whole picture URL in a query parameter. After that creation of the text file is done and downloaded. The next step includes fetching all URLs once all are collected. We utilise the request parameter to download the.txt file once we have all of the URLs in it. For the proposed system, 530 satellite images were used to generate characteristics.

5.2 Work Output

For surveillance, CBIR with deep learning is used to analyse not only what happened at that exact instant, but also what happened in the prior days and how they differ from what happened today. Assume we've gotten a significant number (perhaps millions) of satellite photos depicting a group of militants nearing the border. However, rather than detecting anything, we want to use changes in a specific region's activity as a monitoring tool for analysing changes. We'll need the most comparable photos from the prior actions to compare with the most current ones. It will take much too long to locate the most comparable photos if someone manually compares a query image to millions of other photographs in the folder. By taking input, a dashboard has been given to prevent manual interruptions.

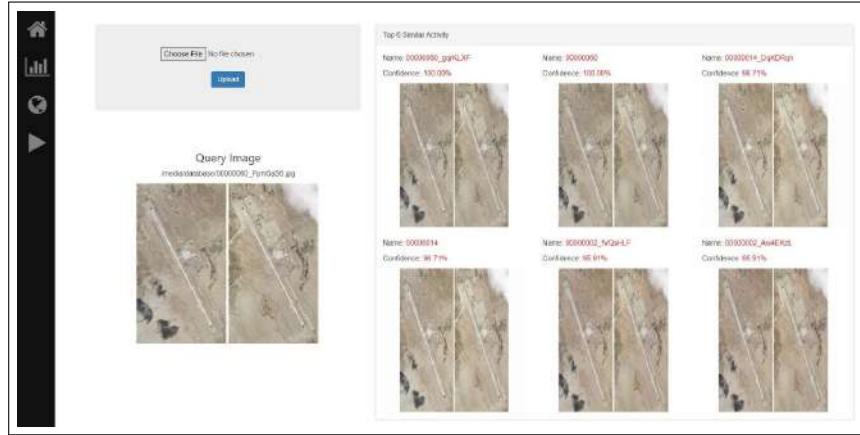


Fig. 7 The dashboard's main screen.

In Figure 8, a satellite picture is given as input, and accordingly received the utmost comparable images linked with that event. The six most similar images were gathered to show the prototype, with the first being highly matched, the second being bit lesser alike than the first, and so on. This dashboard makes use of tabs to allow for easy navigation from one section to the next and, as a consequence, removes the need to refresh the web page in Fig. 7.

A graphical breakdown of the outcome may be shown at the same time in Fig. 8. After that, you'll find the about section and real-time feature extraction. It is possible to see the number and names of images that have been lately posted in Fig. 9. If you submit the same image more than once, it will be re-titled and will be taken as a new picture. After all these the images are stored in a database and further utilized to automate the scheduler. When the scheduled interval arrives, it searches for recently stored images before starting the training process, which eliminates the need for human participation.

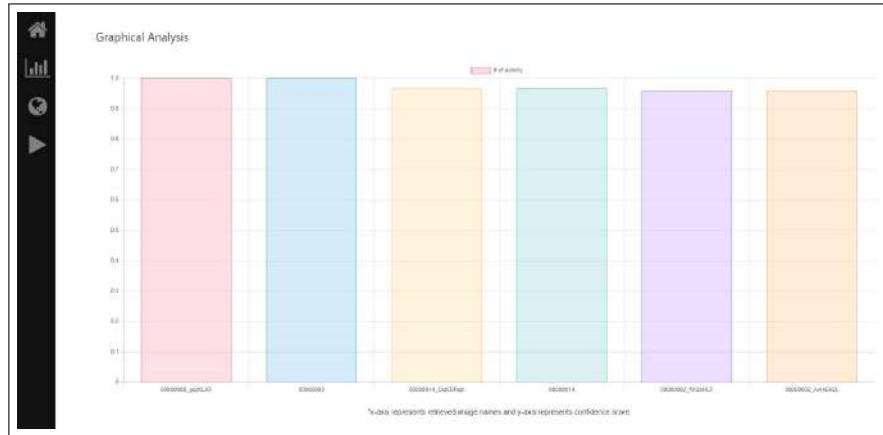


Fig. 8 Real-time graphical analysis of test results in real-time.

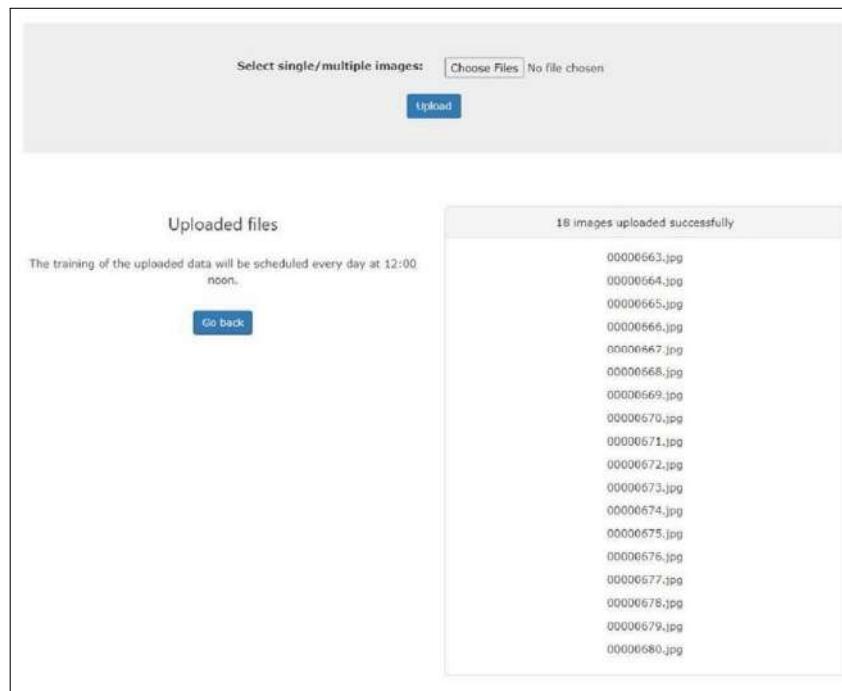


Fig. 9 Real-time feature extraction.

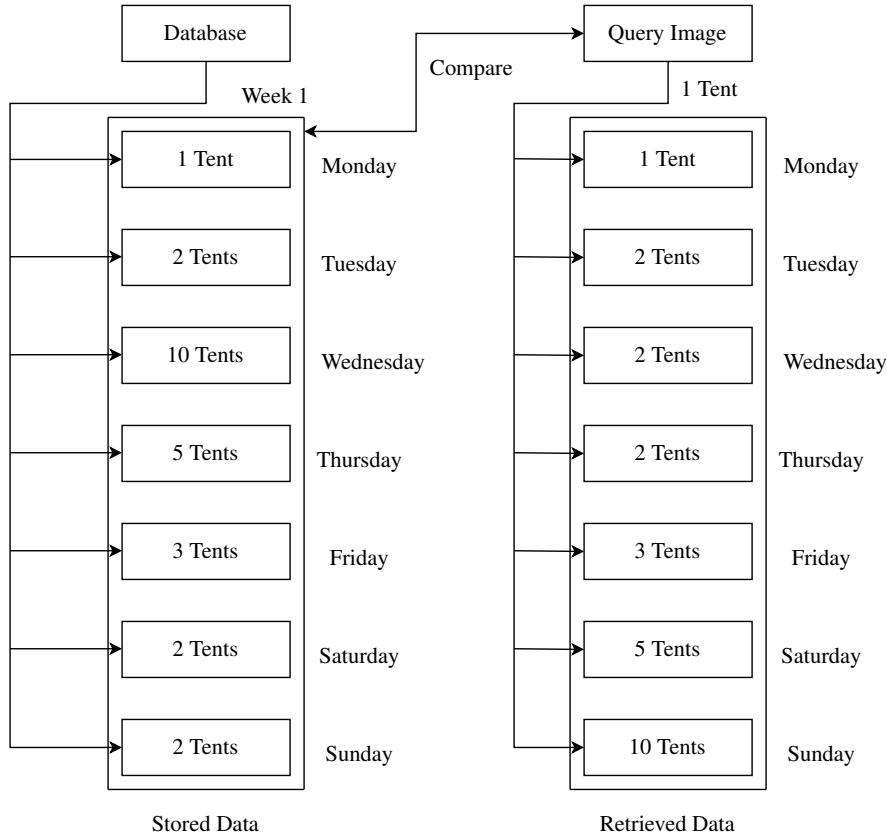


Fig. 10 Example analysis of suspicious activity.

5.3 Analysis

This part will show you how to get the analysed result from the returned images acquired with CBIR and deep learning. A circumstance in which n number of tents are placed in an area for a week has been presented as an example. This example assumes that the subject is well-understood. Fig. 10 shows how input or query pictures can be utilized to recognise doubtful activity on earth faster and more efficiently. The labels in Fig. 10 are only there to help you understand what is happening to check the suspicious activity. Week 1 data may be displayed, with activities such as 1 Tent, 2 Tents, and so on, with Monday through Sunday indicating the day those activities occurred. The results are matched to the given test image from the first day of week 2. Assume the query image has the activity 1 Tent. It is then compared to the database image, which retrieves the most similar images, which can now be analysed to identify the variation in movement, which reveals that tents increased in the early days of the past week and steadily lessened by the completion of the

week. The database visualisations in Fig. 10 are only illustrative. It won't simply be a few photographs in the database; it might be hundreds of thousands. As a result, analysing and making conclusions only based on data is impossible. In Table 2 all the recent work published in area surveillance is utilized to identify the techniques they are using to compare with the proposed method.

Table 2 Comparative analysis of the proposed surveillance system with other state-of-the-art systems

Author	CBIR	DL	Real-time GA	Real-time FE
T.Saba [21]	✗	✓	✗	✗
K. Madasamy et. al [22]	✗	✓	✗	✗
P. Gupta et. al [23]	✗	✓	✗	✗
M. Jouhari et. al [24]	✗	✓	✗	✗
H. Gupta and O. Verma [25]	✗	✓	✗	✗
Proposed Work	✓	✓	✓	✓

The components considered are CBIR, DL, real-time GA, and real-time FE, where DL stands for deep learning, GA for graphical analysis, and FE for feature extraction. A tick indicates that a system has used these parameters, whereas a cross indicates that it has not.

6 Future Work

The suggested system introduces new methods for carrying out surveillance activities, although there is always an opportunity for creativity. As a result, further research may be done on:

1. Enhancement of the feature extraction process of CBIR.
2. Enhancement of the deep neural network architecture.
3. Using detection in CBIR with deep learning when dealing with multi-spectral images.

7 Conclusion

Content-Based Image Retrieval (CBIR) combining deep learning techniques with real-time feature extraction and real-time graphical analysis has never been used in a surveillance system. As a result, the proposed system employs CBIR and deep learning algorithms to identify suspicious behaviour in real-time while extracting data characteristics. It may be used to compare and contrast acts that occurred not

only at that exact time, but also in prior days, and how they differ from what is being done now. The recommended answer is in the shape of a dashboard, which, thanks to its swift navigation capabilities, allows all of these actions to be accomplished quickly. This suspicious analysis technique not only determines whether or not an event has occurred, as categorization systems do but also provides a warning if an occurrence is likely to occur.

References

1. S. Mubareka, D. Ehrlich, B. Ferdinand and F. Kayitakire, "(Settlement location and population density estimation in rugged terrain using information derived from Landsat ETM and SRTM data," International Journal of Remote Sensing, vol. 29, no. 8, pp. 2339-2357, 2008.
2. f. S. B. T. Checchi and J. Palmer, "Validity and feasibility of a satellite imagery-based method for rapid estimation of displaced populations," International Journal of Health Geography, vol. 12, no. 4, pp. 1-12, 2013.
3. S. Brown, "Satellite surveillance may be less of a privacy concern than you think – for now," 29 October 2019. [Online].
Available: <https://www.cnet.com/science/turns-out-satellite-surveillance-only-sounds-like-a-major-privacy-concern/> [Accessed 26 May 2022]
4. R. Venkatesan, "ISRO satellite imageries to monitor suspicious vessels in Indian waters," 8 January 2018. [Online].
Available: <https://www.thehindubusinessline.com/news/isro-satellite-imageries-to-monitor-suspicious-vessels-in-indian-waters/article9972309.ece> [Accessed 21 May 2022]
5. A. Ziemann, "Integrating diverse satellite images sharpens our picture of activity on Earth," 21 December 2021. [Online].
Available: <https://www.lanl.gov/discover/science-columns/top-columns-and-blogs/2021/space-satellite-images.php> [Accessed 25 May 2022].
6. A. Latif, A. Rasheed, U. Sajid, J. Ahmed, N. Ali, N. I. Ratyal, B. Zafar, S. H. Dar, M. Sajid and T. Khalil, "Content-Based Image Retrieval and Feature Extraction: A Comprehensive Review," Mathematical Problems in Engineering, pp. 1-21, 2019.
7. K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," arxiv 14091556, 2014.
8. A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolution Neural Network," Communication of ACM, vol. 60, no. 6, pp. 84-90, 2017.
9. R. Rajkumar and M. V. Sudhamani, "Content based Image Retrieval System using Combination of Color and Shape Features and Siamese Neural Network," International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 9, no. 2, pp. 71-77, 2019.
10. Z. Swati, Q. Zhao, M. Kabir, F. Ali, Z. Ali, S. Ahmed and J. Lu, "Content-Based Brain Tumor Retrieval for MR Images Using Transfer Learning," IEEE Access, vol. 7, pp. 17809 - 17822, 2019.
11. S. J. Hassen, A. Taha and M. M. Selim, "Trademark Image Retrieval using Transfer Learning," Journal of Engineering and Applied Sciences, vol. 14, no. 18, pp. 6897-6905, 2019.
12. P. Sadeghi-Tehran, P. Angelov, N. Virlet and M. Hawkesford, "Scalable Database Indexing and Fast Image Retrieval Based on Deep Learning and Hierarchically Nested Structure Applied to Remote Sensing and Plant Biology," Journal of Imaging, vol. 5, no. 3, pp. 33-54, 2019.
13. O. Mohamed, E. A. Khalid, O. Mohammed and A. Brahim, "Content-Based Image Retrieval Using Convolutional Neural Networks," Lecture Notes in Real-Time Intelligent Systems, 2019.
14. X. Xie, B. Li and X. Wei, "Ship Detection in Multispectral Satellite Images Under Complex Environment," Remote Sensing, vol. 12, no. 5, p. 792, 2020.

15. W. Ao, Y. Fu, X. Hou and F. Xu, "Needles in a Haystack: Tracking City-Scale Moving Vehicles from Continuously Moving Satellite," *IEEE Transactions on Image Processing*, 2019.
16. D. Bloisi, L. Iocchi, M. Fiorini and G. Graziano, "Automatic Maritime Surveillance with Visual Target Detection," *Computer Science Journal*, 2011.
17. D. D. Bloisi, F. Previtali, A. Pennisi, D. Nardi and M. Fiorini, "Enhancing Automatic Maritime Surveillance Systems with Visual Information," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 4, pp. 1-10, 2017.
18. L. X. Q. L. X. and Z. N. (. S. D. C. o. O. S. i. t. B. S. B. o. S. R. S. a. G. J. o. C. R. Bing, "Spatial Distribution Characteristics of Oil Spills in the Bohai Sea Based on Satellite Remote Sensing and GIS," *Journal of Coastal Research*, vol. 90, pp. 164-170, 2019.
19. M. Reggiannini and L. Bedini, "Multi-Sensor Satellite Data Processing for Marine Traffic Understanding," *Electronics*, vol. 8, no. 2, pp. 152-170, 2019.
20. R. K. Tripathi, A. S. Jalal and S. C. Agrawal, "Suspicious human activity recognition: a review," *Artificial Intelligence Review*, vol. 50, no. 2, pp. 283-339, 2018.
21. T. Saba, "Real time anomalies detection in crowd using convolutional long short-term memory network," *Journal of Information Science*, 2021.
22. K. Madasamy, V. Shanmuganathan, V. Kandasamy, M. Y. Lee and M. Thangadurai, "OSDDY: embedded system-based object surveillance detection system with small drone using deep YOLO," *EURASIP Journal on Image and Video Processing*, vol. 19, no. 1, 2021.
23. P. Gupta, B. Pareek, G. Singal and D. V. Rao, "Edge device based Military Vehicle Detection and Classification from UAV," *Multimedia Tools and Applications*, vol. 81, no. 14, p. 19813–19834, 2022.
24. M. Jouhari, A. Al-Ali, E. Baccour, A. Mohamed, A. Erbad, M. Guizani and M. Hamdi, "Distributed CNN Inference on Resource-Constrained UAVs for Surveillance Systems: Design and Optimization," *IEEE Internet of Things Journal*, pp. 1-16, 2022.
25. H. Gupta and O. Verma, "Monitoring and surveillance of urban road traffic using low altitude drone images: a deep learning approach," *Multimedia Tools and Application*, vol. 81, pp. 19683-19703, 2022.

Three-dimensional structure extraction and evaluation of microvessels in cardiac tissue imaged via confocal microscopy

Shotaro Kaneko¹, Yuichiro Arima², Masahiro Migita³, and Masashi Toda³

¹ Graduate School of Science and Technology, Kumamoto University, Kumamoto 860-8555, Japan

² Dept. of Cardiovascular Medicine, Kumamoto University, Kumamoto 860-0811, Japan

³ Center for Management of Information Technologies, Kumamoto University, Kumamoto 860-8555, Japan
228d8710@st.kumamoto-u.ac.jp

Abstract. Because cardiac disease accounts for 15% of all deaths in Japan, there is an urgent need to elucidate its pathogenesis and establish new treatment methods. However, the mechanisms underlying the pathogenesis of cardiac disease are still unclear. With the advancement of computer technology in recent years, elucidation of the pathophysiology at the cellular level in the heart is expected. Many previous studies investigated myocardial tissues, which is a three-dimensional body, in two dimensions. Therefore, in this study, we propose ‘vascular straightness’ and ‘angle between cardiomyocyte nucleus and vascular skeleton’, which quantitatively evaluate the vascular structure and the relationship between cardiomyocytes and blood vessels, respectively, by performing three-dimensional analysis of microscopic images of myocardial tissues of mice. We calculated the proposed indices in wild-type (WT) and knockout (KO) mice and confirmed the differences. In the future, we intend to test for significant differences using statistical analysis, propose indices for the quantitative evaluation of the vessel thickness and surface structure, and classify mice into healthy mice and mice with heart defects based on the proposed indices.

Keywords: Three-dimensional medical imaging, cardiac disease, microvascular.

1 Introduction

Heart disease accounts for 15% of all deaths in Japan, second only to malignant neoplasms (tumours), and the mortality rate is increasing every year [1]. Heart failure accounts for approximately half of all deaths owing to heart disease [1]. Although drug therapy for heart failure has made steady progress compared to the past, it is often ineffective for severe cases of heart failure, and no established treatment other than heart transplantation is currently available [2]. Heart transplantation has been initiated in Japan. However, the number of donors is significantly fewer than that required for transplantation, and it is difficult to imagine that this treatment will be-

come widely available in the future [2]. Therefore, there is an urgent need to elucidate the pathogenesis of heart failure and establish new treatment methods.

With the development of computer technology, the pathophysiology of heart failure is expected to be clarified at the cellular level. Myocardial cells, the main constituents of the heart, have been used to reveal the pathophysiology of heart failure. However, because it is difficult to use human cardiomyocytes, studies using cardiomyocytes from mice, which are also mammals, have been conducted. A representative study reported the cross-sectional area of cell nuclei from myocyte images of mouse cardiomyocytes captured using a microscope under conditions similar to heart failure [3]. However, most studies investigated cardiomyocytes, which are three-dimensional bodies, in two dimensions. Therefore, additional information might be obtained by conducting three-dimensional analysis of tiny cells, such as cardiomyocytes. In addition, because cardiomyocytes and other cells coexist in the heart, it is desirable to classify nucleated cells and analyse their volume and positional relationships as three-dimensional information. Single-cell analysis [4] has revealed the mechanism of cardiomyocyte hypertrophy and failure by machine learning. However, single-cell analysis is expensive and time-consuming, as it isolates cells and focuses on a single cell. In addition, because the cells are isolated, it is impossible to obtain information on their original shape and positional relationship when they exist as cardiac tissues. A representative three-dimensional analysis of nuclei was performed by Alexandr et al. [5]. In their study, modelling, analysis, and classification of cell nuclei and nucleoli were performed in three dimensions. They compared the morphology of serum-starved and proliferating fibroblasts, followed by a comparison of epithelial and mesenchymal human prostate cancer cell lines to classify fibroblasts, and epithelial and mesenchymal cells, respectively. However, the myocardium was not investigated, and only morphological information (for example, volume and surface area) of cell nuclei and nucleoli was utilised.

In this study, we examined a method for three-dimensional image analysis of cellular tissues of the myocardial region. Fig. 1 shows a myocardial tissue microscopy image of the mouse used in this study. The contributions of this study are as follows:

- Regions complemented by missing holes in endothelial cell membrane images were extracted, in which no contrast agent was used, and the areas inside blood vessels and where endothelial cell nuclei exist are not fluorescent.
- To quantitatively evaluate the structure of blood vessels and the relationship between cardiomyocytes and blood vessels, we proposed two indices: “straightness of blood vessels” and “angle between the cardiomyocyte nuclei and blood vessels.”
- We calculated the index values of the proposed method in wild-type (WT) and knockout (KO) mice and confirmed the difference between the two.

2 Method

2.1 Extraction of vascular regions

Fig. 2(a) shows a stained image of a vascular endothelial cell membrane. This image was smoothed using an anisotropic diffusion filter [6, 7] (Fig. 2(b)). The Otsu method [8], a widely used binarisation method, does not successfully extract the vascular endothelial region because of differences in the fluorescent intensity values for different locations, even in the vascular endothelial cell membrane. Therefore, adaptive thresholding [9] was used in this study (Fig. 2(c)). Fig. 2(d) shows the endothelial region image after noise removal and other processing steps for the binarised image. The endothelial cell nuclei and interior of the vessels were not stained, resulting in the appearance of missing holes (Fig. 3(a),(b)). This explains why the centre line cannot be captured satisfactorily when skeletonisation is performed, as described below. Therefore, it is necessary to fill these closed regions. Filling in the closed regions is performed in the xy-, yz-, and zx-planes. If the edges or vertices of a pixel touch, the pixel is connected [10]. Figs. 2(e) and 3(c) show the endothelial region after filling the closed region.

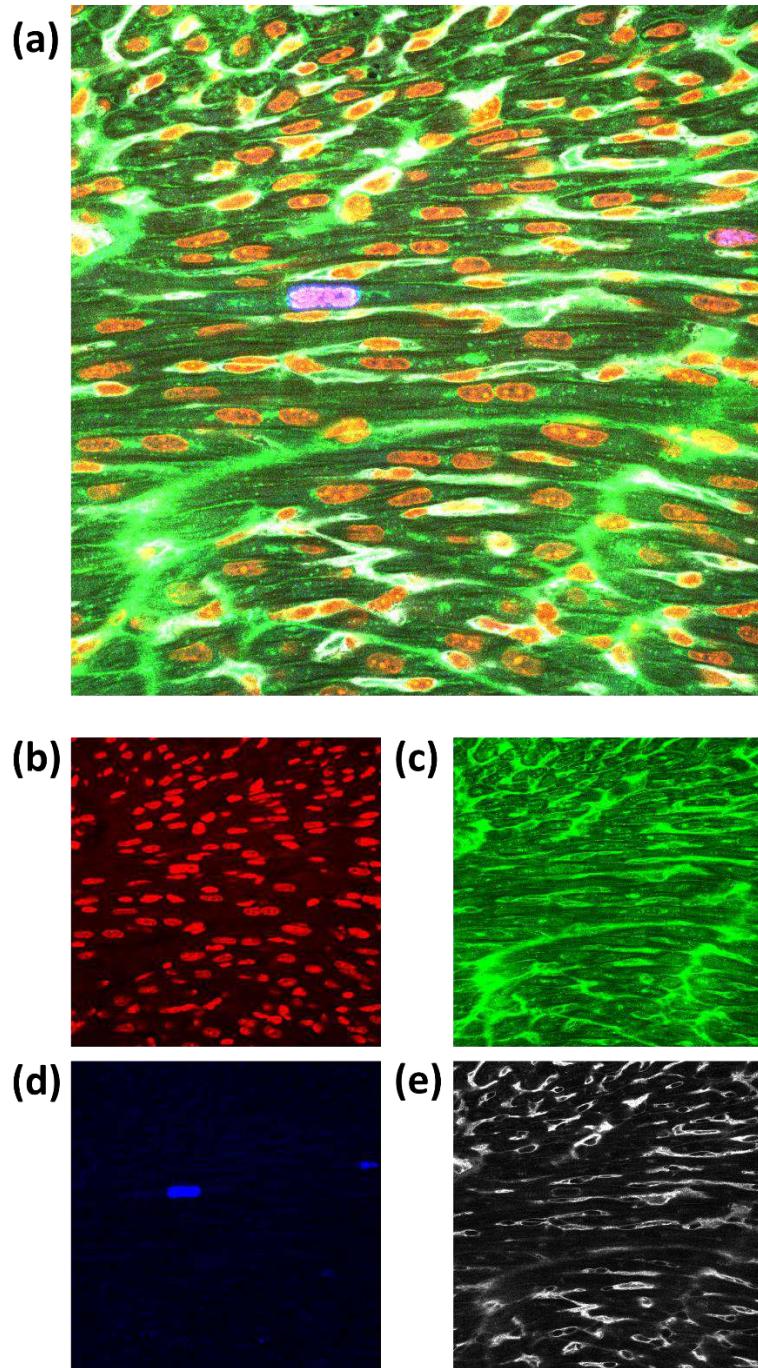


Fig. 1. (a) Myocardial tissue microscopic image. (b) Cell nucleus. (c) Cell membrane. (d) Dividing cell nucleus. (e) Vascular endothelial cell membrane.

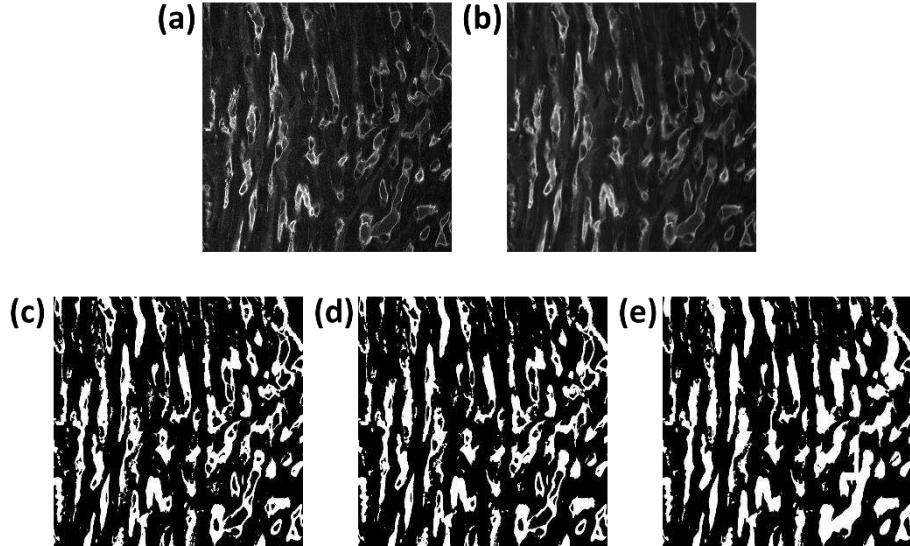


Fig. 2. (a) Vascular endothelial cell membrane. (b) Membrane smoothed with anisotropic diffusion filter. (c) Binarisation with adaptive thresholding. (d) Vascular endothelial area after noise removal and other processes. (e) Vascular endothelial area after hole filling.

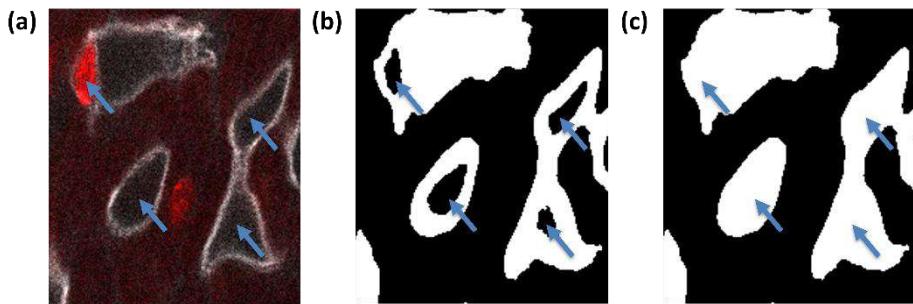


Fig. 3. (a) Cell nuclei (red) and vascular endothelial cell membrane (white). (b) Vascular endothelial area before hole filling. (c) Vascular area after hole filling.

2.2 Blood vessel region skeletonisation and bifurcation point extraction

Skeletonisation [11, 12], a method for determining the centreline of a vessel, was performed on the endothelial region of a vessel after hole filling, extracted in the previous section, to obtain the basic structure of the vessel.

After skeletonisation of the vascular region, branching points of the skeleton were identified. A bifurcation point is a voxel where multiple branches intersect [13, 14]. Fig. 4 shows the extracted vessel region (grey), the vessel skeleton (magenta), and the bifurcation point (cyan). In addition, a single-vessel skeleton is defined here as the end between or from branch points (Fig. 5).

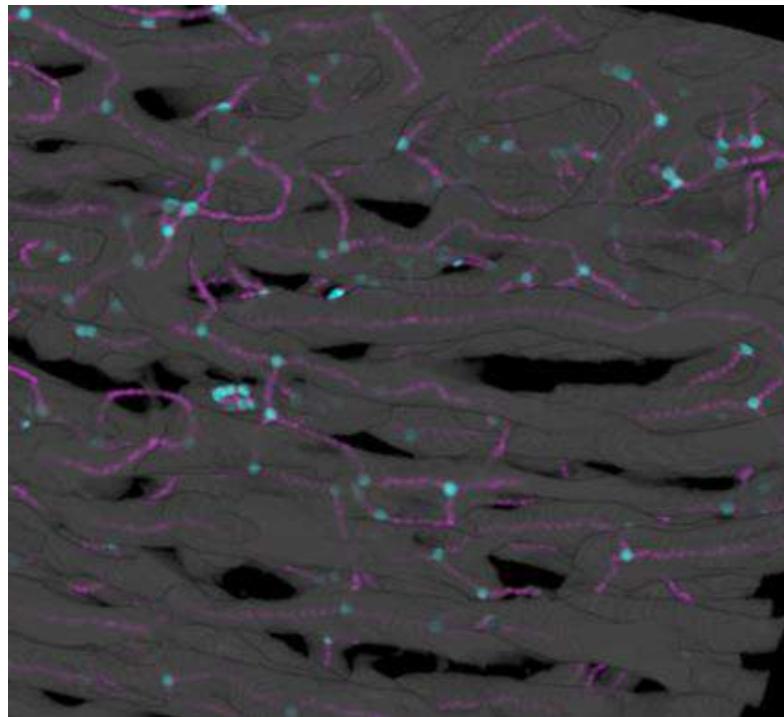


Fig. 4. Extracted blood vessel regions (grey), blood vessel skeleton (magenta), and bifurcation points (cyan).

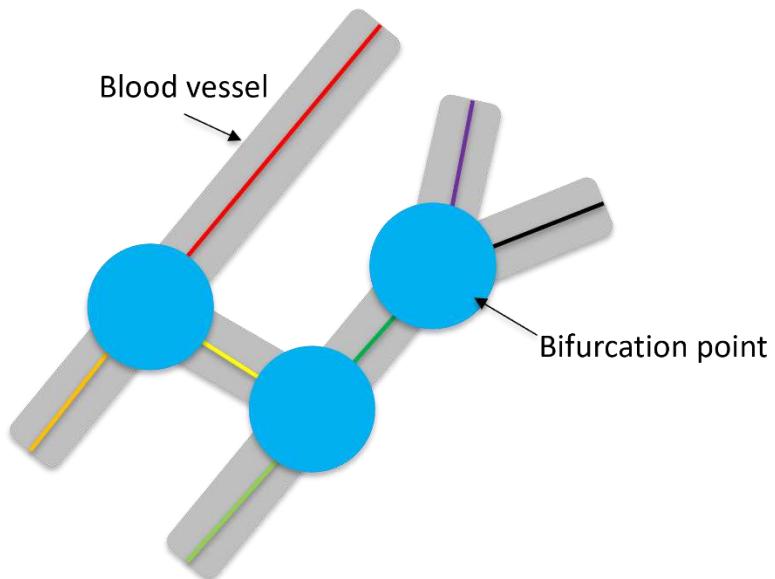


Fig. 5. Diagram of single-blood vessel skeleton. In this figure, there are seven vessel skeletons.

2.3 Cell Nucleus Classification

The myocardial tissue microscopic images used in this study contained vascular endothelial cells, smooth muscle cells, fibroblasts, and other cells, in addition to myocardial cells. Therefore, it is necessary to classify cells by type and identify the cardiomyocytes to analyse the relationship between cardiomyocytes and blood vessels. Currently, it is difficult to extract cellular regions from stained images of cell membranes; therefore, analysis is conducted on cell nuclei, from which it is relatively easy to extract regions. To classify cell nuclei, we utilised a method for identifying vascular endothelial cells from myocardial tissue microscopy images of mice [15]. This method classifies cell nuclei into vascular endothelial cell nuclei and other types of cell nuclei based on the ‘coverage ratio’, which quantitatively expresses how much of the cell nucleus is covered by the vascular endothelial area, and can classify the nuclei into vascular endothelial cell nuclei and other types with approximately 85% accuracy. Vascular endothelial cell nuclei are covered by the vascular endothelium, whereas cardiomyocyte nuclei are not covered by vascular endothelium. Based on these characteristics, this study classified cardiomyocyte nuclei into three clusters: vascular endothelial cell nuclei with high coverage, cardiomyocyte nuclei with low coverage, and other cell nuclei.

2.4 Quantitative index for evaluating three-dimensional vascular structure

Vascular straightness

We propose ‘straightness of blood vessels’ as an index to quantitatively evaluate the three-dimensional structure of blood vessels. The straightness is calculated as follows:

$$s = \frac{d}{L} \quad (1)$$

where d is the Euclidean distance between the two ends of the vessel skeleton, and L is the length of the vessel skeleton.

Therefore, the closer the straightness value is to 1, the straighter the vessel.

Angles between myocardial cell nuclei and vascular skeleton

We propose the ‘angle between the cardiomyocyte nucleus and vascular skeleton’ as a quantitative measure of the relationship between cardiomyocytes and blood vessels. First, the long-axis direction of the cardiomyocyte nucleus was determined. In this study, principal component analysis [16] was used for a group of voxels constituting the myocardial cell nucleus, and the first principal component derived was used as the long-axis vector.

Next, the orientation of the vessel skeleton was determined. The two ends of the voxels comprising a single-vessel skeleton were determined, and the vector between these two points was considered the vector of the vessel skeleton.

Finally, the vessel skeleton corresponding to an individual myocyte nucleus and the angle between the vectors were determined. In the case of myocardial cell nuclei and blood vessels that are significantly far apart, it is difficult to assume that they interact.

In addition, it has been difficult to accurately extract cellular regions from cell membrane images, and the analysis is currently performed using myocardial cell nuclei. Therefore, we defined a region corresponding to each myocardial cell nucleus and focused only on blood vessels within that region. To determine the region corresponding to each myocyte nucleus, we used a three-dimensional Voronoi partition [17] with respect to the centre of gravity of all myocyte nuclei. Because the skeleton is similar to a line passing through the approximate centre of a blood vessel, many cardiomyocyte nuclei have no blood vessels within the three-dimensional Voronoi region when the inside–outside region was determined for the vessel skeleton. Because it is physiologically unlikely that no blood vessels correspond to a cardiomyocyte nucleus, an in/out judgement was performed on the surfaces of the blood vessels. Edge detection [18] was used to assess the vessel surface. For each vessel surface voxel, we assigned a labelling number corresponding to the labelling number of the vessel skeleton at the closest distance. The angles between the long-axis direction of the myocardial cell nucleus and the labelling number of the vessel surface within the three-dimensional Voronoi region and the corresponding vessel skeleton were then calculated. The angle is obtained by [19] as follows:

$$\theta = \cos^{-1} \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}| |\mathbf{b}|} \quad (2)$$

where \mathbf{a} is the direction of the long axis of the cardiomyocyte nucleus, \mathbf{b} is the direction of the vascular skeleton, and $0^\circ \leq \theta \leq 90^\circ$. Fig. 6 shows an example of a cardiomyocyte nucleus and the corresponding blood vessel. The cardiomyocyte nucleus is represented by magenta, the direction of the long axis of the cardiomyocyte nucleus by a black arrow, the three-dimensional Voronoi region by green, the surfaces of the blood vessels within that region by red, and the surface of blood vessels outside the region by blue.

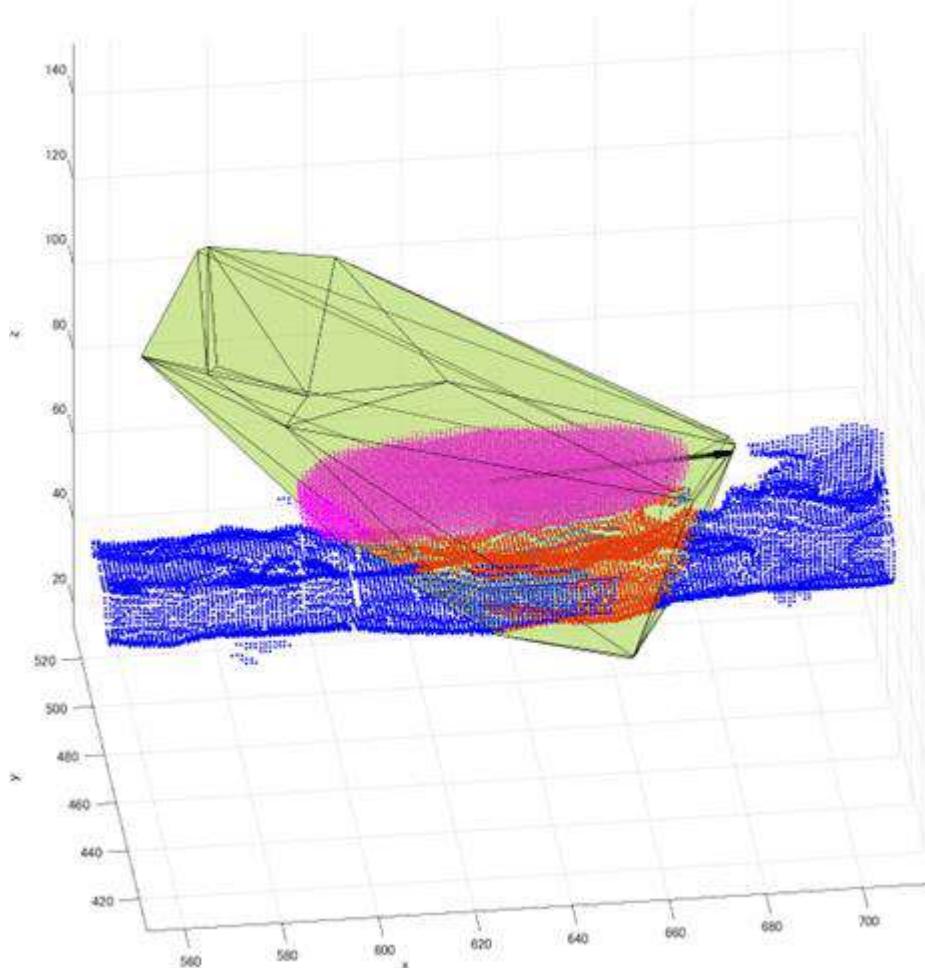


Fig. 6. Examples of myocardial cell nuclei and corresponding blood vessels.

3 Experiment

Table 1 lists the equipment and development environment used in the experiment. In this study, microscopic images of the hearts of neonatal mice were analysed. Microscopic images of cell nuclei, cell membranes, dividing cell nuclei, and vascular endothelial cell membranes were captured, each with fluorescence.

The experiment was performed using myocardial images of WT (normal) and KO mice of the same number of postnatal days. The KO mice were genetically modified to prevent them from producing HMG-CoA Synthase 2, an enzyme essential for synthesising ketone bodies, so that ketones are unavailable, and metabolism is affected [20]. The samples were three WT mice and three KO mice. Three images were captured from different parts of each individual.

The images of cell nuclei and vascular endothelial cells were fluorescent. Table 2 lists the details of the datasets. In this experiment, the straightness of the blood vessel and the angle between the myocardial cell nucleus and vascular skeleton were obtained using the proposed method for the dataset. The analysis was performed for blood vessel skeletons with lengths of 50–99 pixels (9.0188–17.8571 microns) and more than 100 pixels (18.0375 microns).

Table 1. Equipment and development environment

OS	MS Windows 10
CPU	i7-8700K (3.70 GHz)
RAM	40.0 GB
Development environment	MATLAB R2020b

Table 2. Details of dataset

Format	TIFF
Resolution (Width × Height × Depth)	1,024 × 1,024 × 169 (However, the depth of 221107_WT06_01 is 139.)
1 voxel (Width × Height × Depth)	0.1803752 × 0.1803752 × 0.2985004 (microns)
Colour type	Grayscale
Target	Neonatal murine
Shooting location	Myocardial tissue of the left atrium
Shooting object	Cell nucleus, cell membrane, dividing cell nucleus, and vascular endothelial cell membrane
Shooting equipment	Confocal laser scanning microscopy

4 Results

4.1 Results of cell nucleus classification

Table 3 lists the classification values of cell nuclei into three clusters based on coverage: vascular endothelial cell nuclei, myocardial cell nuclei, and other cell nuclei.

Table 3. Results of cell nucleus classification

Sample name	Vascular endothelial cell nuclei	Myocardial cell nuclei	Other cell nuclei	Total
221107_WT04_01	951 (35.0%)	1,176 (43.2%)	593 (21.8%)	2,720
221107_WT04_02	855 (30.7%)	1,417 (50.9%)	511 (18.4%)	2,783
221107_WT04_03	833 (23.5%)	2,150 (60.7%)	557 (15.7%)	3,540
221107_WT06_01	757 (40.6%)	784 (42.1%)	323 (17.3%)	1,864
221107_WT06_02	748 (32.8%)	1,068 (46.8%)	464 (20.4%)	2,280
221107_WT06_03	665 (32.2%)	1,064 (51.6%)	334 (16.2%)	2,063
221117_WT7_01	715 (40.1%)	825 (46.2%)	244 (13.7%)	1,784
221117_WT7_02	721 (33.2%)	918 (42.3%)	530 (24.4%)	2,169
221117_WT7_03	613 (34.3%)	816 (45.7%)	358 (20.0%)	1,787
221107_KO2_01	845 (36.8%)	1,055 (46.0%)	395 (17.2%)	2,295
221107_KO2_02	945 (41.8%)	968 (42.8%)	347 (15.4%)	2,260
221107_KO2_03	722 (48.1%)	647 (43.1%)	131 (8.7%)	1,500
221107_KO05_01	637 (35.5%)	840 (46.8%)	319 (17.8%)	1,796
221107_KO05_02	757 (38.2%)	847 (42.8%)	377 (19.0%)	1,981
221107_KO05_03	519 (33.9%)	720 (47.0%)	294 (19.2%)	1,533
221117_KO2_01	789 (43.5%)	789 (43.5%)	234 (12.9%)	1,812
221117_KO2_02	650 (33.8%)	979 (50.9%)	296 (15.4%)	1,925
221117_KO2_03	638 (33.5%)	781 (41.0%)	487 (25.6%)	1,906

4.2 Results for blood vessel straightness and angle between myocardial cell nucleus and blood vessels

Fig. 7 shows the results for the straightness of the vessels, and Fig. 8. shows the values of the angle between the myocardial cell nucleus and vascular skeleton. Regarding the accuracy of the vascular area extraction, skeletonisation, and bifurcation point extraction, we had an expert examine at some of the data, who confirmed that there were no problems. The straightness visualisation shown in Fig. 9 indicates that the closer the straightness value is to 1, the straighter the blood vessel structure. We compared the differences in the proposed quantitative index between the WT and KO mice and found that both the straightness of blood vessels and the angle between the myocardial cell nuclei and blood vessels differed between samples obtained at different locations in the same individual as well as between individuals. This is attributed to the fact that some images showed blood vessels extending along the z-axis, whereas others showed vessels extending in the xy-plane. Moreover, the direction of the blood vessels is not constant, depending on the area being photographed.

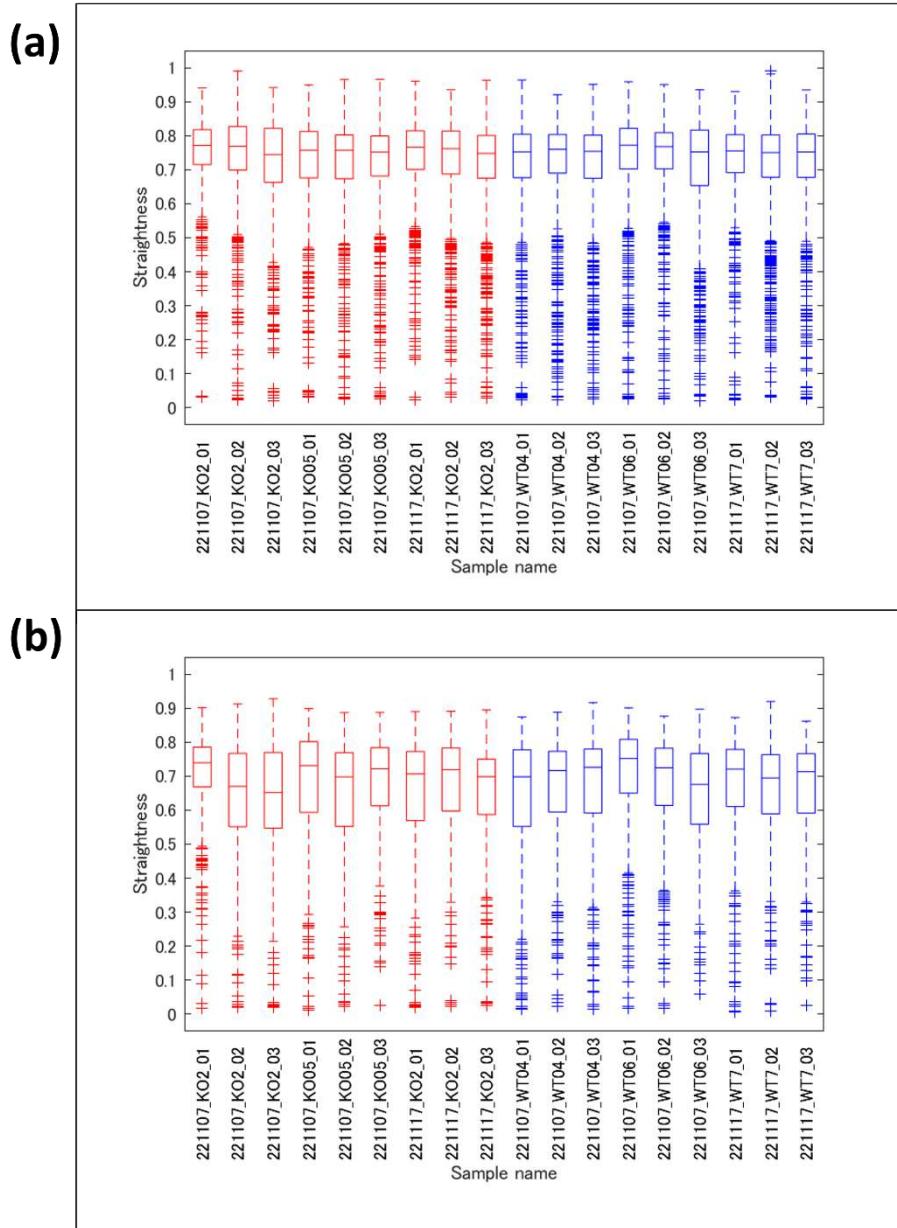


Fig. 7. Vascular straightness. (a) Vascular skeleton length of 50–99 pixels (9.0188–17.8571 microns). (b) Vessel skeleton length exceeding 100 pixels (18.0375 microns).

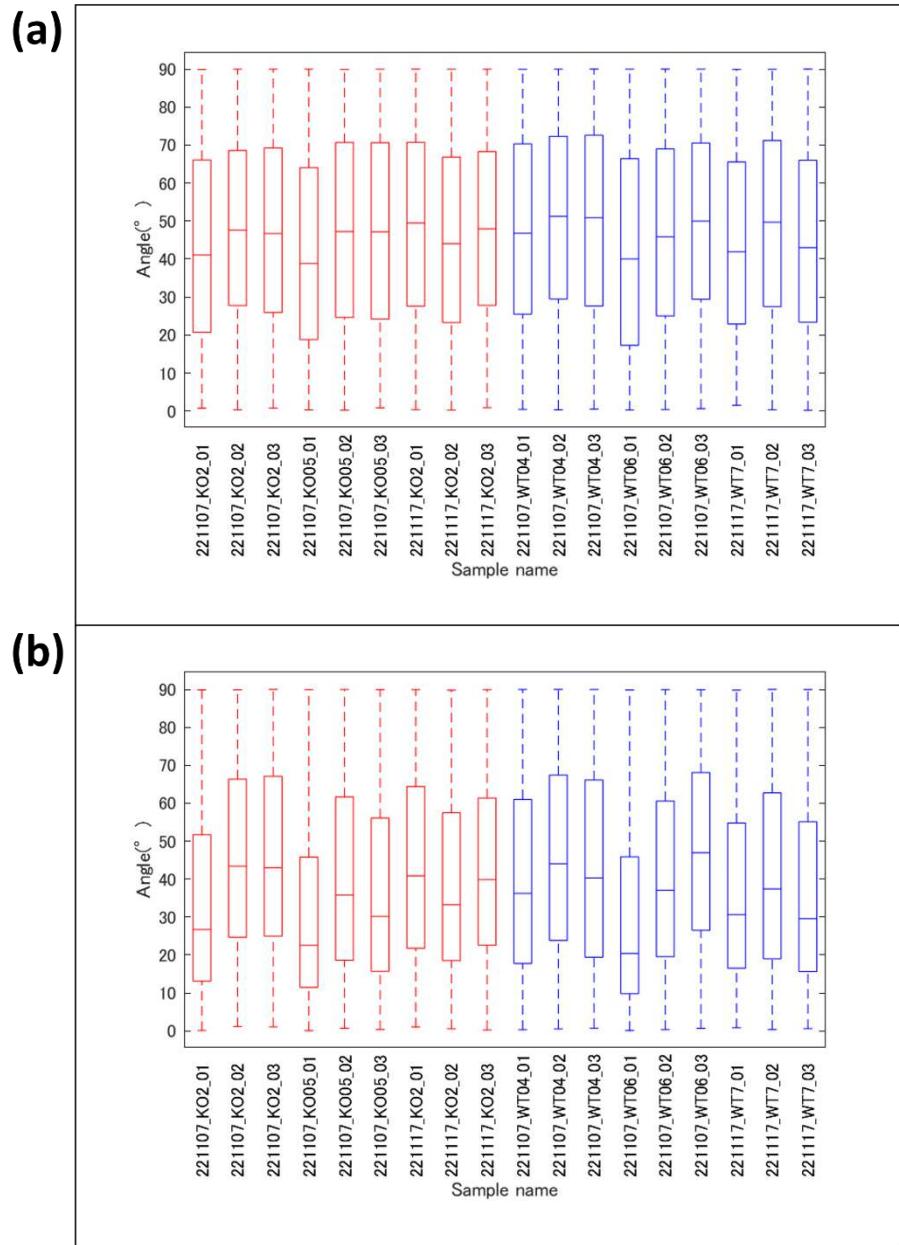


Fig. 8. Angles between myocardial cell nuclei and vascular skeleton. (a) Vascular skeleton length of 50–99 pixels (9.0188–17.8571 microns). (b) Vascular skeleton length exceeding 100 pixels (18.0375 microns).

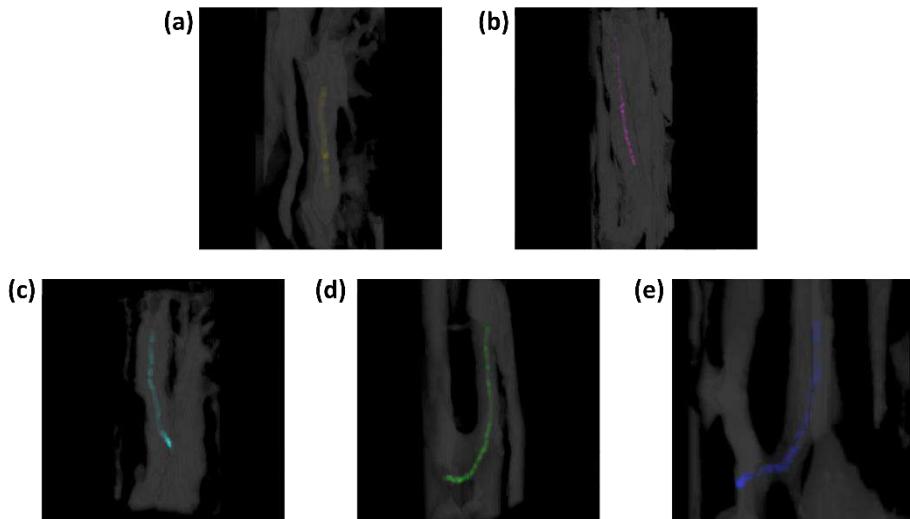


Fig. 9. Examples of straightness visualization. (a) $0.9 \leq s \leq 1.0$. (b) $0.8 \leq s < 0.9$. (c) $0.7 \leq s < 0.8$. (d) $0.6 \leq s < 0.7$. (e) $0.5 \leq s < 0.6$.

5 Conclusion

In this study, we performed a three-dimensional analysis of cellular tissue images of the mouse myocardium, considering the positional relationship between cells and blood vessels. We proposed ‘straightness of blood vessels’ and ‘angle between myocardial cell nucleus and vascular skeleton’ as indices for quantitative evaluation of vascular structure and the relationship between myocardial cells and blood vessels. The results confirmed that differences were observed between WT and KO mice.

In future studies, it will be necessary to conduct a statistical analysis to determine whether significant differences exist between WT and KO mice. In addition, we intend to analyse indices that can be used to quantitatively evaluate the thickness and surface structure of blood vessels and compare them between WT and KO mice. Furthermore, we hope to discriminate between healthy mice and mice with heart defects based on these indices.

References

1. Ministry of Health, Labour and Welfare, Summary of the 2020 Vital Statistics Monthly Report Annual Total (Approximate), <https://www.mhlw.go.jp/toukei/saikin/hw/jinkou/geppo/nengai20/dl/gaikyouR2.pdf>, last accessed 2023/01/23.
2. Ministry of Education, Culture, Sports, Science and Technology (MEXT) Strategic Research on Heart Failure Elucidation of the Pathogenesis of Heart Failure Using Developmental Engineering and Gene and Cell Therapy, https://www.mext.go.jp/a_menu/shinkou/hojyo/1300506.htm, last accessed 2023/01/23.

3. Bao N. P., Wataru K., Shalini A. M., Jesung M., James F. A., Kate L. P., David G., Beverly A. R., Rui C., Joseph A. G., Celio X. S., SuWannee T., Eiichiro M., Michael T. K., Paul M. R., Serena Z., Shibani M., David J. C., Ahmed I. M., Mauro G., Peter S. R., Asaithamby A., Ajay M. S., Luke I. S., Hesham A. S.: The Oxygen Rich Postnatal Environment Induces Cardiomyocyte Cell Cycle Arrest Through DNA Damage Response. *Cell*, Vol. 157, No. 3, pp.565-579 (2014).
4. Nomura S., Satoh M., Fujita T. et al.: Cardiomyocyte Gene Programs Encoding Morphological and Functional Signatures in Cardiac Hypertrophy and Failure. *Nature Communication* 9, 4435 (2018). <https://doi.org/10.1038/s41467-018-06639-7>
5. Kalinin A.A., Allyn-Feuer A., Ade A. et al.: 3D Shape Modeling for Cell Nuclear Morphological Analysis and Classification. *Sci Rep* 8, 13658 (2018). <https://doi.org/10.1038/s41598-018-31924-2>
6. Perona, P., J. Malik. Scale-Space and Edge Detection Using Anisotropic Diffusion. *IEEE® Transactions on Pattern Analysis and Machine Intelligence*. Vol. 12, No. 7, pp. 629-639 (1990).
7. Gerig, G., O. Kubler, R. Kikinis, and F. A. Jolesz. Nonlinear Anisotropic Filtering of MRI Data. *IEEE Transactions on Medical Imaging*. Vol. 11, No. 2, pp. 221-232 (1992).
8. Otsu, N., A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*. Vol. 9, No. 1, pp. 6266 (1979).
9. Bradley, D., G. Roth, Adapting Thresholding Using the Integral Image, *Journal of Graphics Tools*. Vol. 12, No. 2, pp.13-21 (2007).
10. Soille, P., *Morphological Image Analysis: Principles and Applications*, Springer-Verlag, pp. 173-174 (1999).
11. Lee, T.-C., Kashyap, R. L., Chu, C.-N. Building Skeleton Models via 3-D Medial Surface/Axis Thinning Algorithms. *Computer Vision, Graphics, and Image Processing*, Vol. 56, No. 6, pp. 462-478 (1994).
12. Kerschnitzki, M, Kollmannsberger, P, Burghammer, M. et al. Architecture of the Osteocyte Network Correlates with Bone Material Quality. *Journal of Bone and Mineral Research*, Vol. 28, No. 8, pp.1837-1845 (2013).
13. Haralick, R. M., Shapiro, L. G. *Computer and Robot Vision*, Vol. 1, Addison-Wesley, 1992.
14. Kong, T. Y. and Rosenfeld, A. *Topological Algorithms for Digital Image Processing*, Elsevier Science, Inc., 1996.
15. Kaneko, S., Arima, Y., Migita, M., Toda, M. Proposal of a Method to Identify Vascular Endothelial Cells from Images of Mouse Myocardial Tissue. In: Sumi, K., Na, I.S., Kaneko, N. (eds) *Frontiers of Computer Vision. IW-FCV 2022. Communications in Computer and Information Science*, vol 1578. Springer, Cham, 2022. https://doi.org/10.1007/978-3-031-06381-7_12
16. Jolliffe, I. T. *Principal Component Analysis*. 2nd ed., Springer, 2002.
17. Barber, C. B., Dobkin, D. P., Huhdanpaa, H. T. The Quickhull Algorithm for Convex Hulls, *ACM Transactions on Mathematical Software*, Vol. 22, No. 4, p. 469-483 (1996).
18. Canny, J. A Computational Approach to Edge Detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 6, pp. 679-698 (1986).
19. I don't know how to find the angle formed by two vectors, what kind of calculation method is there. <https://jp.mathworks.com/matlabcentral/answers/896337->, accessed on 2023/01/23.
20. Yuichiro Arima - IRCMS - Kumamoto University, https://ircms.kumamoto-u.ac.jp/research/yuichiro_arima/, accessed on 2023/01/23.

Multi-Attributed Face Synthesis for One-Shot Deep Face Recognition

Muhammad Shaheryar^{1[0000-0003-3992-1387]}, Lamyanya
 Laishram^{1[0000-0002-0324-214X]}, Jong Taek Lee^{1[0000-0002-6962-3148]}, and Soon
 Ki Jung^{1[0000-0003-0239-6785]}

School of Computer Science and Engineering, Kyungpook National University,
 Daegu, Republic of Korea
 {shaheryar, yanbalaishram, jongtaeklee, skjung} @knu.ac.kr

Abstract. Nothing is more unique and crucial to an individual's identity than their face. With the rapid improvement in computational power and memory space and recent specializations in deep learning models, images are becoming more essential than ever for pattern recognition. Several deep face recognition models have recently been proposed to train deep networks on enormously big public datasets like MSCeleb-1M [8] and VG-GFace2 [5], successfully achieving sophisticated performance on mainstream applications. It is particularly challenging to gather an adequate dataset that allows strict command over the desired properties, such as hair color, skin tone, makeup, age alteration, etc. As a solution, we devised a one-shot face recognition system that utilizes synthetic data to recognize a face even if the facial attributes are altered. This work proposes and investigates the feasibility of creating a multi-attributed artificial face dataset from a one-shot image to train the deep face recognition model. This research seeks to demonstrate how the image synthesis capability of the deep learning methods can construct a face dataset with multiple critical attributes for a recognition process to enable and enhance efficient face recognition. In this study, the ideal deep learning features will be combined with a conventional one-shot learning framework. We did experiments for our proposed model on the LFW and multi-attributed synthetic data; these experiments highlighted some insights that can be helpful in the future for one-shot face recognition.

Keywords: Deep Learning · Computer Vision · One-Shot Face recognition · Siamese Networks · Image Classification

1 Introduction

The practical significance and great theoretical interest from cognitive scientists have been precisely the reason why facial recognition systems have been the target of such great curiosity and attention for the past few decades, making it impossible to disregard their importance as a non-contact verification method. It has expanded its usage in a variety of digital media, including video indexing,

video analytics, and security departments. Face identification and face verification are often two sub-tasks that make up face recognition. Three phases are involved in each task: face detection, feature extraction, and classification.

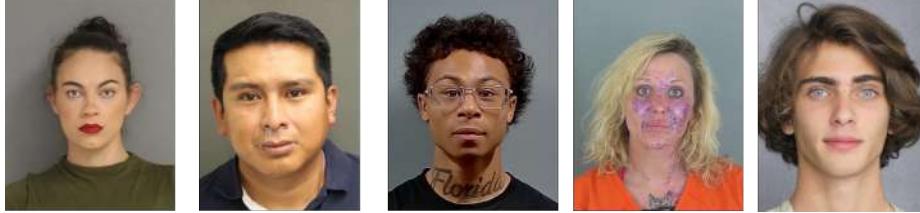


Fig. 1. Examples of Face images in the Smoking Gun's mug shot collection [24]

It is pretty daunting to build a face recognizer, especially with a small dataset. Even though there has been a significant advancement in face detection, problems still prevent the technology from being as accurate as a human. Early research created shallow models with basic facial features, while contemporary face recognition methods are considerably improved and powered by deep CNNs.

A deep convolutional network promises to attain greater accuracy using a straightforward classification approach, but it requires a lot of memory to train. One of the significant challenges is the very volatile and unbalanced amount of training data, where certain classes in the dataset may have a lot of photos. In contrast, others may have relatively few, affecting the quality of the results severely. The latest expansion of methods based on deep learning, including generative adversarial networks (GANs) [10,15], can solve the problem of varying dataset sizes by producing realistic facial images while accepting appropriate control parameters. Additionally, there are other issues, such as various humans having remarkably similar appearances and the fact that the faces of the same person may look very different due to lighting, position, and age variations. Although other deep network approaches can also handle variations in pose, lighting, and facial expressions, their requirement of a significant quantity of annotated data to train the system will always be a considerable drawback. Since GANs continue to be successful in producing artificial data for computer vision tasks [26], a new field of biometric research is beginning to explore how synthetic face images might be produced and utilized to train FR models. Face attribute adjustment is possible with the encoder-decoder architecture by decoding the encoder's latent space representation based on the given attributes. Compared to other synthetic data generation challenges, this research challenges assigning an identity to the synthetically generated faces to render them usable while ensuring variations within identity.

The fundamental pillar of this research is developing a hybrid approach for one-shot face recognition that, while sustaining the true identity, allows for an accurate modification of 14 various multi-attributes of any specified face. One

can utilize our strategy to increase the variety of single faces in a dataset and strengthen face recognition algorithms. This one-shot-based deep face recognition (OS-DFR) method is distinct from typical face synthesizing methods and seeks to learn the synthetic features without giving the original characteristics. Motivated by the ATTGAN [10]’s success in generating realistic facial attributes, OS-DFR integrates the two tasks, one-shot synthetic face generation and face recognition. According to the statistical link between synthetic characteristics and face identification, this method successfully achieves the aim of deep face recognition. It is crucial when only one sample is available for a particular person. Face images generated from the ”MugShots” for evaluating the performance of a recognition job and benchmarking are soon to be proposed as our dataset that includes unconstrained face images. The key contributions to this work are:

- We provided a technique for multi-attributed face synthesis for one-shot face recognition, employing synthetic data to replace augmentation approaches for development of realistic and feature enriched images of a person. To the best of our knowledge, this is the first instance of one-shot facial recognition using multi-attributed synthetic data.
- We empirically verified the effectiveness of the approach for multi-attributed synthetic data for face recognition in the real world.

The remainder of the paper is organized as follows: In Section 2, prior studies on the creation of synthetic data, one-shot face recognition, and the use of synthetic data for deep neural network training are reviewed. A thorough explanation of the suggested technique, including an explanation of the network and the created synthetic dataset, is given in Section 3. The outcomes of our methodology are presented in Section 4, along with experimental settings and details. The ideas and algorithms created in this study are summarized in Section 5, which is then followed by a brief discussion of prospective future work.

2 Related Work

The majority of this section covers the current status of one-shot learning in the literature. The most recent low-shot learning work [23], [29] also garners a lot of interest in the broader image recognition scenario. The authors divided the ImageNet data2 into the base and low-shot (referred to as new in [23]) classes, and the goal is to recognize images from both the base and low-shot classes. Their benchmark job is quite similar to one-shot face recognition but in the broader image recognition domain. Since the domain is really distinct from ours, their approach is pretty different from ours.

Overall, one-shot learning remains an unresolved issue. A natural source of information is obtained from new data in numerous ways through ”data manufacturing” [2]. There have been several works that tackle this issue in recent years. With little data, transfer learning is a viable method that encourages the usage of deep CNNs in several disciplines. [13, 17] shows that by leveraging

information from similar tasks with more enormous datasets, CNN-based transfer learning can produce superior classification results in our work with limited datasets (target domain). In their research [7, 9], the authors proposed CNN-based novel frameworks. The primary focus of their framework was to address an issue in one-shot learning by constructing generative models to build samples to solve the underrepresented classes' problems.

Bromley et al. [4] suggested the idea of Siamese Networks for the signature verification problem, and [16] demonstrated the application of deep convolutional Siamese networks for one-shot tasks with exceptional accuracy. The approach of deep attribute encoding of faces for one-shot face recognition was proposed in another work [14]. They honed a deep CNN for face recognition using particular features of human faces, such as the face's shape, hair, and gender. One-shot face recognition using mix method of Siamese neural network and deep feature encoding was proposed in [6]. [19] demonstrated the application of deep convolutional Siamese networks for one-shot tasks with substantial accuracy. By relying on a similarity function [8] [9] based on pairs of images, this network seeks to build a deep relevant feature representation. In fact, the neural network learns to distinguish between two inputs associated with distinct classes rather than explicitly learning to categorize its input. Moreover, this network focuses on learning embeddings for the similar classes samples and we can learn semantic similarities. In order to construct a trustworthy face recognition system, the method we propose in this study integrates the concept of Deep Convolutional Siamese Networks and synthetic data generation.

The use of synthetic data in face recognition has gained popularity in recent times. The behavior of face image quality generated by [15] has been examined in [31]. Furthermore, Shen et al. [22] concluded that synthetic face images could deceive humans. The excellent quality images that GANs and their various variations [15, 19, 25, 27] can create have attracted more and more attention. To address the insufficient dataset, images generated by MorphGan [20] can severely assist with their data augmentation. There is a spike and improvement in performance by up to 9% by merely augmenting the faces with new expressions and poses – consequently addressing the issue of limited datasets. It is, however, limited to the head and expression of the face image. Three methods based on meta-learning, disentangling, and filtering were described by Zhai et al. [30] to lessen the modal difference between synthetic and real data. Then, they trained face recognition model using a hybrid of a synthetic and real dataset. Recent work proposed in the field of face recognition using synthetic data [3] has come to light, where the authors examine the viability of training face recognition algorithms using a synthetically created face dataset and raise a variety of privacy, legal, and ethical issues in relation to the gathering, use, and sharing of real biometric data.

Face recognition is abstracted into two phases. Extraction of facial features is the first phase, and estimation of the person's identification from the extracted face features is the second. Face recognition has recently paced due to the rapid development of deep convolutional neural networks and put great emphasis on

learning a clear facial feature space where faces of the same person are close to each other and faces of different people are far apart. This representative technique aimed to learn the discriminative face representations directly from the original picture space. In limited circumstances, face recognition performance has significantly increased. In order to obtain the SOTA accuracy of 97.35% , DeepFace { [23] introduced classification loss and three-dimensional normalized alignment processing in 2014 on the LFW dataset [12]. FaceNet [21] achieved 99.63% on LFW using the triplet loss function in 2015. However, there are significant problems for the use of the face recognition system in genuine unconstrained scenarios [18]. One of the most significant issues is that the quality of the input facial image might impact the system's accuracy. Even ArcFace, which is extremely strong, can only attain an accuracy of 63.22% on the RealWorld Masked Face Recognition Dataset (RMFRD) [28]. This result is based on [11], which was obtained when ArcFace was not retrained on this dataset. As a result, how to improve facial recognition in unconstrained real-world settings is now the most pressing topic.

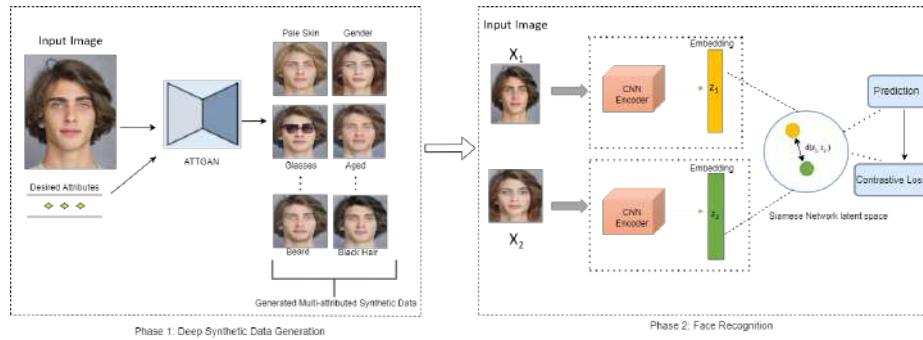


Fig. 2. Overview of the proposed approach. On the left, ATTGAN [10] is used to create synthetic face dataset from the one-shot image with multi attributes. The identity label is assigned to each generated synthetic images. The learning strategy is shown on the right, where the face recognition will be trained on the synthetic dataset using Contrastive loss. The trained model will be used for better face recognition.

3 ONE-SHOT SYNTHETIC FACE RECOGNITION

Deep learning algorithms requiring hundreds or thousands of images to bear effective results have always been one of their vast drawbacks. This section describes the method for creating and using synthetic face images to train a Face recognition model that accounts for the various subject variables, such as hair color, age, mustache etc. Two distinct phases will comprise the One-Shot based synthetic face recognition. Overall architecture can be shown in Figure. 2. In

this section, we outline these processes as well as our suggested technique for one-shot face recognition using synthetic data.

3.1 Deep Synthetic Data Generation

The primary objective of our generative model is to provide valuable auxiliary data for the one-shot classes in order to facilitate one-shot deep face recognition. We can span the feature space for these classes by doing this. We initially generated multi-attributed synthetic face data from one mugshot face image. This step utilizes an Attribute GAN (AttGAN) [10] due its high reliability in altering attributes to the generated image. The attribute classification constraint, the reconstruction loss, and the adversarial loss are combined to generate a unified AttGAN. With the knowledge of the omitted characteristics preserved, this enables alteration of the desired attributes. Overall, the encoder and decoder network's objectives are as follows:

$$\min_{G_{enc}, G_{dec}} L_{enc, dec} = \lambda_1 L_{rec} + \lambda_2 L_{cls_g} + L_{adv_g} \quad (1)$$

These hyper parameters λ_1 , λ_2 and λ_3 are used to balance the losses.

The code for AttGAN implemented in machine learning framework Tensorflow [1] is publicly accessible at <https://github.com/LynnHo/AttGAN-Tensorflow>. For additional information about implementation, please visit the website. We create our synthetic face dataset by creating 14 images for each individual from a mugshot one-shot face image, as depicted in Figure. 3. We have generated 14 images with multiple attributes from just one shot of the person. Another important characteristic of AttGAN is its direct applicability for attribute intensity control. Although AttGAN is taught using binary attribute values (0/1), its basic principle may still be used when testing with continuous attribute values. So, additionally, we produced nine photos for each feature with varying intensities, and we obtained more than 50+ synthetic face images with actual attributes for a single person from a single shot image. With a continuous input value between [0, 1], as seen in Fig. 4, the progressive shift of the generated images is natural and smooth.

3.2 Face Recognition

The convolutional Siamese network utilized in this research is constructed to learn properties of the input images independent of previous domain knowledge using very few samples from a given distribution. One-shot learning can be accomplished using a Siamese network design [6]. The twin networks' shared weights, which need fewer training parameters and reduce the possibility of overfitting, were another factor in the decision to utilize this model. For the investigation, a small labeled support set of classes used for train, test and validation.

In addition to this, several approaches may be investigated while taking into account the loss functions. One that is highly popular uses the softmax loss, whose goal is to increase the probability associated with the correct class. This

straightforward strategy, however, has poor feature derivation performance for the face recognition task. To acquire highly discriminative deep features for face recognition, Euclidean-distance-based loss is preferred because of the maximization of inter-class variance and minimization of intra-class variance. It is the main reason we choose Contrastive loss function shown in eq. 2

$$L = (1 - y) \frac{1}{2} (D_w)^2 + (Y) \frac{1}{2} \{ \max(0, n - D_w) \}^2 \quad (2)$$

where $m > 0$ represents margin, D_w is the distance function between two samples, and Y stands for the output label. The Siamese network produces a distance value. We measured the distance between two image's feature embeddings using the Euclidean distance.

The core idea in this phase is to learn discriminative facial characteristics with a wide gap across classes throughout the training phase. A neural network (with a certain structure) is trained for this phase under a specified loss function, which controls how the network's parameters vary. After getting optimal feature representation from input images, it can be used to perform face verification. The significant conclusions will be how helpful the first step in producing synthetic face characteristics will be for computer vision in the future. The testing data is supplied to the Siamese Network during the testing phase in order to extract facial features, which are then utilized to compute the euclidean distance to conduct face verification and identification. A benefit of this strategy is that by creating synthetic data, the Siamese Network can distinguish between several people who have multiple attributes and can become more resilient to high-level feature fluctuation. The time- and space-complexity of the network can be a drawback.

4 Evaluation Experiments

We carried out assessment studies utilizing two publicly available datasets to assess the efficacy of the proposed OS-DFR approach.

4.1 Datasets

First dataset mugshots (citation: [24] for the initial experiments Due to the scant amount of annotation available for these images, we used some of the dataset's image samples to train the ATTGAN. Since a mugshot is a photographic portrait of a person from the shoulders up and we have just one image of each person, there is a good chance that ATTGAN hasn't been trained on it yet, which is why we chose these mugshot face images. In all of our experiments, 14 attributes that have a significant visual impact are used. These are "Bags Under Eyes," "Bald," "Bangs," "Black Hair," "Blond Hair," "Brown Hair," "Bushy Eyebrows," "Eyeglasses," "Gender," "Mouth Open," "Mustache," "No Beard," "Pale Skin," and "Age," which cover the majority of the attributes used in the previous works depicted in Figure. 3.

Moreover, we have also used the attribute intensity control characteristic of ATTGAN and generated multiple synthetic images for single attributes shown in Figure. 4.



Fig. 3. Editing results of the Facial attributes on the custom one-shot dataset: the first is the original image, and the rest 14 images result from multi-attributed synthetic face images generated by ATTGAN [10]

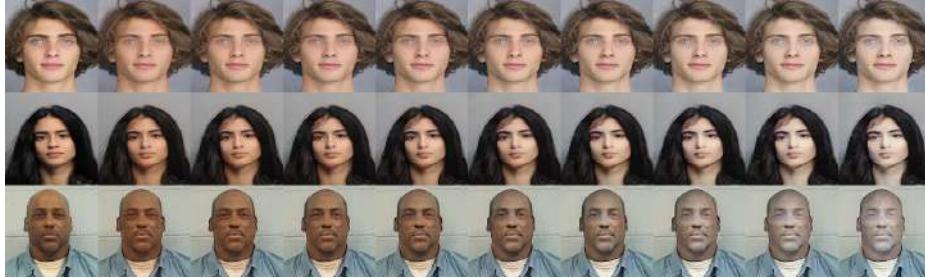


Fig. 4. Illustration of pale skin with several degrees of intensity.

4.2 Network Settings

The network was implemented using PyTorch. On a batch size of 16, we trained networks for 30 epochs. Rmsprop was used as the optimizer, with a learning rate of 0.001. The NVIDIA Titan Xp with 12GB of RAM was used to train and test the system. Two convolutional layers with kernel sizes of 11x11 and 7x7 make up the Siamese network. 2D max pooling immediately follows each layer.

4.3 Results

The network showed poor performance in our initial tests while trying to deal with 5, 10 and 15-way one-shot recognition. After epochs and network layer settings we achieve the the highest accuracy of 78% as shown in fig 7. We tried our experiments with the following settings: Table 1 shows the appropriate resultant

performances. It is clear from that table that the verification work becomes more challenging the more dissimilar the training and testing sets are.

Table 1. Performance based on the characteristics of the training and the testing set

Experiment Data Set	Train accuracy		Val accuracy		Test accuracy	
	LFW	Combined	Synthetic	LFW	Synthetic	Combined
5-Way Shot	98%	93%	65%	73%	65.66%	78%
10-Way Shot	99%	94%	67%	77%	68%	72.50%
15-Way Shot	96.50%	91%	68%	75%	62.29%	66%

There are some serious takeaways from the results. It can be seen that the performance of the model is strongly dependent on how different the images are in both the training and validation sets. If we take an example of just identifying the person from the face that is already present in the database, then the trained model on multi-attributed synthetic data would be enough to support that case.

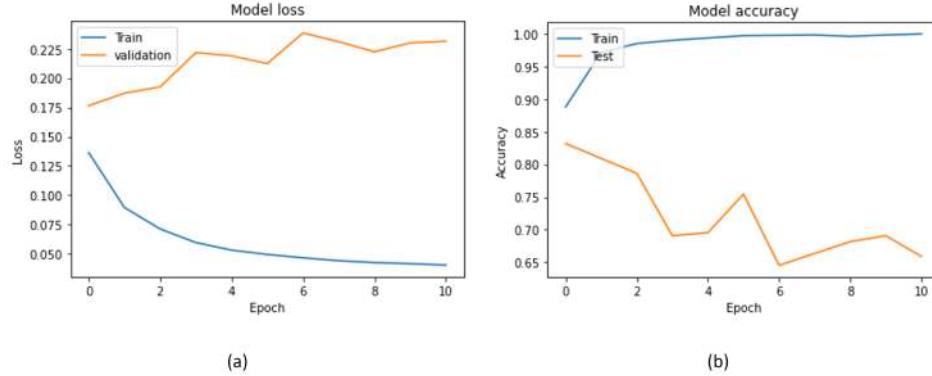


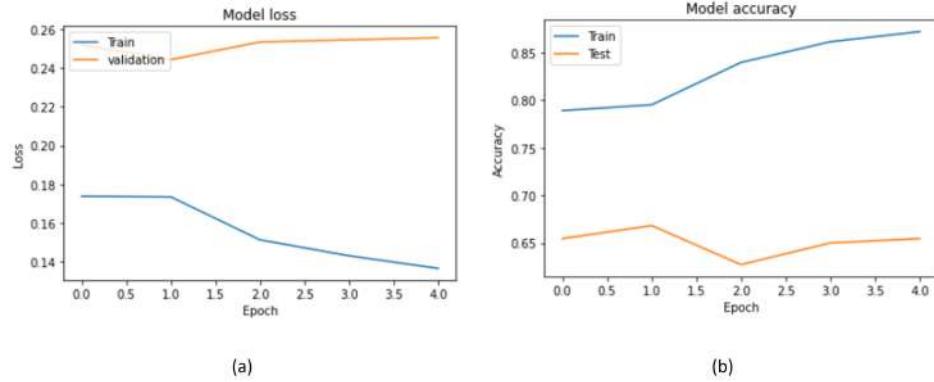
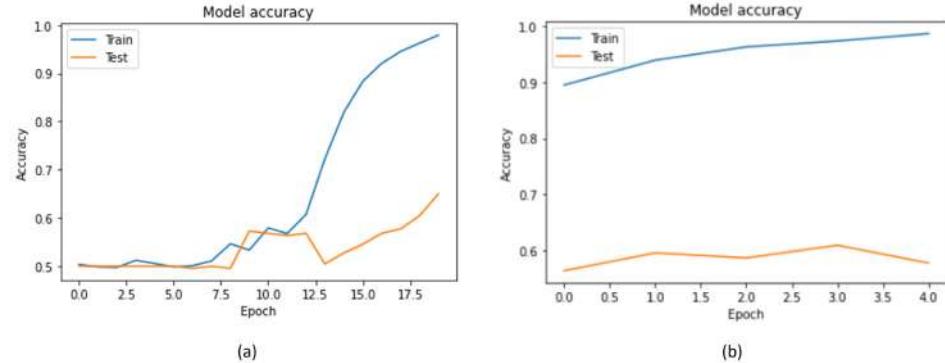
Fig. 5. 10-Way One shot on LFW & Synthetic Test Data (a) Model Loss (b) Model Accuracy

It is demonstrated in fig 6 as an ablations study the accuracy of 10-way shot when the model was solely testes on synthetic dataset.

This particular Siamese network was chosen since it is a basic net that was trained using a contrastive loss with feature normalization at the end and no final linear layer following the computation of feature distance.

5 Limitations and Discussion

Face data augmentation is the most complicated of the other data augmentation techniques. Several techniques, including pose transfer, hairdo transfer,

**Fig. 6.** 10-Way One shot on Synthetic Test Data**Fig. 7.** Accuracy of 5-Way One shot on (a) Model Loss on Combined Data (b) Model Accuracy on Combined Data

expression transfer, cosmetics transfer, and age transfer, have been suggested to change the appearance of an actual face image. In the meantime, the simulated virtual faces can also be improved to match the realism of the genuine ones. We proposed a one-shot synthetic data generation for deep face recognition in this work. The model is based on the image generation capability of GANs, whereby we try to use the data variance of the base set to synthesize more efficient augmented data for one-shot face recognition. The idea was built to aid researchers in making efficient facial recognition technology and minimizing the impact of the obstacle of limited data. Our solution can also identify a person who keeps changing their facial appearance. Our architecture shares the same constraints and is based on synthetic image generation with multiple attributes. Any new findings that enhance the image generation capability with multiple attributes should directly be applicable to our technique. Our approach, irrespective of the current limitations, has shed some light that, with the help of computer graph-

ics, will allow for efficient training and recognition of facial models from just one-shot face images. Future research plan is to use more efficient GAN network for generating high quality multi-attributed synthetic face images and train a deeper face recognition system.

6 Acknowledgement

This study was supported by the BK21 FOUR project (AI-driven Convergence Software Education Research Program) funded by the Ministry of Education, School of Computer Science and Engineering, Kyungpook National University, Korea (4199990214394) and was also supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2019-0-00203, Development of 5G-based Predictive Visual Security Technology for Preemptive Threat Response) and also by the MSIT(Ministry of Science and ICT), Korea, under the Innovative Human Resource Development for Local Intellectualization support program (IITP-2022-RS-2022-00156389) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation).

References

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: {TensorFlow}: a system for {Large-Scale} machine learning. In: 12th USENIX symposium on operating systems design and implementation (OSDI 16). pp. 265–283 (2016)
2. Bart, E., Ullman, S.: Cross-generalization: Learning novel classes from a single example by feature replacement. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). vol. 1, pp. 672–679. IEEE (2005)
3. Boutros, F., Huber, M., Siebke, P., Rieber, T., Damer, N.: Sface: Privacy-friendly and accurate face recognition using synthetic data. arXiv preprint arXiv:2206.10520 (2022)
4. Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., Shah, R.: Signature verification using a "siamese" time delay neural network. Advances in neural information processing systems **6** (1993)
5. Cao, Q., Shen, L., Xie, W., Parkhi, O.M., Zisserman, A.: Vggface2: A dataset for recognising faces across pose and age. In: 2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018). pp. 67–74. IEEE (2018)
6. Chanda, S., GV, A.C., Brun, A., Hast, A., Pal, U., Doermann, D.: Face recognition—a one-shot learning perspective. In: 2019 15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS). pp. 113–119. IEEE (2019)
7. Guo, Y., Zhang, L.: One-shot face recognition by promoting underrepresented classes. arXiv preprint arXiv:1707.05574 (2017)
8. Guo, Y., Zhang, L., Hu, Y., He, X., Gao, J.: Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In: European conference on computer vision. pp. 87–102. Springer (2016)

9. Hariharan, B., Girshick, R.: Low-shot visual recognition by shrinking and hallucinating features. In: Proceedings of the IEEE international conference on computer vision. pp. 3018–3027 (2017)
10. He, Z., Zuo, W., Kan, M., Shan, S., Chen, X.: Attgan: Facial attribute editing by only changing what you want. *IEEE transactions on image processing* **28**(11), 5464–5478 (2019)
11. Huang, B., Wang, Z., Wang, G., Jiang, K., Zeng, K., Han, Z., Tian, X., Yang, Y.: When face recognition meets occlusion: A new benchmark. In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 4240–4244. IEEE (2021)
12. Huang, G.B., Mattar, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In: Workshop on faces in 'Real-Life' Images: detection, alignment, and recognition (2008)
13. Huang, Z., Pan, Z., Lei, B.: Transfer learning with deep convolutional neural network for sar target classification with limited labeled data. *Remote Sensing* **9**(9), 907 (2017)
14. Jadhav, A., Namboodiri, V.P., Venkatesh, K.: Deep attributes for one-shot face recognition. In: European Conference on Computer Vision. pp. 516–523. Springer (2016)
15. Karras, T., Laine, S., Aila, T.: 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)
16. Koch, G., Zemel, R., Salakhutdinov, R., et al.: Siamese neural networks for one-shot image recognition. In: ICML deep learning workshop. vol. 2, p. 0. Lille (2015)
17. Li, X., Pang, T., Xiong, B., Liu, W., Liang, P., Wang, T.: Convolutional neural networks based transfer learning for diabetic retinopathy fundus image classification. In: 2017 10th international congress on image and signal processing, biomedical engineering and informatics (CISP-BMEI). pp. 1–11. IEEE (2017)
18. Masi, I., Wu, Y., Hassner, T., Natarajan, P.: Deep face recognition: A survey. In: 2018 31st SIBGRAPI conference on graphics, patterns and images (SIBGRAPI). pp. 471–478. IEEE (2018)
19. Mokhayeri, F., Kamali, K., Granger, E.: Cross-domain face synthesis using a controllable gan. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 252–260 (2020)
20. Ruiz, N., Theobald, B.J., Ranjan, A., Abdelaziz, A.H., Apostoloff, N.: Morphgan: One-shot face synthesis gan for detecting recognition bias. arXiv preprint arXiv:2012.05225 (2020)
21. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 815–823 (2015)
22. Shen, B., RichardWebster, B., O'Toole, A., Bowyer, K., Scheirer, W.J.: A study of the human perception of synthetic faces. In: 2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021). pp. 1–8. IEEE (2021)
23. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: Closing the gap to human-level performance in face verification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1701–1708 (2014)
24. the smoking gun (1997), <http://www.thesmokinggun.com/mugshots>
25. Viazovetskyi, Y., Ivashkin, V., Kashin, E.: Stylegan2 distillation for feed-forward image manipulation. In: European conference on computer vision. pp. 170–186. Springer (2020)

Multi-Attributed Face Synthesis for One-Shot Deep Face Recognition 13

26. Wang, Q., Gao, J., Lin, W., Yuan, Y.: Learning from synthetic data for crowd counting in the wild. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8198–8207 (2019)
27. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8798–8807 (2018)
28. Wang, Z., Wang, G., Huang, B., Xiong, Z., Hong, Q., Wu, H., Yi, P., Jiang, K., Wang, N., Pei, Y., et al.: Masked face recognition dataset and application. arXiv preprint arXiv:2003.09093 (2020)
29. Wu, Y., Liu, H., Fu, Y.: Low-shot face recognition with hybrid classifiers. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 1933–1939 (2017)
30. Zhai, Z., Yang, P., Zhang, X., Huang, M., Cheng, H., Yan, X., Wang, C., Pu, S.: Demodalizing face recognition with synthetic samples. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 3278–3286 (2021)
31. Zhang, H., Grimmer, M., Ramachandra, R., Raja, K., Busch, C.: On the applicability of synthetic data for face recognition. In: 2021 IEEE International Workshop on Biometrics and Forensics (IWBF). pp. 1–6. IEEE (2021)

Parallax-based Imitation Learning with Human Intervention for Uncertain Insertion Tasks

Yasuhiro Niwa¹, Kunihiro Kato¹,
Hiroaki Aizawa², Yoshiyuki Hatta¹ and Kazuaki Ito¹

¹ Gifu University, 1-1 Yanagido, Gifu City, Gifu 501-1193, Japan

² Hiroshima University, 1-3-2 Kagamiyama, Higashi-Hiroshima City,
Hiroshima 739-8511, Japan
kkato@gifu-u.ac.jp

Abstract. Standard insertion machines require pre-determined component position and posture. If the position and posture change every time, we must solve this problem. Most conventional methods attempted to solve this task by identifying the position and posture. However, these methods require a multi-step strategy following the handmade rule. This paper proposes an imitation learning method to automate the wire insertion task with uncertainties in position and posture. The proposed model learns the motion policy through human demonstrations and maps image data to the robot's action in a single step. Moreover, the model considers the parallax of the stereo images for accurate insertion. In addition, the model outputs the insertion action and recovery action to recover from insertion failures. However, the standard data collection method cannot collect recovery actions, and manual labeling of action classes is essential. This paper proposes a novel data collection method called "Labeling with Human Intervention (LHI)" to tackle this problem. This method automatically generates action labels and collects recovery action with human intervention. We conducted real-space insertion tests and found that our approach achieved 96.3% (104/108).

Keywords: Deep Imitation Learning, Image Processing, Robotics.

1 Introduction

Standard insertion machines based on force control [1] require pre-determined component position and posture. If the position and posture change every time, we must solve this problem. Most conventional methods focused on the uncertainty of the hole position or the component posture. However, these methods require multiple steps through pre-determined rules to identify the position or posture.

In this paper, we apply imitation learning [2] to automate the insertion task shown in Fig 1, where both the hole position and the wire posture are random. In imitation learning, the model learns the motion policy through human demonstrations. The

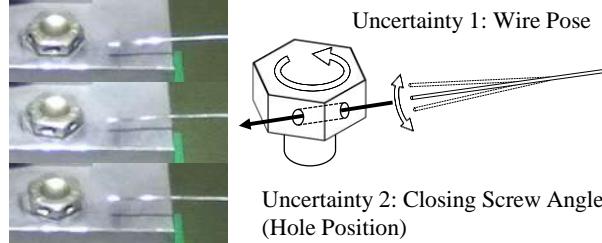


Fig. 1. Insertion Task with Two Uncertainties.

trained model can directly map the input data to the robot's action in a single step without object detection and pose estimation. This paper automates the insertion task via imitation learning using a single model and a single stereo camera.

Our proposed model considers the parallax of stereo images in the feature extractor for accurate wire insertion. In addition, the model outputs insertion action and recovery action to recover from insertion failures. However, the standard data collection method cannot collect the recovery action. Moreover, it is necessary to label action classes manually. This paper proposes a novel data collection method called "Labeling with Human Intervention (LHI)" to tackle this problem. The weak model inserts the wire, and the operator intervenes if the insertion fails. The proposed method collects recovery actions through human intervention and automatically gives action labels. Furthermore, this paper proposes a novel noise added to human actions to expand the input data distribution and reduce the instability of motion generation.

We conducted real-space insertion tests and found that our approach achieved 97.2% (35/36) and improved by 11.1% from the baseline model. Finally, we evaluated our method in a more complex environment with no blackout curtains and automatic grasping, and the success rate was 96.3% (104/108). The results showed that our method could recover from insertion failures.

2 Related Work

2.1 Autonomous Insertion

Previous works focus on the hole position uncertainty or the component posture one. The methods that tackle the hole position insert the component using images [3] or force [4, 5]. These approaches use an object detection model, like YOLOv3 [6], or force sensors attached to the robot to detect the hole position. The methods that tackle the component posture use images [7, 8] or tactile [9]. Image-based methods use contour extraction, triangulation [7], or image processing with deep learning [8]. Tactile-based methods use tactile sensors attached to the robot's fingers to detect the component posture [9]. These methods [3-9] require multi-step processes following pre-determined rules. Another issue is applying completely different approaches for the position and posture. In contrast, this paper tackles these uncertainties using only one model and one stereo camera.

2.2 Imitation Learning

Imitation learning [2] is a method for making autonomous robots without complex multi-step strategies. This method learns the motion policy using human demonstrations. The trained model directly maps the input data to the robot's action. Our approach exploits and extends the basic principle of imitation learning to omit hole detection or pose estimation.

Previous works in imitation learning have achieved complex manipulations, such as grasping a fish [13], needle threading [15], and peeling bananas [18]. Most studies use images [10-18] or force/position data [19-25] as the primary input to generate robot motion. In this paper, we select the stereo images as the input data. In image-based methods, the operator monitors the robot's state using an HMD (Head Mounted Display) and moves the robot remotely. The HMD presents binocular images to the operator, who can recognize the robot's space remotely and three-dimensionally [14-18]. This paper uses two RGB cameras and presents the stereo images to the operator.

Recent works use the linear velocity of the robot's tip as the model output [11-12, 14-18]. The model generates a trajectory by adding up the output of each time step. However, if the output is a single linear velocity, the robot cannot recover from insertion failure [15]. Therefore, our model outputs an insertion action and a recovery action following the model proposed by H. Kim et al. [15]. However, the method [15] needs to label action classes manually for the autonomous action selection. This paper proposes a novel data collection method LHI to avoid manual labeling.

2.3 Covariate Shift

There is a problem called "covariate shift" in the machine learning field. The covariate shift is when the input distribution changes in training and testing. The model inputted unseen distribution data cannot predict appropriate values. In imitation learning, the robot enters unseen situations if the policy model generates wrong actions. Moreover, the input distribution is different from the training data, and the policy model repeats to output incorrect actions. Previous works [26-29] try to expand the input distribution in human demonstrations to solve the covariance shift. In an early study, DAgger (Dataset Aggregation) [26] extends the input distribution by randomly switching between the policy model and the operator. However, this approach has a problem: the operator must keep monitoring the robot until getting a well-learned model. The recent method, DART (Disturbances for Augmenting Robot Trajectories) [27], injects noise into the operator's actions to enlarge the input distribution efficiently. This paper uses the noise injection method to expand training data distribution. The studies about the types of noise change the robot's initial position in each episode [28] or add randomly triangular noise to the operator's steering action [29]. In our method, we change the robot's initial position and add the novel triangular noise improved to three dimensions using a rotation matrix.

3 Method

3.1 Fundamental Principle

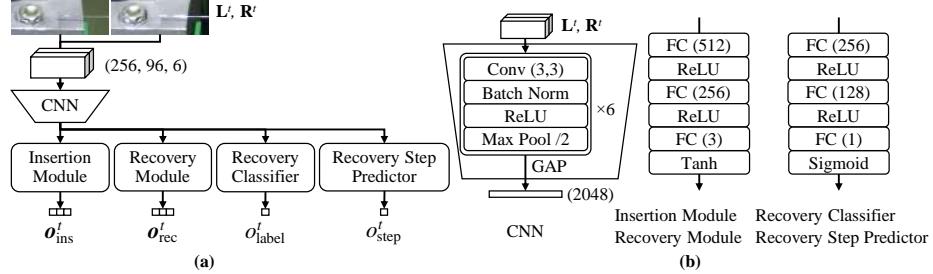
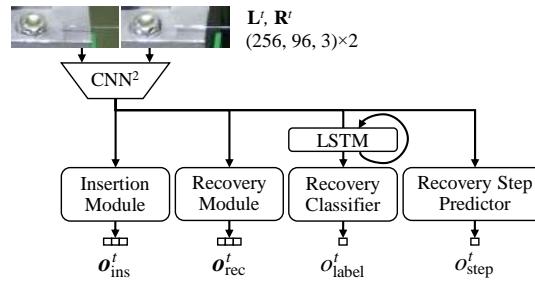
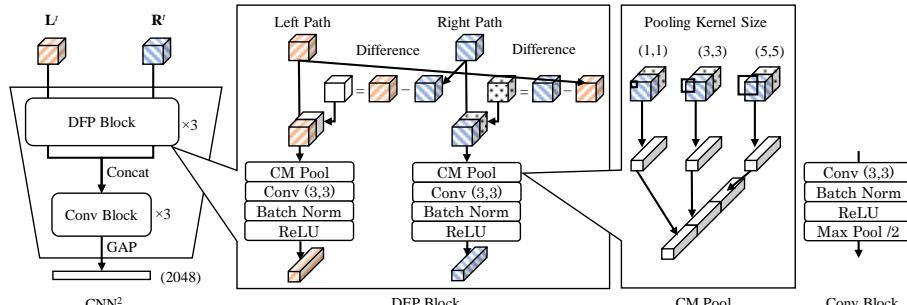
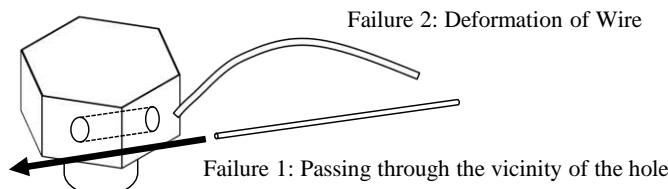
The imitation learning model learns the motion policy through the demonstration set $\mathcal{D} = \{\mathcal{C}^i\}_{i=1}^N$. N is the number of trials. $\mathcal{C}^i = \{(s^t, a^t)\}_{t=1}^L$ is the demonstration represented as the set of combinations of the robot's state s^t and the operator's command a^t at each time t . L is the end time of each demonstration. The model has learnable parameters optimized using standard supervised learning with the input s^t and target signal a^t . The well-learned model outputs the robot's action like the operator's command. Therefore, the model can move the robot instead of the operator. This paper makes the robot insert the wire, expanding this fundamental principle.

3.2 Model Architecture

The proposed model expands the baseline model [15] that outputs insertion and recovery action. Fig. 2 (a) shows the architecture of the baseline model. Fig. 2 (b) shows the details of each module. The baseline model has a standard CNN and requires the concatenation of stereo images. Then, the model transmits the feature to the insertion module, recovery module, recovery classifier, and recovery step predictor. The insertion module outputs a linear velocity o_{ins}^t to insert the wire into the hole. The recovery module outputs another linear velocity o_{rec}^t to pull back the wire. The recovery classifier outputs a probability o_{label}^t to select the appropriate action. The recovery step predictor outputs a consecutive step length o_{step}^t of the recovery action to avoid stagnating the wire. This paper calls the last three modules sub-modules.

Fig. 3 shows the proposed model. There are two improvements over the baseline model. First, the proposed model uses CNN² [30]. The CNN² feature extractor incorporates both binocular and monocular stereoscopic information. Fig. 4 shows the CNN² architecture. The DFP (Dual Feedforward Pathways) block [30] concatenates the differences between left and right features to consider the parallax-based distance perspective. The CM (Concentric Multi-Scale) pooling [30] conducts the multi-scale pooling and connects the outputs in the channel axes to consider the blurry vision caused by distance.

This paper introduces an LSTM layer in the recovery classifier in the second improvement. As shown in Fig. 5, there are two failure patterns in the wire insertion task. The first failure is passing through the outside of hole. The second failure is the bending of the wire with a collision with the screw face. It is crucial to consider the positional relation between the wire and the hole to recover from the first failure. Thus, it is sufficient for the recovery classifier to receive the one-frame image feature. In contrast, it is necessary to consider time series features for wire deformation failure. Therefore, our recovery classifier has an LSTM layer to memorize past wire information.

**Fig. 2.** Baseline Model**Fig. 3.** Proposed Model**Fig. 4.** Feature Extractor CNN²**Fig. 5.** Failure Patterns of Wire Insertion.

3.3 Collecting Insertion Action

This section explains the first step in our approach to collect the operator's insertion action. Fig. 6 shows the loop for collecting the insertion action, and Algorithm 1 shows the details. At first, our system randomly determines the starting position of the slave robot to augment the input distribution. After the slave robot moves to the start position, the operator inserts the wire into the hole remotely. The stereo camera gets binocular images cropped to remove unnecessary information at each time t . The operator watches the slave robot's state remotely and three-dimensionally through the cropped images. The HMD view is the same as the model inputs. Therefore, the operator can confirm that the images have the information for motion generation.

In addition, the operator inputs the linear velocity v^t using the master robot, and the system multiplies a rotation matrix M^t to the action v^t to expand the input distribution. M^t has a function to rotate the direction of v^t , and the system randomly determines the angles using triangular noise [29]. Fig. 7 shows the steps to inject the noise. Fig. 8 shows the two triangular noises α^t and β^t , which have a random amplitude, length, and occurrence probability. The system calculates the rotated action $\tilde{v}^t = M^t v^t$ and updates the slave robot's position by adding up each time output \tilde{v}^t . At each teleoperation loop, our system saves the combinations of the stereo image and the action $((L^t, R^t), v^t)$. At last, we get the demonstration set \mathcal{D}_{ins} , including the operator's insertion actions.

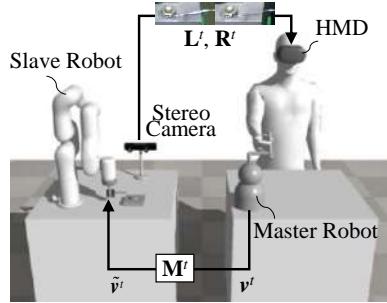


Fig. 6. Manual Mode

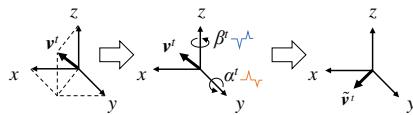


Fig. 7. Rotation of Human Actions

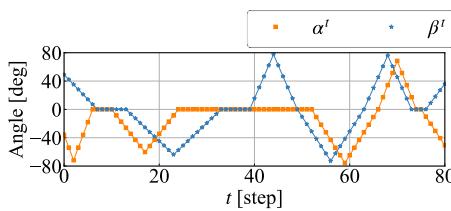


Fig. 8. Triangular Noise

Algorithm 1: Manual Mode

Parameter: Home position p_{home} , positional noise p_{noise} , slave robot position p^t , x -coordinate of the slave robot p_x^t , x -coordinate of the end position p_{end} , trial number N , time $t \leftarrow 0$, demonstration set $\mathcal{D}_{\text{ins}} \leftarrow \{\}$, demonstration $C_{\text{ins}}^i \leftarrow \{\}$.

1. **for** $i = 1$ **to** N **do**
 2. $p_{\text{noise}} \leftarrow$ random values.
 3. $p^t \leftarrow p_{\text{home}} + p_{\text{noise}} \cdot$
 4. $t \leftarrow 0$.
 5. $C_{\text{ins}}^i \leftarrow \{\}$.
 6. **while** $p_x^t < p_{\text{end}}$ **do**
 7. Get (L^t, R^t) from the camera.
 8. Send (L^t, R^t) to the HMD.
 9. Get v^t of the master robot.
 10. Get the rotation matrix M^t .
 11. $\tilde{v}^t = M^t v^t$.
 12. $p^t \leftarrow p^t + \tilde{v}^t$.
 13. $C_{\text{ins}}^i \leftarrow C_{\text{ins}}^i \cup \{((L^t, R^t), v^t)\}$.
 14. $t \leftarrow t + 1$.
 15. **end while**
 16. $\mathcal{D}_{\text{ins}} \leftarrow \mathcal{D}_{\text{ins}} \cup C_{\text{ins}}^i$.
 17. **end for**
 18. **Return** \mathcal{D}_{ins} .
-

3.4 Collecting Recovery Action

In the second step, we collect the recovery action through LHI. Algorithm 2 indicates the details. Fig. 9 shows the LHI loop, in which the operator and a weak model control the slave robot while switching from one to another. The right loop in Fig. 9 indicates the operator control, and the left shows the weak model control. This method first prepares the weak model with the feature extractor and the insertion module trained using \mathcal{D}_{ins} for one epoch. Then, the weak model operates the slave robot. If the weak model falls into errors like Fig. 5, the operator intervenes using the master robot button and pulls back the wire. At this step, the system saves the operator's action as the recovery action and the weak model action as the insertion action. The previous method [15] made the action labels by hand. In contrast, our method can automatically provide them. At last, our system labels the recovery steps for training the recovery step predictor, and we get the demonstration set \mathcal{D}_{rec} , including the operator's recovery actions.

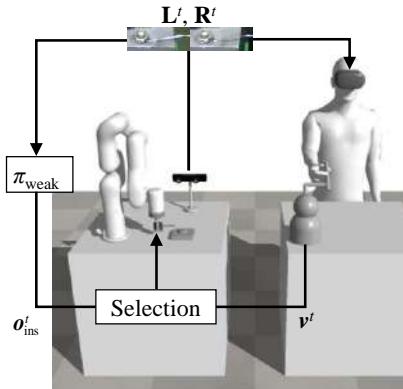


Fig. 9. Semi-autonomous Mode

Algorithm 2: Semi-autonomous Mode (LHI)

Parameter: Home position p_{home} , positional noise p_{noise} , slave robot position p^t , x -coordinate of the slave robot p_x^t , x -coordinate of the end position p_{end} , trial number N , time $t \leftarrow 0$, demonstration set $\mathcal{D}_{\text{rec}} \leftarrow \{\}$, demonstration $C_{\text{rec}}^i \leftarrow \{\}$, weak model $\pi_{\text{weak}}(\cdot)$, the button of the master robot $b^t \leftarrow \text{false}$, action label $c^t \leftarrow 0$.

1. **for** $i = 1$ **to** N **do**
 2. $p_{\text{noise}} \leftarrow$ random values.
 3. $p^t \leftarrow p_{\text{home}} + p_{\text{noise}} \cdot$
 4. $t \leftarrow 0$.
 5. $C_{\text{rec}}^i \leftarrow \{\}$.
 6. **while** $p_x^t < p_{\text{end}}$ **do**
 7. Get $(\mathbf{L}^t, \mathbf{R}^t)$ from the camera.
 8. Send $(\mathbf{L}^t, \mathbf{R}^t)$ to the HMD.
 9. Get v^t of the master robot.
 10. $o_{\text{ins}}^t \leftarrow \pi_{\text{weak}}(\mathbf{L}^t, \mathbf{R}^t)$.
 11. **if** b^t **then**
 12. $p^t \leftarrow p^t + v^t$.
 13. $c^t \leftarrow 1$.
 14. **else**
 15. $p^t \leftarrow p^t + o_{\text{ins}}^t \cdot$
 16. $c^t \leftarrow 0$.
 17. **end if**
 18. $C_{\text{rec}}^i \leftarrow C_{\text{rec}}^i \cup \{((\mathbf{L}^t, \mathbf{R}^t), v^t, c^t)\}$.
 19. $t \leftarrow t + 1$.
 20. **end while**
 21. $\mathcal{D}_{\text{rec}} \leftarrow \mathcal{D}_{\text{rec}} \cup C_{\text{rec}}^i$.
 22. **end for**
 23. Get recovery steps \mathcal{L} .
 24. $\mathcal{D}_{\text{rec}} \leftarrow \mathcal{D}_{\text{rec}} \cup \mathcal{L}$.
 25. **Return** \mathcal{D}_{rec} .
-

3.5 Autonomous Insertion

In the third step, the slave robot insets the wire autonomously. Fig. 10 shows the autonomous control loop, and Algorithm 3 indicates the details. The process is the same as the previous algorithm [15], excepting the initialization of the LSTM. This method first prepares the model, including the sub-modules, trained using \mathcal{D}_{ins} and \mathcal{D}_{rec} . The well-learned model inserts the wire into the hole with occasional recovery.

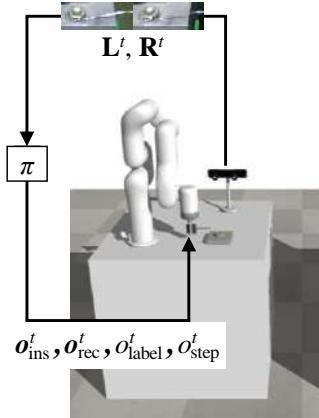


Fig. 10. Autonomous Mode

Algorithm 3: Autonomous Mode

Parameter: Home position p_{home} , slave robot position p' , x -coordinate of the slave robot p_x' , x -coordinate of the end position p_{end} , trial number N , time $t \leftarrow 0$, model $\pi(\cdot)$, selected action o' , threshold c_{th} , recovery steps $k \leftarrow 0$.

```

1. for  $i = 1$  to  $N$  do
2.    $p' \leftarrow p_{\text{home}}$ .
3.    $t \leftarrow 0$ 
4.   while  $p_x' < p_{\text{end}}$  do
5.     Get  $(L^t, R^t)$  from the camera.
6.      $o^t_{\text{ins}}, o^t_{\text{rec}}, o^t_{\text{label}}, o^t_{\text{step}} \leftarrow \pi(L^t, R^t)$ .
7.      $o' \leftarrow o^t_{\text{ins}}$ 
8.     if  $o^t_{\text{label}} > c_{\text{th}}$  and  $k = 0$  then
9.        $k \leftarrow o^t_{\text{step}}$ 
10.      end if
11.      if  $k > 0$  then
12.        Initialize LSTM.
13.         $o' \leftarrow o^t_{\text{rec}}$ 
14.         $k \leftarrow k - 1$ .
15.      end if
16.       $p' \leftarrow p' + o'$ .
17.       $t \leftarrow t + 1$ .
18.    end while
19.  end for

```

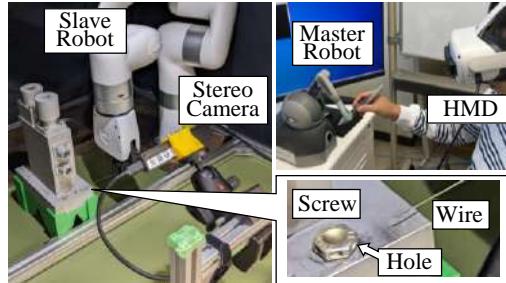


Fig. 11. Equipment

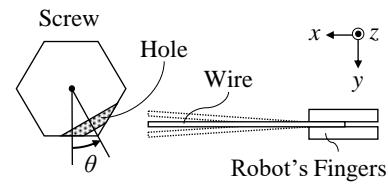


Fig 12. Screw Angle and Wire Pose

4 Experiments

4.1 Equipment

Fig. 11 shows the experimental equipment using the slave robot xArm6, the stereo camera ZED mini, the master robot Geomagic Touch, and HMD Oculus Quest 2. The diameter of the hole is 1.8 mm. The diameter of the wire is 0.8mm. Fig. 12 shows the random range of the screw angle θ and the wire tip. The angle range is [0,30] degrees. The wire tip changes within [-5,5] mm in the axes of y and z.

4.2 Dataset

We collected \mathcal{D}_{ins} by inserting the wire 600 times and randomly divided the data into three parts, 39,248 training data, 3,584 validation data, and 616 test data. Then, we split \mathcal{D}_{rec} from 300 trials in Algorithm 2 into 15,899 training data, 1,570 validation data, and 344 test data.

4.3 Assessment of Feature Extractor

This paper first compares the success rates without the recovery action. We train the feature extractor and the insertion module using \mathcal{D}_{ins} . The baseline model uses a standard 6-layer CNN. In contrast, the proposed model uses a CNN². We optimize these models using Adam (learning rate 10^{-6} , batch size 32) and RMSE and select the weights with the smallest validation loss in 50 epochs. We evaluate 36 insertions with a variety of hole positions and wire postures. Table 1 shows the success rates. The failure that the wire passed through outside of the hole was the most frequent, with nine failures in the baseline model and four in the proposed model. The proposed model using CNN², considering the sense of distance, was able to align the wire tip more accurately than the baseline model.

Table 1. Success Rate without Sub-modules

Policy Model (No Sub-modules)	Success Rate %	Number of Successes times
Baseline Model [15]	75.0	27/36
Proposed Model	83.3	30/36

4.4 Assessment of Recovery Classifier

This section compares the accuracies of the recovery classifier. The feature extractors and the insertion modules have the weights obtained in section 4.3. In contrast, we train the sub-modules using \mathcal{D}_{rec} . We use Adam (learning rate 10^{-5} , batch size 32) and the weighted sum of RMSE and Binary Cross Entropy. Then, the sequence length of the LSTM is five in the proposed model. We select the weights with the smallest validation loss in 50 epochs.

The baseline model predicted the action class with a one-frame image, whereas the proposed model uses five-frame images. Table 2 shows the accuracies with a 0.5 threshold. Fig. 13 shows the outputs of each recovery classifier on the test data in \mathcal{D}_{rec} . The proposed model was superior to the baseline model. In particular, the proposed model could predict the "recovery class" for wire deformations in data ID 271 to 287. The accuracy declines in the proposed model because of mistakes around the data ID 325. Test data around ID 325 showed that the wire deformed despite successful insertion. Therefore, the recovery classifier of the proposed model mistakes by recognizing the deformations of the wire.

Table 2. Accuracy of Test Data

Policy Model	Accuracy %
Baseline Model[15]	82.6
Proposed Model	85.5

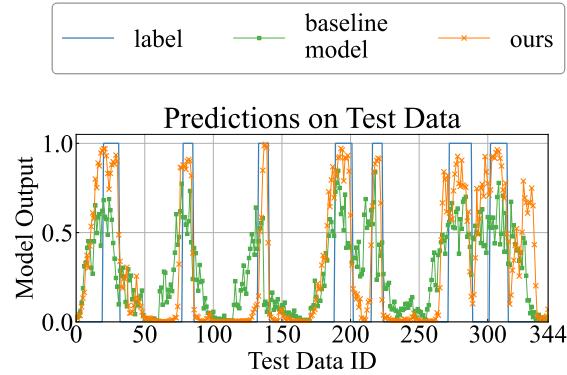


Fig. 13. Prediction Results of Action Labels. 5 demonstrations are consisted of 344 pairs of consecutive evaluation data and inputted into each model in order. The unmarked line shows the correct label, and the vertical value 1.0 indicates the "recovery class," whereas 0.0 indicates the "insertion class." The square line shows the predictions of the recovery classifier of the baseline model. Moreover, the crosses-line shows the predictions of the proposed model. The data numbers 271 to 287 indicate human intervention in the deformation of the wire. While the other cases where the correct labels are 1.0 indicate human intervention for the failure that the wire passed through the outside of the hole.

4.5 Wire Insertion

We conduct experiments with real-space wire insertions of the model having sub-modules. We initialize the model using weights in sections 4.3 and 4.4. The threshold of the recovery classifier is 0.6, 0.7, and 0.8, and Table 3 shows the best result.

In the baseline model, the wire passed outside the hole nine times, and the sub-modules pulled back the wire seven times to avoid insertion errors. However, the recovery classifier could not select the recovery action in the remaining two cases. Other failures included two failures in which the robot could not recover to the normal state and one failure in which the wire bent.

In the proposed model, there was one failure, the wire passed through the outside of the hole. We confirmed the input data in this failure; the wire was unseen without enough light reflection. In other cases, the wire collided and bent four times. The proposed model could pull back and insert the wire successfully, as shown in the case of Fig. 14. However, as we pointed out in the experiment in Section 4.4, the recovery classifier of the proposed model overreacted to the deformation of the wire, and a total of eight times incorrectly selected recovery action amid the appropriate insertion.

Table 3. Success Rate with Sub-modules

Policy Model (Sub-modules Available)	Success Rate %	Number of Successes times
Baseline Model [15]	86.1	31/36
Proposed Model	97.2	35/36

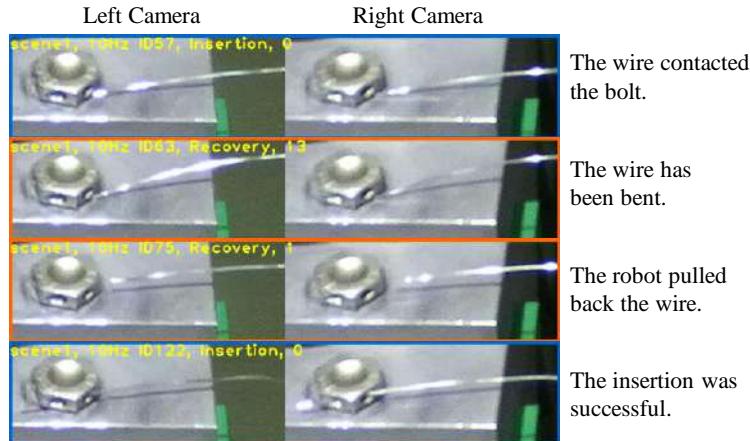


Fig. 14. Success with Pull-back Action

4.6 Wire Insertion (No Blackout Curtains + Automatic Picking)

In experiments 4.3 to 4.5, we used blackout curtains and one fixed light. In addition, we manually attached the wire to the robot's fingers. Finally, this paper performs insertion tests in the experimental setup that assumes real-world use. The new robot system does not have blackout curtains. It is a more complex environment because other lights affect the stereo camera. Moreover, the system can pick up the wire through pre-determined motion (Fig 15), and there is also randomness in the grasping position.

First, this section conducts insertion tests using the proposed model without sub-modules. We train the CNN² and the insertion module using \mathcal{D}_{ins} (600 insertions, 65,135 training data, 5,556 validation data, and 1,350 test data) and Adam (learning rate 10^{-4}). The first row of Table. 4 shows a decline in the success rate. The proposed model tended to fail to cope with more intricate changes in the wire posture. In particular, the proposed model failed when the input was an image, in which the wire tip was challenging to see. Therefore, we improved the proposed model by concatenating no-wire images ($\mathbf{L}^{-1}, \mathbf{R}^{-1}$) into current images (\mathbf{L}', \mathbf{R}'), as shown in Fig. 16. As a result, the success rate improved, as shown in Table 4, and the wire insertions were successful even for images where the wire was difficult to see.

Finally, the proposed model with sub-modules and no-wire image inserts the wire. We train the sub-modules using \mathcal{D}_{rec} (300 insertions, 28,494 training data, 2,868 validation data, and 785 test data), sequence length 10, Adam (learning rate 10^{-4}), and the classifier threshold 0.5. The success rate was 96.3% (104/108), and in three cases, the insertion was successful by pulling back the wire with the recovery action, as shown in Fig. 17 (a) and (b). In four failures, the wire movement stopped halfway and could not entirely pass through the hole. We expect this case can improve by removing small actions from the insertion data.

Table 4. Success Rate (No Blackout Curtains + Automatic Picking)

Policy Model	Test Time (PM) & Weather	Success Rate %	Number of Successes
Proposed Model (No Sub-modules)	1:00-3:00 (sunny)	52.8	19/36
	1:00-3:00 (cloudy)	61.1	22/36
	6:00-9:00 (night)	50.0	18/36
Proposed Model + $(\mathbf{L}^{-1}, \mathbf{R}^{-1})$ (No Sub-modules)	1:00-3:00 (sunny)	80.6	29/36
	1:00-3:00 (cloudy)	86.1	31/36
	6:00-9:00 (night)	91.6	33/36
Proposed Model + $(\mathbf{L}^{-1}, \mathbf{R}^{-1})$ (Sub-modules Available)	1:00-3:00 (sunny)	97.2	35/36
	1:00-3:00 (cloudy)	97.2	35/36
	6:00-9:00 (night)	94.4	34/36

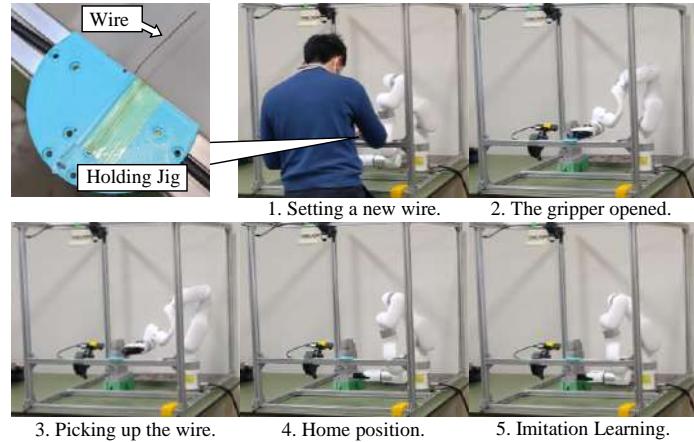


Fig. 15. Automatic Picking using Pre-determined Motion

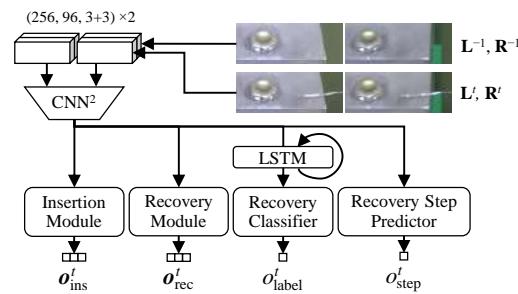


Fig. 16. Proposed Model + No-wire Images

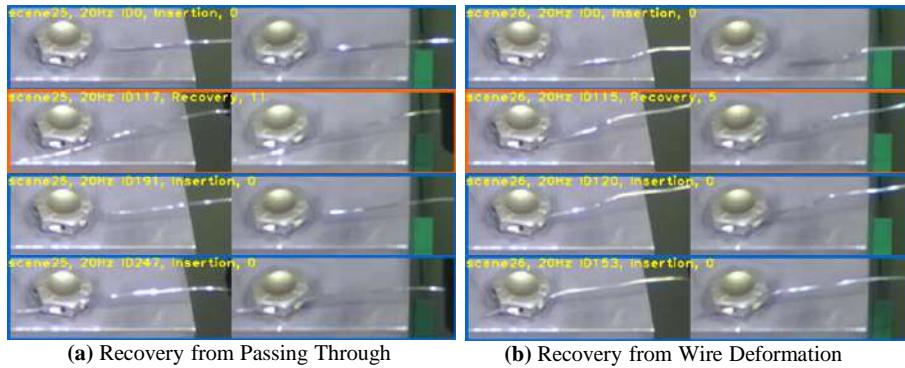


Fig. 17. Success Trials with Recovery Action

5 Conclusion

This paper automated the wire insertion task with uncertain component position and posture by imitation learning. We introduced CNN², LSTM, and no-wire images for

the baseline model. Then we proposed a novel data collection approach LHI to omit manual labeling and to collect the recovery action. In addition, we extend the triangular noise to three dimensions to reduce the covariate shift. We conducted the real-space wire insertion tests. The results showed that the proposed model could insert the wire with a higher success rate and recover from failures even if the wire deformed. In the last experiment, we evaluated our approach with no blackout curtains and automatic grasping and improved the success rate by inputting no-wire images.

The future issue is expanding the random range. The proposed model in this paper only controls the linear velocity of the robot tip. Therefore, it is necessary to improve the model and the teleoperation system by adding rotation commands.

References

1. N. Hogan: Impedance control: An approach to manipulation: Part II-Implementation. In: Journal of Dynamic Systems, Measurement, and Control, Vol. 107, No. 1, pp. 8-16 (1985)
2. A. Hussein, M. M. Gabar, E. Elyan and C. Jayne: Imitation Learning: A Survey of Learning Methods. In: ACM Computing Surveys (CSUR), Vol.50.2, No.21, pp.1-35 (2017)
3. M. Nigro, M. Sileo, F. Pierri, K. Genovese, D. D. Bloisi, Caccavale and F. Caccavale: Peg-in-Hole Using 3D Workpiece Reconstruction and CNN-based Hole Detection. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4235-4240 (2020)
4. S. R. Chhatpar and M. S. Branicky: Search strategies for peg-in-hole assemblies with position uncertainty. In: 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, Expanding the Societal Role of Robotics in the Next Millennium, pp. 1465-1470 (2001)
5. J. C. Triyonoputro, W. Wan and K. Harada: Quickly inserting pegs into uncertain holes using multi-view images and deep network trained on synthetic data. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5792-5799 (2019)
6. J. Redmon and A. Farhadi: YOLOv3: An incremental improvement. In: arXiv:1804.02767 (2018)
7. P. Cirillo, G. Laudante and S. Pirozzi: Vision-Based Robotic Solution for Wire Insertion with an Assigned Label Orientation. In: IEEE Access, Vol. 9, pp. 102278-102289 (2021)
8. D. De Gregorio, R. Zanelli, G. Palli, S. Pirozzi and C. Melchiorri: Integration of Robotic Vision and Tactile Sensing for Wire-Terminal Insertion Tasks. In: IEEE Transactions on Automation Science and Engineering, Vol. 16.2, pp. 585-598 (2018)
9. G. Palli, and S. Pirozzi: A Tactile-Based Wire Manipulation System for Manufacturing Applications. In: Robotics, Vol. 8.2, No. 46 (2019)
10. S. Levine, C. Finn, T. Darrell and P. Abbeel: End-to-End Training of Deep Visuomotor Policies. In: The Journal of Machine Learning Research, Vol. 17.1, pp. 1334-1373 (2016)
11. T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg and P. Abbeel: Deep Imitation Learning for Complex Manipulation Tasks from Virtual Reality Teleoperation. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 5628-5635 (2018)
12. T. Yu, C. Finn, A. Xia, S. Dasani, T. Zhang, P. Abbeel and S. Levine: One-Shot Imitation from Observing Humans via Domain-Adaptive Meta-Learning, In: arXiv:1802.01557 (2018)

13. J. S. Dirtside, E. R. Aye, A. Stahl and J. R. Matthiessen: Teaching a Robot to Grasp Real Fish by Imitation Learning from a Human Supervisor in Virtual Reality. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 7185-7192 (2018)
14. H. Kim, Y. Ohmura and Y. Kuniyoshi: Using human gaze to improve robustness against irrelevant objects in robot manipulation tasks. In: IEEE Robotics and Automation Letters, Vol. 5.3, pp. 4415-4422 (2020)
15. H. Kim, Y. Ohmura and Y. Kuniyoshi: Gaze-based dual resolution deep imitation learning for high-precision dexterous robot manipulation. In: IEEE Robotics and Automation Letters, Vol. 6.2, pp. 1630-1637 (2021)
16. H. Kim, Y. Ohmura and Y. Kuniyoshi: Memory-based gaze prediction in deep imitation learning for robot manipulation. In: arXiv:2202.04877 (2022)
17. H. Kim, Y. Ohmura and Y. Kuniyoshi: Transformer-based deep imitation learning for dual-arm robot manipulation. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 8965-8972 (2022)
18. H. Kim, Y. Ohmura and Y. Kuniyoshi: Robot peels banana with goal-conditioned dual-action deep imitation learning. In: arXiv:2203.09749 (2022)
19. A. Sasagawa, K. Fujimoto, S. Sakaino and T. Tsuji: Imitation Learning Based on Bilateral Control for Human-Robot Cooperation. In: IEEE Robotics and Automation Letters, Vol. 5.4, pp. 6169-6176 (2020)
20. S. Sho: Bilateral Control-Based Imitation Learning for Velocity-Controlled Robot. In: 2021 IEEE 30th International Symposium on Industrial Electronics (ISIE). pp. 1-6 (2021)
21. S. Sakaino, K. Fujimoto, Y. Saigusa and T. Tsuji: Imitation Learning for Variable Speed Object Manipulation, In: arXiv:2102.10283 (2021)
22. T. Kitamura, S. Sakaino, M. Hara and T. Tsuji: Bilateral Control of Human Upper Limbs Using Functional Electrical Stimulation Based on Dynamic Model Approximation. In: IEEJ Journal of Industry Applications, 20009551 (2021)
23. K. Hayashi, S. Sakaino and T. Tsuji: An Independently Learnable Hierarchical Model for Bilateral Control-Based Imitation Learning Applications. In: IEEE Access, Vol. 10 pp. 32766-32781 (2022)
24. Y. Saigusa, S. Sakaino, T. Tsuji: Imitation Learning for Nonprehensile Manipulation through Self-Supervised Learning Considering Motion Speed. In: IEEE Access, Vol. 10, pp. 68291-68306 (2022)
25. H. Kim, Y. Ohmura, A. Naga Kubo and Y. Kuniyoshi: Training Robots without Robots: Deep Imitation Learning for Master-to-Robot Policy Transfer. In: arXiv:2202.09574 (2022)
26. S. Ross, G. J. Gordon and J. A. Bagnell: A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. In: arXiv:1011.0686 (2010)
27. M. Laskey, J. Lee, R. Fox, A. Dragan and K. Goldberg: DART: Noise Injection for Robust Imitation Learning. In: Conference on robot learning, PMLR, pp. 143-156 (2017)
28. L. Ke, J. Wang, T. Bhattacharjee, B. Boots and S. Srinivasa: Grasping with Chopsticks: Combating Covariate Shift in Model-free Imitation Learning for Fine Manipulation. In: 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 6185-6191 (2021)
29. F. Codevilla, M. Muller, A. Lopez, V. Koltun and A. Dosovitskiy: End-to-end Driving via Conditional Imitation Learning. In: 2018 IEEE International Conference on Robotics and Automation (ICRA) (2018)
30. W. Chen and S. Wu: CNN²: Viewpoint Generalization via a Binocular Vision. In: Neural IPS 2019, pp. 1986-1998 (2019)
31. S. Hochreiter and S. Jürgen: Long short-term memory. In: Neural Computation 9(8), pp. 1735-1780 (1997)

A Style-Based Caricature Generator

Lamyanba Laishram^{1[0000-0002-0324-214X]}, Muhammad Shaheryar^{1[0000-0003-3992-1387]}, Jong Taek Lee^{1[0000-0002-6962-3148]} and Soon Ki Jung^{1[0000-0003-0239-6785]}

School of Computer Science and Engineering, Kyungpook National University,
Republic of Korea
 {yanbalalishram, shaheryar, jongtaeklee, skjung}@knu.ac.kr

Abstract. A facial caricature is a creation of new artistic and exaggerated faces which translates into a real image to convey sarcasm or humor while keeping the identity of the subject. In this work, we proposed a new way to create caricatures by exaggerating facial features like the eyes and mouth while keeping the facial contour intact and a realistic style. Our method can be categorized into two steps. First, the facial exaggeration process transformed faces into caricature face images while maintaining facial contours. Second, the appearance style generator is trained in unpaired using the generated caricature faces to produce a facial caricature that can change to any realistic style of our preference. Experimental results show our model produces more realistic and disentangled caricature images as compared to some of the previous methods. Our method can also generate caricature images from real images.

Keywords: Caricature · Style Generator · Generative Adversarial Network

1 Introduction

In the world of comics, animation, posters, and advertising, in particular, artistic portraits are very common in our daily lives. A caricature is a representation of a person whose distinctive features are simplified or exaggerated through sketching or artistic drawings. A facial caricature is a form of art used to convey sarcasm or humor and is used commonly in entertainment.

Applications based on computer vision have a wide range and the creation of caricatures can be done without the need for an artist. Similar to the way an artist approach creating caricatures, a method based on computer vision can also be divided into two stages: (i) identifying the distinct features and exaggerating those features, and (ii) applying styles to the deformed image according to the artist's taste. The separation of these two categories provides flexibility and disentanglement which eventually results in the generation of good-quality caricatures.

Earlier approaches for creating a facial caricature require professional skills to get good results [2]. Traditional artworks tended to emphasize exaggerating facial forms by increasing the shape representation's divergence from the average,

as in the case of 2D landmarks or 3D meshes [3, 27, 14]. With the advancement in applications of computer vision techniques, several automated caricature generations have emerged [33, 11, 4]. Moreover, automatic portrait style transfer based on image style transfer [22, 32, 24] and image-to-image translation [21] have been extensively studied. Recently with the development in the Generative adversarial networks (GANs) [13], the state-of-the-art face generator StyleGAN [19, 20] provides disentangled and high-fidelity artistic images via transfer learning.

In recent years, Deep learning techniques are very successful in performing image-to-image translation by learning from representation data examples [15, 16]. Unfortunately, paired real and caricature are not commonly found. The translation process is not feasible to be trained in a supervised manner and building such a dataset is tedious. One of the readily available caricature datasets is WebCaricature [17], which consists of 6042 caricatures and 5974 photographs from 252 different identities. In our work, we created a set of 10,000 caricature images from the FFHQ dataset [19] for automatic caricature generation which we will discuss in Section 3.

Due to the limited data availability of paired images, most of the research on image-to-image translation in this work is starting to move towards training on unpaired images [5, 16, 40] and learning from unpaired portrait and caricature [4, 35]. However, learning unpaired images can introduce highly varied exaggerations from different artists with divergent styles. Most images will have different poses and scales which might result in difficulty to distinguish facial features. In our method, we performed an unpaired learning approach using a specific caricature design of exaggerating face parts while still maintaining the facial contour of the real image.

We aim to create a method of generating new caricature faces from a real image with realistic details and obtain different stylization results. Our method first modifies a real face into a caricature face and then used that to train a generative model to produce different styles. A summary of our contribution is as follows:

- We proposed a method of generating facial caricature images with big eyes and big mouths using face patches. The method can generate different faces and can apply for multiple style transfers on a specific face.
- Our method is an unpaired learning process of creating a caricature face first from a real face image. A powerful style network is then trained using the generated caricature faces to synthesize different styles transfer.
- Our generated caricature is more realistic and high-quality as compared with the previous methods while still providing a completely disentangling style.

The remainder of our work is organized as follows: The related work in the creation of caricature and style transfer are discussed in section 2. The methodology behind the creation of our caricature and the style transfer are discussed in Section 3. Experimental results are shown and analyzed in Section 4. We finally conclude our work in Section 5.

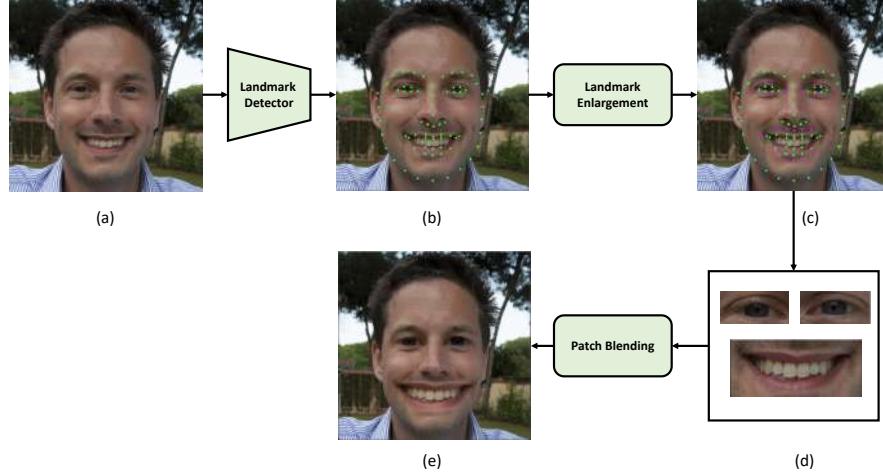


Fig. 1: Facial caricature generation pipeline: (a) input image, (b) facial landmarks using a landmark detector, (c) enlarged landmark location for eyes and mouth regions, (d) segmented and scaled eyes and mouth patches, and (e) final caricature result generated after blending the scaled face patches to the original face image.

2 Related work

In this section, we discuss some of the works related to our paper: caricature creation and style transfer.

2.1 Caricature Creation

The generation of caricature is to identify and exaggerate distinct features of a face while still maintaining the identity of the individual. The creation of caricatures can be performed in three ways: deforming facial attributes, style transfer, or methods using both.

Traditional methods perform by magnifying the deviation from the mean, either by explicitly identifying and warping landmarks [12, 25] or by utilizing data-driven approaches to estimate distinctive facial characteristics [26, 38]. With the advancement in generative networks, some image-to-image translation work [39, 23] has been done to apply transfer style. However, because these networks are unsuitable for techniques with large spatial variation, their outputs have low visual quality.

Cao et al. [4] use two CycleGANs which are trained on image and landmarks space for texture rendering and geometry deformation. WarpGAN [33] can produce better visual quality and more shape exaggeration by providing flexible spatial

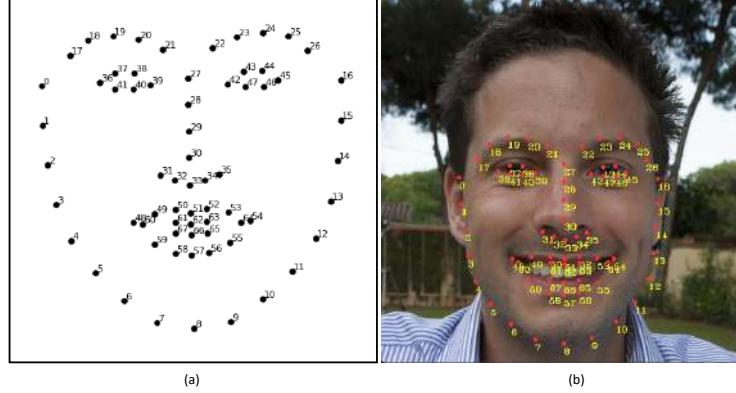


Fig. 2: Facial landmarks: (a) all 68 facial landmark annotations, and (b) test example of generating all 68 facial landmarks performed on FFHQ dataset [19].

variability on both image geometry and texture. CariGAN [4] is a GAN trained using unpaired images to learn the image-to-caricature translation. Shi et al. [33] proposed an end-to-end GAN framework that trains warping and style. Deformation fields are used by AutoToon [11] to apply the exaggerations. AutoToon is trained in a supervised manner using paired data from artist-warp photos to learn warping fields. It maps to only one domain, so it cannot produce diverse exaggerations.

2.2 Style Transfer

One type of image synthesis issue is style transfer, which seeks to create a content image with several styles. Due to the efficient ability to extract semantic features by CNNs [10], numerous style transfer networks are implemented. The initial process of rendering styles is performed by Gatys et al. [8] using hierarchical features from a VGG network [34]. The first neural style transfer approach was put out by Gatys et al. [9] and employs a CNN to transfer the style information from the style picture to the content image. The drawback is that both the style and content of pictures should be similar, which is not the case with caricatures.

A promising area of research has been the use of Generative Adversarial Networks (GANs) [13] for picture synthesis, where cutting-edge outcomes have been shown in applications like text-to-image translation [31] and image inpainting [37]. Using Generative Adversarial Networks (GANs) [13] for image synthesis has been a promising field of study, where state-of-the-art results have been demonstrated in applications ranging from text to image translation [31], image inpainting [37] and many more. Unpaired image translation is accomplished by CycleGAN [40] using a cycle consistency loss. StarGAN [5, 6] uses a single generator to learn mappings between various picture domains. To capture the

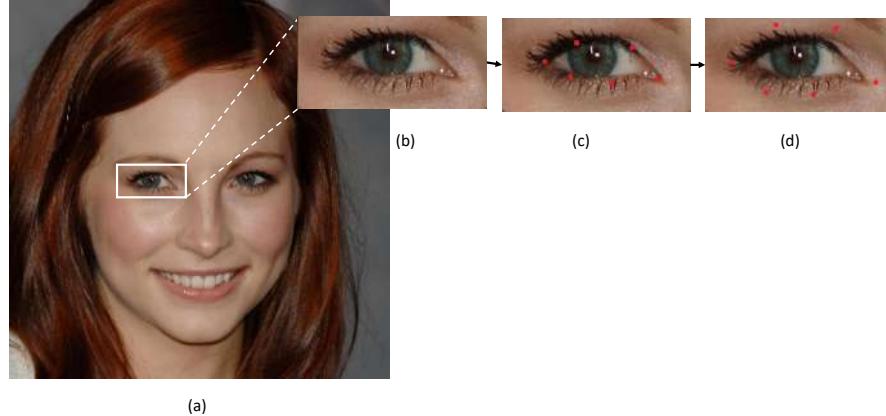


Fig. 3: Generated facial patches: (a) image from the FFHQ dataset [19], (b) visualizing the left eye from our point of view, (c) the landmark positions of the left eye, and (d) increasing the landmark indexes for improving blending results.

geometric transformation, it is challenging to learn photo-to-caricature mapping directly in an image-to-image translation approach.

StyleGAN [19, 20, 18] generates high-fidelity face images with hierarchical channel style control. StyleGAN was refined by Pinkney and Adler [30] using sparse cartoon data, and they discovered that approach was effective in producing realistic cartoon faces. DualStyleGAN [36] offers customizable management of dual styles transfer for both the expanded artistic portrait domain and the original face domain.

3 Methodology

The goal of our method is to generate a caricature face that looks realistic and train a state-of-the-art style generator with our newly generated caricature face. Our method provides completely disentangling styles for the generated caricature faces. Our whole method is sectioned into two steps: face caricature creation and face style generation. Face caricature creation focuses on the creation of exaggerated faces with enlarged eyes and mouths from real faces. Face style generation focuses on the generation of caricature faces with realistic and distinct styles.



Fig. 4: Our generated caricature images from real images.

3.1 Face Caricature Creation

Real-face images are used for the creation of a caricature face. The face images are randomly sampled from the FFHQ dataset [19] which covers diverse gender, races, ages, expressions, poses and etc. The first process is to find the facial landmark points as shown in Figure 2. The pre-trained facial landmark detector inside the dlib library [1] is used to estimate the location map of facial structures on the face. Dlib is a commonly used open-source library that can recognize 68 (x, y) coordinates of the structure of a face image. These 68 landmarks are specifically assigned for each part of the face like eyes, eyebrows, nose, mouth, and face contour.

Our implementation specifically focuses on the enlargement of the eyes and mouth region of the face. The indexes of the landmarks of the eyes can be categorized into two groups, such as the left eye and the right eye. The left eye is represented by indexes 37 to 42 whereas the right eye is by indexes 43 to 48. The mouth area consists of the upper lips and the lower lips. When we consider the mouth as a whole, we take only the top landmarks indexes of the upper lip and the bottom indexes of the bottom lips. Indexes 49 to 60 represent the mouth region. All these landmark positions are shown in Figure 2.

Using the eyes and mouth landmarks indexes, we segmented two eye regions and a mouth region patch. Before creating these patches, the corresponding landmarks are increased to make the interested area of the patches bigger as shown in Figure 3. The patches are then scaled first to a factor of 1.5 and then blend back to the original center location of the eyes and mouth part respectively. The scaled patches regions are patched back to the original image using the

Poisson image editing technique [29] which blends in seamlessly. The blending technique affects the image illumination and the texture. The Poisson seamless editing can be represented as follows:

$$v = \operatorname{argmin}_v \sum_{i \in S, j \in N_i \cap S} ((v_i - v_j) - (s_i - s_j))^2 + \sum_{i \in S, j \in N_i \cap \neg S} ((v_i - t_j) - (s_i - t_j))^2$$

where v is the pixel values of the new image, s is the pixel values of the source image, t is the pixel values of the target image, S is the destination domain, and N_i a set of neighboring pixels of i .

Our final caricature faces are realistic as it is produced from real images. We illustrate some of our caricature datasets in Figure 4.

3.2 Style Transfer Generator

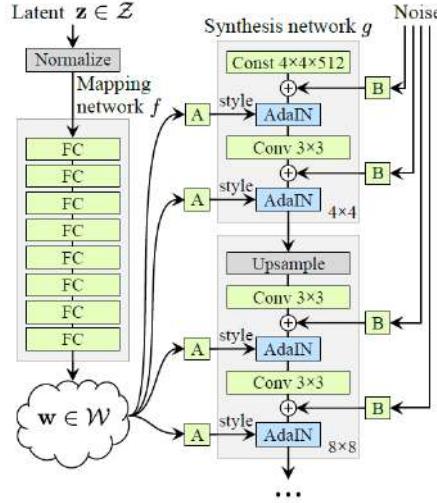


Fig. 5: StyleGAN Architecture [19, 20]

For the style transfer process, we trained a powerful style-based generator called StyleGAN [19, 20]. We trained the generator only using our newly produced face caricature images which have enlarged eye and mouth regions. The architecture of StyleGAN consists of two networks: a mapping network and a synthesis network as shown in Figure 5. A mapping network f is an 8-layer MLP that maps a given latent code $z \in Z$ to produce $w \in W$, defined as $f : Z \rightarrow W$.



Fig. 6: Four examples results of our caricature generation with style generator and each row is one caricature identity with seven different styles.

The synthesis network g are 18 convolutional layers that are controlled through adaptive instance normalization (AdaIN) [7] at each layer with the learned affine transformation “A” of latent code w . A scalable Gaussian noise input “B” is also fed in each layer of the synthesis network g .

The architecture is designed in a way that each style controls only one convolution. Random latent codes can be used to control the styles of the generated images. After we train our new caricature images, the generator can produce caricature images with different styles of facial attributes like skin tone, hair color, shapes, etc. Note that, unlike previous caricature generation, we trained our generator using only our generated caricature faces and no paired data.

3.3 Implementation

Our experiment is implemented with the diverse set of face FFHQ dataset [19]. We collected 55,000 FFHQ images for the face caricature generation and the style transfer training. The image resolution we worked on is 256 X 256. The style generator is trained with the same network architecture and other hyperparameters as the original Stylegan-ADA generator [18]. Our core algorithm is developed using PyTorch 1.7.1 [28] and CUDA 11.3. The experiment is performed using four NVIDIA TITAN Xp GPUs and a batch size of 16.

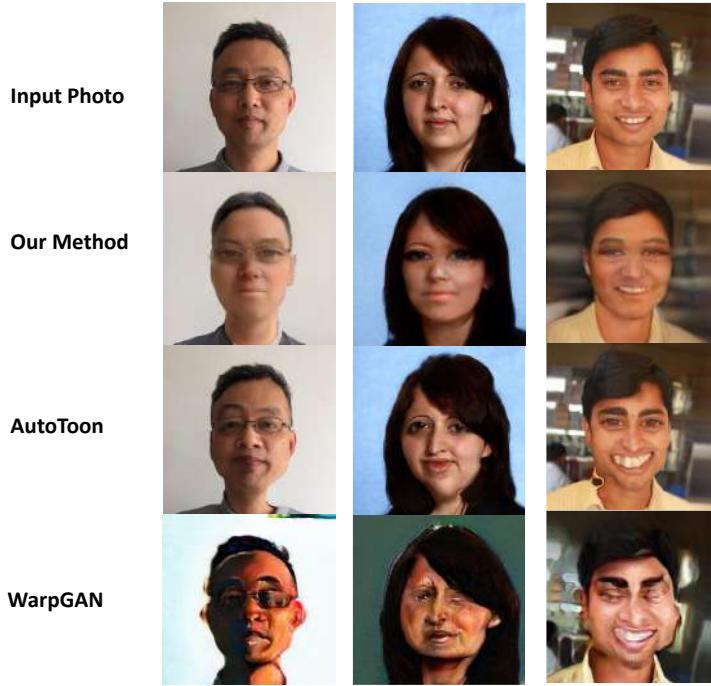


Fig. 7: Comparing our method with WarpGAN [33] and AutoToon [11].

4 Experiments

We explore various possibilities that our caricature generator can perform. Our caricature generator can generate any face type with exaggerated facial parts. Our caricature faces preserve the contour of the face shape. Face caricatures with different poses, hairstyles, face shapes, eye colors, etc can be generated. Figure 6 show different style provided for a specific caricature identity. Each row represents one identity and numerous style techniques can be applied to the generated caricature. This is possible because of the disentanglement nature of the StyleGAN generator. The latent space of StyleGAN is disentangled and we used it to our benefit. The generated images have a realistic style as we produced the caricature faces from the real images. Our caricature generator can also generate a caricature image from a real image. We demonstrate the effectiveness of our method by applying it to a different range of images gathered from publicly available content. These include images characterized by different facial expressions, poses, and illumination.

We qualitatively compare our caricature generation method with the previous caricature creation methods like AutoToon [11] and WrapGAN [33] as shown in Figure 7. We find that all three methods produced very different results. The style of WarpGAN is tightly linked to its warping, which results in irregularities

or deformation of facial features, and the quality of the caricature is degraded considerably. On the other hand, AutoToon exaggerates facial characteristics while maintaining their general quality and consistency in a way that is true to the original image, particularly with regard to specifics like the eyes, ears, and teeth. AutoToon needs paired learning method to generate these results. Our method doesn't change the face contours and exaggerates only the specific face region. The identity information and facial expression are also preserved. Our result looks more realistic as compared to other techniques, yet shows facial deformation. Since our generator is trained only on our caricature faces and not paired images, it is difficult to obtain our generator result directly from real images. We believe that we can improve our results by introducing an encoder to guide the latent space of the generator. This will be our future work.

5 Conclusion

In this paper, we proposed a framework for generating realistic unpaired caricature images. We proposed a new approach to keep the facial contours intact while exaggerating the facial parts like the eyes and mouth regions. We used a powerful style-based architecture to produce a realistic caricature from real face images. Our approach supports flexible controls to change the style of the generated caricature faces. Experimental results demonstrate that the proposed method creates caricatures that are more realistic than other state-of-the-art caricature generation methods. Although our model achieved superior results, there still exist problems that need to be tackled in caricature generation. The caricature generation is limited to all the drawbacks of the StyleGAN architecture. We will further improvements in the caricature generation process in the future.

Acknowledgment

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2019-0-00203, Development of 5G-based Predictive Visual Security Technology for Preemptive Threat Response) and also by the MSIT(Ministry of Science and ICT), Korea, under the Innovative Human Resource Development for Local Intellectualization support program (IITP-2022-RS-2022-00156389) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation).

Bibliography

- [1] dlib c++ library. <http://dlib.net>
- [2] Akleman, E., Palmer, J., Logan, R.: Making extreme caricatures with a new interactive 2d deformation technique with simplicial complexes. In: Proceedings of visual. vol. 1, p. 2000. Citeseer (2000)
- [3] Brennan, S.E.: Caricature generator: The dynamic exaggeration of faces by computer. *Leonardo* **18**(3), 170–178 (1985)
- [4] Cao, K., Liao, J., Yuan, L.: Carigans: Unpaired photo-to-caricature translation. arXiv preprint arXiv:1811.00222 (2018)
- [5] Choi, Y., Choi, M., Kim, M., Ha, J.W., Kim, S., Choo, J.: Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8789–8797 (2018)
- [6] Choi, Y., Uh, Y., Yoo, J., Ha, J.W.: Stargan v2: Diverse image synthesis for multiple domains. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8188–8197 (2020)
- [7] Deng, Y., Tang, F., Dong, W., Sun, W., Huang, F., Xu, C.: Arbitrary style transfer via multi-adaptation network. In: Proceedings of the 28th ACM International Conference on Multimedia. p. 2719–2727. MM ’20, Association for Computing Machinery, New York, NY, USA (2020)
- [8] Gatys, L., Ecker, A.S., Bethge, M.: Texture synthesis using convolutional neural networks. *Advances in neural information processing systems* **28** (2015)
- [9] Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2414–2423 (2016)
- [10] Gatys, L.A., Ecker, A.S., Bethge, M., Hertzmann, A., Shechtman, E.: Controlling perceptual factors in neural style transfer. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3985–3993 (2017)
- [11] Gong, J., Hold-Geoffroy, Y., Lu, J.: Autotoon: Automatic geometric warping for face cartoon generation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 360–369 (2020)
- [12] Gooch, B., Reinhard, E., Gooch, A.: Human facial illustrations: Creation and psychophysical evaluation. *ACM Transactions on Graphics (TOG)* **23**(1), 27–44 (2004)
- [13] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. *Communications of the ACM* **63**(11), 139–144 (2020)
- [14] Han, X., Hou, K., Du, D., Qiu, Y., Cui, S., Zhou, K., Yu, Y.: Caricatureshop: Personalized and photorealistic caricature sketching. *IEEE transactions on visualization and computer graphics* **26**(7), 2349–2361 (2018)

- [15] Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. *science* **313**(5786), 504–507 (2006)
- [16] Huang, X., Liu, M.Y., Belongie, S., Kautz, J.: Multimodal unsupervised image-to-image translation. In: Proceedings of the European conference on computer vision (ECCV). pp. 172–189 (2018)
- [17] Huo, J., Li, W., Shi, Y., Gao, Y., Yin, H.: Webcaricature: a benchmark for caricature recognition. arXiv preprint arXiv:1703.03230 (2017)
- [18] Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., Aila, T.: Training generative adversarial networks with limited data. *Advances in Neural Information Processing Systems* **33**, 12104–12114 (2020)
- [19] Karras, T., Laine, S., Aila, T.: 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)
- [20] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
- [21] Kim, J., Kim, M., Kang, H., Lee, K.: U-gat-it: Unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation. arXiv preprint arXiv:1907.10830 (2019)
- [22] Li, C., Wand, M.: Combining markov random fields and convolutional neural networks for image synthesis. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2479–2486 (2016)
- [23] Li, W., Xiong, W., Liao, H., Huo, J., Gao, Y., Luo, J.: Carigan: Caricature generation through weakly paired adversarial learning. *Neural Networks* **132**, 66–74 (2020)
- [24] Liao, J., Yao, Y., Yuan, L., Hua, G., Kang, S.B.: Visual attribute transfer through deep image analogy. arXiv preprint arXiv:1705.01088 (2017)
- [25] Liao, P.Y.C.W.H., Li, T.Y.: Automatic caricature generation by analyzing facial features. In: Proceeding of 2004 Asia Conference on Computer Vision (ACCV2004), Korea. vol. 2 (2004)
- [26] Liu, J., Chen, Y., Gao, W.: Mapping learning in eigenspace for harmonious caricature generation. In: Proceedings of the 14th ACM international conference on Multimedia. pp. 683–686 (2006)
- [27] Mo, Z., Lewis, J.P., Neumann, U.: Improved automatic caricature by feature normalization and exaggeration. In: ACM SIGGRAPH 2004 Sketches, p. 57 (2004)
- [28] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* **32** (2019)
- [29] Pérez, P., Gangnet, M., Blake, A.: Poisson image editing. In: ACM SIGGRAPH 2003 Papers, pp. 313–318 (2003)
- [30] Pinkney, J.N., Adler, D.: Resolution dependent gan interpolation for controllable image synthesis between domains. arXiv preprint arXiv:2010.05334 (2020)

- [31] Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative adversarial text to image synthesis. In: International conference on machine learning. pp. 1060–1069. PMLR (2016)
- [32] Selim, A., Elgharib, M., Doyle, L.: Painting style transfer for head portraits using convolutional neural networks. ACM Transactions on Graphics (ToG) **35**(4), 1–18 (2016)
- [33] Shi, Y., Deb, D., Jain, A.K.: Warpgan: Automatic caricature generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10762–10771 (2019)
- [34] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
- [35] Wu, R., Tao, X., Gu, X., Shen, X., Jia, J.: Attribute-driven spontaneous motion in unpaired image translation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5923–5932 (2019)
- [36] Yang, S., Jiang, L., Liu, Z., Loy, C.C.: Pastiche master: Exemplar-based high-resolution portrait style transfer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7693–7702 (2022)
- [37] Yeh, R., Chen, C., Lim, T.Y., Hasegawa-Johnson, M., Do, M.N.: Semantic image inpainting with perceptual and contextual losses. arXiv preprint arXiv:1607.07539 **2**(3) (2016)
- [38] Zhang, Y., Dong, W., Ma, C., Mei, X., Li, K., Huang, F., Hu, B.G., Deussen, O.: Data-driven synthesis of cartoon faces using different styles. IEEE Transactions on image processing **26**(1), 464–478 (2016)
- [39] Zheng, Z., Wang, C., Yu, Z., Wang, N., Zheng, H., Zheng, B.: Unpaired photo-to-caricature translation on faces in the wild. Neurocomputing **355**, 71–81 (2019)
- [40] Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2223–2232 (2017)

Detecting Mounting Behaviors of Dairy Cows by Pre-Training with Pseudo Images

Yuta Okuda^{1[0000-0003-1847-1882]}, Yota Yamamoto^{1[0000-0002-1679-5050]},
 Kazuaki Nakamura^{1[0000-0002-4859-4624]}, and Yukinobu
 Taniguchi^{1[0000-0003-3290-1041]}

Tokyo University of Science, Tokyo, Japan
 4621505@ed.tus.ac.jp, {yy-yamamoto, nakamura.kazuaki,
 taniguchi.yukinobu}@rs.tus.ac.jp

Abstract. A key part of improving the productivity of the dairy industry is detecting signs of estrus in dairy cows. In this paper, we propose a method based on deep learning to automatically detect mounting behavior, an indicator of estrus, using cameras installed on the ceiling of the barn. Mounting behavior occurs rarely, and it is virtually impossible to manually collect actual data of mounting events. The novelty of the proposed method lies in the pre-training scheme, which uses pseudo-mounting images generated by overlapping randomly selected cow images that are easily collected. The pre-trained model is then fine-tuned on a small amount of actual mounting data. We introduce a consistency regularization term based on using background replacement images to reduce the influence of background changes in pre-training. We show the effectiveness of the proposed method through experiments that compare the proposed method and previous self-supervised learning methods in terms of their detection accuracy using data collected in actual cattle barns.

Keywords: Self-Supervised Learning · Anomaly Detection · Image Generation.

1 Introduction

Recently, the number of dairy farms in Japan has been decreasing due to the drop in the population of dairy farmers because of aging and the lack of successors [11]. However, the demand for dairy products remains constant. For higher production efficiency, dairy farms are getting larger in scale. To improve dairy production, it is necessary to increase the rate of estrus detection and thus pregnancy. Dairy farmers can efficiently increase the number of dairy cows by artificially inseminating those dairy cows in estrus. The signs of estrus appear in cow behavior (e.g., mounting, accepting mounting, walking around, etc.), that is dairy cows during the estrus period tend to mount other cows and accept mounting, so dairy farmers need to observe each dairy cow carefully. However, the burden of managing individual dairy cows has become significant with the scale of herds.



Fig. 1: Mounting image.

One way to reduce the burden on dairy farmers is to attach acceleration sensors to dairy cows. The acceleration sensor detects signs of estrus by measuring the cow movements. However, the sensor, which is attached to the neck of each cow, is costly and fails often as dairy cows love to rub against the wall. In addition, attaching them to dairy cows causes stress. In this paper, we focus on the individual management of dairy cows by installing cameras on the ceiling of the barn (ceiling camera).

One way to detect signs of estrus is to detect mounting behavior (Fig. 1) in which one dairy cow mounts another dairy cow. Wang et al. [15] used the object detector YOLOv5 to detect mounting behaviors. However, it requires a large amount of data on mounting behavior for training. Unfortunately, it is time-consuming to collect the large amount of training data needed because mounting behavior rarely occurs. Fortunately, it is easy to prepare a large amount of non-mounting data from barn images.

To address the problem, this paper proposes a method of detecting mounting behavior that uses for pre-training i) pseudo-mounting images generated by overlapping randomly selected cow images (which are easily collected), and ii) a consistency regularization loss term based on background replacement images. The pre-trained model is then fine-tuned on a small amount of actual mounting data.

2 Related Work

2.1 Mounting Behavior Detection

Several methods have been developed for detecting the mounting behaviors of cows or pigs. Most of them are based on region features. Nasirahmad et al. [8] use an ellipse fitting technique to locate the position of pigs and detect mounting behaviors from the distance between each pig. Li et al. [5] use Mask R-CNN [3] to detect specific regions of pigs from which mounting behavior is identified. From

the detection results, three features, length around the pig, region of the half of the mask, and distance between the centers of the bounding boxes (BBOX), are selected. The eigenvectors are classified by a kernel extremal learning machine (KELM) to detect mounting behavior. Noe et al. [9] proposed a method for detecting the mounting behaviors of dairy cows that uses object detection and tracking techniques. It extracts segmented regions of dairy cows by using Mask R-CNN while a lightweight tracking algorithm is used to detect mounting behavior. It takes advantage of the fact that the body region of a cow rises up when mounting. These methods detect mounting from just the region features of cows and pigs.

Wang et al. [15] proposed a detection method based on mounting-specific postures (image features) using improved YOLOv5, which has stronger detection ability against complex environments and multi-scale objects. However, this method requires a large amount of manually annotated mounting data.

2.2 Self-Supervised Learning

There are many studies on self-supervised learning to deal with the lack of training data.

CutPaste [4] is an anomaly detection model based on a two-stage framework with self-supervised learning. The first step is to learn a deep representation using data augmentation. Data augmentation is simple: cut an image patch and paste it at a random position in the original image. Next, a classifier builds on the learned deep representation in one class. The original image is defined as normal, and the data augmentation image is defined as abnormal. Inferencing is performed using the Gaussian density estimator to find anomalies, and then Grad-CAM [12] detects the locations of anomalies.

Noroози et al. [10] proposed a method for learning image representations through a task in which models solve jigsaw puzzles. The model is trained by inputting images divided into tiles whose positions are swapped and predicting the original order. Therefore, the images do not need to be labeled. The learned models can be reused for tasks such as object detection and classification.

SimMIM [16] learns an image representation through the task of predicting the original image from a randomly masked image. Masking is done at patch level, that is, a patch is either fully visible or completely masked. The encoder uses the transformer model [2, 7]. The prediction head can be as light as linear, and the image representation is learned by regression of the RGB values.

This paper improves the accuracy for actual tasks by performing self-supervised learning specifically for mounting detection.

3 Proposed Method

The main idea behind our proposed method is to generate pseudo mounting images from dairy cow images (non-mounting images), which are easy to collect.

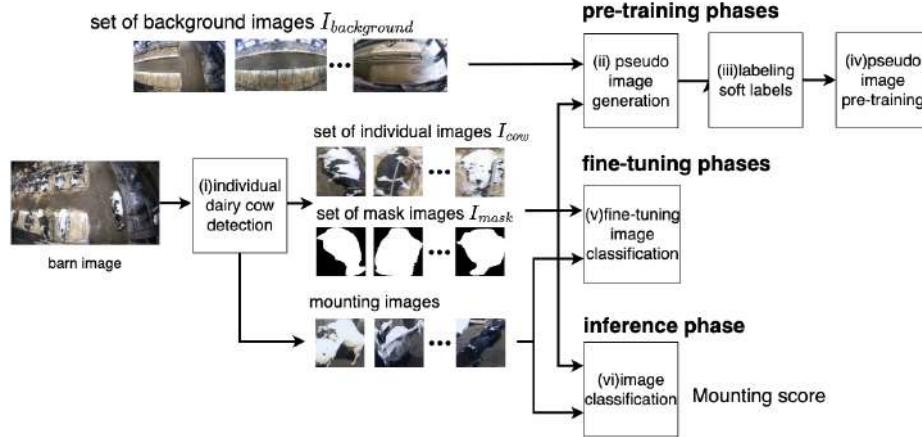


Fig. 2: Proposed method

We draw inspiration from the CutPaste [4] method. However, instead of randomly cutting out rectangular image patches, we use Mask R-CNN to extract cow regions that are then cut out and pasted.

The proposed method is shown in Fig. 2. It comprises six processes: (i) individual dairy cow detection, (ii) pseudo-image generation, (iii) assigning soft labels, (iv) pseudo image pre-training, (v) fine-tuning image classification, and (vi) image classification.

- (i) **Individual dairy cow detection.** Mask R-CNN takes as input a barn image taken by a ceiling camera and detects individual cows $I_{cow} = \{I_1, I_2, \dots, I_n\}$ and the corresponding segmentation masks $M_{cow} = \{M_1, M_2, \dots, M_n\}$. Mask R-CNN used is fine-tuned using barn data.
- (ii) **Pseudo-image generation.** We generate pseudo images to be used for training from individual images I_{cow} and masks M_{cow} . The pseudo-image generation methods are described in Section 3.1.
- (iii) **Assigning soft labels.** We assign soft labels, the likelihood of mounting behavior, to the pseudo images. The method of assigning soft labels is explained in Section 3.2.
- (iv) **Pseudo image pre-training.** The images and labels generated in (ii) and (iii) are used to train the classifier. The structure of the classifier used is described in Section 3.3.
- (v) **Fine-tuning image classification.** The pre-trained classifier model is fine-tuned on a small number of mounting images and a large number of other images. Details of this process are shown in Section 3.4.
- (vi) **Image classification.** The classifier takes an individual image detected by (i) as input and outputs a real number ranging from 0 to 1, which indicates the likelihood of mounting behavior. If the score is higher than a threshold value, the cow image is considered to contain mounting behavior.

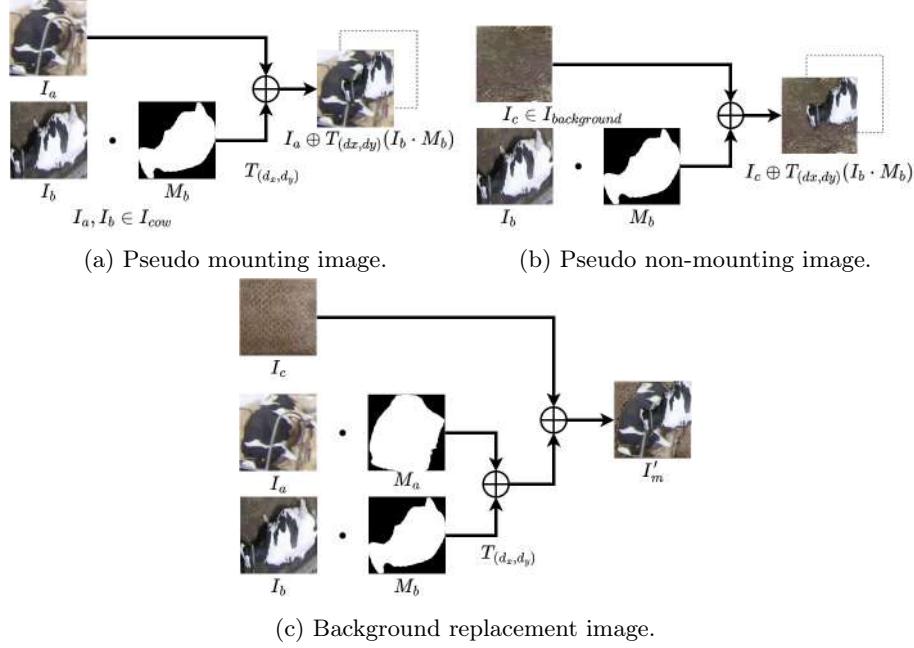


Fig. 3: Pseudo-image generation.

3.1 Pseudo-Image Generation

The method for generating pseudo images is shown in Fig. 3. There are three types of pseudo images: pseudo mounting images, pseudo non-mounting images, and background replacement images. Fig. 4 shows examples of (a) pseudo mounting images, (b) pseudo non-mounting images, (c)(d) background replacement images for (a), (b).

Pseudo-Mounting Images: To offset the paucity of mounting images, pseudo-mounting images are generated. As in Fig. 3(a), pseudo-mounting image I_m is generated as follows:

$$I_m = I_a \oplus T_{(d_x, d_y)}(I_b \cdot M_b), \quad (1)$$

where cow image $I_a, I_b \in \mathcal{I}_{cow}$ is randomly selected, mask image M_a, M_b corresponding to I_a, I_b , $A \oplus B$ is the result of replacing the image value of image A with that of image B , \cdot is the logical product of the images, and $T_{(d_x, d_y)}$ is the operation of shifting the image by (d_x, d_y) . The image I_a to be pasted is called the base image and the image I_b to be pasted is called the overlapping image.

So that the overlapping ratio s is uniformly distributed, we determine displacement vector (d_x, d_y) as follows.:

$$d_x = \pm(1 - x_{ratio})W, \quad d_y = \pm(1 - y_{ratio})H, \quad (2)$$

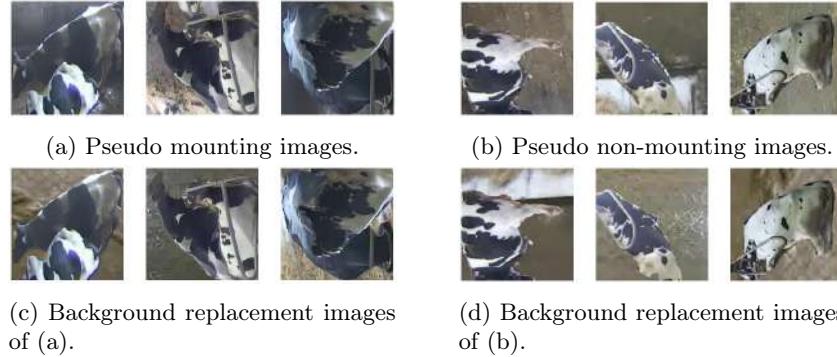


Fig. 4: Example of pseudo images.

where (W, H) indicates image size, sign is chosen randomly, and overlapping ratio $s \sim U(0, 1)$, $x_{ratio} \sim U(s, 1)$, $y_{ratio} = s/x_{ratio}$.

Pseudo Non-Mounting Images: When a classifier is trained with both pseudo-mounting images and real non-mounting images, it learns to detect pasting seams. To avoid this problem, pseudo non-mounting images are generated and added for training. Background images $I_{background}$ are the collection of barn images captured when there are no cows.

The pseudo non-mounting images are generated by selecting one from each dairy cow I_{cow} and one from each background image $I_{background}$. The selected background image is resized to 224×224 after random cropping.

$$I_n = I_c \oplus T_{(d_x, d_y)}(I_b \cdot M_b), \quad (3)$$

where $I_c \in \mathcal{I}_{background}$, $I_b \in \mathcal{I}_{cow}$.

Background Replacement Images: To make the image classifier pay attention to the pose of dairy cows instead of the background, we generate background replacement images from pseudo-mounting and pseudo non-mounting images as follows:

$$I'_m = I_c \oplus (I_a \cdot M_a) \oplus T_{(d_x, d_y)}(I_b \cdot M_b), \quad (4)$$

$$I'_n = I_c \oplus T_{(d_x, d_y)}(I_b \cdot M_b), \quad (5)$$

where I_a, I_b are the same as the generated-pseudo image and $I_c \in \mathcal{I}_{background}$.

3.2 Assigning Soft Labels

The proposed method assigns a soft label to a pseudo image that indicates the likelihood of mounting behavior. We assign soft label y to pseudo-mounting image I_m generated by eq.(1) as follows:

$$y = \frac{|M_a \cap T_{(d_x, d_y)}(M_b)|}{|M_a \cup T_{(d_x, d_y)}(M_b)|}, \quad (6)$$

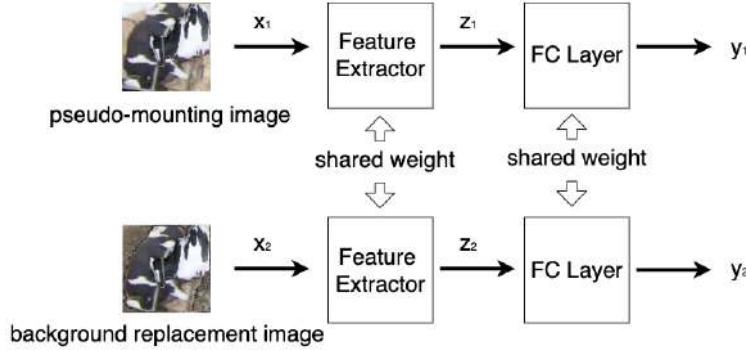


Fig. 5: Image classifier.

which indicates the Intersection over Union (IoU) between the masked regions of two cows. The soft label of the pseudo non-mounting image is always set to 0. The background replacement image is given the same label as the original pseudo image.

3.3 Pseudo image pre-training

We train an image classifier on the pseudo images generated above and the soft labels. The image classifier outputs a score ranging from 0 to 1 indicating the likelihood of mounting behavior. As illustrated in Fig. 5, the classifier model is composed of feature extraction and fully-connected layers (FC Layer).

Feature Extraction: Pseudo image x_1 and the corresponding background replacement image x_2 are input to the feature extractor which outputs feature values z_1, z_2 . The weights of the feature extractors are shared.

FC Layer: Inputs features z_1, z_2 to the FC layer + sigmoid and outputs the score y'_1, y'_2 .

We use two types of loss: binary cross entropy loss L_{score} and consistency regularization loss $L_{consistency}$. The binary cross entropy loss (BCELoss) is defined by:

$$L_{score} = \sum_{n=1}^2 y \log y'_n + (1 - y) \log(1 - y'_n), \quad (7)$$

where y is the value of the soft label of the image. The consistency regularization loss is defined by:

$$L_{consistency} = \|z_1 - z_2\|^2, \quad (8)$$

The loss constrains the classifier to output the same features for a pseudo image and the background replacement image. The total loss is $L = L_{score} + L_{consistency}$.

3.4 Fine-tuning image classification

We fine-tune the pre-trained model on a small number of mounting images and a large number of non-mounting images. The images are assigned binary labels instead of soft ones, 1 for mounting images and 0 otherwise. Focal loss [6] is employed to reduce the effect of data imbalance and is defined by:

$$L_{focal} = -(1 - p_t)^\gamma \log(p_t), \quad (9)$$

where $p_t = y'$ if $y = 1$, otherwise $p_t = 1 - y'$, γ is a focusing parameter.

4 Experimental Settings

4.1 Dataset

We used barn image data captured by 13 ceiling cameras installed in an actual barn. We prepared a dataset of individual images, which were cropped barn images of individuals detected by Mask R-CNN. The dataset was manually annotated and contains images of mounting behaviors. Table 1 shows the number of individual images used for pre-training, fine-tuning, and testing.

Pre-training: We prepared two datasets (pseudo and individual image dataset) for pre-training. For each epoch, both pseudo-mounting and pseudo-non-mounting images were generated with a probability of 50% by the method described in Sec. 3.1. Thirteen empty barn images (no cows) were used as sources of background images.

Fine-tuning and test: The fine-tuning and testing dataset consisted of mounting images and other images. We split the dataset into fine-tuning and test subsets so that temporally consecutive images were not present in either subset.

4.2 Evaluation Metric

The evaluation metric is Area Under the Curve (AUC), which is the value of the area under the ROC curve. The ROC curve is a plot of the true positive rate (TPR) on the vertical axis and the false positive rate (FPR) on the horizontal axis, with varying threshold values. Since the number of mounting images is small, the experiment was conducted using three-fold cross validation.

Table 1: Number of individual images.

	dataset	non-mounting	mounting
pretrain	pseudo image	3349	-
	individual image	23431	-
fine-tuning		5547	
test		5892	124

4.3 Implementation Details

The implementation details are as follows.

Pre-training: We used Adam optimizer with the learning rate of 0.0001. The number of epochs was 100, and the image size was 224×224 . EfficientNet-B0 [14] was used as the feature extractor; it output 1,000 dimensional features and was trained by ImageNet [1].

Fine-tuning: The learning rate was 0.00001, and focusing parameter γ was 2. The other conditions were the same as in pre-training, and the number of epochs was 30.

4.4 Baseline Methods

This paper compares the following five methods: the proposed method, data augmentation, two self-supervised methods, Jigsaw [10] and SimMIM [16], and a bbox-based method.

Proposed Method: To evaluate the impact of pre-training with pseudo-images, we evaluated two methods: with and without pre-training, and with unsupervised learning but using only pseudo-images.

Data Augmentation: We used a simple data augmentation method instead of pseudo-images. Three data augmentations were used: random rotation(90-degree increments), random flip, and color jitter(all parameters ranged from to ± 0.2). The training model was similar to the proposed method.

Jigsaw: Jigsaw [10] is a pre-training method that solves jigsaw puzzles. We compared three different methods: pre-training on a pseudo-image dataset, pre-training on an individual image dataset, and no pre-training. With regard to implementation details, the image was divided into 3×3 and 250 different puzzle patterns. The number of epochs of pre-training was 300. Other details followed those in 4.3.

SimMIM: SimMIM [16] is a pre-training method that predicts the mask portion given to an image. We compared three settings following the Jigsaw test above. We employed the transformer model VIT [2]. For pre-training, images were masked with a probability of 0.6, where the mask patch size was 32. Other details followed those in 4.3.

Bbox-Based Method: Since it is difficult to reproduce the previous method[5] accurately, we implemented a simple method that used the positional relationship of the detected bounding boxes. Taking $B_i (i = 1, 2, \dots)$ to be the bounding boxes of individual dairy cows detected in barn images, we computed score S_i as the maximum of $IoU(B_i, B_j) (j \neq i)$. If score S_i exceeds a threshold, bounding box B_i is judged to contain mounting behavior. The threshold parameter was chosen empirically instead of fine-tuning based on training data.

5 Results

Table 2 shows the experimental results and Fig. 6 shows the ROC curves. Our method achieved the highest AUC value of 0.914 when pre-trained on pseudo-mounting images and fine-tuned on actual mounting and non-mounting images.

Table 2: Experimental results.

method	pre-training	fine-tuning	AUC
Bbox-based method	-	-	0.578
Jigsaw [10]	-	✓	0.777
	pseudo image individual image	✓ ✓	0.799 0.746
SimMIM [16]	-	✓	0.794
	pseudo image	✓	0.772
	individual image	✓	0.750
Data augmentation	-	✓	0.888
Ours	-	✓	0.856
	pseudo image	-	0.759
	pseudo image	✓	0.914

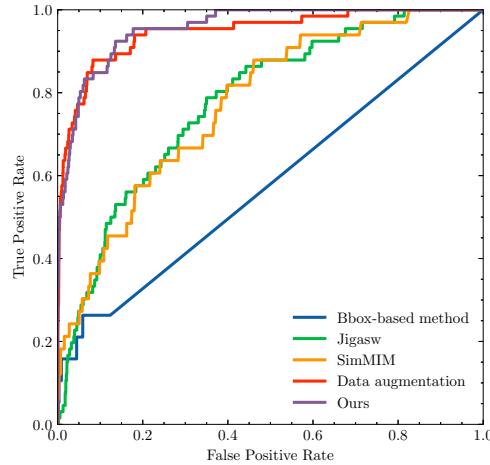


Fig. 6: ROC curves.

By assessing our method in different settings, we see a 15.5 point improvement with fine-tuning, and a 5.8 point improvement with pre-training. This shows the effectiveness of pre-training and fine-tuning. Although pseudo-mounting images are not an accurate representation of mounting images, they aided mounting behavior detection by providing an appropriate task in pre-training.

Comparing pseudo and individual images used for pre-training, we can see that the accuracy of both Jigsaw and SimMIM pre-trained on pseudo images was better than those pre-trained on individual images. It shows that the pseudo-image is more effective in extracting the image features necessary for mounting detection. Whereas, there is no significant difference between the two methods with and without pre-training. This is probably because the pre-training task is very different from mounting detection and does not make up for the lack of data.

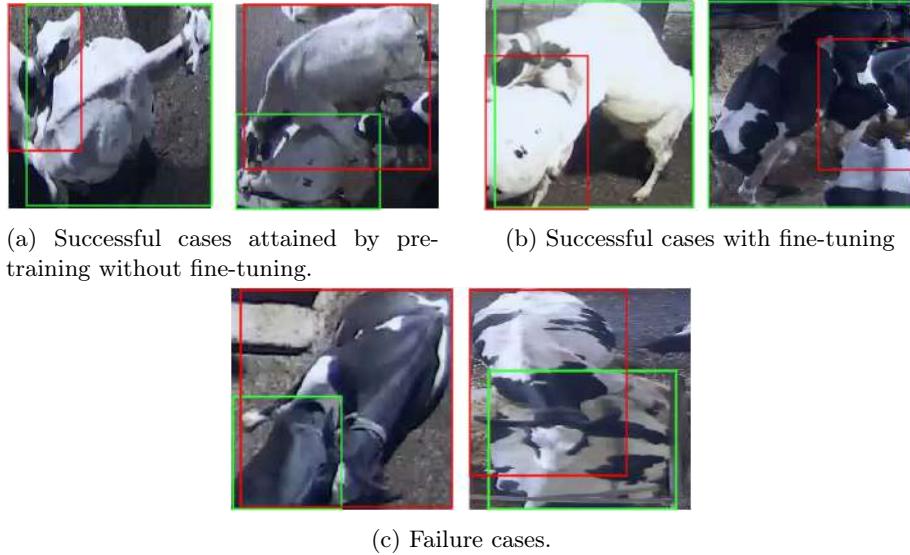


Fig. 7: Example of prediction results (Red bboxes show mounting cows. Green bboxes show cows accepting mounting).

Fig. 7 shows the examples of successful and failure cases. Fig. 7(a) shows the example of images that have already been successfully detected without fine-tuning, trained using only pseudo-images. The positional relationship between the two dairy cows is easy to understand, so the detection could be done by learning only pseudo-images. Fig. 7(b) is an example of failure without fine-tuning. The failure can be caused by the difficulty of generating pseudo-mounting images shot from a side view, as depicted in Fig. 7(b). Fine-tuning is necessary to learn poses that are difficult to generate. The failure cases shown in Fig. 7(c) could be caused by the misidentification of two mottled cows as one cow.

We conducted ablation experiments to evaluate the effect of background replacement images and $L_{consistency}$. Table 3 shows the results of the experiments. Comparing our method in different settings, we see a 4.4 point improvement with background replacement, and a 1.3 point improvement with $L_{consistency}$. The accuracy without background replacement is equivalent to that without pre-training. By training with both pseudo images and background replacement images, it is possible to extract background-independent image features. Furthermore, by adding $L_{consistency}$, common image features such as the positioning of dairy cows could be extracted.

Fig. 8 show the saliency maps generated by Full-Gradient [13] on the pre-trained model without fine-tuning. Comparing the saliency maps obtained by our methods (a), (b), and (c), we can see in Fig. 8(c) that the proposed method (ours(c)) tends to successfully give attention to the body of dairy cows. As in Fig. 8(a) and (b), the model pre-trained without using background replacement

Table 3: Ablation study

method	background replacement	loss	AUC
ours(a)	-	L_{score}	0.857
ours(b)	✓	L_{score}	0.901
ours(c)	✓	$L_{score} + L_{consistency}$	0.914

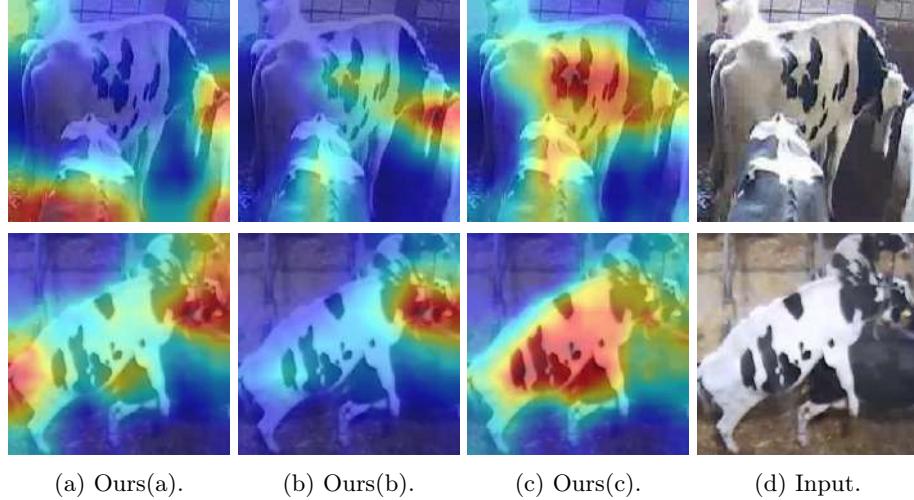


Fig. 8: Results of saliency map.

images or consistency loss gives attention to the background region outside the cow region. The differences in the attention acquired in the pre-training phase may have affected the mounting detection performance after fine-tuning.

6 Conclusions

In this paper, we proposed a method for the automatic detection of cow mounting behavior. The proposed two-phase learning scheme drastically reduces the burden of capturing mounting behaviors, which are a rare occurrence. In the first phase, we train the model on a large number of pseudo-images generated from two dairy cow images. In the second phase, we fine-tune the pre-trained model on a small number of actual mounting images. In future work, we will develop synthesis methods that can generate more realistic mounting behaviors by taking into account dairy cow pose rather than random synthesis. Furthermore, to further reduce the burden of data collection, unsupervised methods that do not require actual mounting images need to be developed.

Acknowledgements The authors thank the members of Tsuchiya Manufacturing Co. Ltd. for helpful discussions and for providing the video data of barns. This work was supported by JSPS KAKENHI Grant Number JP20K12115.

References

1. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
2. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In: International Conference on Learning Representations (2021)
3. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
4. Li, C.L., Yoon, J., Sohn, K., Pfister, T.: CutPaste: Self-Supervised Learning for Anomaly Detection and Localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021)
5. Li, D., Chen, Y., Zhang, K., Li, Z.: Mounting Behaviour Recognition for Pigs Based on Deep Learning. *Sensors* **19**(22) (2019)
6. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
7. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10012–10022 (2021)
8. Nasirahmadi, A., Hensel, O., Edwards, S.A., Sturm, B.: Automatic detection of mounting behaviours among pigs using image analysis. *Computers and Electronics in Agriculture* **124**, 295–302 (2016)
9. Noe, S.M., Zin, T.T., Tin, P., Kobayashi, I.: Automatic detection and tracking of mounting behavior in cattle using a deep learning-based instance segmentation model. *Int. J. Innov. Comput. Inf. Control* **18**, 211–220 (2022)
10. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: European conference on computer vision. pp. 69–84. Springer (2016)
11. Sato, M., Kato, H., Noguchi, M., Ono, H., Kobayashi, K.: Gender differences in depressive symptoms and work environment factors among dairy farmers in japan. *International journal of environmental research and public health* **17**(7), 2569 (2020)
12. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-CAM: Visual explanations from deep networks via gradient-based localization. In: ICCV (2017)
13. Srinivas, S., Fleuret, F.: Full-Gradient Representation for Neural Network Visualization. In: Advances in Neural Information Processing Systems (2019)
14. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International conference on machine learning. pp. 6105–6114. PMLR (2019)

14 Yuta Okuda et al.

15. Wang, R., Gao, Z., Li, Q., Zhao, C., Gao, R., Zhang, H., Li, S., Feng, L.: Detection Method of Cow Estrus Behavior in Natural Scenes Based on Improved YOLOv5. *Agriculture* **12**(9), 1339 (2022)
16. Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., Hu, H.: SimMIM: A Simple Framework for Masked Image Modeling. In: International Conference on Computer Vision and Pattern Recognition (2022)

Classification of Lung and Colon Cancer Using Deep Learning Method

Md. Al-Mamun Provath^{1[0000-0002-1203-3331]}, Kaushik Deb^{1[0000-0002-7345-0999]}, and Kang Hyun Jo^{2[0000-0001-8317-6092]}

¹ Department of Computer Science and Engineering, Chittagong University of Engineering & Technology (CUET), Chattogram 4349, Bangladesh

u1704098@student.cuet.ac.bd, debkaushik99@cuet.ac.bd,

² Department of Electrical, Electronic and Computer Engineering, University of Ulsan acejo@ulsan.ac.kr

*Correspondence: debkaushik99@cuet.ac.bd

Abstract. Cancer seems to have a significantly high mortality rate as a result of its aggressiveness, significant propensity for metastasis, and heterogeneity. One of the most common types of cancer that can affect both sexes and occur worldwide is lung and colon cancer. It is early and precise detection of these cancers which can not only improves the rate of survival but also increase the appropriate treatment characteristics. As an alternative to the current cancer detection techniques, a highly accurate and computationally efficient model for the rapid and precise identification of cancers in the lung and colon region is provided. For the training, validation and testing phases of this work, the LC25000 dataset is used. Cyclic learning rate is employed to increase the accuracy and maintain the computational efficiency of the proposed methods. This is both straightforward and effective which facilitates the model to converge faster. Several transfer learning models that have already been trained are also used, and they are compared to the proposed CNN from scratch. It is found that the proposed model provides better accuracy, reducing the impact of inter-class variations between Lung Adenocarcinoma and another class Lung Squamous Cell Carcinoma. Implementing the proposed method increased total accuracy to 97% and demonstrate computing efficiency in compare to other method.

Keywords: Convolutional Neural Network, Transfer Learning, Lung Cancer Pathology.

1 Introduction

The word cancer is used to describe a large group of diseases that affect various body parts. One of the characteristics that distinguishes cancer is the unrestrained, fast proliferation of aberrant cells that cross their normal borders and have the potential to infiltrate other organs. International Agency for Research on Cancer (IARC) of the World Health Organization (WHO) [1] reports that in 2020, cancer is the greatest cause of death worldwide, accounting for 19 million new cases and approximately 10 million deaths. The main reason for death from cancer is metastasis, which occurs when cancer spreads from its primary place to another organ of the body without the aid of adhesion chemicals. Any organ in the human body could get cancer, but the lung, colon, rectum, liver, stomach, and breast are the most frequently affected organs. The most common cancers that cause deaths in both men and women are colon and lung cancer. Globally, there were 2.21 million new cases of lung cancer in 2020, 1.93 million cases of colorectal cancer, 1.80 million lung cancer-related deaths, and approximately 1 million colorectal cancer deaths [2]. Behaviors as a high body mass index, a drinking habit, or smoking are factors in the development of cancer. Along with genetic ones, there are physical toxins like radiation and UV rays in [2]. When lung cells mutate, they grow uncontrollably and combine into a mass known as a tumor, which is when they turn malignant. The colon, the last part of our digestive system, may develop colon cancer if it has malignant cells. In the majority of cases of colon cancer, a tumor develops as normal cells that line the colon or rectum enlarge out of control.

Without a broad spectrum of diagnostic techniques, cancer detection task is difficult. Patients usually have little or no disease symptoms, but by the time they appear, it is frequently too late. Understanding metastases is a critical topic of cancer research because metastatic illness causes

90% of cancer deaths [3]. Colon cancer frequently metastasizes to the liver, lungs, and peritoneum, while lung cancer frequently metastasizes to the brain, liver, bones, and other areas of the lungs. Although symptoms are commonly linked to the presence of cancer cells in the organ where they spread, the metastatic cells would look under a microscope to be sick primary organ cells [4]. Early detection and appropriate treatment are now the main ways to reduce the frequency of cancer-related mortality [5]. If colon cancer is discovered at Stage 0, for instance, more than 92 percent of patients between the ages of 18 and 73 can live with the appropriate medication, and 83% in Stage 1, 67%, in Stage 2, 11% in Stage 3. The relative lung cancer survival rates are 69%, 50%, 29%, and 8% in [6]. The high cost of screening equipment prevents many people from using them. 70% of deaths caused by cancer in countries other than those with greater incomes [2]. The solution to this issue may lie in a field that has nothing to do with medicine. It is medical field which use deep learning for numerous purposes in [7].

In order to categorize and forecast different kinds of biological signals, machine learning methods have been utilized. Deep Learning (DL) techniques have been developed, allowing machinery that deals with data which are by nature high in dimension including images, videos. A CNN model was developed from scratch to extract features from pathological images, carry out end-to-end training, gradually and accurately categorize the Lung and Colon Cancer pathological images. The hyperparameters were tuned to ensure the best configuration and a learning process cyclical in nature was used to reduce computation and make the model faster.

The following is a list of this paper's key contributions:

1. In order to improve classification performance, a scratch CNN model is developed.
2. The inclusion of the cyclical learning rate approach in the proposed model delivers substantial performance increases and lowers the computational expense.
3. To increase the accuracy and compare accuracy with different transfer learning methods from others method.

2 Related Works

For more than 40 years, researchers have studied the automatic assistant diagnosis of cancer by classifying histopathological images into non-cancerous or malignant patterns for analysis, which is the initial aim of the image analysis system. The complexity of image analysis, however, made it difficult to deal with the complexity of histological images. Approximately 40 years ago [8], investigated the possibility of automatic image processing, but the difficulty of analyzing complex images makes it still difficult today. Back then, implementing machine learning-based computer-aided diagnosis (CAD) required feature extraction as a crucial step. Different cancer ontologies have been looked into in studies by in [9] provide a thorough overview of cancer diagnosis by carrying out tests of various deep learning methods. Additionally, it offers comparisons of the various prominent architectures. The next few paragraphs, briefly discuss the previous works by the researcher.

A representational Sparse in nature Classification (mSRC) technique of diagnosing cancer of lung was described by in [10]. The authors used samples from needle biopsies to automatically segment regions of nuclei numbered 4372 of the diagnosis of cancer in lung. This approach has average classification accuracy of 88.10%. In [11], on the basis of the examination of CT scan images, to classify cancer an approach was followed by authors and which was dealing with CAD. They took six different statistical feature and forward and its reverse propagation are the two types of networks which were used. The comprehensive analysis demonstrates that skewness, when combined with ANN with back-propagation, yields the best classification results. In [12], a classification method which is free of label for grading cancer in colon was published. Different dedifferentiation states of colon cancer and infrared spectral histopathology imaging were used in this work. Random Forest, a supervised learning technique based on Decision Trees (DT), carried out the classification (RF). In [13], a technique was proposed which can analyze colonoscopy video to identify cancer and that can automatically identify polyps from colonoscopy video was described by Yuan et al. They employed AlexNet, a well-known CNN based architecture, for classification, which had an accuracy rate of 91.47%. In [14], a technique for cancer in lung, stage detection was proposed by Masood et

al. The researchers evaluated their model using six different datasets and used CNN and DFCNet in their research. A swarm optimization- based technique for cancer in lung prediction was presented in [15] using images from various sources. A maximum accuracy of 98% was attained using their learning algorithm of choice, the Recurrent Neural Network (RNN).

In order to detect colorectal cancer from colonoscopy videos, A method which is based on neural network was and weights of binary nature was used to classify in [16]. Collected data was evaluated and achieved classification accuracy of more than 90%. In [17] an approach of automatic in nature was developed for detecting lung cancer. They used the Wolf heuristic feature selection approach and bin smoothing for the normalization mechanism. The classifier applied in this study neural network of learning of ensemble kinds was the most intriguing aspect of the study's methodology. Its accuracy was over 99%.

In [18], proposed a CNN model after extracting more than three sets of features, from histopathological images of lung and colon cancer. Authors used convolutional layers of numbered three pooling double times, single batch normalization with dropout for this classification task. Authors have also showed a comparison of related research where the proposed method of 96.33% accuracy the method can identify tissues of desired nature, performing well than other works.

As a result, it can be concluded that the classification of both lung and colon cancer has had a significant impact for a long time. Deep learning models combined with a wide range of configurations have recently exceeded current state-of-the-art methods, as well. There is a huge amount of scope for initiating innovation and development in this developing research field to overcome this.

3 Datasets

3.1 LC25000

This dataset, has images total of 25000 and which are of different types - total five in number [19]. These variations include lung adenocarcinoma, lung squamous cell cancer, benign lung tissue, benign colonic tissue, and lung adenocarcinoma. The authors principally gathered 1250 images of tissues which are of cancer types (250 images of each category). Several techniques were used to increase images of each class (5000 images in each class). Before using the augmentation techniques, to make a square of 768X768 pixels from their original size of 1024X768 cropping was used. The dataset has the nature of compliance, and validation, and use of every image in the dataset is totally free. The dataset's contents are listed in Table 1, along with the class names.

Table 1: LC25000 Dataset Summary

Cancer Type	Samples
Colon Adenocarcinoma	5000
Colon Benign Tissue	5000
Lung Adenocarcinoma	5000
Lung Benign Tissue	5000
Lung Squamous Cell Carcinoma	5000
Total	25000

4 Methods

4.1 Cyclic learning Rate

CNNs are one of the most effective architectural designs for the issue of image classification. To extract the most unique features from an image's pixels, CNNs utilize filtering methods. The most essential hyperparameter to adjust while deep learning deep neural networks is the learning rate, that is well known.

The learning rate can cycle between acceptable boundary values using this strategy rather than monotonically decreasing. Training using learning rates of cyclical in nature rather than choosing

values increases accuracy without the necessity of trial and error method and also frequently requires fewer iterations.

The fundamental idea behind cyclical method is based on the idea that speeding up learning could have both short-term detrimental effects and long-term beneficial outcomes. This discovery inspires the concept that rather of using a stepwise fixed or exponentially declining value, the learning rate should be allowed to vary within a range of values. Due to the fact that the triangle window is the most straightforward function that contains both linear rising and linear decreasing, this led to its adoption which is illustrated in Figure 1.

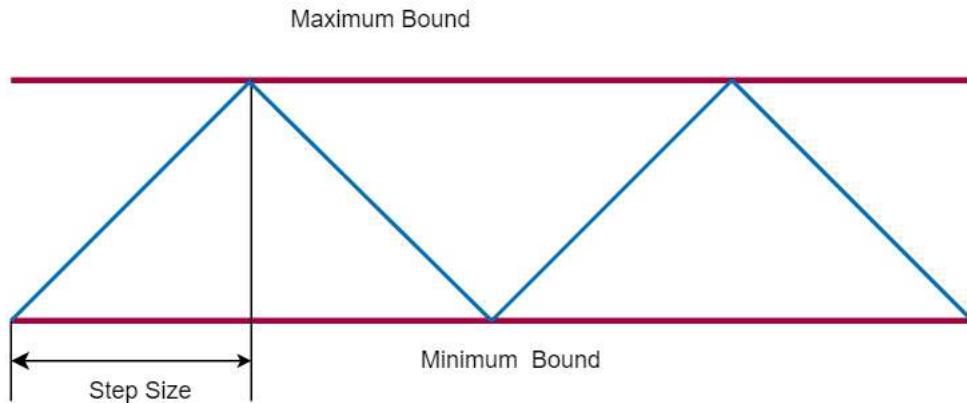


Fig. 1: Triangular Learning Rate Policy

The loss was estimated against the learning rate, and based on the learning rate in Figure 2, the base lr value was adjusted to 0.003 as the loss was declining.

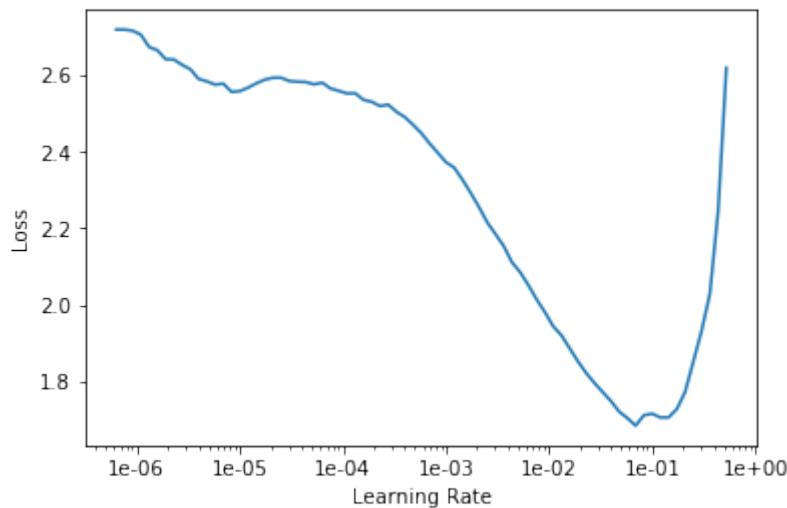


Fig. 2: Loss versus Learning Rate (base learning rate)

The best learning rate is determined after 8 training epochs by re-running the cyclic learning rate. The distinction of the learning rate at the onset of loss diminution and at the juncture where the

loss's declination transforms into irregularity or commences to escalate constitutes optimal limits for specifying the base and maximum learning rate, respectively[20]. In conformity with this, the base learning rate is instituted at the former value, and the maximum learning rate is established at the latter value, as exemplified in Figure 3, where the base lr was set to 0.0045 and max lr was set to 0.0301.

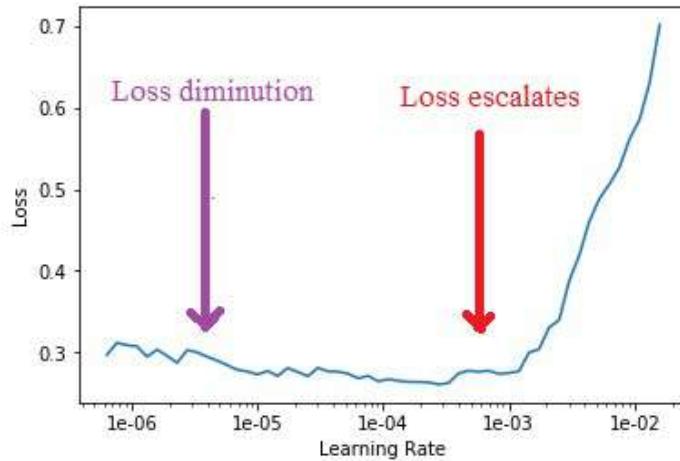


Fig. 3: Loss versus Learning Rate

4.2 Proposed CNN Architecture

In a 3D space with two spatial dimensions (width and height) and one channel dimension, a convolution layer attempts to learn filters. RGB images with a size of 64x64 pixels were used when using this layer. Thus, mapping cross channel correlations and spatial correlations are both accomplished using a single convolution kernel. The input data is divided into three or four smaller areas than the original input space, and any cross channel correlations are then mapped in these smaller 3D spaces using standard 3x3 or 5x5 convolutions. In deep learning frameworks a depth wise separable convolution commonly referred to as a "separable convolution"—consists of a depth wise convolution. A pointwise convolution followed by a spatial convolution carried out individually across each input channel. Despite to separable convolution might implies, this is not to be mistaken with a spatially separable convolution. Although a ReLU non-linearity follows both operations, depths wise separable convolutions are typically performed without nonlinearities.

Each layer of the network can learn more independently due to the layer of batch normalization. It uses normalization to adjust the output of the prior layers. The input layer is scaled during normalization. When batch normalization is used, learning is more successful. To avoid overfitting the model, batch normalization was employed as a regularizer. Three Residual blocks were used in the model. Spatial convolution layers and batch normalization make up the residual block.

Dropouts are a regularization technique that prevents model overfitting. Neurons in the network are modified in some percentage randomly as dropouts are added. When neurons are turned off, the connections to their incoming and outgoing neurons are also disconnected. A pooling method called global average pooling is intended to take the place of fully connected layers in conventional CNNs. One feature map should be produced for each associated classification task category. After creation of the model, the softmax activation function was used to classify the lung and colon histopathology images.

The convolutional layer, batch normalization layer, and residual block were just a few of the layers that constitute a CNN's architecture, as depicted in Figure 4.

Our proposed model's properties are described in Table 2.

Table 2: Property Specification Table of Proposed Model

Specification	Value
Input image	64x64x3
Activation of Conv_2D layers	Relu
Pooling 2D layers	3x3
Output layer activation	Softmax
Optimizer for compilation	Adam

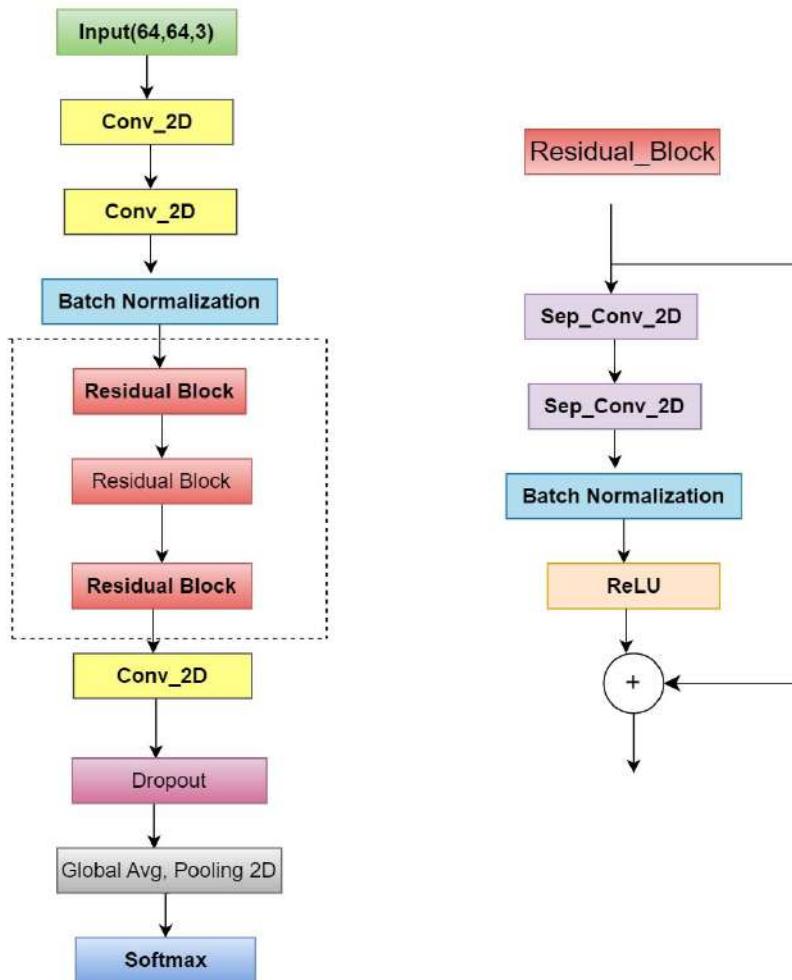


Fig. 4: Proposed Convolutional Neural Network Architecture

5 Result Analysis

In this part, the system configurations have been outlined. The best model's f1-score, recall, and precision are a few of the accuracy metrics that stand out. For further study of the best model, the classification report, confusion matrix, and performance graphs are evaluated.

On LC25000, multiclass classification analyses are performed. The down sampling technique was employed to reduce the size of the image to 64x64. In the LC25000 dataset, various pre-trained transfer learning models were also used. The following are the accuracy comparisons between several transfer learning models and ours:

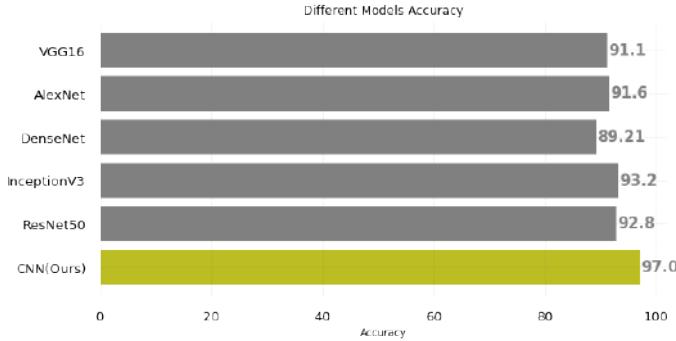


Fig. 5: Accuracy Comparison of Different Models

It is clear from Figure 5 that proposed model's testing accuracy outperforms transfer learning models including those used by other authors. The nature of benchmark datasets and histopathology slides are very different, therefore features at low levels retrieved using transfer learning methods that rely on benchmark datasets are not very relevant in this instance.

An optimizer is a method, such as a function or algorithm, that modifies a neural network's characteristics. It helps to increase accuracy and decrease overall loss. The following three optimizers were utilized in this model: Adam, SGD, and RMSProp are shown in Figure 6.

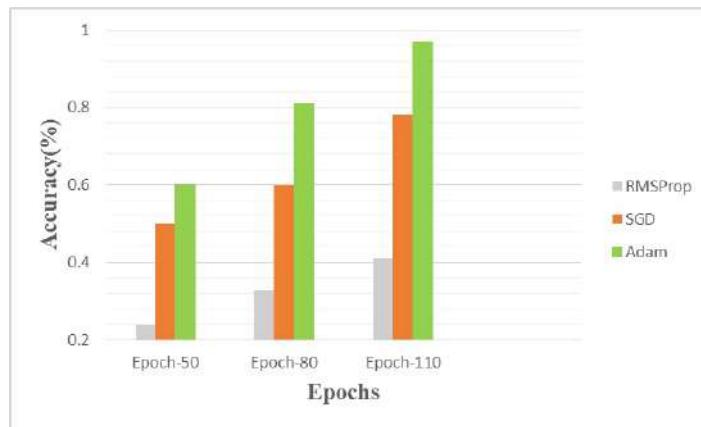


Fig. 6: Optimizer Comparison Curve

Adam's accuracy in the model was the highest according to the following figure and for this Adam was chosen because its accuracy in the model was the highest, as shown by Figure 6.

The samples numbers that are before processing the model hyper tune is the size of each batch. The batch size utilized was shown in Figure 7, and it was determined by the number of samples before processing the model hyperparameter tune.

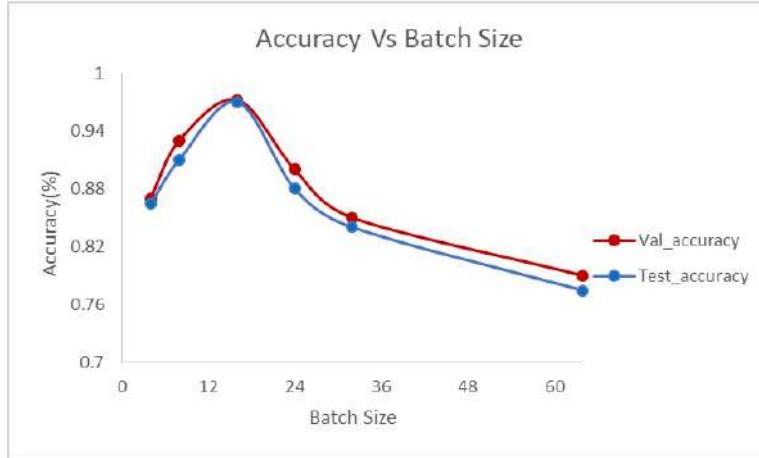


Fig. 7: Accuracy versus Batch Size

Batch size 16 was chosen because it provided the highest accuracy among others, as determined by the comparison curve.

Table 3 explains how the proposed model's hyperparameters were configured.

Table 3: Hyperparameter Configuration of Proposed Model

Hyper Parameter	Range	Optimal Value
Batch Size	4,8,16,24,32,64	16
Epoch	50, 60, 80, 90, 100, 110	100
Optimizer	Adam, RMSProp, SGD	Adam
Dropout	0.20, 0.30, 0.4, 0.50	0.4

5.1 Proposed Model Output

A ratio of 80:10:10 for training, validation, and testing is maintained when the dataset is split in the proposed model. Several parameters are taken into account for performance analysis, as depicted in Figures 8 and 9, to demonstrate the performance of the proposed approach. The accuracy of training and validation was included in Figure 10.

	precision	recall	f1-score	support
0	1.00	0.91	0.96	500
1	0.97	1.00	0.98	500
2	0.93	0.94	0.94	500
3	1.00	1.00	1.00	500
4	0.94	0.97	0.96	500
accuracy			0.97	2500
macro avg	0.97	0.97	0.97	2500
weighted avg	0.97	0.97	0.97	2500

Fig. 8: LC2500 Dataset Classification Report

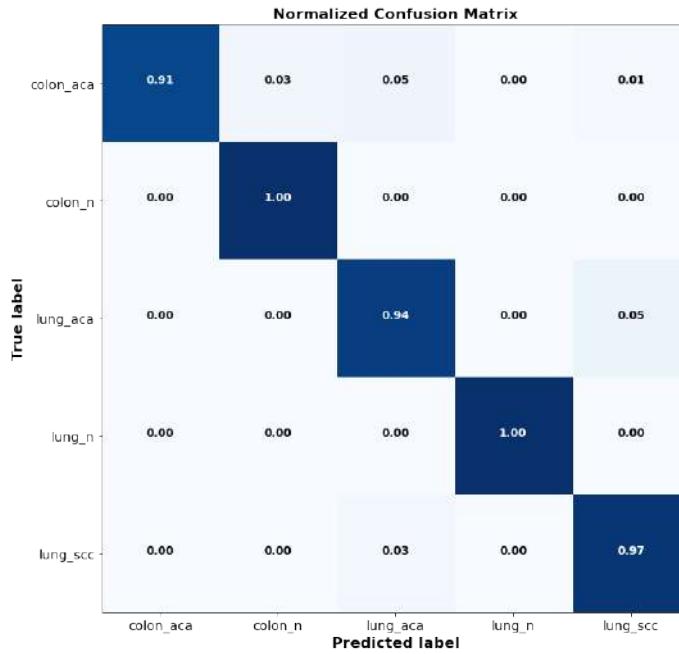


Fig. 9: LC25000 Dataset Normalized Confusion Matrix



Fig. 10: Training and Validation Accuracy Curve

The experiment's results for multiclass classification accuracy were measured and compared to those of other authors, as shown in Table 4. Though the same dataset was not used by all of the authors, as the purpose of the task remains the same, they were compared with the proposed approach. The classification accuracy was 97% and it significantly reduces inter-class variation between two forms of cancer in Lung Adenocarcinoma and Squamous Cell Carcinoma. In prior studies [21], [22] the authors neglected to integrate residual blocks, which are essential components that enable the establishment of deeper network architecture and effectively alleviate the vanishing gradient problem. Additionally, the implementation of skip connections within Residual Networks facilitates the seamless transmission of information across the network, thereby facilitating optimization. The

CNN models in references [22] and [23] suffer from limited feature extraction and overfitting issues due to their shallow network architecture. The proposed model incorporates three ResNet blocks and the use of separable convolution layers within these blocks, which significantly reduces the number of parameters required and results in a compact model with expedited training times. This proposed model reduces 1.95 times training times than described in [24].

Table 4: Author Accuracy Comparison Table

Author	Types of Images	Model	Accuracy	Precision	Recall	F Measure
Y. Shi et al.,[25],13	Biopsy Image	mSRC	88.1	84.6	91.3	86.6
Y. Xu et.al, [26],13	Histopathological	SVMs		73.7	68.2	70.8
Kuruvilla et al.,[27] ,14	CT scan	ANN	93.3		91.4	
Sirinukunwattana K et.al, [28], 16	Histopathological	CNN		78.3	82.7	80.2
Kuepper et.al, [23], 16	Histopathological	RF	95		94	
W. Shen et.al, [24], 17	CT scan	CNN	87.14		93	
Z.Yuan et.al, [13], 17	Colonoscopy	AlexNet	87.14		91.76	
T.Babu et al.,[29],18	Histopathological	RF	85.3			85.2
M.Akbari et al.,[30],18	Colonoscopy	CNN	90.28	74.34	68.32	71.2
Suresh et.al, [21], 20	CT scan	CNN	93.9		93.4	
M.Masud et.al, [22], 21	Histopathological	CNN	96.33	96.39	96.37	96.38
Proposed	Histopathological	CNN	97	97	97	97

It can be concluded from Table 4 that the accuracy of the proposed method surpasses that of other authors.

6 Conclusion

Lung and colon cancer are attributed as mostly caused cancer types among all other types. Early identification of cancer can help patients rate of survival to increase. The prime purpose of the proposed method was to provide a more robust and reliable approach for these two forms of cancer. For this detection, transfer learning was used on a dataset of 25,000 histopathological images of colon and lung tissues. When using the suggested scratch CNN method, accuracy was greatly improved and reached 97%. The proposed methodology offers improved accuracy over current methods for detecting lung and colon cancer while also taking less time and using less computational resources. All the experiments verify the effectiveness of the proposed method regarding the task of detecting cancer. Future incorporation of attention with scratch CNN model to increase the classification accuracy and explore more details of the image.

References

- I. A. for Research on Cancer, "World Fact Sheet," <https://gco.iarc.fr/today/data/factsheets/populations/900-world-fact-sheets.pdf/>, 2020, [Online; accessed 26-June-2022].
- I. H. Organization, "Cancer," <https://www.who.int/news-room/fact-sheets/detail/cancer/>, 2022, [Online; accessed 26-June 2022].
- T. N. Seyfried and L. C. Huysentruyt, "On the origin of cancer metastasis," *Critical Reviews™ in Oncogenesis*, vol. 18, no. 1-2, 2013.
- Verywellhealth, "What Is Metastasis?" <https://www.verywellhealth.com/metastatic-cancer-2249128/>, 2022, [Online; accessed 27-June-2022].
- L. F. Sánchez-Peralta, L. Bote-Curiel, A. Picón, F. M. Sánchez-Margallo, and J. B. Pagador, "Deep learning to find colorectal polyps in colonoscopy: A systematic literature review," *Artificial Intelligence in Medicine*, vol. 108, p. 101923, 2020.
- C. Health, "Cancer Survival Rates," <https://cancersurvivalrates.com/?type=colon&role=patient/>, 2022, [Online; accessed 26-June-2022].
- S. Das, S. Biswas, A. Paul, and A. Dey, "Ai doctor: An intelligent approach for medical diagnosis," in *Industry Interactive Innovations in Science, Engineering and Technology*, S. Bhattacharyya, S. Sen, M. Dutta, P. Biswas, and H. Chattopadhyay, Eds. Singapore: Springer Singapore, 2018, pp. 173–183.

8. K. Doi, "Computer-aided diagnosis in medical imaging: Historical review, current status and future potential," *Computerized Medical Imaging and Graphics*, vol. 31, no. 4, pp. 198–211, 2007, computer-aided Diagnosis (CAD) and Image-guided Decision Support.
9. G. M. te Brake, N. Karssemeijer, and J. H. Hendriks, "An automatic method to discriminate malignant masses from normal tissue in digital mammograms1," *Physics in Medicine & Biology*, vol. 45, no. 10, p. 2843, 2000.
10. Y. Shi, Y. Gao, Y. Yang, Y. Zhang, and D. Wang, "Multimodal sparse representation-based classification for lung needle biopsy images," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 10, pp. 2675–2685, 2013.
11. J. Kuruvilla and K. Gunavathi, "Lung cancer classification using neural networks for ct images," *Computer methods and programs in biomedicine*, vol. 113, no. 1, pp. 202–209, 2014.
12. C. Kuepper, F. Großerueschkamp, A. Kallenbach-Thielges, A. Mosig, A. Tannapfel, and K. Gerwert, "Label-free classification of colon cancer grading using infrared spectral histopathology," *Faraday Discussions*, vol. 187, pp. 105–118, Jan. 2016.
13. Z. Yuan, M. IzadyYazdanabadi, D. Mokkapati, R. Panvalkar, J. Y. Shin, N. Tajbakhsh, S. Gurudu, and J. Liang, "Automatic polyp detection in colonoscopy videos," in *Medical Imaging 2017: Image Processing*, vol. 10133. SPIE, 2017, pp. 718–727.
14. A. Masood, B. Sheng, P. Li, X. Hou, X. Wei, J. Qin, and D. Feng, "Computer-assisted decision support system in pulmonary cancer detection and stage classification on ct images," *Journal of Biomedical Informatics*, vol. 79, pp. 117–128, 2018.
15. R. Selvanambi, J. Natarajan, M. Karuppiah, S. H. Islam, M. M. Hassan, and G. Fortino, "Lung cancer prediction using higher-order recurrent neural network based on glowworm swarm optimization," *Neural Computing and Applications*, vol. 32, pp. 4373–4386, 2020.
16. M. Akbari, M. Mohrekesh, S. Rafiei, S. R. Soroushmehr, N. Karimi, S. Samavi, and K. Najarian, "Classification of informative frames in colonoscopy videos using convolutional neural networks with binarized weights," in *2018 40th annual international conference of the IEEE engineering in medicine and biology society (EMBC)*. IEEE, 2018, pp. 65–68.
17. P. M. Shakeel, A. Tolba, Z. Al-Makhadmeh, and M. M. Jaber, "Automatic detection of lung cancer from biomedical data set using discrete adaboost optimized ensemble learning generalized neural networks," *Neural Computing and Applications*, vol. 32, pp. 777–790, 2020.
18. M. Masud, N. Sikder, A.-A. Nahid, A. K. Bairagi, and M. A. AlZain, "A machine learning approach to diagnosing lung and colon cancer using a deep learning-based classification framework," *Sensors*, vol. 21, no. 3, p. 748, 2021.
19. A. A. Borkowski, M. M. Bui, L. B. Thomas, C. P. Wilson, L. A. DeLand, and S. M. Mastorides, "Lung and colon cancer histopathological image dataset (lc25000)," *arXiv preprint arXiv:1912.12142*, 2019.
20. L. N. Smith, "Cyclical learning rates for training neural networks," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017, pp. 464–472.
21. S. Suresh and S. Mohan, "Roi-based feature learning for efficient true positive prediction using convolutional neural network for lung cancer diagnosis," *Neural Computing and Applications*, vol. 32, no. 20, pp. 15 989–16 009, 2020.
22. M. Masud, G. Muhammad, M. S. Hossain, H. Alhumyani, S. S. Alshamrani, O. Cheikhrouhou, and S. Ibrahim, "Light deep model for pulmonary nodule detection from ct scan images for mobile devices," *Wireless Communications and Mobile Computing*, vol. 2020, pp. 1–8, 2020.
23. C. Kuepper, F. Großerueschkamp, A. Kallenbach-Thielges, A. Mosig, A. Tannapfel, and K. Gerwert, "Label-free classification of colon cancer grading using infrared spectral histopathology," *Faraday discussions*, vol. 187, pp. 105–118, 2016.
24. W. Shen, M. Zhou, F. Yang, D. Yu, D. Dong, C. Yang, Y. Zang, and J. Tian, "Multi-crop convolutional neural networks for lung nodule malignancy suspiciousness classification," *Pattern Recognition*, vol. 61, pp. 663–673, 2017.
25. Y. Shi, Y. Gao, Y. Yang, Y. Zhang, and D. Wang, "Multimodal sparse representation-based classification for lung needle biopsy images," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 10, pp. 2675–2685, 2013.
26. Y. Xu, L. Jiao, S. Wang, J. Wei, Y. Fan, M. Lai, and E. I.-c. Chang, "Multi-label classification for colon cancer using histopathological images," *Microscopy Research and Technique*, vol. 76, no. 12, pp. 1266–1277, 2013.
27. J. Kuruvilla and K. Gunavathi, "Lung cancer classification using neural networks for ct images," *Computer methods and programs in biomedicine*, vol. 113, no. 1, pp. 202–209, 2014.
28. K. Sirinukunwattana, S. E. A. Raza, Y.-W. Tsang, D. R. Snead, I. A. Cree, and N. M. Rajpoot, "Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1196–1206, 2016.

29. T. Babu, D. Gupta, T. Singh, and S. Hameed, “Colon cancer prediction on different magnified colon biopsy images,” in *2018 Tenth International Conference on Advanced Computing (ICoAC)*. IEEE, 2018, pp. 277–280.
30. M. Akbari, M. Mohrekesh, S. Rafiei, S. R. Soroushmehr, N. Karimi, S. Samavi, and K. Najarian, “Classification of informative frames in colonoscopy videos using convolutional neural networks with binarized weights,” in *2018 40th annual international conference of the IEEE engineering in medicine and biology society (EMBC)*. IEEE, 2018, pp. 65–68.

Reproduction of Artwork on Display using Hyperspectral Imaging and Monitor Calibration

Kyudong Sim^{1[0000-0001-6405-0099]} and Jong-II Park^{11[0000-0003-1000-4067]}

¹ Hanyang University, Seoul, South Korea
kdsim@hanyang.ac.kr, jipark@hanyang.ac.kr

Abstract. Reproducing physical artwork in a display screen is difficult due to many limitations in space and equipment. In this paper, we propose a method to make real artwork and monitor artwork the same using hyperspectral imaging and monitor calibration. The spectral reflectance obtained using RGB image with hyperspectral imaging is excellent in color reproduction and can also be used in re-illumination. By using the lighting characteristics of the space where the artwork is located and the color matching function, we can obtain XYZ images from the spectral reflectance. Monitor calibration is a method to obtain the RGB values for outputting XYZ color components, and we can obtain the RGB image of the obtained artwork through monitor calibration. By displaying the obtained RGB image on the monitor, we can confirm that it is similar to the real artwork.

Keywords: Hyperspectral Imaging, Color Correction, Monitor Calibration

1 Introduction

The importance of digitalization of physical artworks is increasing as the number of digital artworks and exhibitions grows. However, reproducing the same color as the physical artwork on a monitor through a simple image capture is not easy, as many conditions such as lighting and color conversion conditions in the location of the image capture are necessary, and those conditions may still limit the situation.

In this paper, we use hyperspectral imaging to obtain the spectral reflectance, and then use monitor calibration to reproduce the color. Hyperspectral imaging is used to obtain the spectral reflectance of the artwork using a conventional RGB camera and color chart. The spectral reflectance is re-illuminated in XYZ color and then converted into RGB color through monitor calibration.

¹ Corresponding author

2 Color reproduction of artwork on the monitor

The reproduction of a artwork through a monitor is carried out through the process shown in Figure 1. Reflectance spectra is obtained from images obtained from multiple illuminations and then re-illuminated into XYZ colors. Then, the artwork in XYZ colors is converted into an RGB image through monitor calibration.

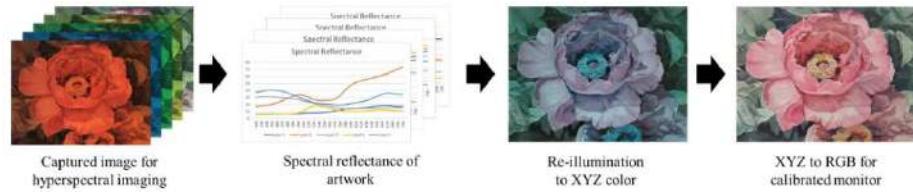


Fig 1. Pipeline of art reproduction methods

2.1 Hyperspectral Imaging

The hyperspectral imaging method for obtaining spectral reflectance using RGB camera and multiple lighting sources can be obtained by using a linear camera model and the PCA basis functions of the spectral reflectance, Parkkinen basis functions. The linear camera model can be expressed as a camera model where the intensity of light is proportional to the RGB values, as shown in equation 1.

$$I = \int p(\lambda)s(\lambda)r(\lambda)d\lambda \quad (1)$$

Here, $p(\lambda)$, $s(\lambda)$, $r(\lambda)$, and I represent the power spectrum of the light source, the camera sensitivity, the object's spectral reflectance, and the RGB values, respectively. spectral reflectance can be expressed as a function of wavelength using the basis functions $b(\lambda)$ and coefficients σ , as shown in equation 2, and can be combined into a single matrix by expressing the function as a wavelength, as shown in equation 3.

$$r(\lambda) = \sum \sigma b(\lambda) \quad (2)$$

$$I = \sigma F \quad (3)$$

By using equation 3, the RGB value and spectral reflectance of an object whose spectral reflectance is known can be obtained by obtaining the function F , and the same lighting can be used to obtain the reflectance coefficient σ . To determine the lighting and camera characteristics, five different lights must be used as shown in Figure 2, and color charts and artwork images are required. By obtaining the lighting conditions and camera characteristics from the color charts with known spectral reflectance, and the same conditions to obtain the artwork image, the spectral reflectance of the artwork can be obtained. This acquired image is relit into XYZ color using the linear camera model, the XYZ color matching function, and the lighting spectrum.



Fig 2. Artworks and color checkers obtained from 5 different lights



Fig 3. The artwork that is re-illuminated into an XYZ image.

2.2 Monitor Calibration

Monitor calibration is the process of determining the relationship between XYZ and RGB colors. Most video data is based on RGB, but in this paper, by generating XYZ images, the effect of lighting can be seen more greatly through direct use of monitor calibration. Monitor calibration is obtained by using a spectroradiometer to acquire the RGBW spectrum of the monitor and then determining the XYZ values through a color matching function. The acquired RGBW XYZ values can be used to create a matrix that transforms from XYZ to RGB.



Fig 4. Reproduced artwork

3 Conclusion

In this paper, we propose a method for reproducing artwork by obtaining the spectral reflectance of the artwork and using XYZ color to re-illuminate it, and using monitor calibration to convert the XYZ image to RGB. This method allows us to reproduce the artwork on a monitor in a way that is similar to the actual object. This was confirmed by confirming that the artwork was reproduced on the monitor in a similar way to the actual object.

References

1. Sim, K.D., Park, J.I., Hayashi, M., and Kuwahara, M.: Artwork Reproduction Through Display Based on Hyperspectral Imaging. In: Proceedings of the International Conference on Human-Computer Interaction, pp. 332-342. Springer, Heidelberg (2022).
2. Park, J.I., Lee, M.H., Grossberg, M.D., and Nayar, S.K.: Multispectral imaging using multiplexed illumination. In: Proceedings of the International Conference on Computer Vision, pp. 1-8. Springer, Heidelberg (2007).

3. Parkkinen, J.P., and Jaaskelainen, T.: Characteristic spectra of Munsell colors. *Journal of the Optical Society of America A*, vol. 6, no. 2, pp. 318-322 (1989).

Game Engine Compatible 3D Clothes Modeling from a Single Image

Soyoung Yoon¹, Sojin Yun¹, and In Kyu Park¹[0000–0003–4774–7841]

Department of Information and Communication Engineering, Inha University
Incheon 22212, Korea
thdud679@gmail.com, sj0524sj@gmail.com, pik@inha.ac.kr

Abstract. In this paper, we propose a clothes modeling technique for 3D human synthesis, as a key ingredient of virtual human modeling. Given a single image, 3D clothes model is reconstructed automatically while being compatible with commercial game engines and graphics tools like Unreal and Maya. Among the pre-selected clothes categories proposed by DeepFashion2 dataset, we classify the clothes types to provide the generic 3D model of clothes. UV map is automatically captured to represent the texture of clothes. Experimental results show that the clothes model is generated with reasonable accuracy and compatibility to Unreal game engine.

Keywords: 3D clothes modeling · Single image · Game engine.

1 Introduction

Metaverse is a combination of ‘Meta’ and ‘Universe’, which means a virtual world linked to reality. In the existing metaverse applications, characters are created by selecting from a few predefined characters and modifying the shape and appearance using the provided assets. With this method, it is difficult to create a 3D virtual human that reflected the variety of human appearances. Therefore, this work proposes a 3D virtual human clothes reconstruction from one’s own single image.

Previously, the work of reconstructing a 3D body model has progressed a lot. Examples of this are [5, 4], which produce 3D surface and texture from images. However, the results of these works are volumetrically modeled with an overall 3D mesh. This integrated mesh is not compatible with the game engine that is the basis of platforms such as Metaverse, so it cannot be used as a character in Metaverse. Therefore, the work of modeling the clothes separately is carried out in [3].

Our previous work [3] estimates the body shape from a single image and models the clothes. However, there are several limitations to this method. For example, very few clothes types are reconstructed, and the texture is often misaligned. Therefore, in this paper, we attempt to solve this problem. This paper focuses on the clothes part and proposes a method for expanding the clothes

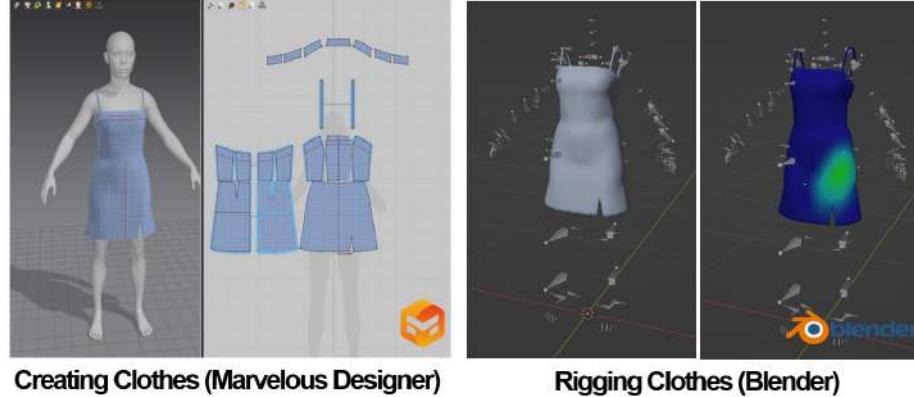


Fig. 1. 3D virtual human custom clothing creation process. Marvelous designer creates clothes that fit the 3D virtual human body type. And then, the clothes are rigged for animation in the Blender.

category and improving the accuracy of texture mapping. To this end, we extract the keypoints for each category of clothing proposed by DeepFashion2 [2] and proceed with a classification of clothes patterns. Thereafter, UV mapping using keypoints and clothes types is performed to represent the 3D clothes from the input image.

2 Proposed Method

2.1 Clothes Geometry Modeling

Essential clothing and body models use files provided by the MetaHuman Creator. However, the basic model of clothes supplied by MetaHuman Creator is limited. 3D tools such as Marvelous Designer and Blender are used to solve these limitations and create various clothes suitable for the 3D virtual human body.

We use Marvelous designer, which is mainly used as a tool for producing 3D clothes. Marvelous designer imports basic 3D virtual human body models as avatars to dress up. Afterward, the clothes are fitted to the body of the 3D virtual human set as an avatar. And the fitted clothes are created as 3D virtual human custom clothes and are exported as an FBX file for compatibility with other software such as blender and unreal engine.

The blender, which is used as a 3D Object editing tool, performs a rigging operation that allows clothes to be animated the same as the body. A 3D virtual human skeleton is combined with the clothes FBX file generated by Marvelous designers, and weight paint is performed on each bone to enable clothes animation. In addition, mesh deformation and UV Map modification are performed to generate clothes optimized for the 3D virtual human body.



Fig. 2. 3D virtual human custom clothing types. Through the process of creating custom clothes, clothes corresponding to 13 categories classified in the dataset DeepFashion2 were composed.

The process of creating custom clothes using two 3D tools can be seen in Fig. 1. And by this process, the results of constructing clothes corresponding to 13 clothes categories can be seen in Fig. 2.

2.2 Clothes Texture Modeling

UV Map is a plane of a 3D model used to wrap a texture. And it can express the texture of clothes. For UV Map generation, the pipeline as shown in Fig. 3 is proposed. In the previous study, clothes parsing was performed from the input image to generate UV Map, and in the case of the top, Pix2Pix was used to blur the boundary. However, inaccuracy was a problem because warping from incorrect points was performed in the UV Map production of the clothes. Therefore, to solve these problems, keypoint detection and clothes pattern classification is added to the existing pipeline.

DeepFashion2 [2] is used as the dataset required for learning clothes keypoint detection. DeepFashion2 has 13 clothes categories, and keypoints proposed for each category are presented according to each clothes characteristic. For learning, clothes classes and bounding boxes, which are results from clothes segmentation, are used together as inputs. And as a model, we proceed with learning using HRNet [6], which has features of maintaining high resolution throughout the entire process.

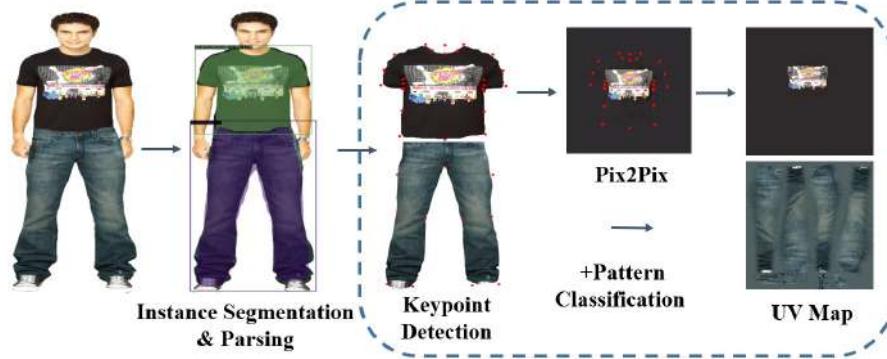


Fig. 3. Clothes UV Mapping Pipeline. When the input image comes in, keypoint detection is performed simultaneously with instance segmentation and parsing. After that, UV Map based Warping is performed with the Pix2Pix and pattern classification results.

The texture of the clothes on the invisible side during UV mapping is inferred. Therefore, according to the characteristics of the clothes, the clothes pattern classification is conducted on whether the pattern is repeated and continued, or not. (Such as the graphic or logo is centered) For the dataset, part of [1] provided by Kaggle was used, and data were organized and learned by image crawling to reflect the diversity of pattern clothes.

A method of generating a clothes UV Map using the above-described information is as follows. With the keypoint coordinate information of the Pix2Pix generation result and the clothes segmentation result, it is mapped by applying a function related to the perspective transformation of OpenCV to the coordinates to be mapped in the UV Map format. In addition, when mapping, it determines whether to reflect the back side according to the clothes pattern classification results and uses a boundary processing function to add naturalness.

3 Experimental Results

Experimental results are shown in Fig. 4. Note that we present the qualitative result only because there is no ground truth in this field. A clothes FBX combination is provided by classifying clothing types among 13 clothes categories from the input image. In addition, according to the UV Map format of the clothes FBX, a UV Map that reflects the texture of the clothes is provided to express the texture of the clothes model.

Looking at the results of Fig. 4, it can be seen that the clothes corresponding to the input image are provided according to the clothes category. However, the generated clothes did not completely match those in the input image. This is because the clothes categories are categorized into 13 categories and generalized, but there are more diverse lengths and fits in reality. Previous work [3] has

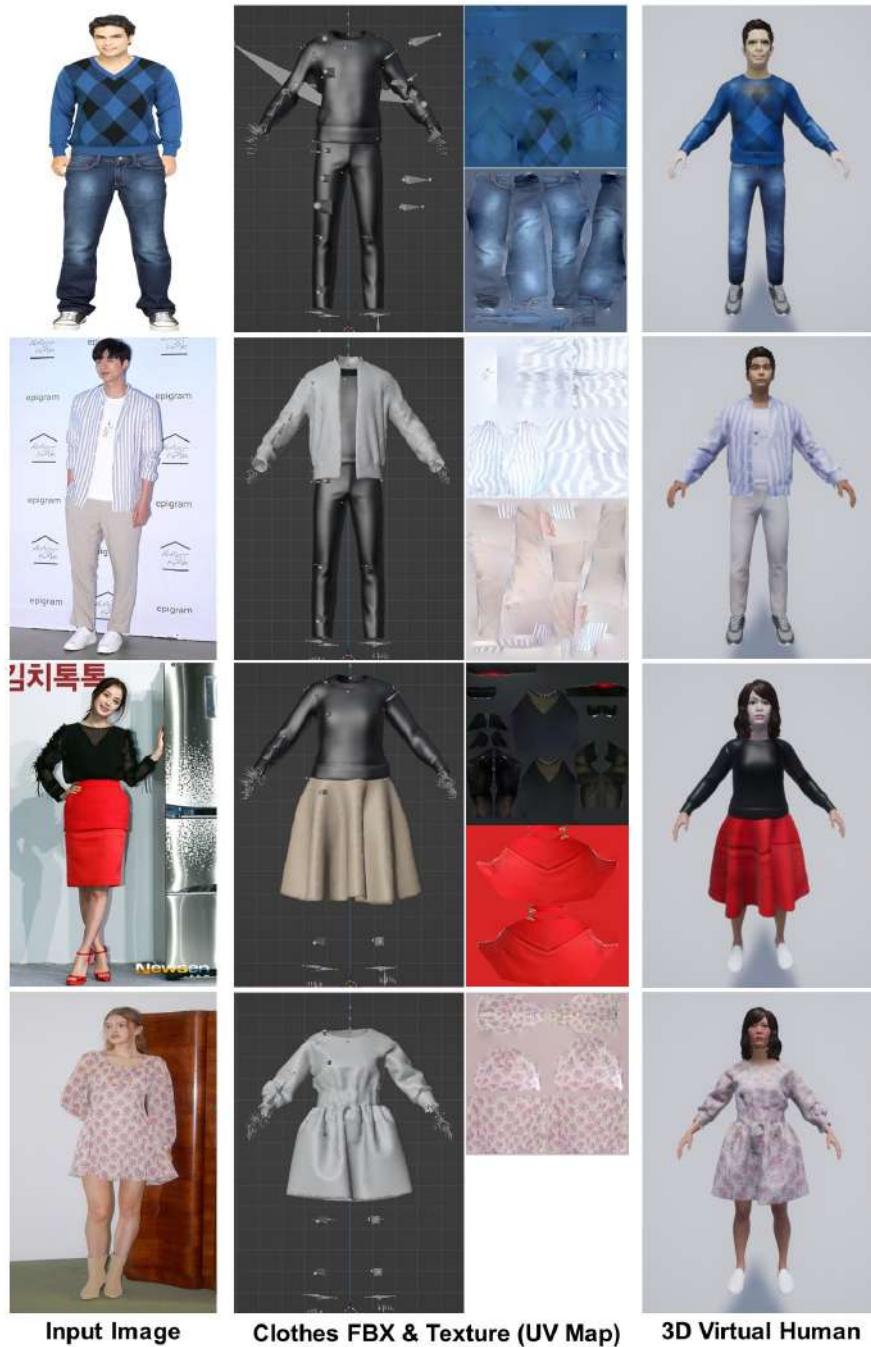


Fig. 4. Results of creating a 3D virtual human. The clothes of the corresponding category are classified and provided from the input image. It also provides UV Map reflecting the texture of the clothes. And the 3D virtual human created with the compatible clothes can be checked on the Unreal Engine.

mentioned errors in UV Map generation due to the inability to find feature points for images that are not frontal or static, or the problem of poor detail because only the center of the clothes is reflected. By checking the UV Map, it can be seen that these problems are solved and the overall details are reflected. However, obstacles such as hands appear as there is no dealing with the occlusion. To overcome this limitation, it seems necessary to handle the hidden side using the information on the occlusion provided by the deepfashion2 Dataset.

Looking at the final 3D virtual human in Fig. 4, the body and face materials are used with costumes to create a complete virtual human being. Although not covered in this paper, the clothes are transformed along with the body according to the body type estimate and the face is also reflected. The 3D virtual human finally created through the automation process can be identified in the Unreal Engine. By performing the clothes FBX rigging work, it is possible to animate the clothes to move the same as 3D virtual human body animation. And these animations can be checked along with the basic animations of the Unreal Engine.

4 Conclusion

This paper presented a clothes modeling method during 3D virtual human restoration from a single image. It was observed that 3D characters that captured the appearance of a real person could be created within the 3D virtualization platform by reflecting the increased types of clothes and detailed textures. We believe it would increase the satisfaction of metaverse users.

Acknowledgements This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2022R1A4A1033549). This work was partly supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-00981, Foreground and background matching 3D object streaming technology development and No.RS-2022-00155915, Artificial Intelligence Convergence Innovation Human Resources Development (Inha University)).

References

1. Fashion product images dataset <https://www.kaggle.com/datasets/paramagarwal/fashion-product-images-dataset>
2. Ge, Y., Zhang, R., Wang, X., Tang, X., Luo, P.: DeepFashion2: a versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5337–5345 (2019)
3. Kim, H.W., Kim, D.E., Kim, Y., Park, I.K.: 3D clothes modeling of virtual human for metaverse. Journal of Broadcast Engineering **27**(5), 638–653 (2022)
4. Remelli, E., Bagautdinov, T., Saito, S., Wu, C., Simon, T., Wei, S., Guo, K., Cao, Z., Prada, F., Saragih, J., Sheikh, Y.: Drivable volumetric avatars using texel-aligned features. In: Proc. ACM SIGGRAPH 2022 Conference. pp. 1–9 (2022)

Game Engine Compatible 3D Clothes Modeling from a Single Image 7

5. Saito, S., Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., Li, H.: Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In: Proc. IEEE/CVF International Conference on Computer Vision. pp. 2304–2314 (2019)
6. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognitionn. pp. 5693–5703 (2019)

Event-Based Reflectance Separation

Ryota Kunimasu¹, Ryo Kawahara^{1[0000-0002-9819-3634]}, and
Takahiro Okabe^{1[0000-0002-2183-7112]}

Department of Artificial Intelligence,
Kyushu Institute of Technology
680-4 Kawazu, Iizuka, Fukuoka 820-8502, Japan
okabe@ai.kyutech.ac.jp

Abstract. In this extended summary, we introduce our proposed method for diffuse-specular separation using an event-based camera. Our setup observes a scene of interest by using a standard camera and an event-based camera through a rotating linear polarizer. Our method combines the pixel values with low temporal resolution and low dynamic range and the events, *i.e.* the asynchronous changes of logarithmic radiance values, and achieves diffuse-specular separation with high temporal resolution and high dynamic range.

Keywords: reflectance separation · event-based camera · polarization

1 Introduction

The reflected light observed on an object surface consists of a diffuse reflection component and a specular reflection component in general. Separating those reflection components is important for preprocessing of various techniques in CV and CG. Because the contrast between specular and diffuse reflection components are large, diffuse-specular separation often requires High Dynamic Range (HDR) images. Therefore, diffuse-specular separation with high temporal resolution is an important issue to be addressed.

In this extended summary, we introduce our proposed method for diffuse-specular separation using an event-based camera. Our setup observes a scene of interest by using a standard camera and an event-based camera through a rotating linear polarizer. Our method combines the pixel values with low temporal resolution and low dynamic range and the events, *i.e.* the asynchronous changes of logarithmic radiance values, and achieves diffuse-specular separation with high temporal resolution and high dynamic range.

2 Related Work

Reflectance separation: For diffuse-specular separation, we can make use of the difference not only in the colors [4] but also in the polarization states [1] of diffuse and specular reflection components; specular/diffuse reflection components are polarized (partially polarized)/unpolarized (weakly polarized). The

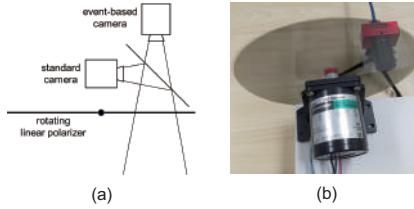


Fig. 1. Our setup for reflectance separation: (a) a sketch and (b) a prototype.

images with different polarization angles can be captured by using a polarization camera [5] as well as a pair of a standard camera and a rotating linear polarizer [6, 3]. However, diffuse-specular separation with high temporal resolution is difficult even if we use a polarization camera. This is because the contrast between specular and diffuse reflection components are large, *i.e.* specular/diffuse reflection components are often saturated/too dark, and therefore HDR images are required.

Event-based camera: In contrast to standard cameras which capture the radiance values of a scene, event-based cameras capture the changes of radiance values. More specifically, they capture the changes of logarithmic radiance values asynchronously, and therefore events have high dynamic range and high temporal resolution [2]. Our study is a novel application of event-based cameras; we utilize it for photometric analysis of diffuse-specular separation based on polarization.

3 Proposed Method

Setup: Fig. 1 (a) shows the sketch of our setup; a scene of interest is observed by a standard camera and an event-based camera through a linear polarizer rotating at high speed. Because specular/diffuse reflection components are polarized/unpolarized, the radiance values of specular/diffuse reflection components seen through the polarizer are variable/constant with respect to the rotational angle of the polarizer. Since event-based camera captures the changes of radiance values, specular reflection components can be detected from the events. Note that we use a standard camera for acquiring diffuse reflection components and for fixing the scales of specular reflection components.

Self-calibration: We self-calibrate the global parameters of our system: the angular velocity of the polarizer ω and the threshold for event occurrence δ ¹. At a point p on a static object, the radiance value $i_p(t)$ at time t is described by

$$i_p(t) = a_p \sin(\omega t + \phi_p) + b_p, \quad (1)$$

where a_p , b_p , and ϕ_p are the amplitude, bias, and phase of the radiance values. First, we estimate ω , a_p , b_p , and $\phi_p^{(s)}$ from the captured images via least

¹ If an event occurs when the radiance value is i_1 , then the next event occurs when the radiance value is i_2 such that $\delta = |\ln i_2 - \ln i_1|$.

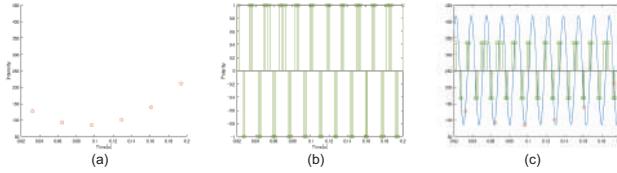


Fig. 2. The result of the self-calibration: (a) the observed pixel values, (b) the observed events, and (c) the estimated radiance values.

squares. Because our setup does not synchronize the devices, the radiance phase $\phi_p^{(s)}$ includes the shift of the time axes between the rotating polarizer and the standard camera. Second, we estimate δ and $\phi_p^{(e)}$ from the events on the basis of the consistency between the acquired events and the radiance values. Here, the event phase $\phi_p^{(e)}$ includes the shift of the time axes between the rotating polarizer and the event-based camera. We use the estimated global parameters ω and δ for the following reflectance separation.

Reflectance separation: When we can use only the images captured by the standard camera, three frames are required for diffuse-specular separation. This is because there are three unknowns in eq.(1): a_p , b_p , and $\phi_p^{(s)}$ except for the global parameter ω . Specifically, we can estimate those unknowns via least squares, and then consider a_p and $(b_p - a_p)$ as the specular and diffuse reflection components respectively. When we can use both the images and events, we can separate reflection components only from a single image and at least neighboring four events. This is because there are four unknowns: a_p , b_p , $\phi_p^{(s)}$, and $\phi_p^{(e)}$ in total². Moreover, if the pixel values at a certain pixel are saturated, we can interpolate/extrapolate the radiance values during saturation from non-saturated pixel values in neighboring frames and the events. Hence, the events increase not only the temporal resolution but also the dynamic range of separation.

4 Experiments

Setup: Fig. 1 (b) shows the prototype of our setup. We used DAVIS346 from iniVation, which can capture coaxial standard images in addition to events. We rotated a linear polarizer in front of the camera at approximately 30 revolutions per second by using a synchronous electric motor.

Results: Fig. 2 shows the result of the self-calibration by using the pixel values and events observed at a point on a static object. We estimated the angular velocity of the rotating polarizer ω and the amplitude, bias, and radiance phase from (a) the pixel values, and then estimated the threshold for event occurrence

² Since the events capture the changes of logarithmic radiance values, n events give only $(n - 1)$ constraints.

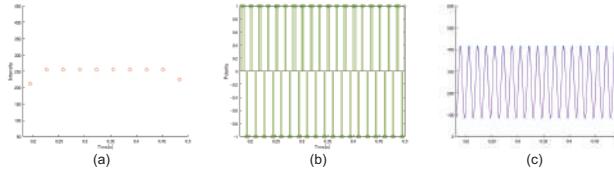


Fig. 3. The result of the interpolation of radiance values during saturation: (a) the observed pixel values, (b) the observed events, and (c) the comparison.

δ and the event phase from (b) the events. In Fig. 2 (c), the observed pixel values, the observed events, and the estimated radiance values are superimposed.

Fig. 3 shows the result of the interpolation of radiance values during saturation: (a) the observed pixel values, (b) the observed events, and (c) the comparison between the estimated and the ground truth radiance values. Here, we observed a point on a static object, and then considered the radiance values estimated from non-saturated pixel values in the other frames as the ground truth. We can find that the estimated radiance values (purple line) almost overlaps the ground truth radiance values (blue line) in (c).

5 Conclusion and Future Work

In this extended summary, we introduced our proposed method for diffuse-specular separation using an event-based camera. Our method combines the pixel values with low temporal resolution and low dynamic range and the events, and achieves diffuse-specular separation with high temporal resolution and high dynamic range. Our future study includes the application to and evaluation on dynamic scenes.

Acknowledgement: This work was supported by JSPS KAKENHI Grant Numbers JP20H00612 and JP21K21319.

References

1. M. Born and E. Wolf, *Principles of Optics*, Pergamon Press, 1959.
2. G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, “Event-based vision: a survey”, *IEEE Trans. PAMI*, Vol.44, No.1, pp.154–180, 2022.
3. Y. Nisaka, R. Matsuoka, T. Amano, and T. Okabe, “Fast separation of specular, diffuse, and global components via polarized pattern projection”, *Frontiers of Computer Vision (IW-FCV2021)*, CCIS1405, pp.294–308, 2021.
4. S. Shafer, “Using color to separate reflection components”, *COLOR Research and Application*, Vol.10, No.4, pp.210–218, 1985.
5. Sony Semiconductor Solutions Corporation, “The principle of reflection removal utilizing polarization and features of polarization image sensor”, White Paper.
6. L. Wolff and T. Boult, “Constraining object features using a polarization reflectance model”, *IEEE Trans. PAMI*, Vol.13, No.6, pp.167–189, 1991.

A Set of Control Points Conditioned Pedestrian Trajectory Prediction*

Inhwan Bae and Hae-Gon Jeon

AI Graduate School, GIST, Gwangju, South Korea
 inhwbanbae@gm.gist.ac.kr and haegonj@gist.ac.kr

Abstract. Predicting the trajectories of pedestrians in crowded conditions is an important task for applications like autonomous navigation systems. Previous studies have tackled this problem using two strategies. They (1) infer all future steps recursively, or (2) predict the potential destinations of pedestrians at once and interpolate the intermediate steps to arrive there. However, these strategies often suffer from the accumulated errors of the recursive inference, or restrictive assumptions about social relations in the intermediate path. In this paper, we present a graph convolutional network-based trajectory prediction. Firstly, we propose a control point prediction that divides the future path into three sections and infers the intermediate destinations of pedestrians to reduce the accumulated error. To do this, we construct multi-relational weighted graphs to account for their physical and complex social relations. We then introduce a trajectory refinement step based on a spatio-temporal and multi-relational graph. In experiments, the proposed network achieves state-of-the-art performance on various real-world benchmarks.

Keywords: Pedestrian Trajectory Prediction · Multi-agent Forecasting

1 Introduction

Predicting the future trajectories of humans in crowds is an important task, especially for social robots, autonomous navigation, and surveillance systems. However, this task is challenging because such predictions require considering the desired destinations of each pedestrian, and the social norms of other moving agents, simultaneously.

Early works have attempted to capture social interactions using handcrafted Langevin equations, however, they often fail to model the complex social interactions that occur in crowded scenes. The recent development of convolutional neural networks (CNNs) and recurrent neural networks (RNNs) combines with social pooling [6] and social attention [15], and has improved understanding of the social interactions among pedestrians. However, these approaches still suffer from severe errors in final destination, because they accumulate the errors inherent to problems in the recursive predictions.

* This paper is a summary presentation of the paper which has been published in AAAI'23 by request of the IW-FCV2023 program committee to share the research results.

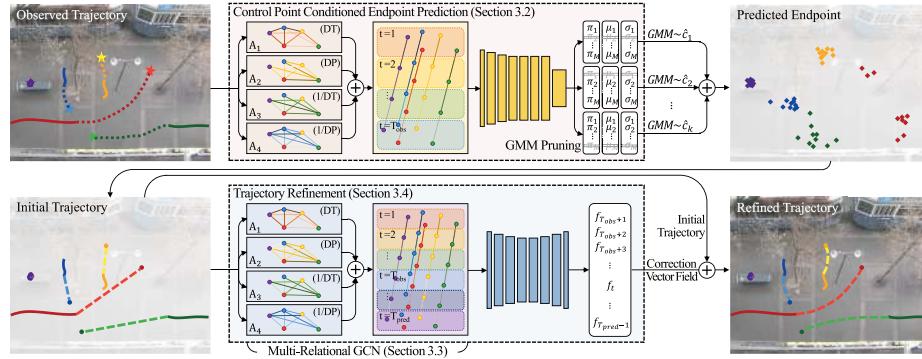


Fig. 1. An overview of our Graph-TERN. First, a control point prediction module takes the observed trajectory $S_{1:T_{obs}}$, and then constructs a multi-relational pedestrian graph. With a spatio-temporal aggregation using GCN and CNN, we can predict hypothetical endpoints \hat{E} through a summation of a set of randomly sampled control points \hat{C} . Second, a trajectory refinement modules takes the predicted endpoint \hat{E} and the observed path $S_{1:T_{obs}}$ to predict a correction vector field. A refined path $\hat{S}_{T_{obs}+1:T_{pred}}$ is obtained by summing the initial trajectory with the correction vector field.

To overcome this issue, Endpoint conditioned trajectory prediction is introduced [2, 4, 3], which infers the hypothetical arrival points and then interpolates their paths like vehicle navigation systems. Although these methods show promising performance improvements, issues remained. (1) Long-term predictions are performed without considering events occurring in the intermediate steps, and (2) social interactions are not regarded in endpoint prediction.

In this paper, we propose a Graph-based pedestrian Trajectory Estimation and Refinement Network (Graph-TERN) using a set of control points that combines the advantages of both step-by-step methods and endpoint conditioned methods. Firstly, we divide each pedestrian's future path into three sections and infer each stochastic goal, called control points. Secondly, we generate realistic future paths by introducing a refinement module that yields correction vector fields. Lastly, we design a multi-relational GCN operator to take account of complex social interactions in both the control point prediction and the trajectory refinement. By effectively incorporating the three modules, our model achieves state-of-the-art results using a variety of public pedestrian trajectory prediction benchmarks.

2 Proposed Method

Graph-TERN consists of two key components: (1) learning the probabilistic distribution for sampling endpoint candidates based on the control point; (2) yielding socially acceptable path prediction using a refinement module. Using an MRGCN framework, we develop a model that can successfully predict future trajectories while considering complex social relations. The overall framework is shown in Figure 1.

2.1 Preliminaries

Problem Definition. Pedestrian trajectory prediction attempts to determine future position sequences from observed position sequences for all agents in a scene. Suppose that there are N pedestrians in a scene at specific time t , and the corresponding positions of each pedestrian $n \in \{1, \dots, N\}$ can be represented as $p_t^n = (x_t^n, y_t^n)$. The trajectory sequence from the first time frame to the observed time T_{obs} can be denoted as $S_{1:T_{obs}}^n = \{p_t^n \in \mathbb{R}^2 | t \in \mathbb{N}, 1 \leq t \leq T_{obs}\}$. The consecutive prediction time frames are represented as $S_{T_{obs}+1:T_{pred}}$.

An additional goal of this work is to estimate a set of potential destinations for each pedestrian, called endpoint E^n . The endpoint \hat{E} can be predicted with the observed sequence $\hat{E}^n = \hat{p}_{T_{pred}}^n | S_{1:T_{obs}}$, then the future trajectories $\hat{S}_{T_{obs}+1:T_{pred}}$ can be inferred from the observed sequence $S_{1:T_{obs}}$ and the predicted endpoint \hat{E} .

Graph Convolutional Network. In general, the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represents a set of nodes \mathcal{V} and edges \mathcal{E} . In a pedestrian graph, the spatio-temporal graph \mathcal{G} consists of a pedestrian node $\mathcal{V} = \{p_t^n | n \in \mathbb{N}, 1 \leq n \leq N, 1 \leq t \leq T\}$ and a set of spatial and temporal edges $\mathcal{E} = \mathcal{E}_t \cup \mathcal{E}_n$. The spatial edge $\mathcal{E}_t = \{a_t^{i,j} | i, j \in \mathbb{N}, 1 \leq i, j \leq N\}$ represents a spatial relation for each pedestrian at a specific time t , and the temporal edge $\mathcal{E}_n = \{a_n^{i,j} | i, j \in \mathbb{N}, 1 \leq i, j \leq T\}$ represents the temporal relation of each pedestrian n within an observed sequence. Node features are aggregated with both spatial and temporal dimensions using GCNs and CNNs [16, 11]. With the node feature $H = \{h_t^n | n \in \mathbb{N}, 1 \leq n \leq N, 1 \leq t \leq T_{obs}\}$ and adjacency matrix $A = \{a_t^{i,j} | i, j \in \mathbb{N}, 1 \leq i, j \leq N, 1 \leq t \leq T_{obs}\}$, the GCN feature update rule is defined as $H' = \sigma(\hat{A}H\hat{W})$. Here, W and \hat{A} indicate the learnable weight matrix and the normalized form with the formula $\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$, respectively. We denote the self-loop added adjacency matrix as $\tilde{A} = A + I$ and diagonal node degree matrix as \tilde{D} from \tilde{A} .

2.2 Control Point Conditioned Endpoint Prediction

Graph Control Point Prediction. The key idea of our model is to use multiple control points when inferring potential endpoints. First, we define a set of control points C based on a displacement in one section, which is equally divided into future sequences $S_{T_{obs}+1:T_{pred}}$ in Figure 2, which is formulated as:

$$\begin{aligned} C^n &= \left\{ c_k^n = p_{T_{obs}+\tau \times k}^n - p_{T_{obs}+\tau \times (k-1)}^n \right\} \\ \text{for } \forall k &\in \{1, \dots, K\}, \quad \tau = \frac{T_{pred} - T_{obs}}{K}, \end{aligned} \quad (1)$$

Next, we present a control point prediction module. We update feature maps for the observed input sequence using a spatio-temporal and multi-relational GCN in Figure 1. We then use a multivariate Gaussian mixture model (GMM) to sample the 2D displacements of a set of control points in a Mixture Density Network (MDN). In contrast to previous works [10], we incorporate our control point prediction into a GCN framework, and in this way, our model provides a socially compliant endpoint that considers intermediate social interactions.

GMM Pruning. Public pedestrian trajectory datasets contain abnormal behaviors of agents. Since statistical models need to allocate a portion of its capacity to ensure the abnormal cases, it is left with relatively less capacity for generating realistic paths. These abnormal cases are considered as out-of-distribution samples drawn far away from the training distribution statistically, and leads to performance drops. The truncation trick is widely used to restrict the distribution of samples, it limits the distributions of reasonable control point candidates can be assigned to effectively feasible areas.

Through the GMM pruning, potential control point candidates can be assigned to effectively feasible areas.

Endpoint Sampling. The final endpoint \hat{e} is determined by summing the set of control points \hat{C} , which is sampled through the probabilistic process $\hat{e}^n = p_{T_{obs}}^n + \sum_{k=1}^K \hat{c}_k^n$. Following the previous study [6], we sample the $L = 20$ endpoints $\hat{E}^n = \{\hat{e}_l^n | l, n \in \mathbb{N}, 1 \leq l \leq L, 1 \leq n \leq N\}$ which represent multi-modality, and feed them into a trajectory refinement module in Section 2.4.

2.3 Multi-Relational Pedestrian Graph

While the GCN has the advantage of imposing physical constraints, conventional GCN-based models use a single relation edge, which makes capturing social relations limited. Due to this reason, the GCN-based approaches have gained less interest than those of the GAT-based approaches whose multi-head attention allows it. In this work, we fully take advantage of the GCN framework by overcoming the limitation through a multi-relational-based kernel function to produce each relational adjacent matrix as:

$$H' = \sigma \left[\text{CNN} \left(\sum_{r=1}^R \hat{A}_r H^T W_r \right)^T + H \right], \quad (3)$$

where R is the number of elements in a set of relations $\mathcal{R} = \{Distance, Displacement, 1/Distance, 1/Displacement\}$, \hat{A}_r is the normalized term of $A_r = \{a^{i,j,r} \in \mathbb{R} | i, j, r \in \mathbb{N}, 1 \leq i, j \leq N, 1 \leq r \leq R\}$, and W_r is a learnable weight matrix. Through this, Our multi-relational graph deals with obstacles, stop and go motion, and group following in very challenging situations.

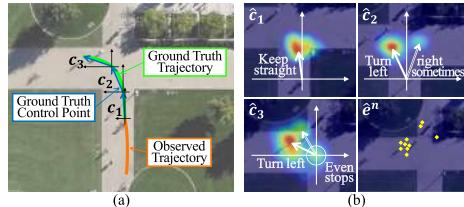


Fig. 2. A set of control points prediction. (a) When a person turns left at the crossroad, three control points are defined based on the displacement. (b) Examples of predicted distributions for the control points and endpoint sampling.

To address this issue, we devise a GMM pruning which cuts off a lower half of the bi-variate Gaussian based on predicted mixing coefficients as follows:

$$M^* = \left\lfloor \frac{M}{2} \right\rfloor, \quad z^* = \underset{z' \subset z, |z'|=M^*}{\text{argmax}} \sum_{z'^\pi \subset z'} z'^\pi. \quad (2)$$

While the GCN has the advantage of imposing physical constraints, conventional GCN-based models use a single relation edge, which makes capturing social relations limited. Due to this reason, the GCN-based approaches have gained less interest than those of the GAT-based approaches whose multi-head attention allows it. In this work, we fully take advantage of the GCN framework by overcoming the limitation through a multi-relational-based kernel function to produce each relational adjacent matrix as:

$$H' = \sigma \left[\text{CNN} \left(\sum_{r=1}^R \hat{A}_r H^T W_r \right)^T + H \right], \quad (3)$$

where R is the number of elements in a set of relations $\mathcal{R} = \{Distance, Displacement, 1/Distance, 1/Displacement\}$, \hat{A}_r is the normalized term of $A_r = \{a^{i,j,r} \in \mathbb{R} | i, j, r \in \mathbb{N}, 1 \leq i, j \leq N, 1 \leq r \leq R\}$, and W_r is a learnable weight matrix. Through this, Our multi-relational graph deals with obstacles, stop and go motion, and group following in very challenging situations.

2.4 Trajectory Refinement

Guided Endpoint Sampling. To jointly train the control point prediction module and trajectory refinement module, we need to decouple them. We define a rule to limit the predicted endpoints at training time. We divide the predicted endpoints into a positive set \mathcal{Y}^+ and a negative set \mathcal{Y}^- using the adaptive threshold parameter Γ .

$$\begin{aligned}\mathcal{Y}^{n+} &= \{\hat{e}_l^n \mid \|\hat{e}_l^n - e^n\| \leq \Gamma\} \\ \mathcal{Y}^{n-} &= \{\hat{e}_l^n \mid \|\hat{e}_l^n - e^n\| > \Gamma\} \\ \text{for } \forall l \in \{1, \dots, L\}, \Gamma &= \frac{\|p_{T_{obs}}^n - p_1^n\|}{T_{obs} \times \gamma},\end{aligned}\quad (4)$$

To ensure that the model converges stably, we only back-propagate gradients for the positive sets using a valid mask Ψ . Here, the element of the valid mask $[\Psi]_{n,l}$ is defined as $[\Psi]_{n,l} = \psi_l^n = \{1 \text{ if } \hat{e}_l^n \in \mathcal{Y}^{n+}, \text{ otherwise } 0\}$. During the initial training phase, the number of positive sets might be extremely small because the candidates are broadly spread. To address this issue, we additionally sample the guided endpoints within the range Γ of the ground-truth endpoint.

Initial Trajectory Prediction. The purpose of establishing an initial trajectory based on the guided endpoints is to make our trajectory refinement module tractable. The simplest way to do this is to connect these control points through linear interpolation. However, we observe that the use of a set of control points in the initial trajectory prediction acts as a hard constraint, even though it is helpful to infer accurate destinations of pedestrians. Therefore, we first generate a single initial trajectory $\tilde{S}_{T_{obs}+1:T_{pred}}$ for one endpoint as below:

$$\begin{aligned}\tilde{p}_{t,l}^n &= p_{T_{obs}}^n + \frac{\hat{e}_l^n - p_{T_{obs}}^n}{T_{pred} - T_{obs}} \times (t - T_{obs}) \\ \tilde{S}_{T_{obs}+1:T_{pred},l} &= \{\tilde{p}_{t,l}^n\} \\ \text{for } \forall t &\in \{T_{obs} + 1, \dots, T_{pred}\}, \\ \forall l &\in \{1, \dots, L\}, \forall n \in \{1, \dots, N\}.\end{aligned}\quad (5)$$

Graph Trajectory Refinement. As a next step, we present a novel refinement module to yield an accurate trajectory from the observed trajectory $S_{1:T_{obs}}$ and the initial trajectory $\tilde{S}_{T_{obs}+1:T_{pred}}$. By concatenating the two trajectories along with the time axis, we can aggregate the social interactions for all time frames using the MRGCN. For social interactions, the correction vector field is computed and the final refined trajectory can be obtained as below:

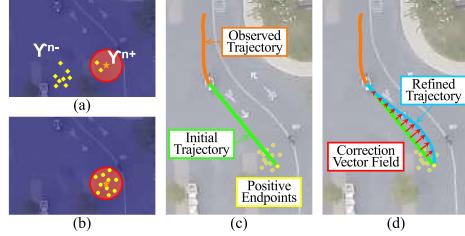


Fig. 3. An example of trajectory refinement. (a) Endpoint candidates are classified into the positive and negative set. (b) Guided endpoints are randomly sampled. (c) The initial trajectory is predicted by linearly interpolating between the observed trajectory and the endpoints. (d) After that, the trajectory is refined by adding a correction vector field in the initial trajectory.

Table 1. Comparison of our Graph-TERN with other state-of-the-art methods on ETH/UCY dataset (ADE/FDE, Unit: meter). The mark † means that the common data-loader in [6] are used. **Bold:** Best, Underline: Second best.

Model	Year	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
Linear Regression	-	1.33 / 2.94	0.39 / 0.72	0.82 / 1.59	0.62 / 1.21	0.77 / 1.48	0.79 / 1.59
Social-LSTM [1]	2016	1.09 / 2.35	0.79 / 1.76	0.67 / 1.40	0.47 / 1.00	0.56 / 1.17	0.72 / 1.54
Social-GAN [6]	2018	0.87 / 1.62	0.67 / 1.37	0.76 / 1.52	0.35 / 0.68	0.42 / 0.84	0.61 / 1.21
STGAT [7]	2019	0.65 / 1.12	0.35 / 0.66	0.52 / 1.10	0.34 / 0.69	0.29 / 0.60	0.43 / 0.83
Social-STGCNN [11]	2020	0.64 / 1.11	0.49 / 0.85	0.44 / 0.79	0.34 / 0.53	0.30 / 0.48	0.44 / 0.75
PECNet [†] [10]	2020	0.65 / 1.13	0.22 / 0.38	0.35 / 0.57	0.25 / 0.45	<u>0.18</u> / <u>0.31</u>	0.33 / 0.57
Trajectron++ [†] [15]	2020	0.61 / 1.03	0.20 / 0.28	<u>0.30</u> / <u>0.55</u>	0.24 / <u>0.41</u>	<u>0.18</u> / 0.32	<u>0.31</u> / <u>0.52</u>
Causal-STGAT [5]	2021	0.60 / 0.98	0.30 / 0.54	0.52 / 1.10	0.32 / 0.64	0.28 / 0.58	0.40 / 0.77
TPNMS [9]	2021	<u>0.52</u> / <u>0.89</u>	0.22 / 0.39	0.55 / 1.13	0.35 / 0.70	0.27 / 0.56	0.38 / 0.73
Causal-STGCNN [5]	2021	0.64 / 1.00	0.38 / 0.45	0.49 / 0.81	0.34 / 0.53	0.32 / 0.49	0.43 / 0.66
SGCN [16]	2021	0.63 / 1.03	0.32 / 0.55	0.37 / 0.70	0.29 / 0.53	0.25 / 0.45	0.37 / 0.65
LBEBM [†] [13]	2021	0.62 / 1.16	0.19 / 0.35	0.37 / 0.67	<u>0.23</u> / 0.43	0.19 / 0.36	0.32 / 0.59
DMRGCN [2]	2021	0.60 / 1.09	0.21 / 0.30	0.35 / 0.63	0.29 / 0.47	0.25 / 0.41	0.34 / 0.58
STT [12]	2022	0.54 / 1.10	0.24 / 0.46	0.57 / 1.15	0.45 / 0.94	0.36 / 0.77	0.43 / 0.88
Graph-TERN	-	0.42 / 0.58	0.14 / 0.23	0.26 / 0.45	0.21 / 0.37	0.17 / 0.29	0.24 / 0.38

$$\begin{aligned} \hat{p}_{t,l}^n &= \tilde{p}_{t,l}^n + f_{t,l}^n \\ \hat{S}_{T_{obs}+1:T_{pred},l} &= \{\hat{p}_{t,l}^n\} \cup \hat{E} \\ \text{for } \forall t &\in \{T_{obs} + 1, \dots, T_{pred} - 1\}, \\ \forall l &\in \{1, \dots, L\}, \quad \forall n \in \{1, \dots, N\}. \end{aligned} \tag{6}$$

Unlike existing methods which use social interactions based only on observations, our refinement module allows a more complex social relation because our MRGCN captures such relations even with the interpolated points and the endpoint.

2.5 Loss Functions

We maximize an expectation to train the control point prediction module. We sum the probabilistic density functions of all the predicted control point distributions and pedestrians. The loss function Θ_w is defined as:

$$\Theta_w = \sum_{n=1}^N \sum_{k=1}^K -\log \left[\sum_{m=1}^M \hat{\pi}_{m,k}^n \frac{\exp\left(-\frac{(c_k^n - \hat{\mu}_{m,k}^n)^2}{2(\hat{\sigma}_{m,k}^n)^2}\right)}{\sqrt{2\pi} \hat{\sigma}_{m,k}^n} \right] \tag{7}$$

In addition, we minimize the trajectory refinement loss Θ_r . The loss is based on a mean square error (MSE) of an average displacement between a refined trajectory and a ground truth trajectory, and is formulated as below:

$$\Theta_r = \sum_{n=1}^N \sum_{l=1}^{2L} \sum_{t=T_{obs}+1}^{T_{pred}-1} \psi_l^n \left[(x_{t,l}^n - \hat{x}_{t,l}^n)^2 + (y_{t,l}^n - \hat{y}_{t,l}^n)^2 \right] \tag{8}$$

Finally, the loss function Θ of the entire network is defined as a weighted sum of the control point loss and the refinement loss: $\Theta = \Theta_w + \lambda \Theta_r$, where λ is a scale factor between the control point prediction error and the trajectory refinement error, and is empirically set to $\lambda = 1$

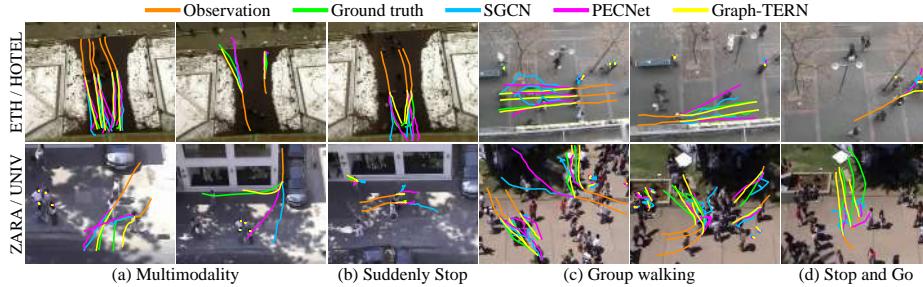


Fig. 4. Visualization of prediction results. We compare Graph-TERN, SGCN and PECNet, whose results are reproduced with the pre-trained network. To aid visualization, trajectories with the best ADE on 20 samples are reported.

3 Experiments

We evaluate our Graph-TERN using two real-world public datasets: ETH [14] and UCY [8]. In ETH and UCY datasets, there are five different scenes (ETH, HOTEL, UNIV, ZARA1, and ZARA2) with various complex social interactions such as collision avoidance, group movement, and people stopping. We follow the standard evaluation strategy.

We compared our Graph-TERN with other state-of-the-art works with two performance metrics: average displacement error (ADE) and final displacement error (FDE) in meter scale on both ETH and UCY datasets Table 1. The results indicate Graph-TERN provides the best performance on nearly all of the measures and datasets. Of particular note, in ETH and HOTEL set, there are many people who abruptly stop walking. Graph-TERN handles this case well by learning the probability of people stopping in the control point prediction as in Figure 2(b). For UNIV set with very crowded scenes, our multi-relational GCN synergizes well with the initial trajectory prediction and the refinement module by considering complex social relations. This infers accurate multimodal and stop-and-go predictions based on the well-estimated group movements in Figure 4.

4 Conclusion

We present a set of control points prediction and a refinement network for pedestrian trajectory prediction. The control point prediction allows the accurate computation of the final destinations of pedestrians and the refinement provides a socially acceptable trajectory. By incorporating a MRGCN, our model achieves SOTA results by modeling complex social interactions in real-world scenes¹.

¹ [Remarks] This paper is a summary presentation of the paper which has been published in AAAI'23 by request of the IW-FCV2023 program committee to share the research results.

References

1. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social lstm: Human trajectory prediction in crowded spaces. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
2. Bae, I., Jeon, H.G.: Disentangled multi-relational graph convolutional network for pedestrian trajectory prediction. Proceedings of the AAAI Conference on Artificial Intelligence (AAAI) (2021)
3. Bae, I., Park, J.H., Jeon, H.G.: Learning pedestrian group representations for multi-modal trajectory prediction. In: Proceedings of European Conference on Computer Vision (ECCV) (2022)
4. Bae, I., Park, J.H., Jeon, H.G.: Non-probability sampling network for stochastic human trajectory prediction. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
5. Chen, G., Li, J., Lu, J., Zhou, J.: Human trajectory prediction via counterfactual analysis. In: Proceedings of International Conference on Computer Vision (ICCV) (2021)
6. Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A.: Social gan: Socially acceptable trajectories with generative adversarial networks. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
7. Huang, Y., Bi, H., Li, Z., Mao, T., Wang, Z.: Stgat: Modeling spatial-temporal interactions for human trajectory prediction. In: Proceedings of International Conference on Computer Vision (ICCV) (2019)
8. Lerner, A., Chrysanthou, Y., Lischinski, D.: Crowds by example. Computer Graphics Forum **26**(3), 655–664 (2007)
9. Liang, R., Li, Y., Li, X., Tang, Y., Zhou, J., Zou, W.: Temporal pyramid network for pedestrian trajectory prediction with multi-supervision. Proceedings of the AAAI Conference on Artificial Intelligence (AAAI) (2021)
10. Mangalam, K., Girase, H., Agarwal, S., Lee, K.H., Adeli, E., Malik, J., Gaidon, A.: It is not the journey but the destination: Endpoint conditioned trajectory prediction. In: Proceedings of European Conference on Computer Vision (ECCV) (2020)
11. Mohamed, A., Qian, K., Elhoseiny, M., Claudel, C.: Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
12. Monti, A., Porrello, A., Calderara, S., Coscia, P., Ballan, L., Cucchiara, R.: How many observations are enough? knowledge distillation for trajectory forecasting. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
13. Pang, B., Zhao, T., Xie, X., Wu, Y.N.: Trajectory prediction with latent belief energy-based model. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
14. Pellegrini, S., Ess, A., Schindler, K., Van Gool, L.: You'll never walk alone: Modeling social behavior for multi-target tracking. In: Proceedings of International Conference on Computer Vision (ICCV) (2009)
15. Salzmann, T., Ivanovic, B., Chakravarty, P., Pavone, M.: Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In: Proceedings of European Conference on Computer Vision (ECCV) (2020)
16. Shi, L., Wang, L., Long, C., Zhou, S., Zhou, M., Niu, Z., Hua, G.: Sgcn: Sparse graph convolution network for pedestrian trajectory prediction. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021)

Format-Compatible 3D Metahuman Modeling from a Single Image

Sojin Yun¹, Soyoung Yoon¹, and In Kyu Park^{1[0000-0003-4774-7841]}

Department of Information and Communication Engineering, Inha University
Incheon 22212, Korea
sj0524sj@gmail.com, thdud679@gmail.com, pik@inha.ac.kr

Abstract. In this paper, we propose a method for generating 3D metahumans reflecting the body shape and clothes texture estimated from a single general image. The body and clothes mesh are automatically transformed in the blender, using body shape estimates obtained through SMPL-X. The obtained clothes' UV Map and mesh reflecting the body shape are automatically called to the Unreal Engine to create a metahuman. The generated metahuman can be seen as an animation implemented on Blueprint of the Unreal engine.

Keywords: Format-compatible · 3D metahuman · Single image.

1 Introduction

Metaverse is a combination of Meta and Universe, and means a virtual world linked to reality. Recently, more and more people are communicating in digital spaces since COVID-19, centering on the generation familiar with the digital environment. In the current existing metaverse, it is difficult to create the same metahuman that is 3D Virtual Humans on metaverse, because you choose the limited created type of characters and clothes. Therefore, this work proposes a 3D metahuman generation method from one's own single image.

Recent studies [13, 15] have modeled clothes with the body volumetrically using voxels or point cloud, which is not compatible with game engines and authoring tools. In this paper, metahumans can be imported to Unreal engines, the commonly used game engines. Our previous work [7] estimates the body type with one image and models the clothes to generate metahumans, but there are limitations in research. The gender and clothes categories of the model are limited, so various types of characters can't be reflected. In addition, UV Map generation is not accurate for images that are not frontal and static. Also, both the body shape deformation code in the blender and the metahuman generation and animation in the unreal engine are manually performed. This study aims to make up for these limitations.

The main contributions can be summarized as follows.

- A proof of concept for format-compatible and animatable 3D metahuman generation from a single image
- Automatic shape and texture modeling for widely-used commercial game engines, *e.g.* Unreal engines

2 Proposed Method

Fig. 1 shows the proposed pipeline that generates 3D metahumans through body shape and clothes texture extracted from a single image.

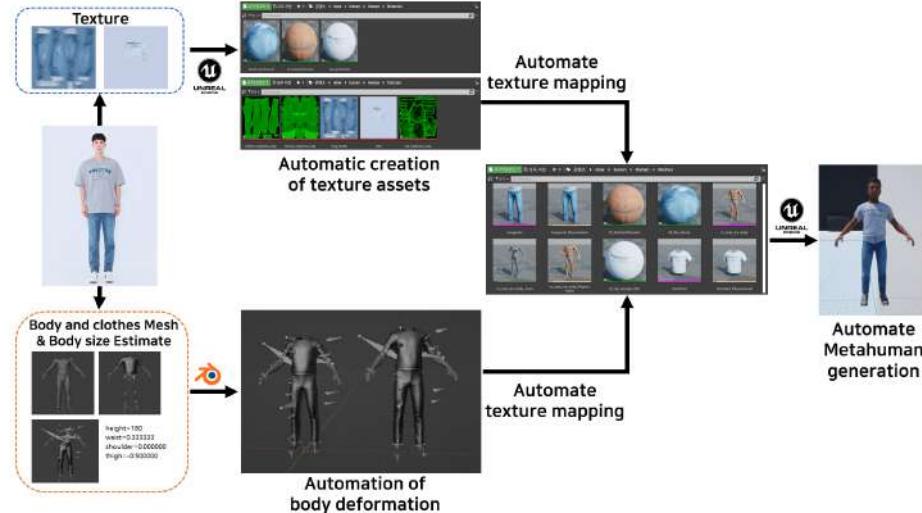


Fig. 1: 3D metahuman generation pipeline with body and clothing textures estimated from a single image. Test image is from [1].

2.1 3D Face Reconstruction

We follow the general pipeline that was proposed in our previous work [10]. Basic face rendering features utilize 3DMM [5] and CNN. Landmarks are detected on faces, and 3DMM coefficients are estimated using them. Alignment between 2D and 3D landmarks is performed via RBF to restore a rough 3D face. With CNN-based fine grained depth information estimation, wrinkles and other elements are included in the 3D face. Face texture is also generated using landmarks and CNN by removing hair and ears and filling the background of the image with skin color and texture. It generates a low resolution diffuse reflection texture from a single basic face image. [10]

Additionally, cavity map, normal map, and roughness map are used for natural and realistic face rendering. When using only one sheet of reflection texture, there is a problem that is not realistic when texture mapping to the face mesh model. Therefore, additional cavity map and normal map representing light reflection and skin irregularities, and roughness maps representing the resulting skin roughness, are used. Through the above additional information, realistic skin expression that the face mesh model cannot express is possible. [8, 9]

2.2 3D Body Reconstruction

We follow the general pipeline that was proposed in our previous work [14]. Extracting body type information from a single image utilizes SMPL [11] and SMPL-X [12]. Detailed information of the person is estimated using SMPL-X from the input image. It is converted to SMPL for measuring body type parameters and posture parameters of body models. In this conversion, the SMPL model modified to a straight posture is obtained by changing only the value of a specific posture parameter without converting the body shape parameter. Using the SMPL model, which modified the posture, the dimensions of the height, shoulder, waist, and thigh set as the main parts of the body are measured. The basic body model provided by the metahuman creator is converted through the measured size. [9, 14]

We use body size estimates to deform the mesh of the body and clothes. The body and clothes files provided by the Metahuman Creator consist of mesh and skeleton. For each skeleton, an affected set of vertices is defined. As the skeleton rotates, the affected set of vertices moves together. Therefore, all vertex sets defined in the skeleton set are calculated and selected. Thereafter, the deformation is performed on the selected set of vertices. The Python code of [7] deforms the mesh in the blender. Adjusting the height uses a function that scales in the z-axis direction relative to the origin. For body parts other than height, the operation of reducing or enlarging the surface based on the normal vector is performed.

By adding from [7], the import and export process of the mesh file is automatically performed, and the mesh deformation is also automatically performed. The biggest difference from the existing method is that it transforms the necessary mesh files one by one. Using a method of importing and deforming all meshes at once, it is impossible to map textures separately from the Unreal engine because the export function stores all the meshes currently in place. Therefore, the mesh for the body, top, and bottom is all recalled separately, and each is deformed and exported. In between this process, each modified and exported mesh is deleted so that it does not affect other meshes.

2.3 3D Cloth Reconstruction

The basic costume clothes mesh uses a Metahuman Creator. Various clothes that are not provided, such as skirts, dresses, and outerwear, are produced using 3D tools such as Marvelous Designer and Blender. The basic metahuman is set as an avatar on Marvelous Designer and clothes are fitted to the body. Metahuman's skeleton is planted in this, combined with the costume, and weight paint is performed on each bone to enable costume animation.

From the input image, the clothes segmentation is carried out into 13 costume categories provided by Deepfashion2 [6]. From this result, keypoint tailored to the characteristics of each costume is detected. With the coordinate information of the keypoint, perspective transformation mapping is performed with coordinates to be mapped to the UV Map. The texture of the invisible side is reflected in the UV Map by distinguishing whether it is pattern or logo-centered. [16]

2.4 3D Metahuman Generation

First, we perform the task of loading files of various types of extensions, such as obj, fbx, jpg, and png. Textures have extensions such as jpg, png, and bmp, and are stored in the ‘Textures’ folder upon loading.

The imported texture file is used to create the Synthesized Asset. Synthesized asset is a type of material instance that allows for more sophisticated texture mapping by setting normal, roughness, and cavity. The created synthesized asset is stored in the “Materials” folder, and the mesh of the body, top, and bottom is stored in the “Meshes” folder. Each mesh is mapped in texture with a set synthesized set. The final metahuman is implemented in the viewport of the Unreal engine. The final metahuman is implemented in the viewport of the Unreal Engine by reflecting the position and rotation. At this time, meshes such as shoes, faces, and hair are also called in to create the final metahuman [4].

3 Experimental Results

The methods proposed by the paper are automatic conversion of body and clothes mesh and automatic generation of metahuman within an unreal engine. Body and clothes mesh is converted separately and then stored to facilitate texture mapping when meta-humans were generated. Mesh to which the texture is mapped is created in Viewport of the Unreal Engine where you can check the appearance and animation of the asset, to confirmation of the final appearance.

Fig. 2 shows the experimental results of the method proposed by the paper. Unlike previous studies [7], the experiment is conducted with a general image. UV Maps without mixing background colors is created even though they is not static, and female models and clothes such as skirts is added to create more diverse metahumans. Also, unlike the manual experiments on the blender and unreal engines, metahumans are generated through automation codes. When using the automation process, body shape conversion takes about 4 seconds and metahuman generation takes about 30 seconds. Using the existing code [7], body shape conversion takes about 10 seconds each for the body, top, and bottom, taking a total of 30 seconds. Metahuman generation takes about 5 minutes if you are familiar with the Unreal engine because all the work is done manually. Using the code proposed by the paper, time can be greatly reduced compared to previous studies and the results can be easily checked.

Animation can be implemented because it uses the same skeleton as the existing metahuman. However, the animation is done manually on Blueprint. Blueprint that implements the animation of the basic metahuman is duplicated to add the necessary body and costume assets to the component. The same animation is applied by setting it to the child class where the legging work has been done. Application of animation to the generated metahuman can be found in Fig. 3.

The metahuman’s face and hands generated tend to be larger than the entire body. Through SMPL and SMPL-X, the body and costume mesh is converted

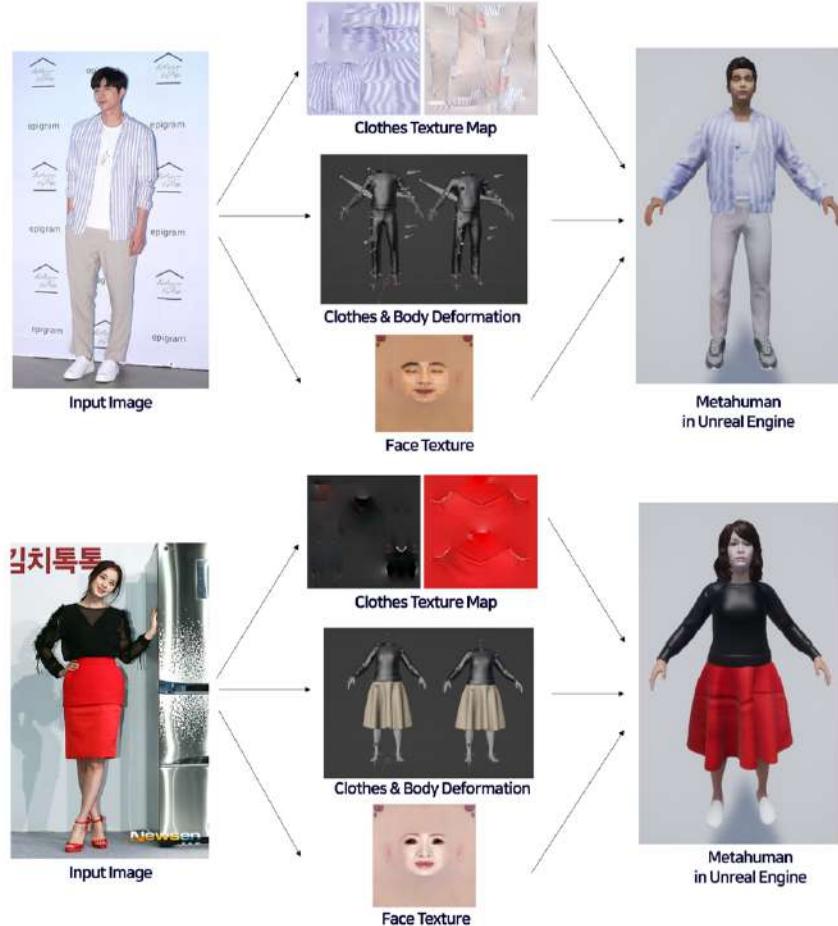


Fig. 2: Experimental results on typical images. Extract costume UV Map and body type estimates from general images. Body estimates are used for body shape deformation of body mesh. The final metahuman is created by mapping the clothes UV Map to the deformed body mesh. In this process, body shape transformation and metahuman generation are implemented by automation code. Test images are from [2, 3]

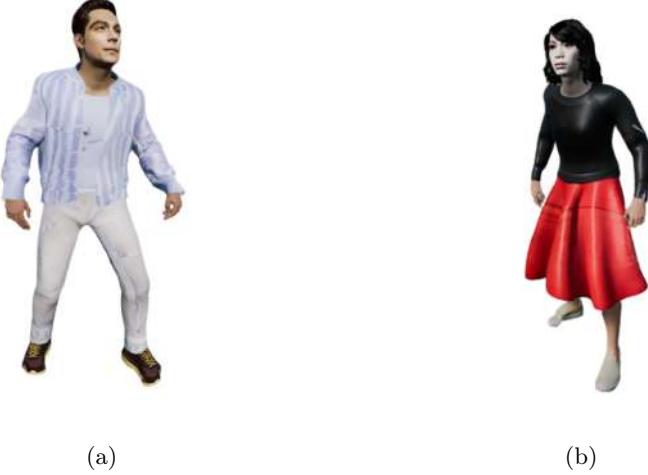


Fig. 3: Application of animation to the generated metahuman. It uses the same skeleton as the basic metahuman, so that basic metahuman animations such as running and jumping can be executed through the direction keys. (a) Gong Yoo. (b) Kim Tae Hee

from the body dimensions of the input image, but the accuracy is poor because the ratio of the basic metahuman is not correct. In addition, it tends to be less accurate because the face texture is applied to the face mesh of the existing metahuman, not the face mesh used in [10], for the animation of the face.

4 Conclusion

In this paper, we proposed a method for generating 3D metahumans through body shape and clothes textures estimated from a single image. The resultant model could be imported onto a popular game engine like Unreal and authoring tool like Maya for metaverse applications. The model was observed to be animatable on those platforms.

Acknowledgements This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2022R1A4A1033549). This work was partly supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-00981, Foreground and background matching 3D object streaming technology development and No.RS-2022-00155915, Artificial Intelligence Convergence Innovation Human Resources Development (Inha University)).

References

1. <https://www.musinsa.com/app/goods/2546982>, Accessed on December 12, 2022
2. https://www.newsen.com/news_view.php?uid=201310311519262910, Accessed on December 12, 2022
3. <http://osen.mt.co.kr/article/G1110913218>, Accessed on December 12, 2022
4. Unreal Python API Documentation <https://docs.unrealengine.com/4.27/en-US/PythonAPI/>, accessed August 24, 2022
5. Blanz, V., Romdhani, S., Vetter, T.: Face identification across different poses and illuminations with a 3D morphable model. In: Proc. of IEEE International Conference on Automatic Face and Gesture Recognition. p. 202–207 (August 2002)
6. Ge, Y., Zhang, R. and Wang, X., Tang, X., Luo, P.: Deepfashion2: a versatile benchmark for detection, pose estimation. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR). p. 5337–5345 (2019)
7. Kim, H.W., Kim, D.E., Kim, Y., Park, I.K.: 3D clothes modeling of virtual human for metaverse. Journal of Broadcast Engineering **27**(5), 638–653 (December 2022)
8. Kim, Y.: A study on multimodal facial texture generation for 3d human modeling. Master's thesis, Inha University (August 2022)
9. Kim, Y., Park, I.K.: Controllable facial micro-element synthesis using segmentation maps. In: Proc. IEEE International Conference on Automatic Face and Gesture Recognition (January 2022)
10. Lee, J., Lumentut, J.S., Park, I.K.: Holistic 3D face and head reconstruction with geometric details from a single image. Multimedia Tools and Applications **81**(26), 38217–38233 (November 2022)
11. Loper, M., N. Mahmood, J.R., Pons-Moll, G., Bla, M.J.: SMPL: A skinned multi-person linear model. ACM Trans. on Graphics **34**(6), 248:1–248:16 (Nov 2015)
12. Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A.A.A., Tzionas, D., Black, M.J.: Expressive body capture: 3D hands, face, and body from a single image. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR). p. 10975–10985 (2019)
13. Saito, S., Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., Li, H.: PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization. In: IEEE International Conference on Computer Vision(ICCV). pp. 2304–2314 (2019)
14. Santoso, J., Williem, , Park, I.K.: Holistic 3D body reconstruction from a blurred single image. IEEE Access **10**, 115399–115410 (Nov 2022)
15. Wu, B., Wang, Z., Wang, H.: A GPU-based multilevel additive schwarz preconditioner for cloth and deformable body simulation. ACM Trans. on Graphics **41**(4), 63:1–63:14 (July 2022)
16. Yoon, S., Yun, S., Park, I.K.: Game engine compatible 3D clothes modeling from a single image. In: Proc. International Workshop on Frontiers of Computer Vision (February 2023)

YOLO5PKLot: A Parking Lot Detection Network Based on Improved YOLOv5 for Smart Parking Management System

Duy-Linh Nguyen^[0000-0001-6184-4133], Xuan-Thuy Vo^[0000-0002-7411-0697],
 Adri Priadana^[0000-0002-1553-7631], and Kang-Hyun Jo^[0000-0002-4937-7082]

Department of Electrical, Electronic and Computer Engineering, University of Ulsan,
 Ulsan 44610, South Korea
 ndlinh301@mail.ulsan.ac.kr, xthuy@islab.ulsan.ac.kr,
 priadana@mail.ulsan.ac.kr, acejo@ulsan.ac.kr

Abstract. In recent years, the YOLOv5 network architecture has demonstrated excellence in real-time object detection. For the purpose of applying in the smart parking management system, this paper proposes a network based on the improved YOLOv5, named YOLO5PKLot. This network focus on redesigning the backbone network with a combination of the lightweight Ghost Bottleneck and Spatial Pyramid Pooling architectures. In addition, this work also resizes the anchors and adds a detection head to optimize parking detection. The proposed network is trained and evaluated on the Parking Lot dataset. As a result, YOLO5PKLot achieved 99.6% mAP on the valuation set with only fewer network parameters and computational complexity than others.

Keywords: Convolutional neural network (CNN) · Ghost Bottleneck · Smart parking management system · Parking lot detection · Parking lot dataset · YOLOv5.

1 Introduction

Currently, there are about 1.45 billion cars in the world and it is increasing every year. The report in [18] predicts that in 2023 the number of vehicles sold is about 71 million units. The rapid increase both in the number and type of vehicles has led to the expansion of parking lots in supermarkets, shopping malls, city offices, etc. Automated operations to manage and distribute parking spaces are essential. For a long time ago, researchers and engineers have been designing automated parking management systems. These techniques are mainly based on various types of sensors to determine the status of parking spaces such as ultrasonic [19], infrared [5], geomagnetic [21], and wireless [20]. This type of parking usually requires the installation and maintenance of each sensor per parking space. Therefore, these methods increase the cost quite a lot when deployed in large-scale parking lots. In general, the sensing methods achieve high prediction but have a large cost. From the above analysis along with the development of

the computer vision field, this paper proposes a vision-based parking lot detection network. This work improves the famous object detection network YOLOv5 by focusing on redesigning the backbone network and adding a new detection head to increase the object detection ability. This network uses lightweight architectures in Ghost Bottleneck (Ghost) to greatly reduce network parameters and computational complexity, serving real-time applications on low-computing devices. The paper provides several main contributions as follows:

- 1 - A modified Ghost Bottleneck block is proposed to apply to the backbone of YOLOv5.
- 2 - Redesigns YOLOv5 backbone network with a combination of lightweight Ghost Bottleneck and Spatial Pyramid Pooling (SPP) architectures.
- 3 - Adds a detection head with new anchor sets to improve the prediction task.

The remainder of the paper is distributed as follows: Section 2 introduces the techniques related to parking lot detection. Section 3 details the proposed techniques. Section 4 presents and analyzes the experimental results. Section 5 concludes the issue and future development orientation.

2 Related work

2.1 Traditional-based method

These methods are implemented through two main steps, feature extraction, and parking classification. The feature extraction process generates one or more feature vectors using traditional techniques. Specifically, [1, 8] used Local Phase Quantization (LPQ) and Local Binary Patterns (LBP) as feature extractors and classifiers using Support Vector Machine (SVM). Later, the authors developed new methods based on the change of camera and parking areas [3, 2]. [4] applies Quaternionic Local Ranking Binary Pattern (QLRBP) for feature extraction, Support Vector Machine (SVM), and k-nearest neighbors (k-NN) are used for classification. In [10] the LBP and the Histogram of Oriented Gradients (HOG) were used as feature extractors for the SVM classifier. The advantage of the above methods is that it is easy to implement, but the accuracy is not high.

2.2 Machine learning-based method

With the remarkable development of object detection networks in the computer vision field, smart parking management systems are also developed based on popular networks. [9] refines the YOLOv3 architecture by adding residual blocks to the original network to improve feature extraction for parking classification. [6] designs a lightweight version of YOLOv3 with MobileNetV2 architecture to improve parking classification. A faster R-CNN two-stage detection network was applied in [15] with different camera angles and parking changes. [17, 16] exploit the power of the Mask R-CNN network to extract individual cars and then classify parking conditions. Generative Adversarial Networks (GANs) are

also used to directly detect occupancy and vacancies using drone imagery [14]. The advantage of machine learning methods is high detection and classification accuracy but requires networks to reach a certain depth and complexity to ensure operation in real parking conditions.

3 Methodology

Fig. 1 details the proposed parking lot detection network. This network is refined based on the original YOLOv5 architecture [13] with three main modules: backbone, neck, and head.

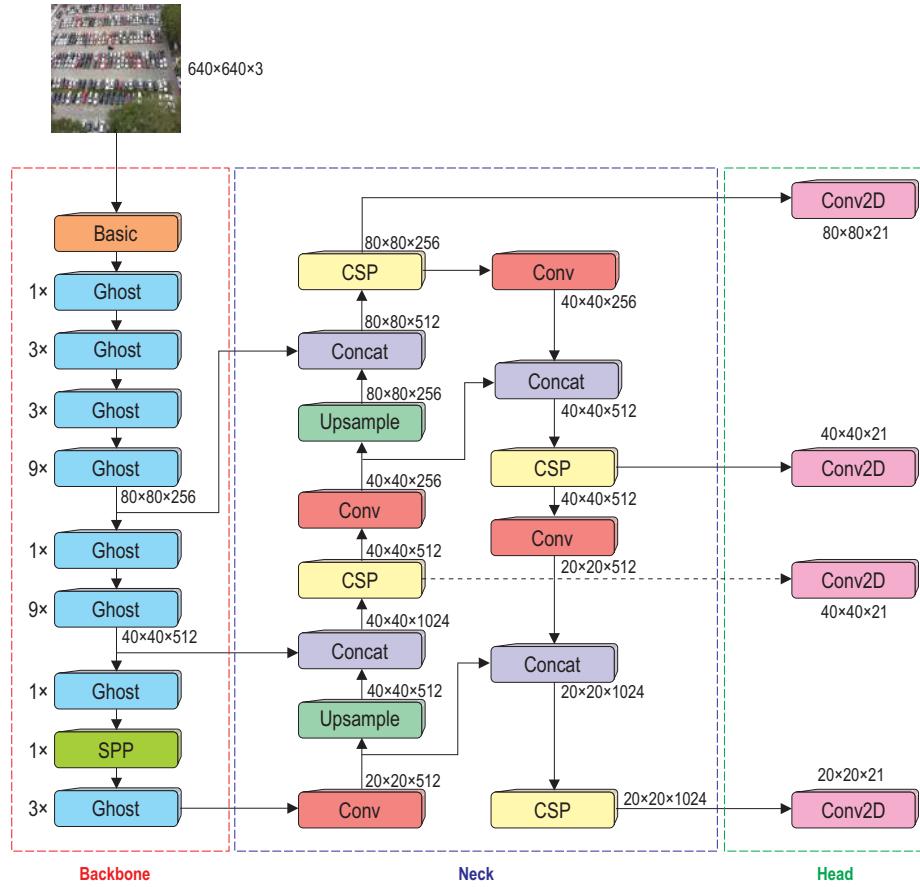


Fig. 1. The proposed parking lot detection network (YOLO5PKLot).

3.1 Proposed network architecture

The backbone module follows the design of the backbone in the YOLOv5 architecture, but this work changes a few essential modules. The techniques are applied to reduce a lot of the network parameters and computational complexity but still ensure good feature extraction. Specifically, the Focus module in YOLOv5 is replaced by the Basic module shown in Fig. 2. This module is designed with two main branches. One branch consists of 3 Conv blocks contiguously, the another branch is a Max pooling layer that is attached after the first Conv block in the first branch. The output feature maps from these two branches are concatenated and followed by another Conv block.

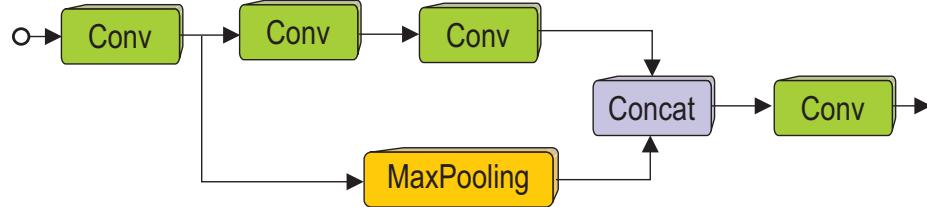


Fig. 2. The Basic module architecture.

The design of the Conv block is described in Fig. 3 with one convolution operation (Conv2D), one batch normalization (BN), and one Sigmoid Linear Unit (SiLU) activation function.



Fig. 3. The Conv block design.

Next, this design replaces all CONV blocks and Cross Stage Partial modules (Bottleneck CSP) in YOLOv5 with an architecture inspired by Ghost Bottleneck [11], named Ghost. This module is built on top of the GhostConv module (Fig. 4(a)) combined with the Squeeze and Excitation (SE) attention module [12] for stride of 1 (Fig. 4(b)) and added with the depthwise separable convolution layer (DWConv) [7] for stride of 2 (Fig. 4(c)).

Finally, this study also changes the Kernel size of the Maxpooling in the Spatial Pyramid Pooling (SPP) module from 5×5 , 9×9 , and 13×13 to 3×3 , 5×5 , and 7×7 . The changed SPP module is shown as in Fig. 5.

The neck module still reuses YOLOv5's Path Aggregation Network (PAN) architecture to combine the current feature maps with previous feature maps in the first stage. Multi-scale feature maps are generated with enriched information.

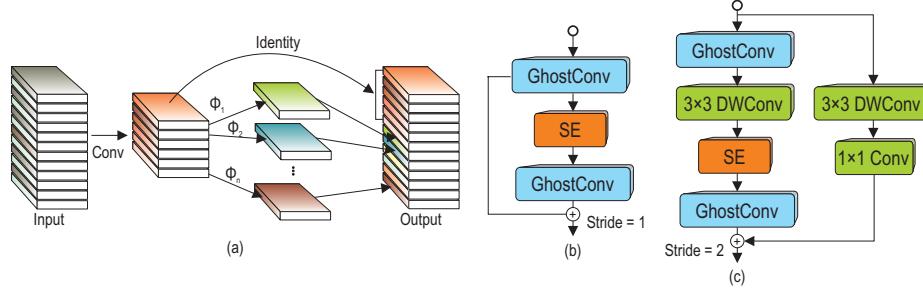


Fig. 4. (a) Ghost convolution, (b) Ghost Bottleneck module with the stride of 1, and (c) Ghost Bottleneck module with the stride of 2.

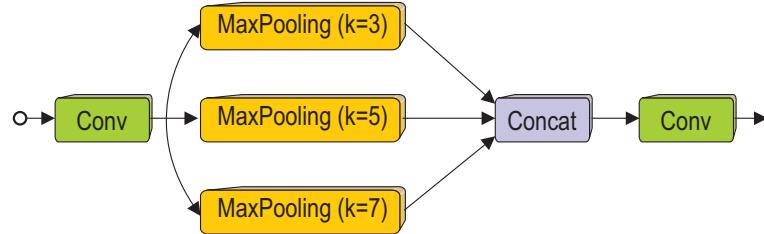


Fig. 5. The Spatial Pyramid Pooling (SPP) module.

These are the input of the detection heads. The CONV block in this module is replaced by the new Conv described above.

The detection head module utilizes three heads from the YOLOv5 architecture with feature maps from PAN neck including $80 \times 80 \times 256$, $40 \times 40 \times 512$, and $20 \times 20 \times 1024$. To increase detection ability, this work adds a detection head at a feature map of size $40 \times 40 \times 1024$ in the early stage of the PAN module. The study also resizes all anchor sizes to be suitable for the size of the objects in the PKLot dataset. The details of the detection heads and the anchor's designs are shown in Table 1.

Table 1. Heads and anchors design.

Head	Input	Anchors	Ouput	Object
1	$80 \times 80 \times 1024$	(4, 5), (8, 10), (13, 16)	$80 \times 80 \times 21$	Small
2 (Added)	$40 \times 40 \times 1024$	(10, 13), (16, 30), (33, 23)	$40 \times 40 \times 21$	Medium
3	$40 \times 40 \times 512$	(30, 61), (62, 45), (59, 119)	$40 \times 40 \times 21$	Medium
4	$40 \times 40 \times 1024$	(116, 90), (156, 198), (373, 326)	$20 \times 20 \times 21$	Large

3.2 Loss function

The loss function used in this paper is defined as follows:

$$\text{Loss} = \lambda_{box} L_{box} + \lambda_{obj} L_{obj} + \lambda_{cls} L_{cls} \quad (1)$$

Where L_{box} is the bounding box regression loss using CIOU loss, L_{obj} is the object confidence score loss using Binary Cross Entropy loss, and L_{cls} is the classes loss also using Binary Cross Entropy loss to calculate. λ_{box} , λ_{obj} , and λ_{cls} denote balancing parameters.

4 Experiments

4.1 Dataset

This experiment uses the Parking Lot Dataset [8] to train and evaluate the performance of the proposed network. This dataset is proposed by authors from the Federal University of Parana. The PKLot dataset contains 12,416 high-resolution images (1280×720 px) extracted from cameras of three different parking lots. The images were taken in sunny, cloudy, and rainy day conditions. Parking spaces are labeled as occupied and empty classes. To perform the experiment, this dataset was split into three subsets: training (8,691 images), evaluation (2,483 images), and testing (1,242 images). To be adaptive to the training process, this work reduces the image size to 640×640 px and converts the standard PKLot dataset format to YOLOv5 format.

4.2 Experimental setup

This study uses the original code of YOLOv5 [13] to generate modifications based on the Python programming language and the Pytorch framework. The proposed network is trained, evaluated, and tested on a GeForce GTX 1080Ti 11GB GPU. The Adam optimization is used. The learning rate is initially set to 10^2 and the final by 10^5 . The momentum start at 0.8 and then increased to 0.937. The training process goes through 300 epochs with a batch size of 32. The balancing parameters $\lambda_{cls}=0.5$, $\lambda_{box}=0.05$, and $\lambda_{obj}=1$, respectively. Several data augmentation methods are applied such as flip up-down, flip left-right, mixup, and mosaic. The speed testing process conducts on the PKLot test set with the image size of 640×640 , a batch size of 32, a confidence threshold of 0.5, and an IoU threshold of 0.5.

4.3 Experimental result

To evaluate the performance, this experiment performs training and evaluation from scratch YOLOv5 (n, s, m, l, x) versions and the proposed network. Besides, this work also compares with other previous networks that have been conducted on the PKLot dataset. As a result, the proposed network achieves

99.6% mean Average Precision (mAP). The results shown in Table 2 demonstrate that the network outperforms previous networks with 4.9% mAP when compared to the best competitor (GAN in [14]). When compared with tiny versions of YOLOv5, the proposed network achieves comparable performance to YOLOv5s and YOLOv5n while the network parameter (4,155,700 parameters) is only half that of YOLOv5s and more than two times that of YOLOv5n. In terms of computational complexity, the YOLO5PKLot network is only 2.8 GFLOPs, the smallest of all the comparison networks. The proposed network also achieves the best inference speed of 2.9 ms on the PKLot test set. The qualitative results of the proposed network on the PKLot dataset are shown in Fig. 6.

Table 2. Comparison result of proposed detection network with retrained YOLOv5 and other networks on PKLot dataset.

Model	Parameter	Weight	GFLOPs	mAP	Inference time
YOLOv5x	86,224,543	169.3 MB	204.2	99.7	20 ms
YOLOv5l	46,636,735	91 MB	114.3	99.7	16 ms
YOLOv5m	21,060,447	42.5 MB	50.4	99.7	13 ms
YOLOv5s	7,050,367	14.4 MB	15.3	99.6	11 ms
YOLOv5n	1,766,623	3.8 MB	4.2	99.6	3.1 ms
YOLOv3 [9]	N/A	N/A	N/A	93.3	N/A
Faster R-CNN [15]	N/A	N/A	N/A	91.9	N/A
Mask R-CNN [16]	N/A	N/A	N/A	92.0	N/A
GAN [14]	N/A	N/A	N/A	94.7	N/A
YOLO5PKLot	4,155,700	8.6	2.8	99.6	2.9 ms

With the outstanding ability in calculation and inference time as mentioned above, YOLO5PKLot can be applied in parking lot management systems with available low computing devices such as CPU and edge devices. However, during testing, the YOLO5PKLot also revealed several weaknesses that made the accuracy decrease when detecting objects in bad weather conditions such as the parking lot being obscured, the parking lot is far away from the camera, and different camera angles. Several detection mistakes are shown in Fig. 7.

4.4 Ablation study

Ablation study 1 conducted training and evaluation of proposed networks with backbone using the Basic and Ghost modules combined with an SPP module as standard. Then replace the Basic module with the Focus module, replace the SPP with the Spatial Pyramid Pooling - Fast (SPPF) module, and completely remove the SPP architectures for comparisons. The results in Table 3 show that when replacing the Basic module with the original Focus module, it increases the computational complexity to 9.2 GFLOPs while the network parameter reduces a few and just still maintains the mAP at 99.6 %. When replacing an SPP

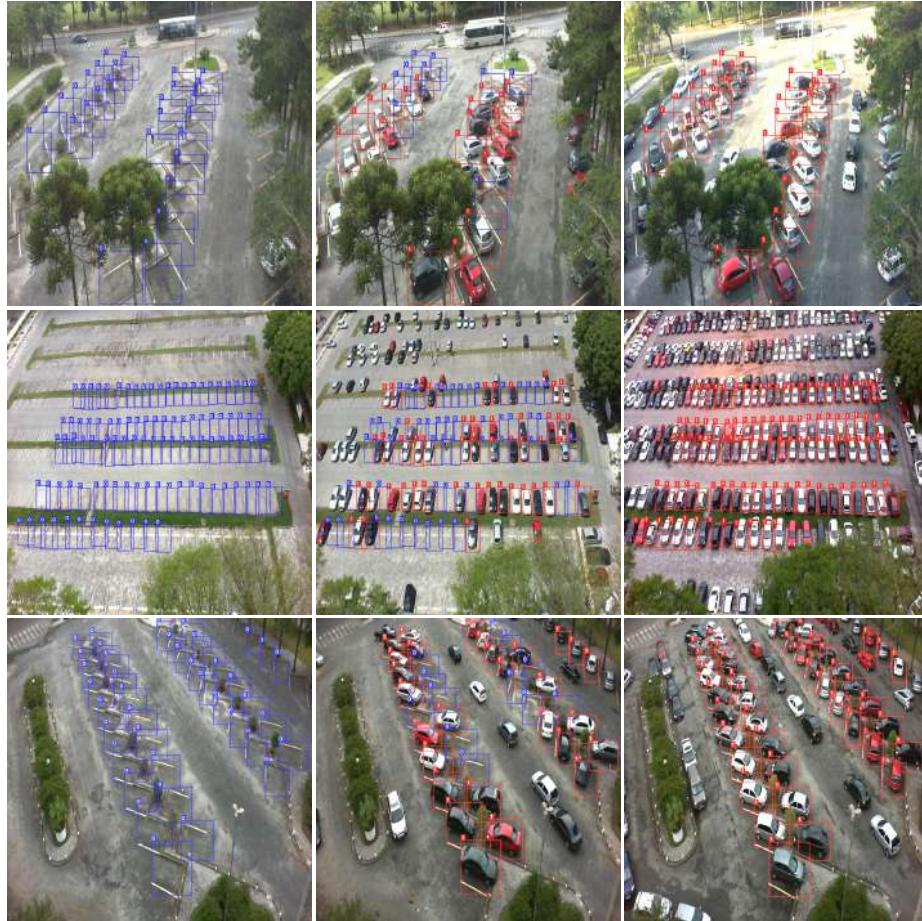


Fig. 6. The qualitative results of the proposed network on the test set of PKLot dataset with IoU threshold = 0.5. The numbers denote the classes: 0 is space-empty, and 1 is space-occupied.

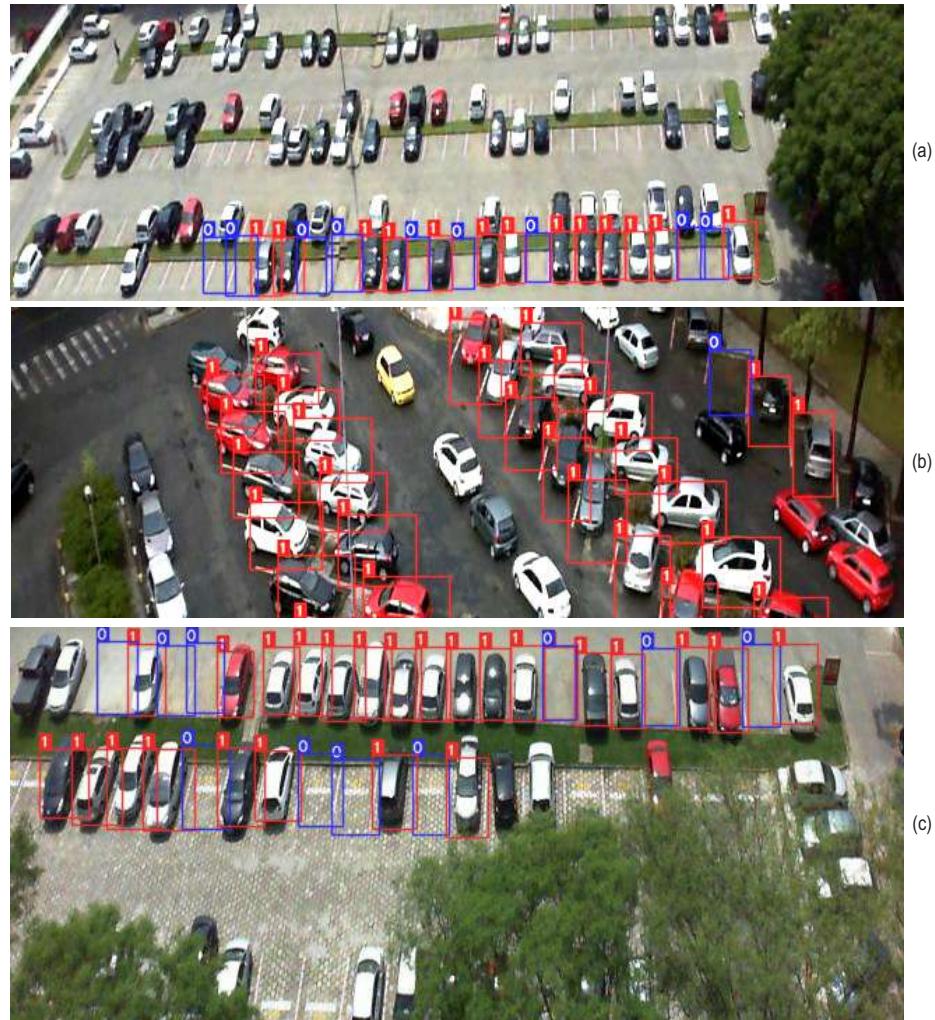


Fig. 7. Several detection mistakes in parking lot detection. (a) The parking lots are at a far distance, (b) The parking lots are obscured from each other, and (c) The parking lots are obscured by other objects (trees).

module with an SPPF module, the efficiency is similar at 99.6% mAP with the same parameters. When using a Basic module and all Ghost modules in the backbone, the network performance decreases by 0.2% mAP.

Table 3. Ablation studies with different backbone designs on the PKLot dataset.

Module	Proposed network			
Basic	✓		✓	✓
Focus		✓		
Ghost	✓	✓	✓	✓
SPPF			✓	
SPP	✓	✓		
Parameter	4,155,700	4,150,964	4,155,700	3,635,615
Weight (MB)	8.6	8.7	8.6	7.6
GFLOPs	2.8	9.2	2.8	2.8
mAP	99.6	99.6	99.6	99.4

In another ablation study, this work compared the performance of three and four detection heads. From the results in Table 4, it can be seen that adding a detection head increases the detection ability by 0.1% mAP. The network parameter increased slightly and the computational complexity remained the same at 2.8 GFLOPs.

Table 4. Ablation studies with different head numbers on the PKLot dataset.

Head	Parameter	Weight (MB)	GFLOPs	mAP@
3	4,150,303	7.6	2.8	99.5
4	4,155,700	8.6	2.8	99.6

5 Conclusion

This paper presents a method to improve the YOLOv5 architecture for parking lot detection in smart parking management systems. The proposed network consists of three main parts: backbone, neck, and head modules. The backbone is redesigned using lightweight architectures to reduce network parameters and computational complexity. The neck is optimized with the addition of activation functions behind convolution operation and BN. The head module added a new detection head and resized the anchors to increase the detection performance of the network. In the future, this network will be further developed with attention modules to address the network's weaknesses when detecting far-distant parking spaces and adapting to different camera angles.

Acknowledgement

This results was supported by "vanishing Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(MOE)(2021RIS-003)

References

1. Almeida, P., Oliveira, L.S., Silva, E., Britto, A., Koerich, A.: Parking space detection using textural descriptors. In: 2013 IEEE International Conference on Systems, Man, and Cybernetics. pp. 3603–3608 (2013). <https://doi.org/10.1109/SMC.2013.614>
2. Lisboa de Almeida, P.R., Oliveira, L.S., Souza Britto, A.d., Paul Barddal, J.: Naïve approaches to deal with concept drifts. In: 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC). pp. 1052–1059 (2020). <https://doi.org/10.1109/SMC42975.2020.9283360>
3. Almeida, P.R., Oliveira, L.S., Britto, A.S., Sabourin, R.: Adapting dynamic classifier selection for concept drift. Expert Systems with Applications **104**, 67–85 (2018). <https://doi.org/https://doi.org/10.1016/j.eswa.2018.03.021>, <https://www.sciencedirect.com/science/article/pii/S0957417418301611>
4. Antoni Suwignyo, M., Setyawan, I., Wirawan Yohanes, B.: Parking space detection using quaternionic local ranking binary pattern. In: 2018 International Seminar on Application for Technology of Information and Communication. pp. 351–355 (2018). <https://doi.org/10.1109/ISEMANTIC.2018.8549756>
5. Chen, H.C., Huang, C.J., Lu, K.H.: Design of a non-processor obu device for parking system based on infrared communication. In: 2017 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW). pp. 297–298 (2017). <https://doi.org/10.1109/ICCE-China.2017.7991113>
6. Chen, W., Sheu, Peng, Wu, L., Tseng: Video-based parking occupancy detection for smart control system. Applied Sciences **10**, 1079 (02 2020). <https://doi.org/10.3390/app10031079>
7. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)
8. de Almeida, P.R., Oliveira, L.S., Britto, A.S., Silva, E.J., Koerich, A.L.: Pklot – a robust dataset for parking lot classification. Expert Systems with Applications **42**(11), 4937–4949 (2015). <https://doi.org/https://doi.org/10.1016/j.eswa.2015.02.009>, <https://www.sciencedirect.com/science/article/pii/S0957417415001086>
9. Ding, X., Yang, R.: Vehicle and parking space detection based on improved yolo network model. Journal of Physics: Conference Series **1325**, 012084 (10 2019). <https://doi.org/10.1088/1742-6596/1325/1/012084>
10. Dizon, C.C., Magpayo, L.C., Uy, A.C., Tiglao, N.M.C.: Development of an open-space visual smart parking system. In: 2017 International Conference on Advanced Computing and Applications (ACOMP). pp. 77–82 (2017). <https://doi.org/10.1109/ACOMP.2017.29>
11. Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., Xu, C.: Ghostnet: More features from cheap operations. CoRR **abs/1911.11907** (2019), <http://arxiv.org/abs/1911.11907>

12. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. CoRR **abs/1709.01507** (2017), <http://arxiv.org/abs/1709.01507>
13. Jocher, G., Chaurasia, A., Stoken, A., Borovec, J., NanoCode012, Kwon, Y., Michael, K., TaoXie, Fang, J., imyhxy, Lorna, Yifu), Wong, C., V, A., Montes, D., Wang, Z., Fati, C., Nadar, J., Laughing, UnglvKitDe, Sonck, V., tkianai, yxNONG, Skalski, P., Hogan, A., Nair, D., Strobel, M., Jain, M.: ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation (Nov 2022). <https://doi.org/10.5281/zenodo.7347926>
14. Li, X., Chuah, M.C., Bhattacharya, S.: Uav assisted smart parking solution. In: 2017 International Conference on Unmanned Aircraft Systems (ICUAS). pp. 1006–1013 (2017). <https://doi.org/10.1109/ICUAS.2017.7991353>
15. Martín Nieto, R., García-Martín, , Hauptmann, A.G., Martínez, J.M.: Automatic vacant parking places management system using multicamera vehicle detection. IEEE Transactions on Intelligent Transportation Systems **20**(3), 1069–1080 (2019). <https://doi.org/10.1109/TITS.2018.2838128>
16. Mettupally, S.N.R., Menon, V.: A smart eco-system for parking detection using deep learning and big data analytics. In: 2019 SoutheastCon. pp. 1–4 (2019). <https://doi.org/10.1109/SoutheastCon42311.2019.9020502>
17. Sairam, B., Agrawal, A., Krishna, G., Sahu, S.P.: Automated vehicle parking slot detection system using deep learning. In: 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC). pp. 750–755 (2020). <https://doi.org/10.1109/ICCMC48092.2020.ICCMC-000140>
18. Scotiabank: Number of cars sold worldwide from 2010 to 2022, with a 2023 forecast (in million units). <https://www.statista.com/statistics/200002/international-car-sales-since-1990/>, note = Accessed: Jan. 01, 2023. [Online]. Available: <https://www.statista.com/statistics/200002/international-car-sales-since-1990/>
19. Shao, Y., Chen, P., Tongtong, C.: A grid projection method based on ultrasonic sensor for parking space detection. pp. 3378–3381 (07 2018). <https://doi.org/10.1109/IGARSS.2018.8519022>
20. Yuan, C., Qian, L.: Design of intelligent parking lot system based on wireless network. In: 2017 29th Chinese Control And Decision Conference (CCDC). pp. 3596–3601 (2017). <https://doi.org/10.1109/CCDC.2017.7979129>
21. Zhou, F., Li, Q.: Parking guidance system based on zigbee and geomagnetic sensor technology. In: 2014 13th International Symposium on Distributed Computing and Applications to Business, Engineering and Science. pp. 268–271 (2014). <https://doi.org/10.1109/DCABES.2014.58>

Texture Synthesis Based on Aesthetic Texture Perception Using CNN Style and Content Features

Yukine Sugiyama¹, Natsuki Sunda¹, Kensuke Tobitani^{1, 2[0000-0002-3898-8435]} and
Noriko Nagata^{1[0000-0002-2037-1947]}

¹Kwansei Gakuin University, Sanda, Hyogo669-1337 Japan

{ggs53875, nagata}@kwansei.ac.jp

²University of Nagasaki Nishi-Sonogi, Nagasaki 851-2195 Japan
tobitani@sun.ac.jp

Abstract. We propose a texture synthesis method that controls the desired visual impression by using CNN style features and content features. Diversifying user needs has led to the personalization of products according to individual needs. In the custom-made garment service, users can select and combine fabrics, patterns, and shapes of garments prepared in advance to design garments that meet their tastes and preferences. Controlling the visual impressions should allow the service to provide designs that better match the user's preferences. In image synthesis, controllable texture synthesis was performed with style and content; however, few previous study controls images based on impression (including aesthetics). In this study, we aim to synthesize textures with desired visual impressions by using style and content features. For this purpose, we first (1) quantify the affective texture by subjective evaluation experiments and (2) extract style features and content features using VGG-19 from pattern images for which evaluation scores are assigned. The explanatory variables are style and content features, and the objective variables are evaluation scores. We construct an impression estimation model using Lasso regression for each of them. Next, (3) based on impression estimation models, we control the visual impressions and synthesize textures. In (2), we constructed highly accurate visual impression estimation models using style and content features. In (3), we obtained synthesis results that match human intuition.

Keywords: Impression, Style, Content, Lasso Regression

1 Introduction

In product design, there is a growing interest in visual impressions. Visual impressions refer to the impression evoked by the surface properties of materials and are considered important in evaluating the quality and desirability of a product. In addition, the customization and personalization of products are becoming more common as the Internet spreads and users need to diversify. One example is a custom-made clothing service. In this service, users can design clothes according to their tastes by selecting and combining fabrics, patterns, and shapes of clothes prepared in advance. However, developing

2 Y.Sugiyama et

a system that supports users in creating their original designs is necessary to promote further personalization. However, creating original designs is difficult for users who do not know design. This study proposes a method to automatically synthesize texture images based on the user's desired visual impressions information. These techniques will enable design support based on human preferences, satisfaction, and other emotional values.

2 Previous Research

Long-standing studies on texture analysis have been closely linked to texture. Gatys et al. proposed an image transformation algorithm focusing on style features and content features extracted from VGG-19[1], a convolutional neural network used for object recognition. The proposed method produces images in which the style image's style is transferred to the shape and structure of the objects depicted in the content image and shows highly accurate results. This study suggests that style features retain more color and pattern information in the image, while content features retain more shape information [2, 3].

Previous studies have used style and content for controllable texture synthesis [4-8], but there are no previous studies that have used aesthetic (impression)-based control.

3 Proposed Method

In this study, we construct a model to estimate the visual impression using style and content features and synthesize the textures with the desired visual impressions' control. Figure 1 presents an overview of the proposed method. The first step is to extract style and content features from the pre-trained middle layer of VGG-19. Next, we construct a visual impression evaluation model by formulating the relationship between the evaluation points assigned to the pattern image and the extracted style features and content features, respectively, using Lasso regression. Finally, based on the constructed model, style and content features are calculated and textures synthesized, which control visual impressions.

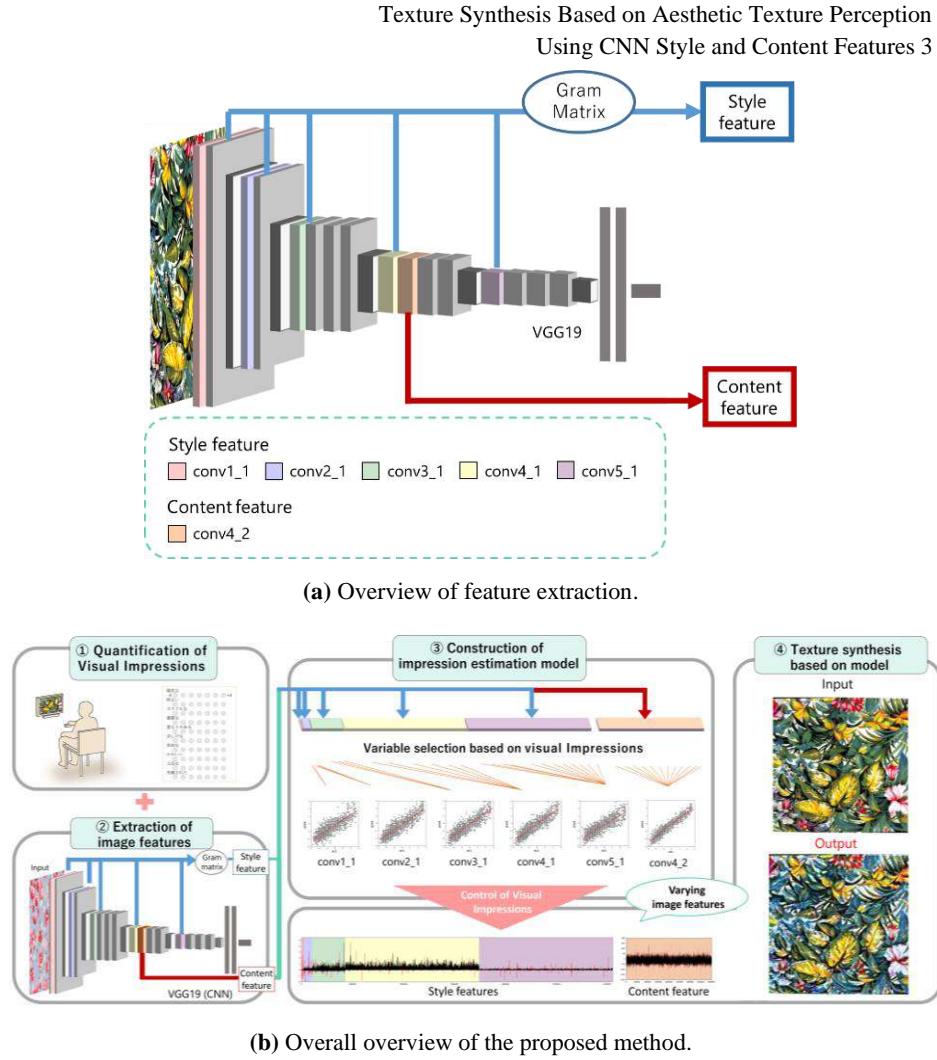


Fig. 1. Overview of the proposed method.

3.1 Extraction of image features

We extract image features to build impression evaluation models. Style features are cross-correlation matrices (Gram Matrix) of feature maps extracted from the middle layer of VGG19. The content features are feature maps extracted from the middle layer of VGG19. Style features are extracted from conv1_1, conv2_1, conv3_1, conv4_1, and conv5_1 based on the work of Gatys [2, 9]. The feature dimensions are 64×64, 128×128, 256×256, 512×512, and 512×512, respectively. Content features are extracted from Conv4_2. The number of feature dimensions is 28×28×512.

4 Y.Sugiyama et

3.2 Construction of a visual impression estimation model

We formulate the relationship between visual impressions and style and content features. Lasso regression is used in the formulation. Lasso regression is a penalized regression model in which the L1 regularization term is used to construct a regression model while preventing overlearning by setting the unselected variables to 0. We use Lasso regression because the explanatory variables are high-dimensional and excessive learning is expected. Objective variables are the evaluation points, explanatory variables are style features and content features, and Lasso regression is used to construct visual impressions' evaluation models.

3.3 Texture synthesis

Based on the model constructed in section 3.2, style features and content features with controlled visual impressions are calculated, and textures are synthesized. The texture synthesis process is completed in five different steps. (i) Extract style features and content features from the input image. (ii) Control the extracted image features. Equation 1. does the control The extracted image features are denoted as P_{original} , the regression coefficients obtained by building the model are transformed to fit the shape of the image features as $\hat{\omega}_{\text{lasso}}$ and the weights are denoted as S . (iii) Style and content features are extracted from the output images. (iv) Calculate the errors of style features and content features from the features of input and output images. Hereafter, the error of the style feature is denoted as style loss (L_{style}) and that of the content feature as content loss (L_{content}). (v) The sum of style loss plus weight α and content loss plus weight β is denoted as L_{total} . Update the output image to minimize equation (2) L_{total} . Iterate (iii) to (v) up to 300 times.

The image features are controlled by Equation 1. By applying the weight parameter S to the regression coefficients of the Lasso regression, the part of the image for which no variable is selected is kept at 0, and only the values for the part of the image strongly related to the affective texture are changed.

$$P_{\text{controlled}} = P_{\text{original}} \times (1 + \hat{\omega}_{\text{lasso}} \times S) \quad (1)$$

$$L_{\text{total}} = \alpha \times L_{\text{style}} + \beta \times L_{\text{content}} \quad (2)$$

In sections 4 and 5, we describe specific experiments in detail.

4 Experiment 1: Quantification of Visual Impressions

4.1 Collection and selection of evaluation terms

We collected and selected evaluation words related to the visual impressions evoked by the patterns. For the experimental method, we conducted free description and goodness-of-fit experiments based on the method of Tobitani [10]. Finally, a total of 10 evaluation words were selected for the subjective evaluation experiment: "cheerful," "bright," "colorful," "complex," "multilayered," "cool-looking," "free," "cute," "elegant," and "sophisticated"[11].

Texture Synthesis Based on Aesthetic Texture Perception
Using CNN Style and Content Features 5

4.2 Subjective evaluation test

We conducted a subjective evaluation experiment to quantify the visual impressions evoked by clothing patterns. Participants observed the stimuli presented on an LCD monitor and rated each evaluation word based on the degree to which it was true or false using a 7-point scale consisting of “not very true,” “not true,” “somewhat true,” “neither true nor false,” “somewhat true,” “true,” and “very true.” The participants were undergraduate and graduate students, male and female. We obtained rating data of 5 to 10 persons per stimulus and per rating word and scored each rating scale in 1-point increments, with -3 points for “not very much” and 3 points for “very much,” and defined the calculated mean value as the rating score (teacher data) for each stimulus and rating word [12]. Figure 2 displays the top 5 patterns with the highest evaluation scores for each evaluation term. From the figure, we confirmed that these evaluation scores aligned with human intuition.



Fig. 2. Top 5 images with the highest evaluation scores.

5 Experiment 2: Texture synthesis using visual impression estimation models

5.1 Extraction of image features

We extracted image features and identified style and content features from the 1158 pattern images that were given evaluation points, which were subjected to visual impressions quantification in section 4.

5.2 Construction of a visual impression estimation model

Objective variables are the evaluation points, explanatory variables are style features and content features, and Lasso regression is used to construct visual impressions’

6 Y.Sugiyama et

evaluation models. The penalty parameter of Lasso regression is the value obtained when K-split cross-validation minimizes the mean squared error. K=11 by Sturges' rule. As for the content features, since the total number of variables selected by this method was small, we added variables by entering values 0.8 times the selected coefficient of determination for variables with a high correlation (correlation coefficient of 0.8 or higher) with the selected variables.

Tables 1 and 2 show the coefficients of determination for each constructed model. In the model using style features, the average coefficient of determination of the five models was more than 0.5 for seven out of ten words, confirming that a highly accurate visual impressions evaluation model could be constructed. In Table 2, the coefficient of determination was 0.5 or higher for 9 out of the 10 words. For the words with low coefficients of determination, “free,” “elegant,” and “refined,” the variation of evaluation scores is slight (Fig. 3). This means that the relationship between visual impressions and image characteristics cannot be modeled precisely. Therefore, in the following texture synthesis, we will perform texture synthesis for the seven words, excluding these evaluation words.

Table 1.
(a) Coefficients of determination for impression estimation models constructed using style features.

evaluation term	conv1_1	conv2_1	conv3_1	conv4_1	conv5_1	average
cheerful	0.582	0.699	0.628	0.694	0.648	0.650
bright	0.711	0.784	0.760	0.801	0.695	0.750
colorful	0.330	0.565	0.608	0.695	0.603	0.560
complex	0.229	0.530	0.543	0.623	0.642	0.513
multilayered	0.167	0.488	0.570	0.673	0.661	0.512
cool-looking	0.699	0.775	0.776	0.809	0.716	0.755
free	0.172	0.386	0.408	0.487	0.332	0.357
cute	0.372	0.550	0.501	0.568	0.549	0.508
elegant	0.229	0.317	0.393	0.460	0.411	0.362
sophisticated	0.138	0.198	0.212	0.305	0.393	0.249

Texture Synthesis Based on Aesthetic Texture Perception
Using CNN Style and Content Features 7

(b) Coefficients of determination for impression estimation models constructed using content features.

evaluation term	Conv4_2
cheerful	0.713
bright	0.803
colorful	0.728
complex	0.800
multilayered	0.832
cool-looking	0.887
free	0.616
cute	0.692
elegant	0.554
sophisticated	0.371

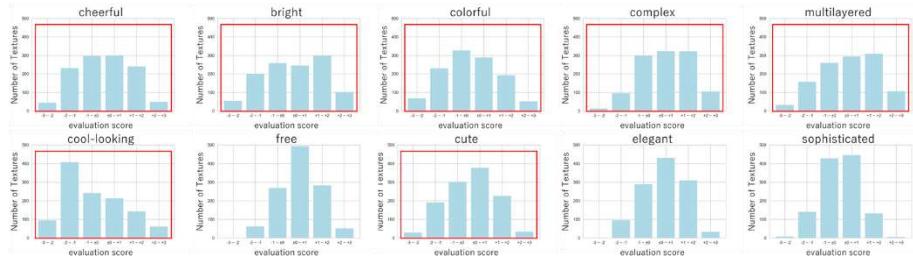


Fig. 3. Distribution of evaluation points (vertical axis: number of images, horizontal axis: evaluation points).

5.3 Texture synthesis

Based on models constructed in section 5.2, style features and content features with controlled visual impressions were calculated and textures were synthesized.

8 Y.Sugiyama et

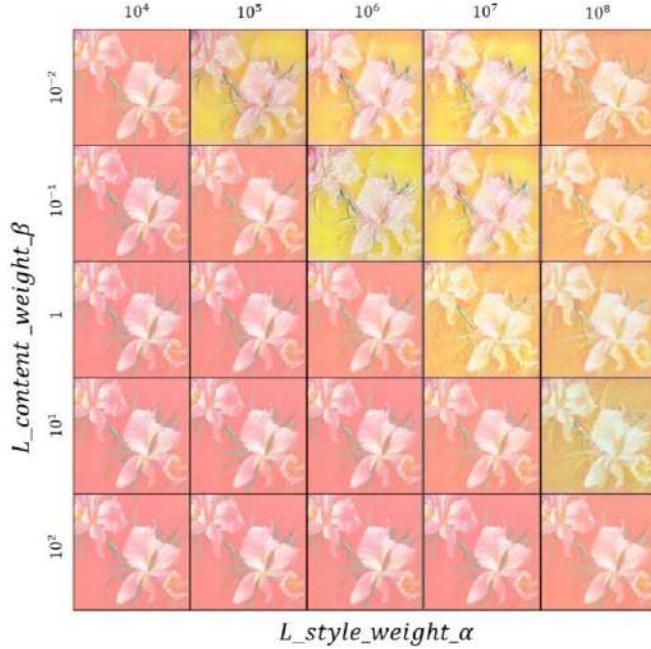
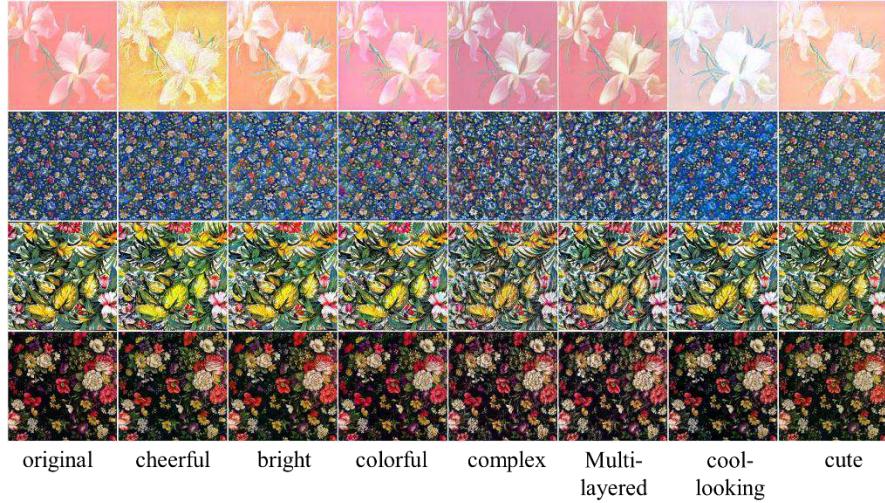


Fig. 4. synthesized results with varying weights α and β for the evaluation word “cheerful.”

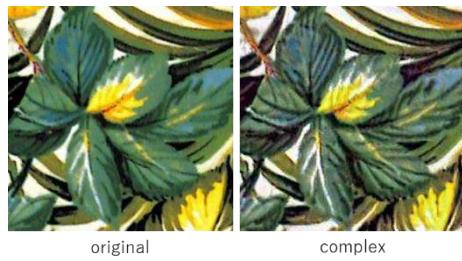
The results of synthesized images by varying weights α and β are shown in (Fig. 4). The input image size was set to 224×224 , and $S=10$ for the control of style features and $S=10^4$ for the control of content features. These images are shown to change in accordance with changes in the α and β weights.

Next, we observe the changes in the image when one of the values of α and β is fixed. First, (Fig. 5(a)) shows the case where $\alpha=10^6$ and $\beta=1$, and the style loss weights increase. In the case of “cheerful,” the entire image is yellowish. In the case of “bright,” the brightness seems to have increased, and in the case of “colorful,” the saturation seems to have increased, respectively, as appropriate. In addition, “complex” emphasized the veins of leaves (Fig. 5(b)), and ”multilayered ”emphasized shadows and other elements to give a visual impression of depth (Fig. 5(c)). In addition, the image in the ”cool-looking” group was tinted bluish, and in the “Cute” group, the brightness was increased while the saturation was decreased, giving the image a pastel tone.

Texture Synthesis Based on Aesthetic Texture Perception
Using CNN Style and Content Features 9



(a) Synthesized results for the case $\alpha=10^6$, $\beta=1$.

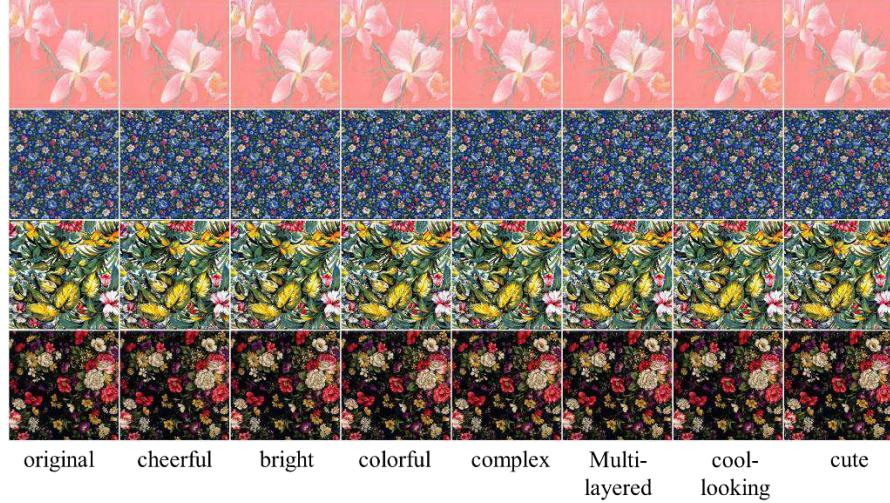


(b) Synthesized results for “complex.”



(c) Synthesized results for “multilayered.”

10 Y.Sugiyama et



(d) Synthesized results for $\alpha=1, \beta=10^6$.

Fig. 5. Texture synthesis results.

On the other hand, (Fig. 5(d)) shows the case where the content loss weights are increased to $\alpha=1$ and $\beta=10^6$. Changes in texture were observed in the “complex” and “multilayered” cases. In both cases, shading is emphasized, and light areas are especially emphasized in the “multilayered” case. However, for the evaluation terms in general, it was found that the images with larger content loss weights showed more minor changes than those with larger style loss weights.

In this section, we only control the style features that were found to be particularly effective and synthesize the textures. In the next section, we examine the validity of the synthesized images.

6 Experiment 3: Verification

We quantitatively verify whether the visual impressions evoked by the synthesized images are significantly improved compared to the original images by focusing on exaggeration. In this section, only the control of style features, which showed appropriate changes in the previous section, is subject to verification, and parameters of $\alpha=10^6$, $\beta=1$, and $S=10$ are employed.

6.1 Construction of experimental dataset

First, we constructed a dataset for the effectiveness experiment.

Stimuli were selected from 2878 unknown images in the dataset. We first estimated (i) the visual impressions of the source images. Using the model constructed in Chapter 6, we calculated the evaluation score for each pattern by inputting the style features

Texture Synthesis Based on Aesthetic Texture Perception
Using CNN Style and Content Features 11

extracted from the original images. Next, (ii) patterns with high/medium/low evaluation points in common for all words (7 words) were extracted. The patterns were arranged in the order of the highest score for each word and divided into three groups: high, medium, and low. Then, the patterns in the high, medium, and low-rank groups for all the evaluation terms were extracted, resulting in 51, 7, and 14 patterns in this order, respectively. Finally, we selected patterns that satisfied (iii) “stability of synthesis” and “visibility of exaggeration.” We synthesized 35 images (5 times for 7 words = 35 images) and selected 10 patterns as stimuli that satisfied each criterion shown in Table 2.

Table 2. Criteria for “stability of Synthesis” and “visibility of exaggeration.”

stability of Synthesis	<ul style="list-style-type: none"> • No image is blacked out. • The same quality is produced at least 4 out of 5 times for all words. • The structure of the pattern is established.
Visibility of exaggeration	<ul style="list-style-type: none"> • The change is easy to see compared to the original image.

6.2 Effectiveness verification experiment

Next, we conducted an effect verification experiment. Participant participants were asked to observe the stimulus pairs presented on an LCD monitor and to answer which of the four evaluation words was true for each word using a four-trial scale consisting of “left,” “more or less left,” “more or less right,” and “right.” The Total number of trials per participant was 280, using the experimental dataset constructed in Section 8.1. The participants were 10 undergraduate and graduate students (5 males and 5 females, aged 23.3 ± 1.19 years). To eliminate the influence of the order effect, the order in which the stimulus pairs were presented was randomized for each participant, and the order of the evaluation words was randomized for each trial.

6.3 Results and Discussion

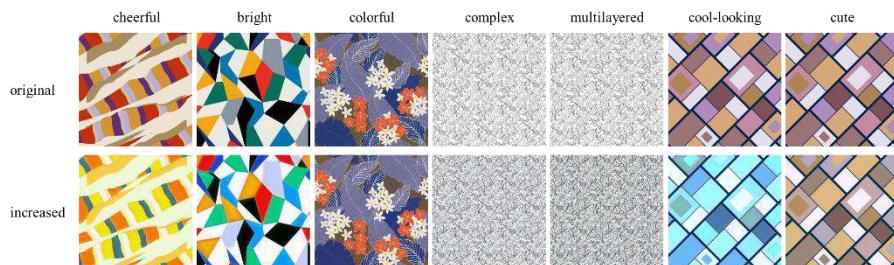
The validity of this method was verified by conducting a statistical analysis of the data obtained in section 8.2 and obtaining the psychological scale values. Multiple comparisons were conducted using the yardstick method to evaluate whether there was a statistically significant difference between each stimulus. As a result, we confirmed that the psychometric values of the synthesized images with exaggerated visual impressions qualities were significantly higher than those of the original images in the proportions shown in Table 3. Figure 7 shows the patterns with the greatest increase and the patterns with the greatest decrease in the psychometric scale values. Figure 7(a) confirms that the images produced by both patterns generally met people’s intuition in terms of visual impressions. In Figure 7(b), the psychometric scale value decreased, but the visual impression did not change. In particular, with the “cool-looking” case, the psychometric scale value increased for all images. Figure 8 presents the changes in the psychological

12 Y.Sugiyama et

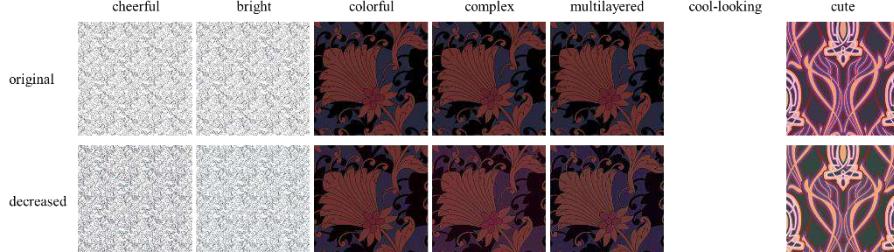
scale values. These results indicate that the proposed texture synthesizing method may be effective at exaggerating the desired visual impression.

Table 3. Percentage of patterns with significantly increased psychological scale values.

Exaggerated visual impressions	p<.01(**)	p<.05(*)
cheerful	0.7	0.8
bright	0.7	0.8
colorful	0.1	0.1
complex	0.3	0.4
multilayered	0.3	0.4
cool-looking	0.9	0.9
cute	0.0	0.0



(a) Image with the most significant increase in psychological scale value.



(b) Image with the most significant decrease in psychological scale value.

Fig. 6. Changes to psychometric scale values and texture images.

The three words “cheerful,” “bright,” and “cool-looking” significantly increased the psychometric scale values for most patterns. Since these are low-order visual impressions qualities perceived from color information, they varied regardless of the pattern’s taste. The psychometric values of “complex” and “multilayered” patterns increased significantly in about half of the patterns. The patterns that showed a significant increase were those with delicate patterns, and the lines became thicker and more three-

Texture Synthesis Based on Aesthetic Texture Perception
Using CNN Style and Content Features 13

dimensional according to the patterns. On the other hand, the patterns with larger scales did not show such changes, which may explain the non-significance of the results.

Next, the psychological scale value of “colorful” increased for most patterns but was non-significant for all of them. One of the reasons for this is that the degree of change was smaller than the other words. At the present stage, the weight parameter in Equation 7.4 is unified as $S=10$ for all words, but for “colorful,” we confirmed that the degree of change approaches the other words by setting a more significant value of S (Fig. 8). Therefore, further study is needed to adjust the parameters.

Additionally, none of the patterns significantly increased in the “cute” category, and half of the patterns significantly decreased in the “cute” category. One of the possible reasons for this is the influence of the original images. It is assumed that patterns with low saturation in the original images are faced with a decrease in saturation, and the balance of the color scheme perceived as “cute” is lost. Furthermore, since “cute” is a higher-order sensory quality consisting of various elements, it is also considered affected by individual differences. Therefore, we conducted a factor analysis of each participant’s psychological scale of “cute” for each image. Due to the factor analysis, three factors were extracted, with a cumulative contribution rate of 60.2%. Table 4 shows the factor loadings matrix after rotation, and Table 5 shows the factor correlation matrix.

14 Y.Sugiyama et

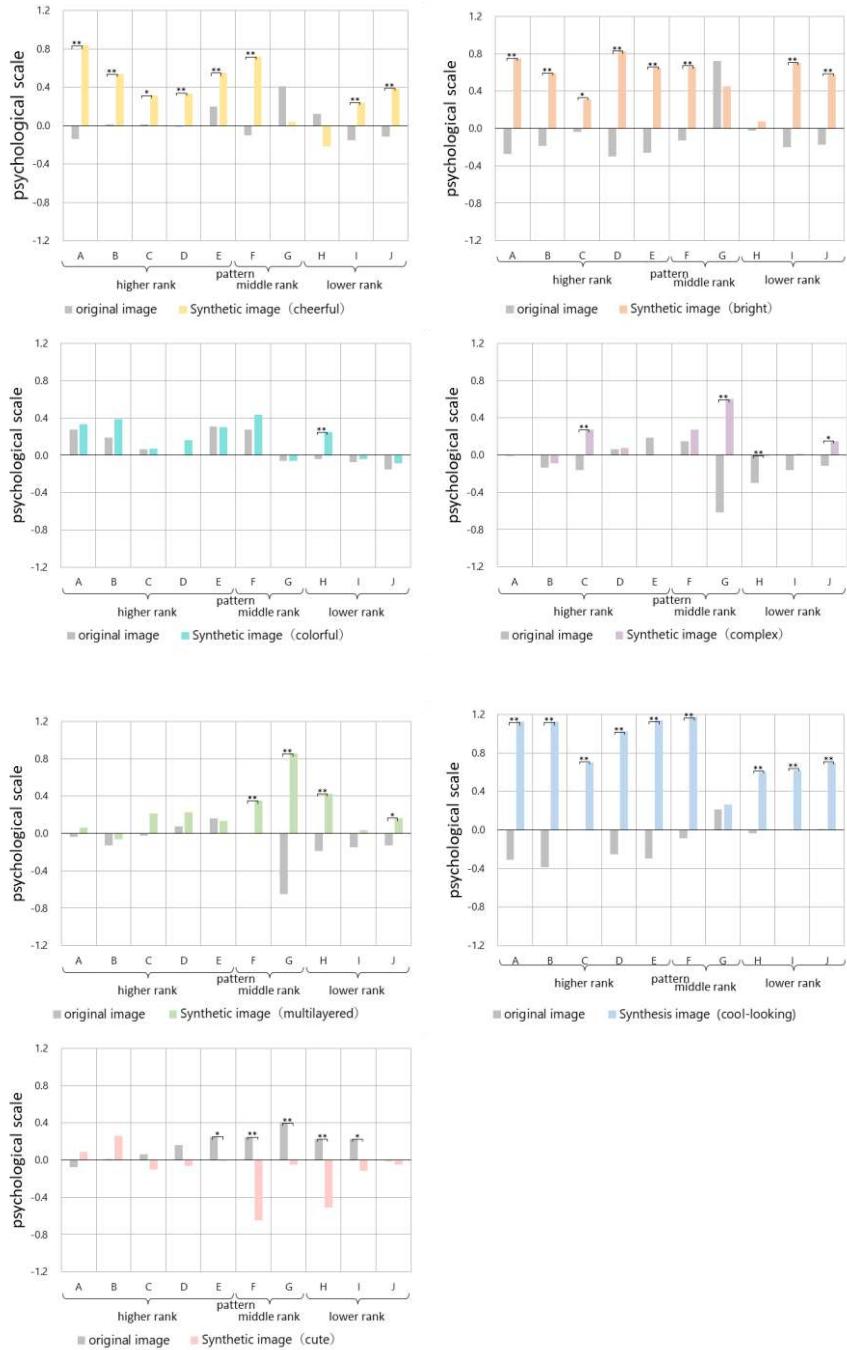


Fig. 7. Change in psychological scale values.

Texture Synthesis Based on Aesthetic Texture Perception
Using CNN Style and Content Features 15

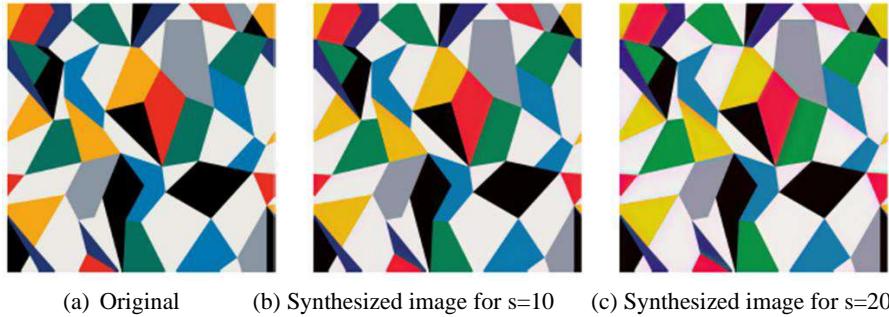


Fig. 8. Comparison of different values of the magnification parameter in “Colorful.”

Table 4. Factor loadings matrix after rotation

participant	factor		
	F1	F2	F3
No. 1	1.178	-0.322	0.013
No. 9	0.704	0.240	-0.168
No. 7	0.550	0.285	0.031
No. 8	0.502	0.326	0.022
No. 10	0.462	0.061	0.403
No. 5	0.034	0.784	0.075
No. 6	0.175	0.655	-0.086
No. 2	-0.159	0.536	0.073
No. 4	-0.062	-0.036	0.853
No. 3	-0.013	0.155	0.542

Table 5. factor correlation matrix

factor	F1	F2	F3
F1	1	0. 665	0. 494
F2	0. 665	1	0. 466
F3	0. 494	0. 466	1

Comparing the factor scores for each image revealed that participants had different evaluation tendencies with each factor. Participants with the “F2” factor tended to rate “cute” highly for the synthesized images with “cheerful” and “bright” exaggerated. Participants with the “F3” factor tended to rate the original and synthesized images with

16 Y.Sugiyama et

the exaggerated “colorful” highly. Participants with the “F1” factor tended to rate the original image, and the “cool-looking” exaggerated image lower than those with the “F2” factor and the “F3” factor. This suggests that the model should be expanded to consider future evaluation tendencies differences among individuals.

7 Conclusion

In this study, we proposed a method for Synthesizing texture images of clothing patterns with desired visual impressions. Our research makes it possible to synthesize controllable textures to affect visual impressions. First, (1) subjective evaluation experiments were conducted on pattern images to quantify the visual impression. We obtained evaluation scores for 10 words that express the visual impressions for the image dataset collected from floral patterns. Next, (2) style and content features were extracted from the pattern images used in the subjective evaluation experiment using the pre-trained VGG19. Then, we constructed a visual impressions evaluation model by formulating the relationship between the quantified visual impressions, the extracted style features, and the content features using regression. As a result, we could model visual impressions with high accuracy while selecting features that are mainly strongly related to visual impressions. Finally, (3) based on the obtained model, we calculated the image features when the desired visual impressions quality is controlled and synthesized images by optimizing the model to minimize the error between the features and the original images. (4) To verify the validity of the proposed method, we synthesized unknown images with the desired exaggerated visual impressions. It was found that the changes in the images synthesized using the content features were smaller than those synthesized using the style features. In addition, the experiment demonstrated the method’s effectiveness in which the emotional quality evoked by the synthesized images was significantly improved compared to the original images.

As future research topics, we will extend the model to a higher-order visual impression consisting of various elements, such as “cute,” to consider individual differences. In addition, we will quantitatively verify the degree to which the degree of exaggeration of the visual impressions quality changes by adjusting the weight parameters set when changing the style features according to the taste of the words and patterns.

References

1. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv, 1409–1556 (2014).
2. Gatys, L. A., Ecker, A. S., Bethge, M.: Image style transfer using convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2414–2423 (2016).
3. Wang, P. Li, Y., Vasconcelos, N.: Rethinking and improving the robustness of image style transfer. In: Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Nashville, pp. 124-133 (2021).

Texture Synthesis Based on Aesthetic Texture Perception
Using CNN Style and Content Features 17

4. Yu, N., Barnes, C., Shechtman, E., Amirghods, S., Lukac, M.: Texture mixer: A network for controllable synthesis and interpolation of texture. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12164–12173 (2019).
5. Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X., Yang, M.: Diversified texture synthesis with feed-forward networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3920–3928 (2017).
6. Yang, S., Wang, Z., Wang, Z., Xu, N., Liu, J., Guo, Z.: Controllable artistic text style transfer via shape-matching GAN. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 4442–4451 (2019).
7. Chen, H., Zhao, L., Wang, Z., Zhang, H., Zuo, Z., Li, A., Xing, W., Lu, D.: DualAST: Dual style-learning networks for artistic style transfer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 872–881 (2021).
8. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10684–10695 (2022).
9. Takemoto, A., Tobitani, K., Tani, Y., Fujiwara, T., Yamazaki, Y., Nagata, N.: Texture synthesis with desired visual impressions using deep correlation feature. In: 2019 IEEE International Conference on Consumer Electronics (ICCE), pp. 1–2. IEEE, Las Vegas (2019).
10. Tobitani, K., Matsumoto, T., Tani, Y., Fujii, H., Nagata, N.: Modeling of the relation between impression and physical characteristics on representation of skin surface quality. The Journal of The Institute of Image Information and Television Engineers 71(11), 259–268 (2017).
11. Mori, T., Uchida, Y., Komiyama, J.: Relationship between visual impressions and image information parameters of color textures. Journal of the Japan Research Association for Textile End-uses 51(5), 433–440 (2010).
12. Sunda, N., Tobitani, K., Tani, I., Tani, Y., Nagata, N., Morita, N.: Impression estimation model for clothing patterns using neural style features. Proceedings of the Springer International Conference on Human-Computer Interaction, pp. 689–697 (2020).

Emotion Recognition by using optimised deep features

Irfan Haider, Guee-Sang Lee*, Hyung-Jeong Yang, and Soo-Hyung Kim

Department of Artificial Intelligence Convergence
 Chonnam National University, Gwangju 61186, South Korea
irfan_haider99@hotmail.com, gslee@jnu.ac.kr, hjang@jnu.ac.kr, and shkim@jnu.ac.kr

Abstract. This research solves the fundamental high-dimensional classification problem in machine learning. A classifier's performance may suffer as the number of attributes in the data is too large since there are fewer training samples available. Limiting the number of features through feature selection is one approach to overcoming this difficulty. Unlike earlier methods, ours proposes selecting features based on knowledge of how they will interact. Our approach employs the transfer learning with Residual Neural Network to extract the deep features first and select the optimal features by using principal component analysis (PCA) and T-distributed Stochastic Neighbor Embedding (t-SNE). Our approach is efficient and is able to feed the optimal features to the classifiers instead of feeding irrelevant information. Experiment Results on two high dimensional datasets shows the performance of our approach in the form of reduction of time and overall cost.

Keywords: Transfer Learning · Residual Neural Network · PCA and t-SNE.

1 Introduction

In recent years, it has become increasingly crucial to be able to read a person's emotional state. Human emotion recognition has garnered attention in several fields, including but not limited to human-computer(8), academia, and medical. Effective role of feelings in interpersonal communication is an impossibility. Emotions have a crucial role in regular human conversation. Sensors can learn about a person's mental state by listening to their voice and reading their facial expressions and body language. The dash et al. (2) states that nonverbal cues, such as voice tone and body language, make up 38% and 55%, respectively, of daily communication, whereas verbal cues account for only 7%.

Emotions can be read from the face, the tone of voice, and the body language of a person. Researchers have found that facial expressions are the most effective means of communicating feelings. Evidence of them can be presented in many forms, some of which are readily apparent to the naked eye and others of which are not.

Research on emotions draws from several fields, including psychology and computer science. In psychological words, it's a condition that influences one's way of thinking, feeling, and behaving, as well as one's level of contentment with life (9). In contrast, in the area of computer science, it can be identified in the form of visual, auditory, and textual data. It's not simple to extract feelings from any of these signals. Emotions, whether happy, negative, or neutral, are the primary means by which humans express themselves to one another. It's commonly recognized that words like "cheerful," "happy," and "excited" are used to portray positive emotions, while words like "hate," "anger," "fear," "depression," and "sad" are used to indicate negative ones. Social media platforms like Facebook, Instagram, and others have become the primary means by which people share information and communicate their emotions (10). They provide a wide range of outlets for expressing inner states.

The constraint of dimensionality can be overcome in pattern recognition by feature selection. Feature selection refers to the steps used to narrow down a large feature set to a manageable subset (1) (2). Features that are particular to a given class and do not overlap with other features make up the best feature set. For high-dimensional data in particular, feature selection is a crucial pre-processing step in machine learning since it reduces the cost of data gathering, aids in the discovery of crucial traits, and improves classification accuracy. In the recent century, high-dimensional data has proliferated, making feature selection more important than ever.

Filter methods, wrapper methods, and embedding methods are the three main types of feature selection strategies used in supervised learning. Methodologies for filtering data (3) (4) assess the value of a subset based on key features, such as information-based measures, distances, or statistical data. These techniques are particularly effective since they do not rely on a learning classifier to select the relevant characteristics. Consequently, it is a feature selection approach for high-dimensional data sets, although the outcomes are subpar. In contrast, wrapper techniques assess the usefulness of a given classifier to measure the importance of a narrowed subset and during search (5) (6). Wrapper approaches for high-dimensional data are time-consuming, but they outperform filter techniques for the same amount of extracted features (7). Each of these approaches, however, has its own limitations. Compared to the wrapper method, which relies on a centralized mechanism for evaluating and selecting features, the filter method relies on decentralized, independent evaluation to arrive at its final decision (6). The embedded approaches (8) (9) interact with the specific structure of a classifier, like the support vector machine (SVM) classifier or even the decision tree classifier, to select a set of characteristics that will be useful for classification. Consequently, this approach can restrict just few classifiers.

2 Related Work

The development of deep learning has substantially enhanced the precision of facial emotion identification. To solve the difficulties of emotion recognition from

facial expressions, many new Convolutional Neural Network (CNN) models have been invented recently. It is a top network in its industry. The building blocks of a convolutional neural network (CNN) are convolutions, activation layers, and pooling layers. Computing terminal devices can now interpret the variations in human emotions to an extent, resulting in more diversity in human-computer communication [11], thanks to the development of Artificial Intelligence technologies like pattern recognition and computer vision. The primary goal of face recognition (FER) is to assign a certain emotional state to a particular facial expression. Feelings can be identified by parsing a face image for recognizable traits and using those elements to identify the subject's emotional state. Face photos require some processing before being fed into a convolutional neural network (CNN) or other machine learning classifier. Existing techniques include the viola-jones algorithm[15], the histogram of gradients [14], the histogram equalization [13], the linear discriminant analysis[12], the histogram of discrete wavelets [12], etc.

Manual feature extraction excels at identification in controlled lab settings but struggles in real-world settings with factors like occlusion and lighting. Recently, there has been a lot of interest in using deep convolutional neural networks for feature extraction[16], which has improved the accuracy of face emotion identification. With its novel strategy to Deep Neural Network optimization, the Deep Residual Network[17] (Deep ResNet) has made significant strides in the field of image recognition. The two-stage classical learning technique used in earlier study on emotion recognition has been abandoned. In the initial phase, we employ image processing methods for feature extraction. Conversely, in the second phase, we used a classic machine learning classifier like Support Vector Machine (SVM) to identify feelings. Weighted random forest (WRF)[18] is one way FER has utilized to glean the most important features of image compositions. Hasani and Mahoor[19] employed a novel network called ResNet-LSTM, which integrates lower highlights to LSTMs, to capture Spatio-temporal data. Because of its improved feature extraction capabilities, the deep learning network has become the most preferred approach to FER. In the wavelet domain, Nigam et al.[14] provided a four-step procedure for efficient FER based on the histogram of oriented gradients (HOG) (face processing, domain transformation, feature extraction and expression recognition). The scientists used a tree-based multi-class SVM to categorize the HOG features obtained by the discrete wavelet transformation and then applied those findings to facial emotion identification. The CK+, JAFFE, and Yale datasets were used for training and testing. Three datasets have been examined, with the results showing an accuracy of 90%, 71.43%, and 75% in the test set.

After extensive research into the Facial Expression Recognition issue, Minaee et al. presented an Attentional Convolutional Neural Network [20] as an alternative to simply adding more layers/neurons. In addition, they proposed using a visualization tool that, using the classifier's output, can zero in on crucial regions of the face for discerning a variety of emotions. An integral aspect of their design is a spatial transformer network that performs an efficient transformation

to encase the input and provide an appropriately transformed output. For the 7 classes they were tasked with classifying, they used the FER2013 dataset and achieved an accuracy rate of 70.02 percent.

The authors utilized the Residual Masking Network⁽²¹⁾ to zero in on deep architecture via the attention mechanism. To improve feature maps, they trained a segmented network to zero in on only the data it needed to make an informed call. They split their work into two sections: the residual masking block, which includes a residual layer, and also the ensemble approach for the conjunction with seven separate CNNs. Their final accuracy on the FER2013 dataset test set was 74.14%. Using a feature extraction network and a pre-trained model, Pu and Zhu⁽²²⁾ created a FER framework. Using a supervised learning technique called residual block optical flow, we may extract useful features.

Inception is used as a classifier for its innovative design. In tests on CK+ and FER2013 datasets, they were able to improve accuracy to 95.74 and 73.11 percent, respectively. Chowanda⁽²³⁾ developed a separable CNN to address the issue of how much computing power is needed to train and analyses CNNs for emotional recognition. As part of the experiment, we compared four different kinds of networks against one another. Networks both with and without modular separation, with and without flattening and completely connected layers, and with and without resorting to global average pooling. They got an accuracy of 99.4 percent on the CK+ dataset with their proposed architecture, and it was faster and had fewer parameters. In our method we used a novel method to choose the optimal number of feature to feed the classifier instead of feeding all the information for this purpose we use PCA and t-SNE. Our method showed efficient performance on two datasets. Remaining paper is consist of proposed method, experiment & results and conclusion.

3 Proposed Method

There are four layers on masking network, each Residual Masking Block includes a Residual Layer and a Masking Block that performs its function on features of varying sizes. After being passed through a 3x3 convolutional layer with stride 2, an input image of size 224x224 will be passed through a 2x2 max-pooling layer, reducing its spatial size to 56x56. We optimised it with triplet loss function and feed the deep features to PCA and t-SNE for dimension reduction. By feeding we obtained the optimal deep features with only the most impact 20 dimensions remained. The following four Residual Masking Blocks turn the feature maps derived from the preceding pooling layer into feature maps with four different spatial sizes: 56 by 56, 28 by 28, 14 by 14, and 7 by 7. At the very end, the network employs a pooling layer to average the inputs and a softmax layer with 7 inputs to generate outputs that map to 7 different expressions (6 emotions and one neutral state). The proposed method is showed in [Figure 1](#)

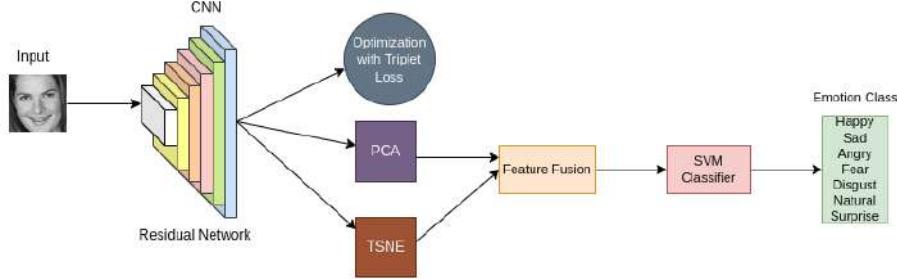


Fig. 1: Proposed Method

4 Experiment and Results

4.1 Dataset

In our experiment three public dataset CK+48, JAFFE and FER2013 are used. At ICML 2013's Challenges in Representation Learning, the first widely used dataset, FER2013(24), was unveiled. As can be seen in Figure 2, there are a total of 35887 greyscale (48x48) photos inside this collection. To train the model, we used 28709 photos; to validate it, we used another 3850; and to test it, we used 3589. Google's image search API gathers all of these pictures and assigns labels for anger, disgust, fear, happiness, sadness, surprise, and neutral. This data set is commonly used for benchmarking various FER approaches that involve deep learning.

Dataset	Type	#Sample	#Feature	#Classes
CK+48	Face image	981	20	7
JAFFE	Face image	213	20	7

Table 1: Number of Features, Samples, and Class in Each DataSet

4.2 Experimental Setup

In order to train with pre-trained models from ImageNet, the original training images are scaled to 224 x 224 and converted to RGB. In addition, over-fitting is avoided by augmenting training photos. Rotating by a factor of(25) is one of the augmentation techniques, along with a left-right flip. For each experiment, if the validation accuracy does not improve by at least eight steps after 50 epochs, the experiment is terminated. Scheduler reduces learning rate by a factor of 10 if validation accuracy does not improve over five consecutive epochs while using

a batch size of 48 and an initial learning rate of 0.0001. Network-agnostic experiments are run with the same hyperparameters, preprocessing, augmentation, and evaluation metrics as one another Pytorch is used for the experiments, and the graphics card used is a RTX 3090.

Processing time in the actual application is tested using a desktop computer equipped with a AMD® Ryzen 7 2700x eight-core processor \times 16 CPU, a Graphics Processing Unit (GPU) GTX 1050Ti, and 24GB of RAM. The suggested network can handle 100 face-containing frames per second with the current setup. This finding gives us confidence in the practicality of our application in real time.

	CK+48
Proposed method	99.66%
Deep Features + SVM classifier	98.83%

Table 2: Performance comparison between our proposed pipeline with previous works on CK+48 dataset.

We conduct two experiments on CK+48 to assess the effectiveness of the proposed method with one experiment using the proposed method and the other feeding deep features to the SVM classifier directly. The dataset is split into training and validation subsets with 7/3 ratio. Table 2 shows that the proposed method increases accuracy on CK+48 with almost 1%. We also provide the training curve and confusion matrix in Figure 2. Following the confusion matrix, the worst performance is in the "fear" class and all the other classes achieve almost perfect performance.

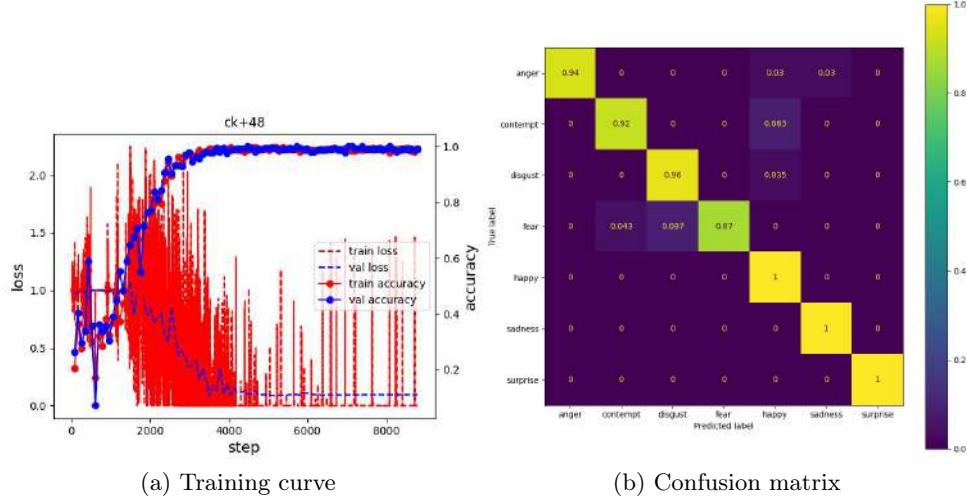


Fig. 2: Training curve and confusion matrix on CK+48

	JAFFE
Proposed method	98.79%
Minaee and Abdolrashidi (2019)	92.80%
Khaireddin and Chen (2021)	73.28%
Aouayeb et al. (2021)	94.83%
Boughida et al. (2022)	96.30%
Shaik and Cherukuri (2022)	97.46%

Table 3: Performance comparison between our proposed pipeline and previous works on the JAFFE dataset.

We also experiment on JAFFE dataset to compare our proposed method with other previous works. We split the dataset into training and validation subset with ratio of 7/3. The Table 3 shows that our proposed method outperforms all the previous works and achieves the SoTA result on JAFFE dataset with 98.79%. The confusion matrix in Figure 3 also shows the near-perfect performance on all emotion classes in JAFFE dataset.

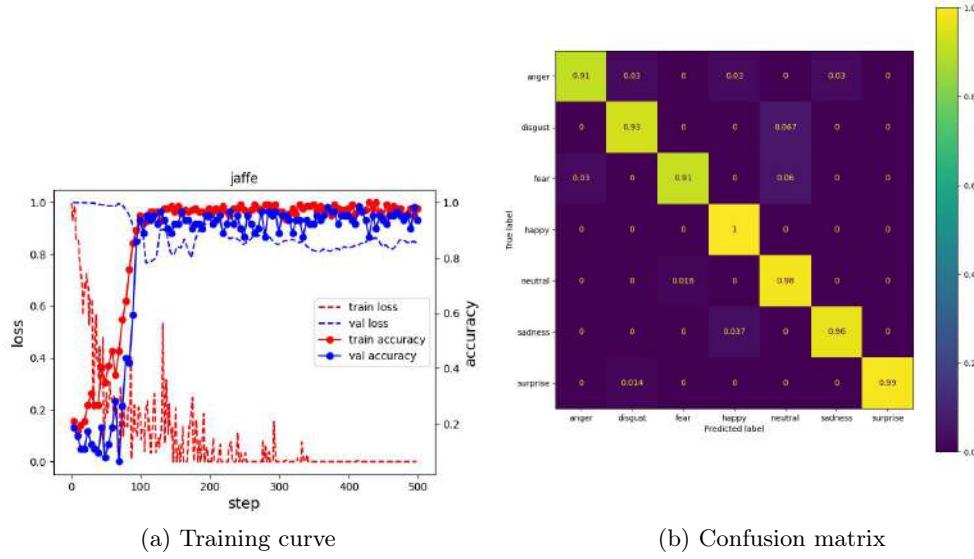


Fig. 3: Training curve and confusion matrix on JAFFE

To further evaluate our proposed method, we perform the proposed method on a larger and more complex dataset than previous ones and show that the proposed method can work well on a small amount of the original FER2013 dataset. We conduct experiments on 5%, 10% and 15% of FER2013 datasets. The experiment results are showed in Table 4. The results shows that our proposed method can achieve the comparable results with other previous SoTA works.

	5% FER2013	10% FER2013	15% FER2013
Proposed method	53.45%	57.24%	61.76%
Barsoum et al. (2016)	53.37%	57.43%	60.11%
Khaireddin and Chen (2021)	53.56%	58.43%	61.32%
Pham et al. (2021)	54.12%	60.60%	61.60%

Table 4: Performance comparison between our proposed pipeline and previous works on the FER2013 dataset.

5 Conclusion

This research improves upon existing methods for recognizing facial expressions by introducing a new feature selection idea, which is implemented by using transfer learning with a Residual Neural Network. In this Residual Neural Network, different Masking Blocks are applied throughout Residual Layers to improve the optimal features selection method by using the PCA and t-SNE. The experimental results obtained using the CK+48 and JAFFE dataset proved that the proposed methods outperformed the state-of-the-art results and the most popular classification algorithms. The proposed method will be further developed with the goal of assessing the model's generalization using the largest existing classification dataset. To further boost network performance in vision tasks like classification and detection, we will investigate varying network parameters and attempt to reduce the number of model parameters required for these tasks. We intend to construct a whole system and put it through its paces in a public rehearsal setting.

Acknowledgements This research was supported by the Bio Medical Technology Development program of the National Research Foundation(NRF) funded by the Korean government (MSIT) (NRF-2019M3E5D1A02067961) and by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education(NRF-2018R1D1A3B05049058 NRF-2020R1A4A1019191).

Bibliography

- [1] Kwak, Nojun, and Chong-Ho Choi. "Input feature selection for classification problems." *IEEE transactions on neural networks* 13, no. 1 (2002).
- [2] Dash, Manoranjan, and Huan Liu. "Feature selection for classification." *Intelligent data analysis* 1, no. 1-4 (1997).
- [3] Bommert, Andrea, Xudong Sun, Bernd Bischl, Jorg Rahnenfuhrer, and Michel Lang. "Benchmark for filter methods for feature selection in high-dimensional classification data." *Computational Statistics Data Analysis* 143 (2020).
- [4] Nguyen, Hoai Bach, Bing Xue, Ivy Liu, and Mengjie Zhang. "Filter based backward elimination in wrapper based PSO for feature selection in classification." In *2014 IEEE congress on evolutionary computation (CEC)*, pp. 3111-3118. IEEE, 2014.
- [5] Mustaqeem, Anam, Syed Muhammad Anwar, Muhammad Majid, and Abdul Rashid Khan. "Wrapper method for feature selection to classify cardiac arrhythmia." In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 3656-3659. IEEE, 2017.
- [6] Zhang, Jixiong, Yanmei Xiong, and Shungeng Min. "A new hybrid filter/wrapper algorithm for feature selection in classification." *Analytica chimica acta* 1080 (2019).

- [7] Labani, Mahdieh, Parham Moradi, Fardin Ahmadizar, and Mahdi Jalili. "A novel multivariate filter method for feature selection in text classification problems." *Engineering Applications of Artificial Intelligence* 70 (2018).
- [8] R. Cowie et al., "Emotion recognition in human-computer interaction," *IEEE Signal Processing Magazine*, vol. 18, no. 1, pp. 32–80, 2001.
- [9] B. Parkinson and A. S. R. Manstead, "Current Emotion Research in Social Psychology: Thinking About Emotions and Other People," *Emotion Review*, vol. 7, no. 4, pp. 371–380, Jul. 2015.
- [10] S. F. Waterloo, S. E. Baumgartner, J. Peter, and P. M. Valkenburg, "Norms of online expressions of emotion: Comparing Facebook, Twitter, Instagram, and WhatsApp," *New Media Society*, vol. 20, no. 5, pp. 1813–1831, May 2017.
- [11] V. R. LeBlanc, M. M. McConnell, and S. D. Monteiro, "Predictable chaos: a review of the effects of emotions on attention, memory and decision making," *Advances in Health Sciences Education*, vol. 20, no. 1, pp. 265–282, Jun. 2014.
- [12] Y. Zhu, "Research on the Human-Computer Interaction Design in Mobile Phones," 2020 International Conference on Computing and Data Science (CDS), Aug. 2020.
- [13] N. Chervyakov, P. Lyakhov, D. Kaplun, D. Butusov, and N. Nagornov, "Analysis of the Quantization Noise in Discrete Wavelet Transform Filters for Image Processing," *Electronics*, vol. 7, no. 8, p. 135, Aug. 2018.
- [14] D. A. Pitaloka, A. Wulandari, T. Basaruddin, and D. Y. Liliana, "Enhancing CNN with Preprocessing Stage in Automatic Emotion Recognition," *Procedia Computer Science*, vol. 116, pp. 523–529, Jan. 2017.
- [15] S. Nigam, R. Singh, and A. K. Misra, "Efficient facial expression recognition using histogram of oriented gradients in wavelet domain," *Multimedia Tools and Applications*, vol. 77, no. 21, pp. 28725–28747, May 2018.
- [16] N. Deshpande and S. Ravishankar, "Face Detection and Recognition using Viola-Jones algorithm and Fusion of PCA and ANN," vol. 10, no. 5, pp. 1173–1189, 2017.
- [17] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep Feature Extraction and Classification of Hyperspectral Images Based on Convolutional Neural Networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 10, pp. 6232–6251, Oct. 2016.
- [18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," openaccess.thecvf.com, 2016.
- [19] Z.-S. Liu, W.-C. Siu, and J.-J. Huang, "Image super-resolution via weighted random forest," *IEEE Xplore*, Mar. 01, 2017.
- [20] B. Hasani and M. H. Mahoor, "Spatio-Temporal Facial Expression Recognition Using Convolutional Neural Networks and Conditional Random Fields," *IEEE Xplore*, May 01, 2017.
- [21] S. Minaee, M. Minaei, and A. Abdolrashidi, "Deep-Emotion: Facial Expression Recognition Using Attentional Convolutional Network," *Sensors*, vol. 21, no. 9, p. 3046, Apr. 2021.
- [22] L. Pham, T. H. Vu, and T. A. Tran, "Facial Expression Recognition Using Residual Masking Network," *IEEE Xplore*, Jan. 01, 2021.

- [23] L. Pu and L. Zhu, "Differential Residual Learning for Facial Expression Recognition," 2021 The 5th International Conference on Machine Learning and Soft Computing, Jan. 2021.
- [24] Goodfellow et al., "Challenges in Representation Learning: A Report on Three Machine Learning Contests," Neural Information Processing, pp. 117–124, 2013.
- [25] Y. S. Teo et al., "Benchmarking quantum tomography completeness and fidelity with machine learning," New Journal of Physics, vol. 23, no. 10, p. 103021, Oct. 2021.
- [26] E. Barsoum, C. Zhang, C. C. Ferrer, and Z. Zhang, "Training deep networks for facial expression recognition with crowd-sourced label distribution," Proceedings of the 18th ACM International Conference on Multimodal Interaction, Oct. 2016.
- [27] Wang, K., Peng, X., Yang, J., Meng, D. & Qiao, Y. Region Attention Networks for Pose and Occlusion Robust Facial Expression Recognition. *IEEE Transactions On Image Processing*. **29** pp. 4057-4069 (2019)
- [28] Siqueira, H., Magg, S. & Wermter, S. Efficient Facial Feature Learning with Wide Ensemble-based Convolutional Neural Networks. *ArXiv*. [abs/2001.06338](https://arxiv.org/abs/2001.06338) (2020)
- [29] Zhou, H., Meng, D., Zhang, Y., Peng, X., Du, J., Wang, K. & Qiao, Y. Exploring Emotion Features and Fusion Strategies for Audio-Video Emotion Recognition. *2019 International Conference On Multimodal Interaction*. (2019)
- [30] Happy, S. & Routray, A. Automatic facial expression recognition using features of salient facial patches. *IEEE Transactions On Affective Computing*. **6** pp. 1-12 (2015)
- [31] Fard, A. & Mahoor, M. Ad-Corre: Adaptive Correlation-Based Loss for Facial Expression Recognition in the Wild. *IEEE Access*. pp. 1-1 (2022)
- [32] Farzaneh, A. & Qi, X. Facial Expression Recognition in the Wild via Deep Attentive Center Loss. *Proceedings Of The IEEE/CVF Winter Conference On Applications Of Computer Vision (WACV)*. pp. 2402-2411 (2021,1)
- [33] Khaireddin, Y. & Chen, Z. Facial Emotion Recognition: State of the Art Performance on FER2013. *CoRR*. [abs/2105.03588](https://arxiv.org/abs/2105.03588) (2021), <https://arxiv.org/abs/2105.03588>
- [34] Savchenko, A., Savchenko, L. & Makarov, I. Classifying emotions and engagement in online learning based on a single facial expression recognition neural network. *IEEE Transactions On Affective Computing*. **13**, 2132-2143 (2022)
- [35] Bodapati, J., Srilakshmi, U. & Veeranjaneyulu, N. FERNet: a deep CNN architecture for facial expression recognition in the wild. *Journal Of The Institution Of Engineers (India): Series B*. **103**, 439-448 (2022)
- [36] Oguine, O., Oguine, K., Bisallah, H. & Ofuani, D. Hybrid Facial Expression Recognition (FER2013) Model for Real-Time Emotion Classification and Prediction. *ArXiv Preprint ArXiv:2206.09509*. (2022)
- [37] Shaik, N. & Cherukuri, T. Visual attention based composite dense neural network for facial expression recognition. *Journal Of Ambient Intelligence And Humanized Computing*. pp. 1-14 (2022)

- [38] Boughida, A., Kouahla, M. & Lafifi, Y. A novel approach for facial expression recognition based on Gabor filters and genetic algorithm. *Evolving Systems*. **13**, 331-345 (2022)
- [39] Qi, Y., Zhou, C. & Chen, Y. NA-Resnet: neighbor block and optimized attention module for global-local feature extraction in facial expression recognition. *Multimedia Tools And Applications*. pp. 1-19 (2022)
- [40] Abdulsattar, N. & Hussain, M. Facial Expression Recognition using Transfer Learning and Fine-tuning Strategies: A Comparative Study. *2022 International Conference On Computer Science And Software Engineering (CSASE)*. pp. 101-106 (2022)
- [41] Minaee, S. & Abdolrashidi, A. Deep-Emotion: Facial Expression Recognition Using Attentional Convolutional Network. *Sensors (Basel, Switzerland)*. **21** (2019)
- [42] Aouayeb, M., Hamidouche, W., Soladié, C., Kpalma, K. & Séguier, R. Learning Vision Transformer with Squeeze and Excitation for Facial Expression Recognition. *ArXiv*. **abs/2107.03107** (2021)
- [43] Barsoum, E., Zhang, C., Canton-Ferrer, C. & Zhang, Z. Training deep networks for facial expression recognition with crowd-sourced label distribution. *Proceedings Of The 18th ACM International Conference On Multimodal Interaction*. (2016)
- [44] Khaireddin, Y. & Chen, Z. Facial Emotion Recognition: State of the Art Performance on FER2013. *ArXiv*. **abs/2105.03588** (2021)
- [45] Pham, L., Vu, T. & Tran, T. Facial Expression Recognition Using Residual Masking Network. *2020 25th International Conference On Pattern Recognition (ICPR)*. pp. 4513-4519 (2021)

Monitoring Students' Classroom Attention on Digital Platform

Hirotoshi IBE and Hiromasa NAKATANI

International Professional University of Technology in Nagoya, 450-0002 Japan
 {ibe.hiro, nakatain.hiro}@iput.ac.jp

Abstract. Most of the learning was shifted from traditional classes held physically to remote and online classes held virtually with the rise of COVID-19. Given the high degree of freedom and efficiency for the participants, online learning is expected to continue to grow post-pandemic. However, there is a noticeable difference between traditional teaching and online teaching. Though students' attention is always the key to keep the quality of learning, it is difficult even for experienced teachers to judge their attention when conducting online teaching. Teachers should always watch the online screen during the session, and also hard to utilize non-visual senses to assess the engagement. Therefore, we propose using a computer vision technique and develop a system that supports the teacher by gauging students' attention level. In this paper, we apply the technique to a Zoom class session and present preliminary results to demonstrate the feasibility of the proposed method.

Keywords: Online Class, Attention Level, Face Detection

1 Introduction

In online teaching, teachers need more efforts to recognize the states of their students that are sitting at home in front of computer. They must always observe numbers of students simultaneously on the screen during the session. With the traditional teaching, teachers could gauge students' attention by students' facial expressions, postures, and utilizing senses we all have. With online teaching, it requires considerable experience understanding the students' attention on limited information based on sound and visual. In addition, most educational institutions have limited online teaching experiences so far.

Although it is difficult for humans to assess students' attitude only over the screen, it is much easier for computers to assess when conducting online teaching. In addition, with the progress in computer vision technology, machines can enhance the teacher's task to conduct students' assessment. Researchers have developed various methods for recognizing facial expressions [1] and behaviors [2, 3] of students in the real classroom scene.

However, the scenes they have dealt with are only from traditional physical classes, but not from online classes. So, we need to develop such procedures that are specific

to online classes. We develop a method to support a teacher by gauging students' attention level. In this paper, we apply it to a Zoom class session and present preliminary results to demonstrate the feasibility of the proposed method.

2 Method

Assuming conducting an online teaching case, we build a system that monitors students' attention utilizing digital information via online meeting platform. Students are sitting in front of their web camera, and a teacher can see their faces on his computer screen. The system monitors students' face, and judges students' attention to be waning if their face often leaves the screen.

The procedure is as follows:

- Step 1. Feed the picture from the camera into face detection function.
- Step 2. If a face is detected, then show the result on the screen. Then, go back to step 1.
- Step 3. If no face is detected, then calculate the ratio of times when a face was detected for a certain period time. If the number is less than a priori given threshold, then notify the warning on the screen. Then, go to step 1.

At step1, we use Viola-Jones method [4] based on Haar-Like features for face detection. We use OpenCV 4.7.0 and a file of haarcascade_frontalface_alt2.xml for training frontal face images [5].

At step 2, letting n be the current frame number of the image, we set

$$d_n = 1 \text{ if a face is detected by the system, and } d_n = 0 \text{ otherwise.}$$

At step 3, we calculate the following ratio r_n , that is the ratio of times when a face was detected in the last n_0 frames:

$$r_n = \frac{1}{n_0} \sum_{i=0}^{n_0-1} d_{n-i} .$$

Then, if

$d_n = 1$ and $r_n < r_\theta$ (r_θ : a priori given threshold),
the system displays a caution on the screen.

3 Experimental Results

In this section we demonstrate experimentally the performance of the proposed method. We have collected our dataset from our Zoom class sessions. The video sequences are recorded at 25 frame per second and each frame is 1280×720 pixels in size. Figure 1 shows an example of one frame, where 25 images are captured and laid out in 256×144 pixels each.

We set the parameters in this experiment as follows: $n_0 = 750$, i.e. it takes 30 seconds to measure the ratio that a face is showing on the screen; and $r_\theta = 0.6$, i.e. if a face is not showing in more than 18 seconds out of 30 seconds, a student's attention is judged as being waning.



Fig. 1. Example of online class image

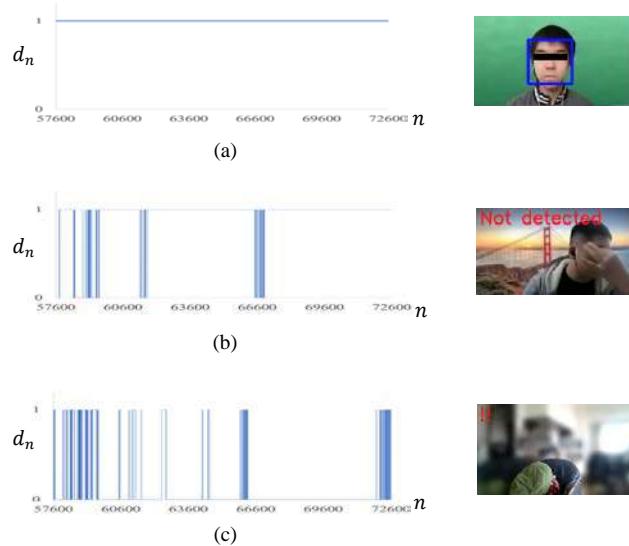


Fig. 2. Changes of students' attention

Figure 2 shows examples of the resulting responses from the face detection function. Three graphs on the left show whether or not a face is showing at frame n . Student of (a) was stable with little motion and his face was detected during that period. Student of (b) occasionally moved his head aside or covered his face, and then his face was not detected. Since student of (c) left his position, the system displayed a warning.

4 Conclusion and discussion

This research applies computer vision technology and builds an environment that supports online classes by realizing the function of recognizing students' attention. In this paper, we applied the proposed technique to a Zoom class session and presented preliminary results to demonstrate the feasibility of the proposed method.

There still remain several problems for the system to measure students' attention accurately. For example, when a student moves their head even for taking notes or solving quizzes, the system could judge their attention level was low. At present, the system cannot tell whether a person is taking a nap or not. To solve such problems, the system needs to recognize meaning of human motions. Those are left for future work.

Our ultimate goal is to enhance education utilizing new opportunities that was created by the accelerating shift to the online teaching, and we are hoping that there will be increasing support in this field.

References

1. Canedo, D., Trifan, A., Neves, A. J. R.: Monitoring students' attention in a classroom through Computer Vision. In: Proceedings on 16th International Workshops of Practical Applications of Agents, Multi-Agent Systems (PAAMS 2018), pp. 371–378 (2018).
2. Abdallah, T. B., Elleuch, I., Guermazi, R.: Student behavior recognition in classroom using deep transfer learning with VCG-16. Procedia Computer Science 192, pp. 951–960 (2021).
3. Zheng, Z., Liang, G., Luo, H., Yin, H.: Attention assessment based on multi-view classroom behavior recognition. IET Computer Vision - Wiley Online Library, <https://onlinelibrary.wiley.com/doi/full/10.1049/cv2.12146> (2022).
4. Viola, P., Jones, M.: Rapid Object Detection using a Boosted Cascade of Simple Features, In: Proceedings of 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR2001), I-511-I-518, pp.8-14 (2001).
5. <https://github.com/opencv/opencv/tree/master/data/haarcascades>, last accessed 2023/01/14.

Patent Image Retrieval Using Cross-entropy-based Metric Learning

Kotaro Higuchi, Yuma Honbu, and Keiji Yanai

Department of Informatics, The University of Electro-Communications
1-5-1 Chofugaoka, Chofu-shi, Tokyo, 182-8585, Japan
`{higuchi-k,honbu-y,yanai}@mm.inf.uec.ac.jp`

Abstract. Intellectual property work covers a wide range of areas. In particular, prior art literature searching in the patent field requires finding documents that can be used to determine novelty and inventive steps from a vast amount of past literature. Concerning this search practice, research and development of a drawing search technology that directly searches drawings, and essential information about inventions, has long been desired. However, patent drawings are described as black-and-white abstract drawings, and their modal characteristics are very different from those of natural images, so they have yet to be explored. This study achieved higher accuracy than the previous one by introducing InfoNCE and ArcFace in the DeepPatent dataset instead of the conventional Triplet. In addition, we developed an application that enables users to search for patent drawings using any images. Our architecture can be applied to patent drawings and many other modal-like drawings, such as mechanical drawings, design patents, trademarks, diagrams, and sketches.

Keywords: Patent Image Retrieval · Metric Learning · Search Application

1 Introduction

The patent field has developed a combination of natural language processing that can easily interact with textual information and patent classification information. High-quality search has been a long-standing challenge in patent practice since patent search requires a person skilled in the technical field, both in making queries and in assigning classifications.

Since the advent of ResNet [6], research on image recognition of natural images has made dramatic progress. However, the development of patent drawing retrieval, which is described as an abstract drawing, has been challenging, and no de facto method or system has yet to emerge [1].

In this study, we develop a search application based on metric learning, which has rapidly developed in recent years, and demonstrate patent drawing retrieval. In particular, for metric learning, we employ cross-entropy-based methods such as InfoNCE [16] and ArcFace [3] instead of the conventional Triplet loss.

Briefly, our main contributions are to:

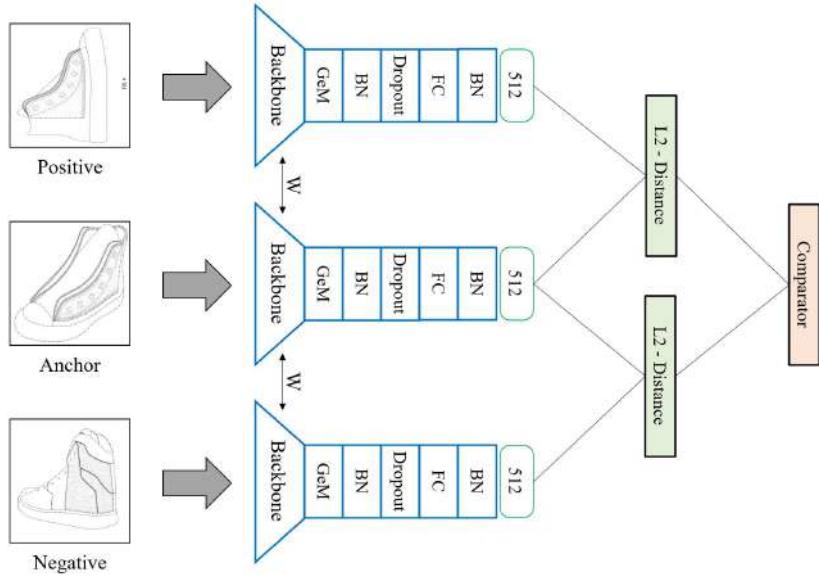


Fig. 1: The proposed architecture. Positive and Anchor are from USD0811075S1; Negative is from USD0811070S1. Both diagrams are cited from the drawings included in DeepPatent [11].

- achieve higher accuracy than previous papers by using cross-entropy-based metric learning methods on a dataset of patent drawings, which has been challenging to achieve in the past.
- develop an application that can search patent drawings using the proposed model (see Figure 1). There has yet to be a successful example of such an application in the past.

2 Related Works

Conventional Patent Searches and Issues Conventionally, patent searches have been performed using various search tools such as EspaceNet [4], making full use of text or patent classification. An example of an operation screen is shown in Figure 2.

However, conventional patent searches are based on the following two assumptions: (1) in the past, the annotator has assigned an appropriate classification to the patent, and (2) the searcher has learned appropriate terms (textual queries) that capture the essence of the patent.

In other words, both the person who assigns the classification and the person who performs the search need to be familiar with the technical field. In addition, the accuracy of patent classification is less than 100% due to the nature of cutting-edge technology being applied. Furthermore, the assignment of clas-

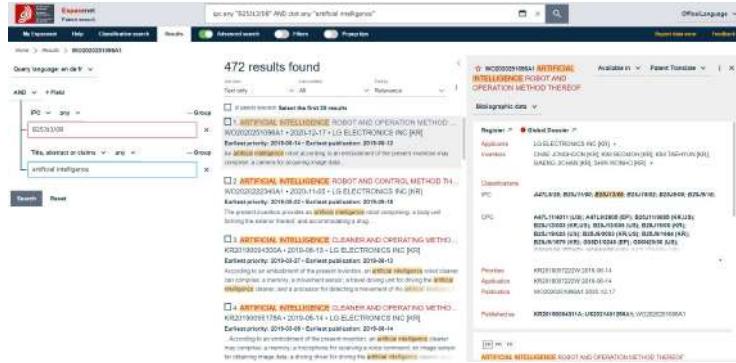


Fig. 2: EspaceNet. Users can perform text-based searches using the bibliography of the invention, the name of the invention, and the patent classification.

sifications and the creation of queries can sometimes be challenging, even for intellectual property specialists.

In conventional patent search, there is a need to search for shapes that are difficult to be expressed in words. However, there are many cases where such needs are challenging to be solved by text queries, and people concerned have to visually check several thousand to several tens of thousands of patent drawings.

Research in computer vision Because of the situation mentioned above, some research has been conducted on patent drawing retrieval based on deep learning techniques [9] [18]. However, the network in this study performed a primary task of estimating eight types of International Patent Classification (IPC, from A to H) and an auxiliary task of classifying nine types of drawings. Therefore, the image representation was strongly related to the existing patent classification, and the problem of omission of classification remained.

In addition, since the same IPC label is assigned to all drawings in the same application, there is an inherent issue of acquiring a sparse image representation. Kucer et al. [11] showed that patent image retrieval is possible by distance learning using ResNet50 [6] as the backbone, together with the DeepPatent dataset described below. However, there was still room for proof-of-concept for whether the model obtained in the paper could be developed as an actual application. Triplet depends on the sample selection in the batch; therefore, it is unpredictable whether a high accuracy can be obtained. For this reason, in this study, we experimented with a cross-entropy-based method using Triplet as a baseline.

Significance of drawings in patent practice Patent drawings are essential in practice [14], and in Japan, there are many cases in which patent drawings are the focus of infringement judgments. There are also precedents indicating that the composition (e.g. shape) can be read from the drawings [8]. Considering the above, the importance of drawing searches in patent practice is high.

3 Methods

We use the architecture shown in Figure 1 for a patent image retrieval system as a baseline for the DeepPatent dataset [11]. The Triplet network [2] [7] consists of three instances (with shared parameters) of the same forward propagation network. The system takes three samples from a batch, the network computes two distances between Anchor and Positive, and between Anchor and Negative. It calculates distances and adds a margin of m to update the model for using the three samples simultaneously.

Triplet method, however, has a drawback that it can handle one positive pair and one negative pair at the same time. This makes it difficult to use negative pairs effectively which exist much more than positive pairs in general. To resolve this problem, in this work, we use cross-entropy-based methods which can train many negative pairs at the same time. As cross-entropy-based methods, we employ InfoNCE [16] and ArcFace [3].

3.1 InfoNCE

InfoNCE [16] is one of the most popular methods used in self-supervised learning. Unlike Triplet, InfoNCE uses many Anchor-Negative pairs for each Anchor-Positive pair in sampling within a batch. The loss function L_i of infoNCE is shown below.

$$L_i = -\log \frac{e^{q \cdot k_+ / \tau}}{e^{q \cdot k_+ / \tau} + \sum_{i=0}^K e^{q \cdot k_i / \tau}}$$

3.2 ArcFace

ArcFace [3] achieves metric learning that increases the variance between classes by adding the following to the Softmax Cross-Entropy loss function used in the classification problem: 1) normalization of weights and features, and 2) a margin for the correct class. The loss function L_a of ArcFace is show below.

$$L_a = -\log \frac{e^{s \cos(\theta_{y_i} + m)}}{e^{s \cos(\theta_{y_i} + m)} + \sum_{j=1, j \neq y_i}^N e^{s \cos \theta_j}}$$

4 Experiments

4.1 DeepPatent Dataset

Kucer et al. [11] published a dataset containing more than 350,000 U.S. Design Patents in the public domain. The dataset is available from the project's Google Drive, and each drawing is assigned a publication number and a drawing number. Table 1 shows the details, which consist of 45,000 cases, divided into 70% as train, 15% as test, and 15% as validation.

The published DeepPatent dataset is not subject to copyright restrictions and is in the public domain, as stated by the U.S. Patent and Trademark Office (USPTO) [20].

Based on the above, we adopted the DeepPatent dataset as our benchmark for patent image retrieval.

Table 1: DeepPatent dataset

DeepPatent	figures	classes
Train	254,787	33,364
Test	38,834	6,927
Validation	44,815	5,888

4.2 Evaluation index

To evaluate the retrieval system, we used mAP score. mAP is the average precision APs from computing each query. Note that the AccuracyCalculator of the Pytorch Metric Learning [15] was used to implement the accuracy measurement.

4.3 Comparison to baseline

In the architecture shown in Figure 1, we compare baseline and metric learning with the proposed methods. Specifically, we made experiments with the baseline ResNet+Triplet, the EffNet [19]+Triplet, the EffNet+infoNCE [16], and the EffNet+ArcFace [3]. We used Hard Negative Mining [21] for Triplet training.

Table 2: mAP Score Comparison

Method	mAP
DeepPatent baseline [11]	0.379
EffNet + InfoNCE [16]	0.447
EffNet + ArcFace [3]	0.622

5 Results and Discussions

5.1 Comparison to baseline

Table 2 shows the results of the comparison between the baseline and the proposed method for the architecture shown in Figure 1.

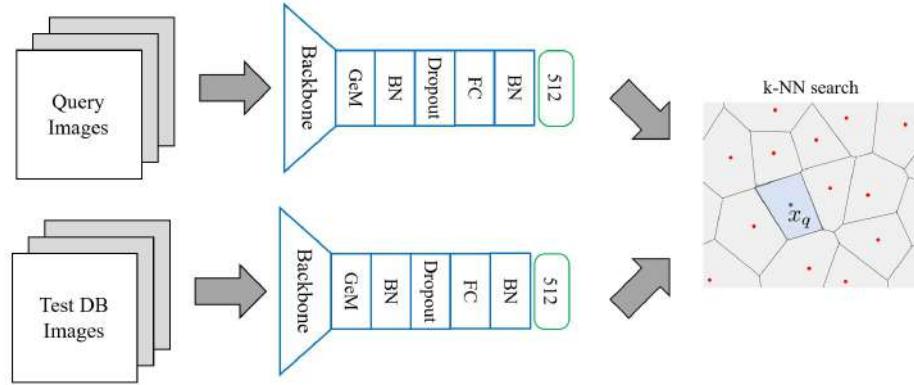


Fig. 3: Inference and retrieval architecture. The system performs indexing by Faiss.

The proposed method achieves a score of mAP=0.622. This score is higher than the conventional method because we select a more appropriate backbone and method for the patent drawing task (more complex than ResNet in this task). In addition, Triplet uses only the m parameter of margin for metric learning, which is easy to control, and the amount of VRAM memory usage is relatively small (within a few GB at batch size=512). Furthermore, the infoNCE can achieve a higher score because the Negative sample is larger than the Triplet. With the ArcFace method, we achieve a value that stands out above.

5.2 Search Application and Practitioners' Opinions

We implemented a patent drawing search application using the indexing method, Faiss [10], as shown in Figure 3. The index file size was about 80 MB, which was large enough to deploy on a server or in the cloud.

We implemented the system as a Web application using Streamlit on an on-premise Ubuntu server. Figure 4 shows the application screenshots.

Afterward, we received feedback from practitioners on the usability of the application. Overall, the feedback was favorable, and we have received positive comments from the users, who look forward to future research development. The following comments are parts of the feedback we received.

- The operation is more straightforward than expected. However, the accuracy still needs to be up to practical use.
- The search is easy because it is possible to search only by image query.

6 Conclusion

With the proposed method, we achieved an mAP score higher than that of previous papers and also realized a patent drawing retrieval application. For

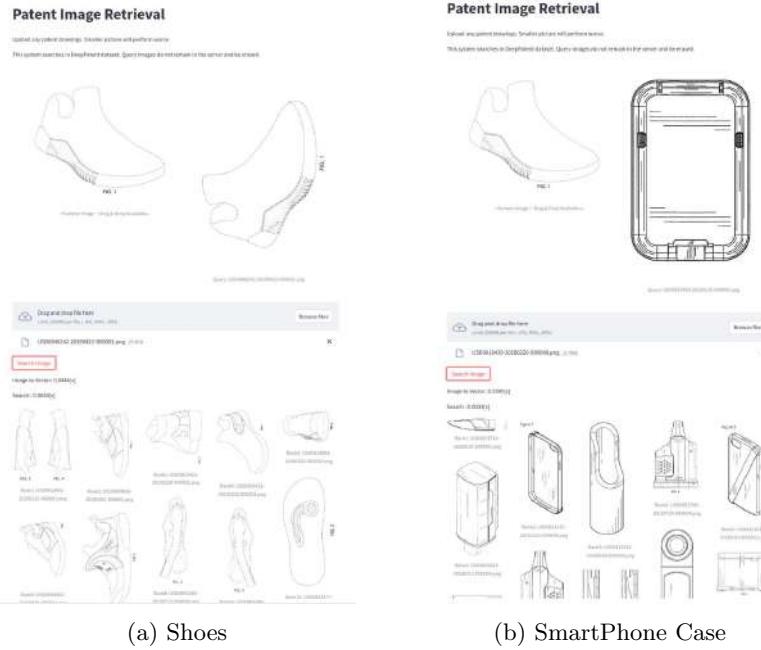


Fig. 4: The screenshots of the developed patent image search system. Users can search for patent drawings by dragging and dropping any images. (a) Search results for a shoe. The search result is good and hits similar patent drawings of shoes. (b) Search results for a smartphone case. The search results are questionable, and future improvements are needed in search accuracy.

many years, numerous issues in the patent field have required the appearance of patent drawings, and our proposal will help solve these issues.

This research has revealed two significant possibilities: first, deep metric learning is possible on patent drawing datasets, and second, by combining a machine learning framework and trained models, a patent drawing application can be developed.

Future work Future work is to improve the mAP score and search accuracy. Where feedback from practitioners has revealed that the accuracy could be better at the level of practice, we recognize that there is a large room for growth in accuracy, as there are various other methods for backbone and metric learning. Examples include SwinTransformer [13] [12], and other losses [5] [17].

The above results show that patent image retrieval has many unexplored areas. A wide range of research is expected to be conducted in the future because it can be applied not only to patent drawings but also to similar modal drawings, sketches, trademarks, designs, utility models, mechanical drawings, and flowcharts.

References

1. Bhattacharai, M., Oyen, D., Castorena, J., Yang, L., Wohlberg, B.: Diagram image retrieval using sketch-based deep learning and transfer learning. In: CVPR. pp. 174–175 (2020)
2. Cao, R., Zhang, Q., Zhu, J., Li, Q., Li, Q., Liu, B., Qiu, G.: Enhancing remote sensing image retrieval using a triplet deep metric learning network. International Journal of Remote Sensing **41**(2), 740–751 (2020)
3. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: CVPR. pp. 4685–4694 (2019)
4. European Patent Office: Espacenet (2023), <https://worldwide.espacenet.com/>
5. Gordo, A., Almazan, J., Revaud, J., Larlus, D.: End-to-end learning of deep visual representations for image retrieval. International Journal of Computer Vision **124**(2), 237–254 (2017)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385 (2015)
7. Hoffer, E., Ailon, N.: Deep metric learning using triplet network. In: Similarity-Based Pattern Recognition. pp. 84–92. Springer International Publishing (2015)
8. Intellectual Property High Court: Case number 2014(ke)10274 (2014), https://www.ip.courts.go.jp/app/hanrei_jp/detail?id=4185
9. Jiang, S., Luo, J., Pava, G., Hu, J., Magee, C.: A convolutional neural network-based patent image retrieval method for design ideation. In: IDETC-CIE. vol. 83983. American Society of Mechanical Engineers (2020)
10. Johnson, J., Douze, M., Jegou, H.: Billion-scale similarity search with gpus. Proc. of IEEE Transactions on Big Data **7**(03), 535–547 (2021)
11. Kucer, M., Oyen, D., Castorena, J., Wu, J.: Deppatent: Large scale patent drawing recognition and retrieval. In: WACV. pp. 2309–2318 (January 2022)
12. Liu, Z., *et al.*: Swin transformer: Hierarchical vision transformer using shifted windows. ICCV pp. 9992–10002 (2021)
13. Liu, Z., *et al.*: Swin transformer v2: Scaling up capacity and resolution. In: CVPR. pp. 12009–12019 (2022)
14. Ministry of Justice: Japanese law translation, patent act, article70(2) (2023), <https://www.japaneselawtranslation.go.jp/en/laws/view/4097>
15. Musgrave, K., Belongie, S., Lim, S.: A metric learning reality check. In: ECCV. Lecture Notes in Computer Science, vol. 12370, pp. 681–699. Springer (2020)
16. Oord, A., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
17. Revaud, J., Almazan, J., Rezende, R., de Souza, C.: Learning with average precision: Training image retrieval with a listwise loss. In: ICCV (2019)
18. Shalaby, W., Zadrozy, W.: Patent retrieval: A literature review. Knowledge and Information Systems **61**, 631–660 (2019)
19. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: ICML. pp. 6105–6114 (2019)
20. United States Patent and Trademark Office: Terms of use for uspto websites (2023), <https://www.uspto.gov/terms-use-uspto-websites>
21. Xuan, H., Stylianou, A., Liu, X., Pless, R.: Hard negative examples are hard, but useful. In: ECCV. pp. 126–142. Springer (2020)

Pre-training of Pneumonia Classifier for Chest CT images using Fractal Database

Yuken Yoshioka¹, Daichi Ikefuji², Tomokazu Funatsu¹, Takashi Nagaoka³,
 Takenori Kozuka⁴, Mitsutaka Nemoto⁵, Takahiro Yamada⁶,
 Yuichi Kimura^{7,8}, Kazunari Ishii^{4,6}, and Hitoshi Habe^{7,8}

¹ Graduate School of Science and Engineering, Kindai University, Japan

² Department of Informatics, Faculty of Science and Engineering, Kindai University, Japan

³ Department of Computational Systems Biology, Faculty of Biology-Oriented Science and Technology, Kindai University, Japan

⁴ Department of Radiology, Faculty of Medicine, Kindai University, Japan

⁵ Department of Biomedical Engineering, Faculty of Biology-Oriented Science and Technology, Kindai University, Japan

⁶ Institute of Advanced Clinical Medicine, Kindai University Hospital, Japan

⁷ Department of Informatics, Faculty of Informatics, Kindai University, Japan

⁸ Cyber Informatics Research Institute, Kindai University, Japan

Abstract. Recently, the number of images for pre-training of deep learning models has been increasing, and large-scale data sets contain inappropriate images such as ethically inappropriate images, copyright infringement, and labeling errors. A method to solve these is by using a fractal database that generates images by mathematical formulas without using natural images. Our goal is to show that the classification accuracy obtained by pre-training with fractal images is comparable to natural images. In the experiments, we compare the performance on the tasks to classify CT images of COVID-19 pneumonia and regular pneumonia.

Keywords: Image classification · Fractal Image · Pre-training · CNN · CT image

1 Introduction

Imaging diagnosis by doctors is essential for the detection of disease. However, if image diagnosis is performed by doctors, even doctors sometimes fail to detect lesions from large amounts of data. This would cause a problem of delays in treatment. Image recognition using deep learning would enable us to find diseases early.

When we perform image recognition using deep learning(DL), first, we perform pre-training using a large, public dataset, and then the DL model is updated by fine-tuning using the data of the application field. In recent years, the required number of datasets for pre-training has been increasing to improve the accuracy of DL, and the creation cost of datasets, including image collection

and annotation, has also been increasing. Furthermore, there are unignorable problems in the public dataset, such as labeling errors[1].

To overcome such problems, Kataoka et al. propose to generate a large amount of training data based on a mathematical model and to use the data for pre-training of DL[2]. Kataoka et al.[2] have shown that pre-training using a formula-driven database for natural image classification tasks yields results that are comparable to those of conventional large-scale natural image databases such as ImageNet. In this study, we use a fractal image database, a mathematical model-based database. And we demonstrate its effectiveness for CT image classification tasks. In the following discussion, we call the fractal image database Fractal Database(FDB).

2 Generating a Fractal Database

In this section, we describe a method for generating fractal Database[2].

2.1 Generating Fractal Images

IFS (Iterated Function System) is a model for generating the point set $X = \{x_1, x_2, \dots, x_K\}$. IFS is defined by a set of transformations $w_i : X \rightarrow X$ and corresponding probabilities p_i in the complete metric space X (Eq. (1)). where N represents the number of pairs (w_i, p_i) .

$$IFS = \{X; w_1, w_2, \dots, w_N; p_1, p_2, \dots, p_N\}. \quad (1)$$

Using the IFS, a fractal $S = \{x_t\}_{t=0}^{\infty} \in X$ is constructed by the random iteration algorithm.

The transformation w is defined as:

$$w_i(x; \theta_i) = \begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix} x + \begin{bmatrix} e_i \\ f_i \end{bmatrix}. \quad (2)$$

$\theta_i = (a_i, b_i, c_i, d_i, e_i, f_i)$ is 6 parameters for rotation and shifting. It generates a point set to depict a fractal image in the two-dimensional Euclidean space. p_i is the probability for selecting the transformation w , and is calculated as follows:

$$p_i = \frac{|detA_i|}{\sum_{i=1}^N |detA_i|}, \quad A_i = \begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix}. \quad (3)$$

The following procedure obtains the point set X that depicts a fractal image.

- I Determine the number N of pairs (w_i, p_i) from a discrete uniform distribution on $[2, 3, \dots, 8]$.
- II The parameters a_i, b_i, \dots, f_i in the transformation w_i in Eq. (2) are randomly selected from the continuous uniform distribution on $[-1, 1]$ and the probability p_i is determined by Eq. (3).

- III Set the initial value to $x_0 = (0, 0)^T$. Choose the transformation w_i from w_1, \dots, w_N with according to the probability p_i . Apply it to the position x_{t-1} to obtain the new position x_t .
- IV By repeating (III) K times, we obtain a point set $X = \{x_1, x_2, \dots, x_K\}$.

When we draw a fractal image from the point set, we randomly choose a value of 0 to 127. And draw the values at the 3×3 region centered on the obtained point set $X = \{x_1, x_2, \dots, x_K\}$. Get the maximum and minimum x and y coordinates of the point set X to normalize the size into a pre-defined image size. Finally, calculate the pixel filling rate. If it is above a certain threshold value, it is used as an image for pre-training.

We assign the same category label for images generated by the same IFS. IFS is characterized by a set of parameters and their corresponding probabilities, expressed as $\theta = \{(w_i, p_i)\}_{i=1}^N$. When we create a data set of n classes, we repeat steps (I)-(IV) at n times. In this research, we use FDB having 1000 classes. The procedure for generating more data for each class is described in the next section.

2.2 Data Augmentation

In section 2.1, we generated one image per class. These classes are related by fractal parameters a_i, b_i, \dots, f_i . Because this is insufficient for training, it is necessary to augment the data for each class, as is often performed in the standard training procedure. We apply the two types of augmentation methods in this paper shown in Figure 1. The first one is based on the original paper[2]. The other one is the image-based augmentation conventionally performed in the standard training process[3].

Formula-based Augmentation As used in [2], we increase the data by the three types of augmentation methods as follows.

- I Changing the parameter set of IFS.
- II Rotate generated fractal images.
- III Changing 3×3 patch patterns for drawing fractal images.

We carry out (I) 25 times, (II) 4 times, and (III) 10 times. Finally, we generate the database containing 1000 images per class. We call this database FDB1k-1k.

Image-based Augmentation The data augmentation described in Section 2.2 draws fractal images for each parameter of IFS. This process is quite time-consuming. Instead of this, we apply simple data augmentation for training[3]. As is the standard data augmentation process, we apply affine transformations and color transformations using random numbers to each fractal image using the RandomAffine function and ColorJitter function in PyTorch libraries. This would enable us to obtain sufficient number of training data for pre-training as is the previous section. We call the augmented data One-instance Fractal Data Base (OFDB). More specifically, we use OFDB1k and OFDB10k data sets which contains 1000 classes and 10000 classes respectively.

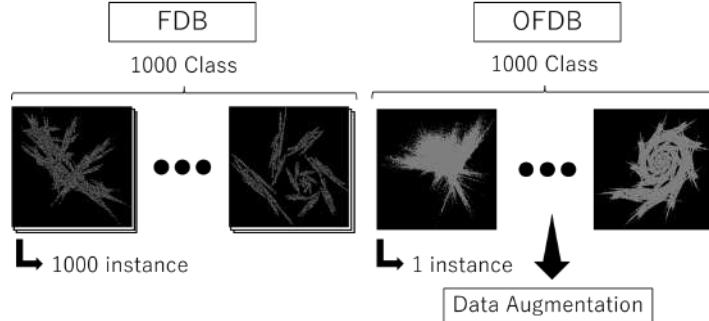


Fig. 1: FDB and OFDB data sets

3 Training Pneumonia Discriminator

This Section describes the procedure for training pneumonia discriminator using FDB1k-1k, OFDB1k, and OFDB10k.

3.1 Pre-training

For comparing the pre-training results of fractal data sets, we use ImageNet Version 2 (IN_V2). IN_V2 is a pre-trained model of 1000 class classification using ResNet50 as the backbone network. Table 1 shows the pre-training conditions. As mentioned in Section 2.2, FDB1k-1k need to generate point set for each images, i.e. 1000 sets for each class. On the other hand, OFDB uses PyTorch function for data augmentation, which realizes efficient computation. Therefore, it has the advantage of short learning time.

Table 1: Pre-training conditions

	FDB1k-1k	OFDB1k	OFDB10k
Hours	168	8	27
Epochs	90	9000	900
Class	1000		10000
Network		ResNet-50	
Batch_size		64	

3.2 Fine-tuning

Next, we perform fine-tuning to adapt the pre-trained models to the pneumonia data. In this paper, we aim the distinguishes between two classes of regular pneumonia and COVID-19 pneumonia. We set the fine-tuning conditions: the epochs number of 90, the batch size of 64, and the learning rate of 0.01.

4 Experiment

We conduct the experiment to examine the effectiveness of the fractal database.

4.1 Experimental Setup

Figure 2 show the two examples of CT images for experiments. (a) is normal pneumonia and (b) is COVID-19 pneumonia. These images were provided by Kindai University Hospital under the permission of the Ethics Committee, Kindai University Faculty of Medicine. Using a data set containing these images, we perform a two-class classification of COVID-19 pneumonia or other. Because COVID-19 pneumonia has frosted-glass shadows in CT images, those features would be an essential cue for classification. 10-fold cross-validation is used for verification. The number of images used is 7092 for regular pneumonia and 6621 for COVID-19 pneumonia.

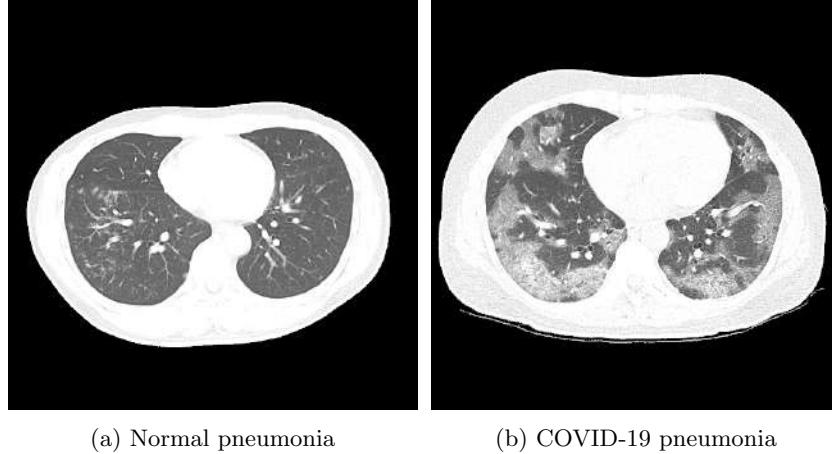


Fig. 2: Example of normal pneumonia and COVID-19 pneumonia

4.2 Results

Table 1 and Figure 3 show the results of 10-fold cross-validation. In Table 1, we show the average of the highest accuracy and recall rate for the test data among 90 trials. The accuracy of IN_V2 was the highest at 90.84%, followed by OFDB1k at 86.43%, OFDB10k at 85.27%, and FDB1k-1k at 85.53%. However, we have to note that decreasing the false-negative, i.e., missing COVID-19 cases, is crucial for diagnosis. From this point of view, the OFDB1k and OFDB10k yield slightly better results than IN_V2. Figure 3 shows the box plot of accuracy and recall during 10-fold cross-validation. This also shows that there is almost no difference between the recall values for OFDB1k, OFDB10k, and IN_V2, while the accuracy of IN_V2 is higher than others.

Table 2: 10-fold cross-validation(%)

	FDB1k-1k	OFDB1k	OFDB10k	IN_V2
Accuracy	85.53	86.43	85.27	90.84
Recall	91.54	95.51	95.26	95.48

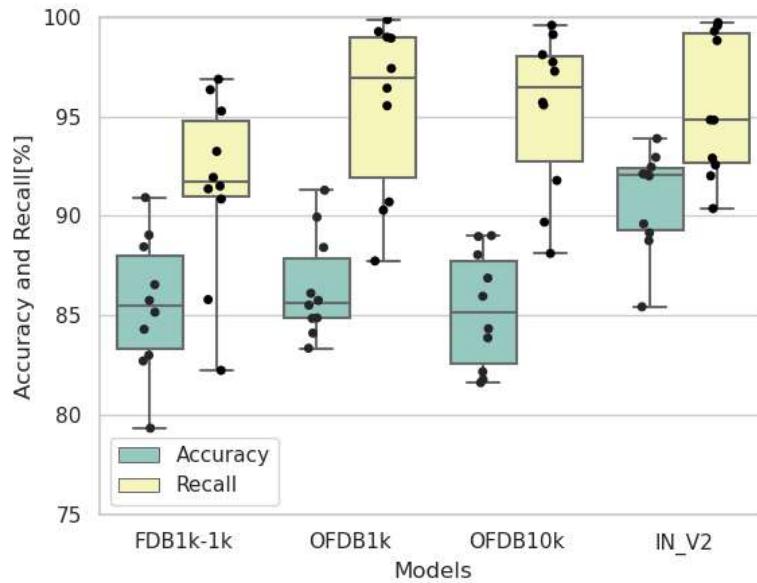


Fig. 3: Box plot of accuracy and recall during 10-fold cross-validation

4.3 Visual Explanations for Decisions

Figure 4 shows the visualization of the reasons behind predictions by LIME[5]. This shows the parts that significantly contribute to the classification. While the green color represents the part that has a higher predicted probability of the correct label, the red color shows a lower probability. From this figures, we can see that the models trained by the fractal database, i.e., FDB and OFDB, focus on the larger regions in the lung area.

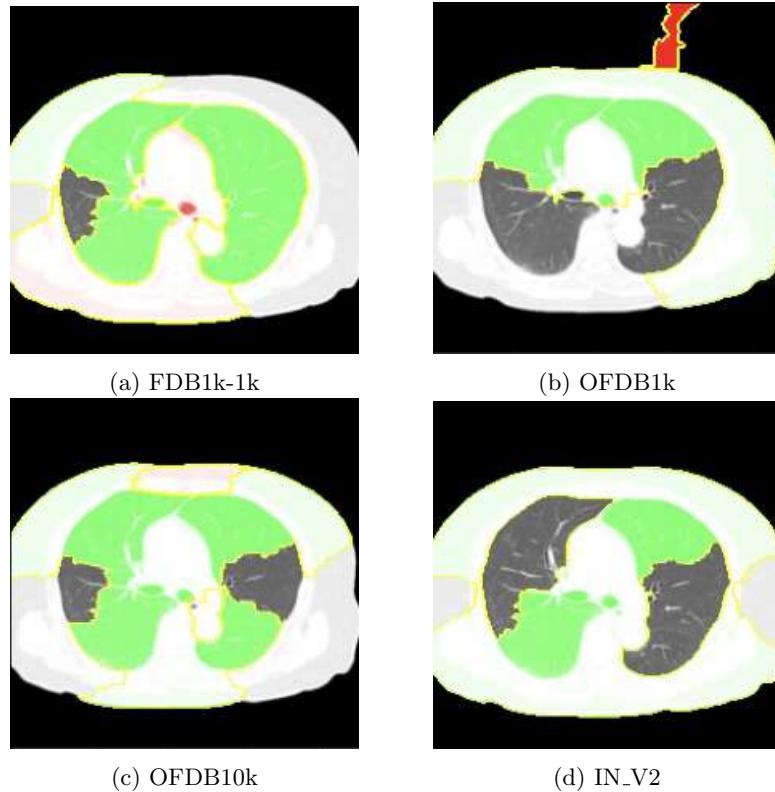


Fig. 4: Visualization of attention area by LIME

5 Conclusion

Although the accuracy of classification using FDB and OFDB are lower than IN_V2, the recall values of OFDB are almost the same as the IN_V2 model. OFDB has the advantage that this has less computational costs. It needs only one single image for each class. Data augmentation can be efficiently conducted by PyTorch libraries.

From section 4.3, FDB and OFDB focus more correctly in the lungs. This implies that the fractal-based database has the potential to achieve higher performance if we have sufficient data for fine-tuning because the model trained by FDB and OFDB focuses on appropriate regions.

In future work, we will investigate the appropriate model to generate a database for pre-training. In the current implementation, we use the model same as the original paper. It is expected that higher performance can be obtained by exploring database generation models that are suitable for the data in the application domain. Additionally, there are still the room for optimizing the hyper-parameters for pre-training, which helps to obtain more reliable models.

This work was partly supported by JSPS KAKENHI Grant Number JP21H05302 and All-Kindai University support project against COVID-19 provided by Kindai University.

References

1. Lucas Beyer, Olivier J. Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. 2020. “Are We Done with ImageNet?” arXiv [cs.CV]. arXiv. <http://arxiv.org/abs/2006.07159>.
2. Hirokatsu Kataoka, Kazushige Okayasu, Asato Matsumoto, Eisuke Yamagata, Ryosuke Yamada, Nakamasa Inoue, Akio Nakamura, and Yutaka Satoh. 2022. “Pre-Training Without Natural Images.” International Journal of Computer Vision 130 (4): 990–1007.
3. Ryo Nakamura, Ryu Tadokoro, Hirokatsu Kataoka, Can visual features be learned with only one generated image per category?, MIRU2022(in Japanese).
4. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. “Deep Residual Learning for Image Recognition.” In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 0:770–78.
5. Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. ““Why Should I Trust You?”: Explaining the Predictions of Any Classifier.” In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1135–44. KDD ’16. New York, NY, USA: Association for Computing Machinery.

Advanced Video Inpainting method using Residual Query Connection

Youngjun La¹[0000-0002-0426-7939] and Jong-Il Park¹[0000-0003-1000-4067]

Hanyang University, Department of Computer Science, Republic of Korea
 {yjla,jipark}@hanyang.ac.kr

Abstract. In this paper, we propose a method to enhance the performance of video inpainting using Residual Query Connection. Video inpainting is a method of visually filling in the damaged regions in each video frame. Recently, Transformer-based video inpainting has shown remarkable performance, however, it has a drawback of slow model speed when using video as input. A simple way to increase the model speed is to decrease the number of Transformer blocks used. Our proposed method adds local feature information by performing Multi-head Self-Attention within the residual connection of the Query. As evidenced by 2% and 2.6% improvement in LPIPS and FID, our approach is shown to slightly improve degraded performance by reducing the number of Transformer blocks. Additionally, we measure performance per 100K iterations and conduct qualitative evaluations, demonstrating the effectiveness of our proposed method.

Keywords: Computer Vision · Deep Learning · Video Inpainting · Vision Transformer.

1 Introduction

Video inpainting is a method of filling in damaged areas in each video frame in a realistic manner. Video inpainting is widely used in various fields such as video completion and object removal [11]. Although image inpainting has greatly advanced, performing inpainting on each frame can result in inconsistent inpainting results [12, 17, 18]. Recently, deep learning-based video inpainting research has been actively conducted to achieve temporal consistency [3, 19, 6, 7, 5, 2].

Among them, the optical flow-based video inpainting model applies pixel propagation to achieve temporal consistency [16, 3]. However, this pixel propagation requires manual operations such as Poisson blending, making it slow. To solve this, E²FGVI (End-to-End Framework for Flow-Guided Video Inpainting) modifies the model structure to enable feature propagation instead of pixel propagation, resulting in improved video inpainting speed [9]. However, E²FGVI still has a slow speed of 0.095 seconds per frame on an NVIDIA GeForce RTX 2080ti GPU. A simple way to increase the speed of this model is to reduce the number of Transformer blocks used in the model. By reducing the number of

blocks from 8 to 4, the model speed increases to 0.068 seconds per frame. However, this approach reduces the depth of the model and therefore the inpainting performance decreases.

This paper proposes Residual Query Connection to improve the degraded inpainting performance. The proposed method adds local feature information by connecting the extracted query from the local window to the residuals during the Multi-head Self-Attention of E²FGVI. This approach can improve the inpainting performance that has been degraded by reducing the number of Transformer blocks in the existing model. Furthermore, by conducting qualitative evaluations, it is shown that the proposed method is effective in diverse damaged video scenes generated by the DAVIS dataset, thus it can be applied in various fields.

The sequence of this paper is as follows: In Section 2, E²FGVI model structure, which is used as the basic model of the proposed method, is introduced. In Section 3, the proposed method, Residual Query Connection, is explained. In Section 4, the experimental method is described, in Section 5, the results and analysis of the experiment are presented. Finally, in Section 6, the conclusion and future research directions of this paper are presented.

2 E²FGVI

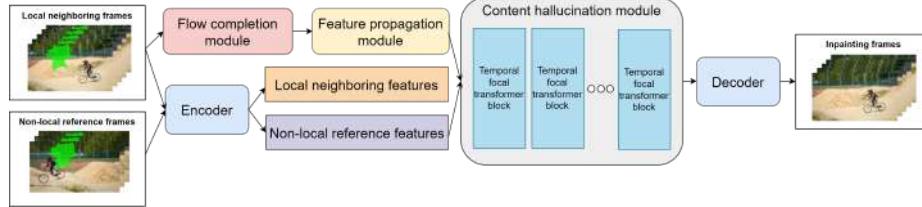


Fig. 1. E²FGVI Model Structure

E²FGVI (End-to-End Framework for Flow-Guided Video Inpainting) is an end-to-end video inpainting model in which modules are performed in the order of flow completion, feature propagation, and content hallucination [9]. The model structure is shown in Figure 1.

The components of the proposed model are as follows. First, the encoder is used to lower the resolution of the features for computational efficiency. Next, the flow completion module is used to complete the optical flow of the local neighboring frames. The flow completion module uses the lightweight model SPyNet [14]. Then, the feature propagation module is used to bidirectionally propagate the local neighboring features based on the completed optical flow. The feature propagation module uses deformable convolution [21]. Next, the multi-layer temporal focal transformer is used to perform content hallucination. The temporal focal transformer is an extension of the Focal Transformer for

videos, which improves the interaction between local and global features [10]. Finally, the decoder is used to increase the size of the features and output the video inpainting results. This model serves as the base model for the proposed method.

3 Residual Query Connection

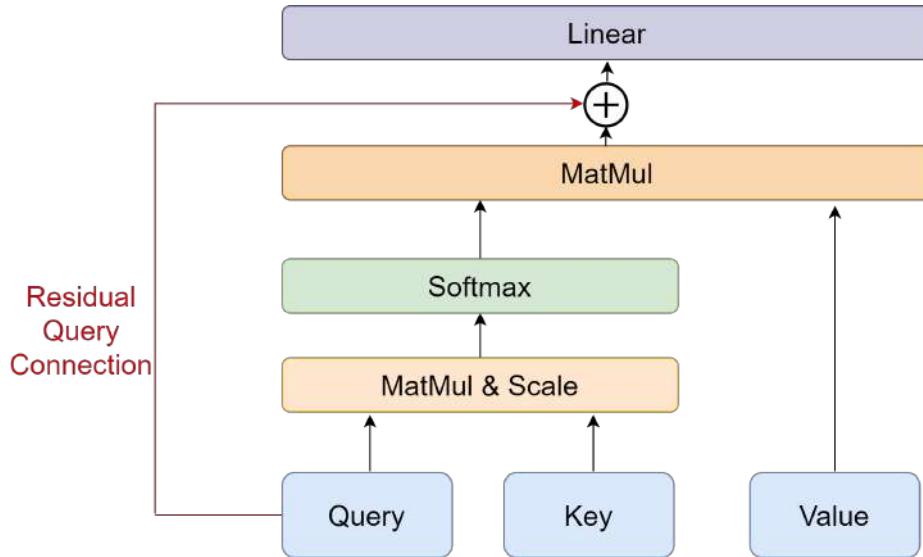


Fig. 2. Multi-head Self-Attention with Residual Query Connection

Residual Query Connection is a method for connecting the residual of the Query in Multi-head Self-Attention of the base model's temporal focal transformer. As shown in Figure 2, the proposed method adds Residual Query Connection to the existing Multi-head Self-Attention. This method is applied by modifying the Residual pooling connection used in previous studies [8]. The difference between ours and previous is that the previous research was used in pooling attention to combine local features before attention, while this research used it in window attention. Specifically, our method connects the entire Query, not the Query that has been pooled. This method has the effect of adding additional local feature information of the Query. This improves the inpainting performance for damaged videos that require more local feature information.

4 Experimental method

In this paper, we conduct the training in the same way as previous studies that used the E²FGVI model [9]. The difference is that our proposed method is added,

and to increase the model speed, we reduce the number of Transformer blocks used in the previous study from 8 to 4. The training data used is the video object segmentation dataset YouTube-VOS with 3471 samples [15]. Commonly, the batch size is 7, and the learning rate starts at 0.0001 and is reduced by a factor of 10 when the number of iterations reaches 400K. The optimizer used is Adam with $\beta_1=0$ and $\beta_2=0.99$. The total number of iterations is 500K, and we use 4 NVIDIA GeForce RTX 2080ti GPUs for training. The model's input consists of 5 consecutive local neighboring frames and 3 randomly extracted global reference frames. The resolution of each video frame is 432x240. The loss function used is the same as previous studies, which is the optical flow loss function L_{flow} and the loss function L_{rec} that calculates the pixel-wise difference between the inpainted video \hat{Y} and the original video Y using the L1 distance [9]. Additionally, we use the loss function L_D of the discriminator that focuses on local and global features between neighboring frames and the adversarial loss function L_{adv} of the generator [1]. The weights of L_{flow} , L_{rec} , and L_{adv} were 1, 1, and 10^{-2} , respectively.

$$L_{flow} = \sum_{t=1}^{T-1} \left\| \hat{F}_{t \rightarrow t+1} - F_{t \rightarrow t+1} \right\|_1 + \sum_{t=2}^T \left\| \hat{F}_{t \rightarrow t-1} - F_{t \rightarrow t-1} \right\|_1 \quad (1)$$

$$L_{rec} = \left\| \hat{Y} - Y \right\|_1 \quad (2)$$

$$L_D = E_{x \sim P_Y(x)}[ReLU(1 - D(x))] + E_{z \sim P_{\hat{Y}}(z)}[ReLU(1 + D(z))] \quad (3)$$

$$L_{adv} = -E_{z \sim P_{\hat{Y}}(x)}[D(z)] \quad (4)$$

5 Experimental results

The experiments and evaluations are conducted in the same way as previous studies using the E²FGVI model [9]. The experimental data used is 50 DAVIS video object segmentation dataset and 50 randomly masked frames for video damage [13]. Also the global reference frames extracted uniformly at sampling rate of 10, and 5 preceding and succeeding local neighboring frames are used as inputs of the model. For quantitative metrics, we use LPIPS (Learned Perceptual Image Patch Similarity) and FID (Frechet Inception Distance) that evaluate the visual quality of the video from a human perspective [20, 4]. Additionally, we also use E_{warp} (flow warping error) to measure temporal consistency [6]. LPIPS uses a pre-trained VGG or AlexNet to extract the features of the original video and the video that is being inpainted and compare their differences. We use AlexNet for feature extraction. And FID calculates the distribution of features from the original video and the inpainted video and measures the Wasserstein distance between the two distributions. The mean and covariance of each distribution are calculated by a pre-trained Inception-V3 model.

Experimental results, as shown in Table 1, reveal that decreasing the number of Transformer blocks from 8 to 4 results in a degradation in overall performance, but increases the model speed by approximately 40%. And when applying the

proposed method to the model using 4 blocks, LPIPS and FID improve by approximately 2% and 2.6%, respectively. E_{warp} remains similar when using 4 blocks. This shows that the proposed method is an efficient way to improve performance while maintaining the model speed. To demonstrate the validity of the proposed method, we measure LPIPS and FID per 100K iterations as shown in Table 2, 3. It is observed that LPIPS and FID are improved at all measured iterations, which indicates that the proposed method is effective in improving the video inpainting performance.

As shown in Figure 3, qualitative evaluations are also conducted. After applying the proposed method, it is observed that the shape of lines and patterns are improved, resulting in reduced distortion of the video and visually plausible results. This shows that the proposed method is effective in various damaged video scenes, and can be applied in various fields.

Table 1. Quantitative results of Video Inpainting

Dataset	DAVIS				
	Model	LPIPS ↓	FID ↓	$E_{warp} \times 10^{-2} \downarrow$	Runtime (s/frame)
8 Blocks	0.0402	11.907	0.1315	0.095	
4 Blocks	0.0445	12.958	0.1340	0.068	
Our	0.0436	12.628	0.1336	0.068	

Table 2. LPIPS measurement per 100K iterations

Dataset	DAVIS				
	LPIPS ↓				
Model \ Iterations	100K	200K	300K	400K	500K
4 Blocks	0.0593	0.0516	0.0503	0.0484	0.0445
Our	0.0560	0.0494	0.0481	0.0458	0.0436

Table 3. FID measurement per 100K iterations

Dataset	DAVIS				
	FID ↓				
Model \ Iterations	100K	200K	300K	400K	500K
4 Blocks	17.143	15.137	14.525	14.114	12.958
Our	16.465	14.619	13.967	13.028	12.628

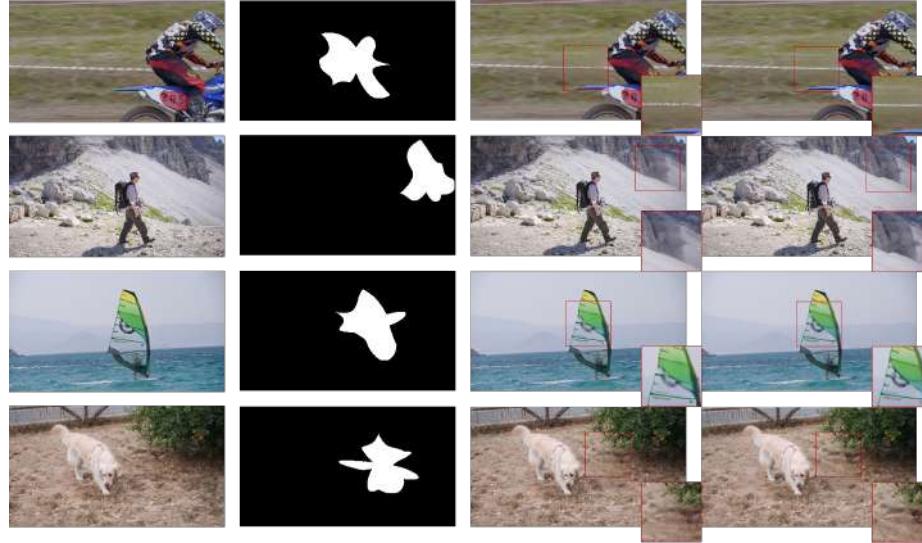


Fig. 3. Qualitative results of Video Inpainting. (First Column) Original Image, (Second Column) Mask for damaging the image, (Third Column) Inpainting results before using proposed method, (Forth Column) Video Inpainting results after using proposed method.

6 Conclusion

In this paper, we propose a method to improve video inpainting performance compared to the existing model E²FGVI. The proposed method adds query to the Multi-head Self-Attention result to add local feature information. This method shows that it is an effective way to inpaint various natural images by improving the degraded performance as the number of Transformer blocks is reduced. However, the model speed is still not fast. If we further reduce the model size to increase the model speed, the model performance significantly degrades, leading to the need for a lightweight video inpainting model. In future research, we will focus on models that can operate in mobile environments while maintaining video inpainting performance.

References

1. Chang, Y.L., Liu, Z.Y., Lee, K.Y., Hsu, W.: Free-form video inpainting with 3d gated convolution and temporal patchgan. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9066–9075 (2019)
2. Chang, Y.L., Liu, Z.Y., Lee, K.Y., Hsu, W.: Learnable gated temporal shift module for deep video inpainting. arXiv preprint arXiv:1907.01131 (2019)
3. Gao, C., Saraf, A., Huang, J.B., Kopf, J.: Flow-edge guided video completion. In: European Conference on Computer Vision. pp. 713–729. Springer (2020)

4. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems* **30** (2017)
5. Kim, D., Woo, S., Lee, J.Y., Kweon, I.S.: Deep video inpainting. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 5792–5801 (2019)
6. Lai, W.S., Huang, J.B., Wang, O., Shechtman, E., Yumer, E., Yang, M.H.: Learning blind video temporal consistency. In: *Proceedings of the European conference on computer vision (ECCV)*. pp. 170–185 (2018)
7. Lee, S., Oh, S.W., Won, D., Kim, S.J.: Copy-and-paste networks for deep video inpainting. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 4413–4421 (2019)
8. Li, Y., Wu, C.Y., Fan, H., Mangalam, K., Xiong, B., Malik, J., Feichtenhofer, C.: Mvitzv2: Improved multiscale vision transformers for classification and detection. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 4804–4814 (2022)
9. Li, Z., Lu, C.Z., Qin, J., Guo, C.L., Cheng, M.M.: Towards an end-to-end framework for flow-guided video inpainting. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 17562–17571 (2022)
10. Liu, R., Deng, H., Huang, Y., Shi, X., Lu, L., Sun, W., Wang, X., Dai, J., Li, H.: Fuseformer: Fusing fine-grained information in transformers for video inpainting. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 14040–14049 (2021)
11. Oh, S.W., Lee, S., Lee, J.Y., Kim, S.J.: Onion-peel networks for deep video completion. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 4403–4412 (2019)
12. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 2536–2544 (2016)
13. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 724–732 (2016)
14. Ranjan, A., Black, M.J.: Optical flow estimation using a spatial pyramid network. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 4161–4170 (2017)
15. Xu, N., Yang, L., Fan, Y., Yang, J., Yue, D., Liang, Y., Price, B., Cohen, S., Huang, T.: Youtube-vos: Sequence-to-sequence video object segmentation. In: *Proceedings of the European conference on computer vision (ECCV)*. pp. 585–601 (2018)
16. Xu, R., Li, X., Zhou, B., Loy, C.C.: Deep flow-guided video inpainting. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 3723–3732 (2019)
17. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Generative image inpainting with contextual attention. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 5505–5514 (2018)
18. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Free-form image inpainting with gated convolution. In: *Proceedings of the IEEE/CVF international conference on computer vision*. pp. 4471–4480 (2019)
19. Zeng, Y., Fu, J., Chao, H.: Learning joint spatial-temporal transformations for video inpainting. In: *European Conference on Computer Vision*. pp. 528–543. Springer (2020)

8 Youngjun La and Jong-Il Park

20. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
21. Zhu, X., Hu, H., Lin, S., Dai, J.: Deformable convnets v2: More deformable, better results. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9308–9316 (2019)

Utilization of Temporal Detection Consistency for Improving the Multi-Object Tracking

Abhyudaya Singh Tak^[0000-0002-2504-2253] and Soon Ki Jung^[0000-0003-0239-6785]

Kyungpook National University, Daegu, South Korea
 {abhyudaya, skjung}@knu.ac.kr

Abstract. Various different improvements have been made in the field of multi-object tracking for separate tracker and detector algorithms. It is often shown that the joint trackers are more robust than the separate trackers but this research is conducted to show how much the separate tracker models have been improved and how much more they can still be improved. Various different research has been done to identify the current problems in the separate trackers. An algorithm to improve the detection using the past frame data for the separate tracker and detector scenario, and an algorithm to evaluate the consistency of the detector using previous and current frame tracker data have been developed. The results show that the algorithms seem to have improved the performance of not only the detection but also the entire real-time tracking model. In this paper, the results are visualized and performance is evaluated using the benchmark MOT datasets.

Keywords: Computer vision · Multi-object tracking · Bounding box.

1 Introduction

Multi-Object Tracking (MOT) plays an essential part in video understanding. It aims to discover and track all specific classes of objects frame by frame. The tracking-by-detection paradigm [4], [17], [5] has dominated the multi-object tracking task a number of times. It performs by spotting various objects per frame and formulates the MOT problem as a data association task. Benefiting from high-performing object detection models, these tracking-by-detection styles have gained favor due to their excellent performance. Still, these approaches generally bear multiple calculations having performance-costly building blocks, similar to a detector and an embedding model. To crack this problem, several recent methodologies integrate the detector and embedding model into a unified architecture. Also, training these joint detection and tracking models appears to produce better results compared to the separate one [14]. Therefore, these styles (joint trackers) achieve similar or indeed better preciseness compared to tracking-by-detection ones (separate trackers). The success of joint trackers has motivated experimenters to design unified tracking fabrics for colorful factors e.g., discovery, stir, embedding, and association models. But these have their

own problems such as competition among their own components and less open source material and data to train these models. That is the sole reason to choose the StrongSORT [9], which is the new improved version of the old DeepSORT [27] and OC-SORT [8] methods for use in this paper.

Extensive research was conducted to identify the current problems in the tracking by-detection paradigm and multi-object tracking field where some valid points were discovered. A problem in the separate tracker models was encountered where the detection model is entirely independent of the tracking task and is also not aware of the detected objects in the previous frames which can be a waste of the potential information that a detector can use to improve detection as well as the tracking performance, so an idea to build an algorithm for improving the results of the detection using the information that the detector has already come across from detection in the previous frame was identified. It can be a vast area to research but it has been narrowed down to two tasks, improving the detection in subsequent frames using the information of the previous frame as well as fixing multi-class bounding boxes for the same object using the same algorithm.

The evaluation of the multi-object trackers has also come a long way from where they were a few years ago. There are separate evaluation techniques for detectors as well as separate evaluation techniques for trackers but for this paper, we implemented an idea to find the consistency mistakes of the detector using the tracker information of the previous frame, and the explanation and results can be found in the later sections of this paper.

The main contributions of this paper are as follows:

- A method is presented to improve the detection provided by the detector by utilizing the previous-frame information as well as solving the issue of multi-class detection for the same object in the multi-object tracking for real-time applications.
- A new algorithm is developed to evaluate the detector with the help of a tracker that checks the bounding box size mistakes made by the detector in real-time.
- The method is evaluated by conducting many experiments on the MOT16 and MOT17 benchmark datasets to show how the method has improved the detection from the detector as well as the tracker's performance when used in real-time.

2 Related Work

There have been various developments in multi-object tracking and visual object tracking fields over the past couple of years. This paper explores problems in the tracking-by-detection paradigm, a sub-field in 2D multi-object tracking. There have also been developments in evaluating these models, be it developing new and improved metrics or the creation of new algorithms to test these models in a different manner. With this related works section, some sections and works

related to or close to the field this paper is based on are covered before we dive deep into our methodology and experimental results sections later in this paper.

2.1 Visual-Object Tracking and Datasets

To understand this tracking field more research was conducted in the field of visual-object tracking. A research work [24] covers up mostly every basic thing there is to know about the said field, from the introduction of the field to various available benchmark datasets for training and evaluating the visual object tracking models.

Visual Object Tracking is a task where the objective of the tracker is to locate or track a particular entity or object regardless of the challenges in all of the frames of a video, where the tracker is only given the location of that entity or object in the first frame. It is important to note that the tracker may have to overcome various challenges due to different factors affecting the video during the training and testing phases. Also, it is worth noting that the tracker might not perform the same on all the benchmark datasets currently available cause that hugely depends on how the tracker has been implemented, what training datasets were used, and what benchmark datasets were used to test its performance. Some trackers might perform well in a particular benchmark dataset but might not perform well in others. Some of the flagship benchmark datasets keep updating every year for a good performance evaluation, forcing developers to take a more generalized approach. It is a prime computer vision area and has been proven to be a fundamental concept for a lot of applications like sports [16], autonomous robots [21], self-driving vehicles [12], surgery [7], AR [1], etc.

The research work [2] revisited the challenges in visual object tracking and compared two of the most significant benchmark datasets, OTB and VOT, respectively. The aim was to get a better understanding of which protocols are preferred for what tracking objective and to ultimately compare the two datasets mentioned above. To check for the robustness of a particular tracker they introduced a new concept of mirror tracking which could help in identifying the overfitting scenarios.

2.2 3D Multi-Object Tracking

This area of tracking is also catching a lot of wind as could be seen in the recent development of this field. Authors are focusing on 3D bounding box tracking for better visualization and increased accuracy in tracking.

The research work [26] discusses a new accurate, simple, and real-time 3D MOT baseline based on SORT [4] (Simple Online and Real-Time Tracking) and, a new evaluation metric based on CLEAR [3] MOTA (Multi-Object Tracking Accuracy) and MOTP (Multi-Object Tracking Precision). They did the evaluation on the KITTI [13] dataset and achieve state-of-the-art performance on 3D MOT by creating a 3D extension to the official KITTI 2D MOT evaluation.

One research work for 3D MOT in the tracking-by-detection paradigm, Simple-Track [18] takes a very basic but effective approach of breaking down all the components of the tracking-by-detection model and analyzing each module to find out their failure cases or shortcomings and then proposing corresponding solutions for them followed by combining them into a simpler baseline model which achieves state-of-the-art results.

2.3 MOT Evaluation Metrics

We have worked with the CLEAR [3] MOT metric in this paper to the sub-metric CLEAR MOTA (Multi-Object Tracking Accuracy) and MOTA+ [23] for evaluation of the proposed algorithm.

$$MOTA+ = 1 - ((|FN| + |FP| + |IDSWS| + |IDTRs|)/|gtDet|). \quad (1)$$

The MOTA+ in Equation (1) introduces IDTRs (identity transfers) error ratio, which is the transpose of the IDSWS (identity switches). In short, it finds the error cases where the id assigned in the previous frame is transferred to a different object in the next frame.

3 Methodology

This section discusses the models that were chosen and used for the research, the problem statements explaining the problem, and solutions to those problems mentioned along with their flowcharts for better visual explanations. This section is further divided into several subsections for an organized explainable approach.

3.1 Detector

The detector model used in this paper is YOLO [19], [20], [6] specifically YOLOv5. The reason for choosing this version of YOLO over the more recently released ones is the fact that the main focus of this research is on real-time usage and version 5 of YOLO assists that goal because of its good accuracy with good real-time speeds to work with. YOLO, when released was a major breakthrough in the field of object detectors. Framing of the object detection by the model was done as a regression problem instead of re-purposing classifiers to perform the detection task. Although, it was noted that it needs a GPU to work under real-time conditions. Figure 1 shows how the data is stored in YOLOv5 after the prediction which will be used and explained later in this paper.

3.2 Tracker

Both DeepSORT [27] and StrongSORT [9] were researched while working on this paper. StrongSORT was chosen after thorough research because of the increased accuracy over the previously proposed DeepSORT method. Below are

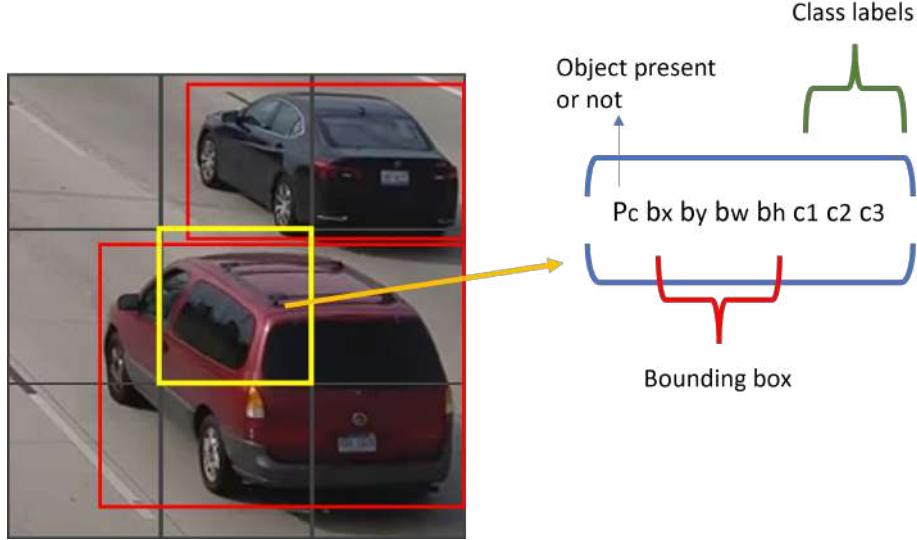


Fig. 1. Data after the YOLO model’s prediction.

some points mentioned stating the differences between the two methods and the slightly altered version of the StrongSORT that is used in this paper instead of the original one. The authors of StrongSORT claimed that DeepSORT underperforms because of its outdated techniques, rather than its tracking paradigm. They simply equip DeepSORT with advanced components, achieving SOTA on MOT17 and MOT20 benchmarks. Also, for our work, we changed the BoT feature extractor [15] used in the original work with a lightweight feature extractor OSNet [29] achieving almost similar mAP (mean average precision) on the person re-id benchmark dataset while having a significantly lower number of parameters as shown in Table 1. Some differences have been stated below between Deep-

Table 1. Feature extractor model comparison.

No.	Method	mAP	Parameters
1	BoT (Resnet backbone) [15]	85.9	25.6M (98MB)
2	OSNet [29]	84.9	2.2M
3	ShuffleNet [28]	65.0	2.2M
4	MobileNet V2 [22]	69.5	3.5M

SORT and the newer StrongSORT to show why choosing the latter over the former was a better choice:

- A stronger appearance feature extractor is used instead of a simple CNN.

- They replace the feature bank in DeepSORT with a feature updating strategy [25].
- Adopt ECC [11] for camera motion compensation.
- Change the vanilla Kalman filter to the NSA Kalman algorithm [10].
- Changed the cascading algorithm of DeepSORT with the vanilla global linear assignment.

The other tracker that was used to carry out the experiments is OC-SORT [8] which is the Observation-Centric SORT and as the name suggests is also based on the famous baseline SORT [4]. OC-SORT still remains simple, online, and real-time but significantly improves upon the SORT method. Their main contribution to developing OC-SORT is making it robust for tracking under occlusions and non-linear motion. Even while keeping the method simple as the baseline they achieve state-of-the-art results in some metrics at the time their paper was published. As per their research results, their method also falls short like it did in SORT when the video has low frame rates and when the object motion is fast but they still achieved great results without over-complicating the baseline and suggest that some authors have tried solving the problems mentioned above by using the center distance [30] or by adding appearance similarity [27] into their respective tracking models.

3.3 Improving Detection in Subsequent Frames Using the Previous Frame

An object detector while making its prediction on an image frame is generally unaware of the predictions it had made previously even if the previously predicted frames were part of a collection of frames from the same scene, which could be perceived as the loss of useful information.

Problem Statement.

- It was noted that since in the tracking-by-detection paradigm the tracker is dependent on the detector for all the detections in a particular frame, sometimes the detected bounding box could be missing from the final output even though that particular object was detected in the previous frame as shown in Figure 3. The reason for this particular issue was that the detector could not correctly identify the object and gave it a confidence score below the threshold, hence these detections were removed from the final output. A part of this paper explains the solution to this problem and has been explained further.
- The other issue was that more than one bounding box of different classes was sent by the detector for the same object in some frames as shown in Figure 2.

Since the detection sent by the detector are independent of the past frames sent by the same detector, it was a great opportunity to utilize the past frame information. These issues have been researched and solved using the same algorithm described in Figure 5 along with some visualizations to understand the working of the same.



Fig. 2. Same objects having two different detected classes. (Two boxes of different color shades depict two different classes).

Post Processing in YOLOv5. After the YOLO algorithm makes its predictions as shown in Figure 1, these detections go to a post-processing function for the final processing cause in the current state they are unusable. That post-processing function is responsible for filtering out multiple things:

- Unwanted detections.
- Detections that have confidence scores below the threshold.
- Overlapping/redundant detections.

Unwanted detections and detections that have confidence scores below the threshold are filtered out using a confidence threshold. Overlapping or redundant detections are filtered out using the Non-Max Suppression function. In the end, the final best detections are returned back from this function for passing to the tracker for tracking. To solve the issue of missing detections in the subsequent frames, this function had to be changed/updated using new techniques to make it aware of the various detections it is disregarding or filtering out to regain the missing detections in the frame T if the particular object was detected in the previous frame, frame T-1.

First, the candidates are selected in the function based on their object score using the confidence threshold as shown in Figure 4. Here it is made sure that most of the currently detected objects are not disregarded by setting a small object confidence threshold (1st) and removing the original object confidence check. Later in the function, object scores and class confidence scores are multiplied to get a final confidence score and are again checked against the confidence threshold (2nd), which is still required to take care of unwanted detections. Also, it is still needed to find the missing detections from these rejected values, which is done with our developed algorithm visualized in Figure 5.

The overlap between the final previous frame (T-1) detection tensor and the current frame (T) detection tensor is checked. With this overlap check, it is noted if any of the previous frame detections do not overlap (0% IoU) with the current

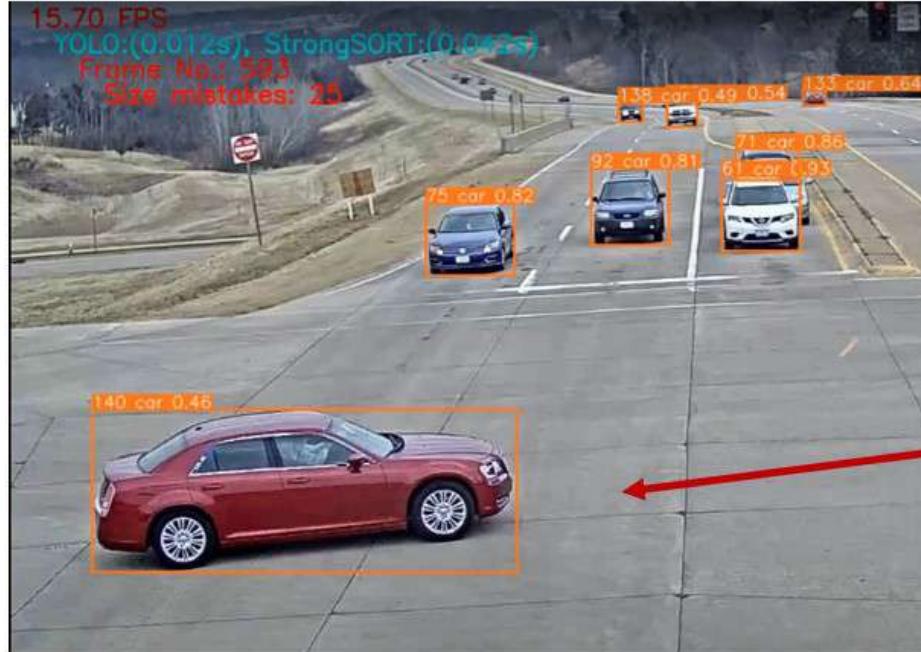
Frame 593: Conf- 0.46Frame 594:

Fig. 3. Missing detection in the subsequent frame even though the object was detected in the previous frame.

```
tensor([[False, False, False, ..., False, False, False]], device='cuda:0')
```

Fig. 4. Candidates Selection

frame unfiltered detections and then these detections are saved in a new tensor ‘N1’. Now all the rejected detections by the (2nd) threshold check in a new tensor ‘LCX’ are saved. After that, the overlap between tensors LCX and N1 is checked and the detections having some amount of overlap from the LCX tensor are saved in a new tensor ‘N2’. Then the N2 tensor is concatenated with the unfiltered detection tensor of the current frame. In the end, this unfiltered tensor is passed through the Non-Max Suppression function which removes redundant detection values. These values are passed to the tracker and also saved as previous frame final detections for the next iteration.

By using the first overlap check information between the final previous frame detections and current frame detections the detector’s mistake of passing two bounding boxes for the same object in the current frame due to them being of different classes is also corrected by using the previous frame detected class result of that object. It is also made sure not to correct this mistake while the object emerges from any side of the frame. The reason is if the object has not properly emerged into the frame the detector might make a mistake in incorrectly identifying it as something else and due to our algorithm that incorrect result might carry forward in future frames, for example, a small part of the front of the car might be identified as a motorcycle or a person which is undesired to be carried forwarded to the future frames.

One concern regarding this algorithm was the speed compromise that could arise from the use of this algorithm during the tracking task. Hence, the speed and performance are kept well preserved by the use of computation time-preserving techniques used throughout the algorithm. One such technique was the use of list comprehension instead of the use of for-loops since many of the tasks in the algorithm involved the need of creating lists/tensors, which made a huge difference in the speed performance. Also, only a single frame (previous frame) information was used for our algorithm because of the fact that as the number of objects and frame comparisons increase, the overall computation overhead is also significantly increased which ultimately decreases the speed of the tracking task which is unwanted for real-time applications. The results of the developed algorithm can be found later in this paper in the results section.

3.4 Algorithm to Evaluate Detector’s Consistency in Bounding Box Size Assignment

Problem Statement. It was noted that the detector could make a mistake in keeping the bounding box size consistent in the subsequent frames. As far as it is known, no other evaluation metric captures this inconsistency of size difference in the bounding boxes. The size difference is mainly noticed when the work is done with the real-time applications of these models.

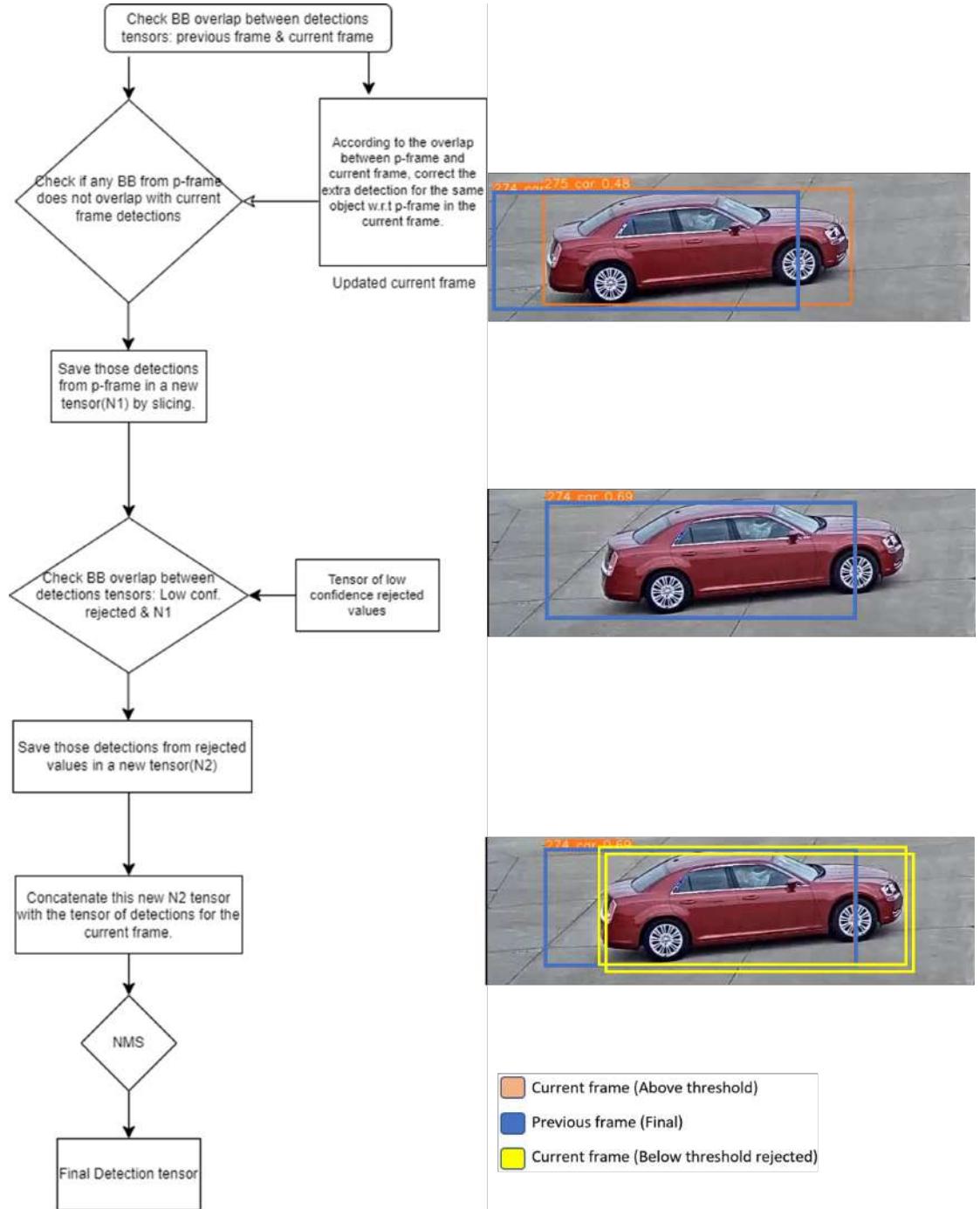


Fig. 5. Algorithm based on previous-frame to reduce the missing detection problem and to solve the multi-class detection for the same object problem.

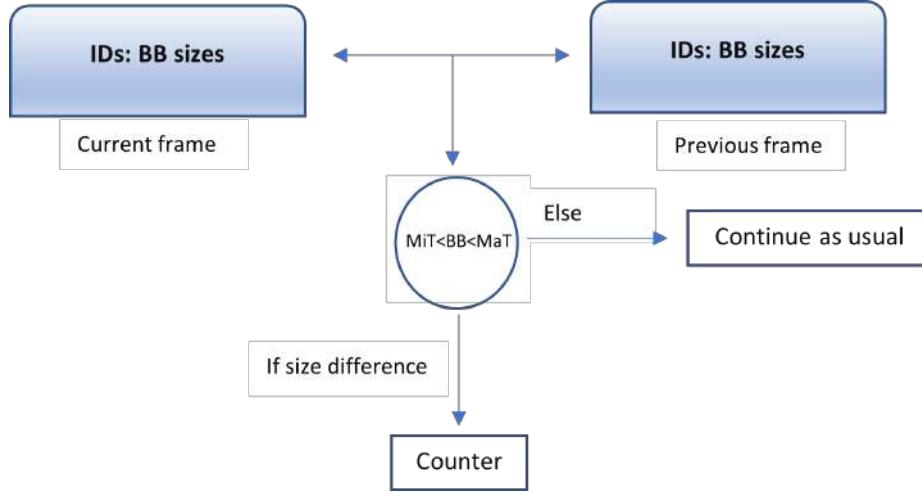


Fig. 6. Depicts the bounding box size difference identifier algorithm's working. (Note - MiT: Minimum Threshold, BB: Bounding Box, MaT: Maximum Threshold).

As already stated in the problem statement above that the reason for developing this algorithm is that it has not come to the observation if any research work has been conducted on checking how many times the detector made an inconsistency mistake while maintaining the bounding box sizes in the subsequent frames, especially in real-time situations with the help of a tracker. Currently, in this algorithm, visualized in Figure 6, the information in two frames, current frame T and previous frame T-1 are utilized. The ID information assigned and provided by the tracker for each and every detection is utilized and then the IDs and the corresponding width and height of each bounding box are stored for the comparison of frames T and T-1. A minimum and maximum threshold value are used which can be adjusted according to the scene by the user to measure the amount of size difference that occurs in subsequent frames. In the end, if the value exceeds or is beneath the set threshold then a counter counts how many times the mistake was made until the scene/video runs out in real-time. By this with the help of a tracker, a user can easily identify the performance and consistency of the detector in a real-time scenario without carrying out the evaluation of the detector separately. The result of the same can be seen later in the results section of this paper.

4 Experiments and Results

This section conveys the results from all of our developed algorithms with resulting real-time frames of a scene, tables, and various different generated data. To run and test our model, we utilize python with the deep learning framework

PyTorch and all computations are performed on a desktop with an Intel(R) Core(TM) i7-9700 CPU @ 3.00GHz and Nvidia Geforce RTX 2080 Super GPU.

4.1 Improved Tracking

Figure 7 shows the improved detection results on the same subsequent frames shown in Figure 3 after applying the algorithm proposed in this paper. In Figure 3 the detection box for the pointed object in frame 594 cannot be seen because the confidence of that object was below the threshold which was set as **0.45** and was not sent by the detector to the tracker, hence it was not displayed initially. But in Figure 7 it could be clearly seen that after applying the algorithm the detection box is visible even though the confidence of the pointed object is still below the threshold. Also, Figure 8 shows examples of other low-confidence bounding boxes during the inference after applying our developed algorithm. Figure 9 shows the improvement made using the algorithm into resolving the multi-class detections for the same object across subsequent frames as it was depicted in Figure 2. Table 2 and 3 show the improvement made by the algorithm in both MOTA/MOTA+ [23] score and True Positive matches using two different trackers and benchmark datasets. It also mentions Mostly Tracked (MT), Partially Tracked (PT), and Mostly Lost (ML) trajectory details before and after applying the algorithm. The improvements have been represented in the Tables using the green font for easy visualization of the performance boost.

Table 2. Evaluation results on MOT16 dataset.

Tracker	MOTA	MOTA+	True Positive	MT	PT	ML
StrongSORT (before)	62.374	61.271	3,515	10	12	3
StrongSORT (after)	64.040	62.936	3,586	11	12	2
OC-SORT (before)	61.080	60.263	3,437	8	14	3
OC-SORT (after)	62.431	61.271	3,510	8	14	2

MT: Mostly Tracked, PT: Partially Tracked, ML: Mostly Lost.

Table 3. Evaluation results on MOT17 dataset.

Tracker	MOTA	MOTA+	True Positive	MT	PT	ML
StrongSORT(before)	60.601	59.587	3,514	8	14	4
StrongSORT(after)	63.061	61.671	3,669	11	12	3
OC-SORT(before)	61.728	60.714	3,463	8	14	4
OC-SORT(after)	62.423	61.239	3,534	8	14	4

Since the main algorithm proposed in this paper is accommodated in the post-processing function of the YOLO model, a speed analysis of the overall tracking model was made in Table 4 with two different scenes having comparisons

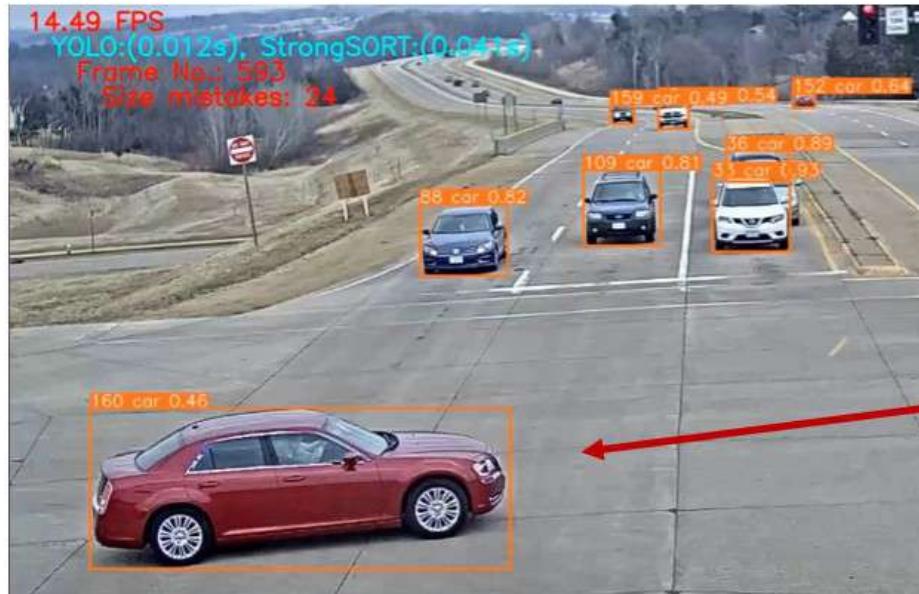
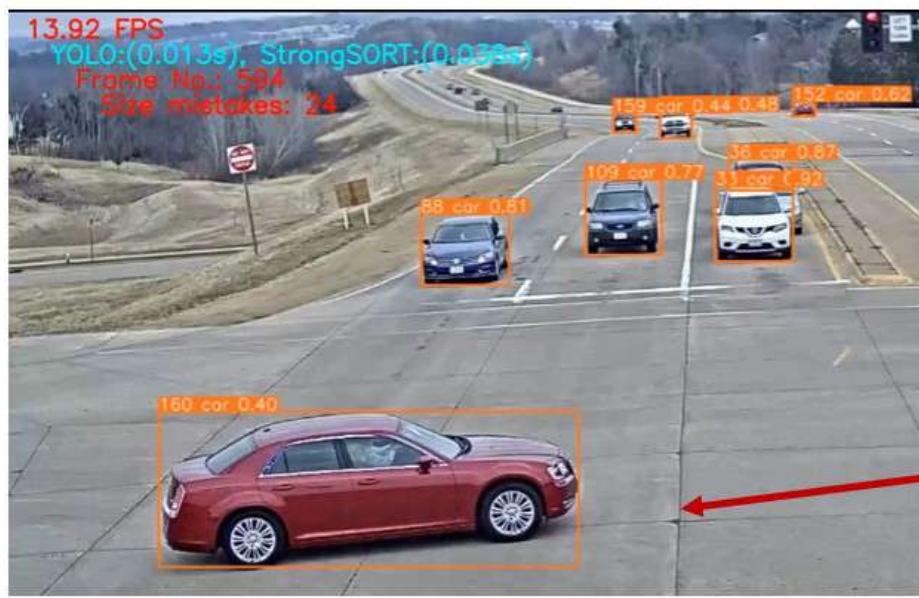
Frame 593: conf- 0.46Frame 594: conf- 0.40 (below Thres.)

Fig. 7. Bounding boxes in subsequent frames after applying the algorithm.



Fig. 8. Low confidence valid bounding boxes in the current frame after applying the algorithm.



Fig. 9. Multi-class detection for the same object issue improvement.

made using unoptimized and optimized versions of our algorithm showing the optimized version achieving almost the similar frames per seconds results as the base version even with added computations because of the implemented algorithm.

Table 4. Model speed analysis

Scene Type	Avg. FPS	Post Processing
Traffic (base)	10.01	2.00ms
Traffic (unoptimized)	6.19	18.07ms
Traffic (optimized)	9.13	10.10ms
Pedestrian (base)	9.10	2.00ms
Pedestrian (unoptimized)	6.20	14.21ms
Pedestrian (optimized)	8.50	9.90ms

Figure 10 shows some failed cases after using the algorithm using the same set of frames. Yellow circles represent good cases and red circles represent bad cases. In the first scene (left), two cars that were detected before applying the algorithm are not detected afterward but instead, we are able to fix other detection mistakes. In the second scene (right), the algorithm fails to detect the marked object in the figure but it can be seen that there is some bounding box size consistency improvement after applying the algorithm.

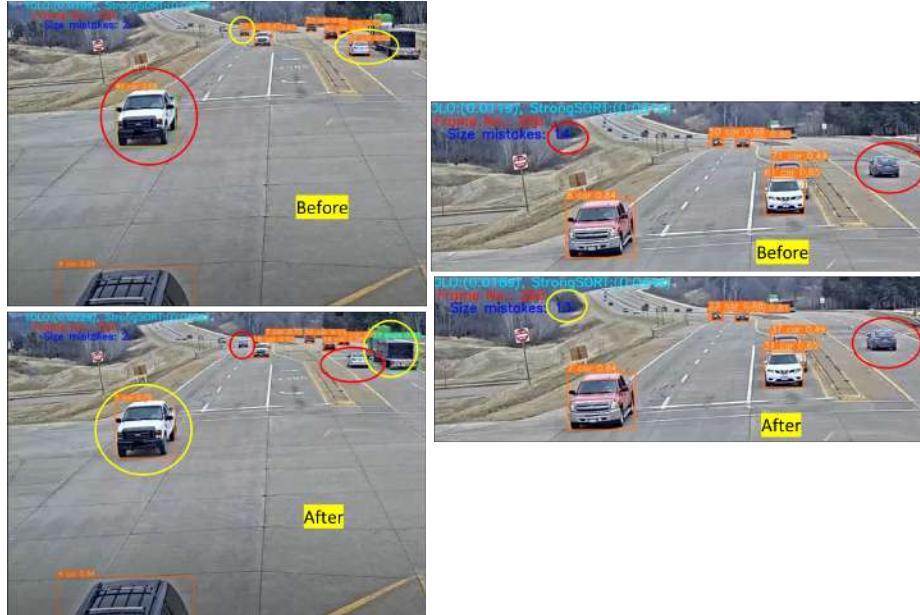


Fig. 10. Some failed cases.

4.2 Results for Bounding Box Size Difference Error Calculation

Figure 11 clearly shows that when the detector made a mistake in keeping the size of the detected object consistent in the subsequent frames our proposed algorithm worked and the mistake was recorded in the top left. Notice how the bounding box size is visibly changed in those two frames due to the misidentification.

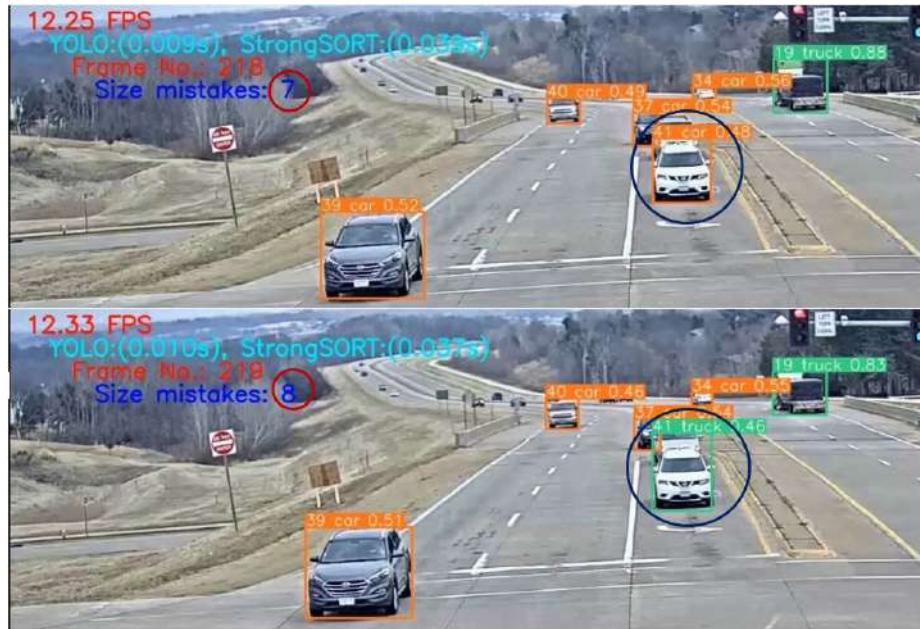


Fig. 11. Bounding box size difference inconsistency in subsequent frames is registered after applying the algorithm.

5 Conclusion

It is concluded that all the experiments were conducted sincerely while keeping the end goals in mind all the time.

- The algorithm to improve the detections using previous frame information was developed and tested to show an improvement over the base evaluation value.
- The detector was evaluated for the bounding box consistency check and the working has been shown using the subsequent frames where the detector had made a mistake and was noted down by the algorithm according to the said threshold.
- Various experiments have been conducted to show various results and findings of all the methods proposed in this paper.

Future Work. Since in this paper we have exclusively focused on YOLO detectors for improving the tracking model, in the future, we would like to work towards evolving the algorithm to incorporate various different detector models and present a qualitative and quantitative comparison with other techniques that are developed to improve the MOT.

Acknowledgements This study was supported by the BK21 FOUR project (AI-driven Convergence Software Education Research Program) funded by the Ministry of Education, School of Computer Science and Engineering, Kyungpook National University, Korea (4199990214394).

References

1. Ababsa, F., Maudi, M., Didier, J.Y., Mallem, M.: Vision-based tracking for mobile augmented reality pp. 297–326 (2008)
2. Bei, S., Zhen, Z., Wusheng, L., Liebo, D., Qin, L.: Visual object tracking challenges revisited: Vot vs. otb. Plos one **13**(9), e0203188 (2018)
3. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: the clear mot metrics. EURASIP Journal on Image and Video Processing **2008**, 1–10 (2008)
4. Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple online and realtime tracking pp. 3464–3468 (2016)
5. Bochinski, E., Eiselein, V., Sikora, T.: High-speed tracking-by-detection without using image information pp. 1–6 (2017)
6. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020)
7. Bouget, D., Allan, M., Stoyanov, D., Jannin, P.: Vision-based and marker-less surgical tool detection and tracking: a review of the literature. Medical image analysis **35**, 633–654 (2017)
8. Cao, J., Weng, X., Khirodkar, R., Pang, J., Kitani, K.: Observation-centric sort: Rethinking sort for robust multi-object tracking. arXiv preprint arXiv:2203.14360 (2022)
9. Du, Y., Song, Y., Yang, B., Zhao, Y.: Strongsort: Make deepsort great again. arXiv preprint arXiv:2202.13514 (2022)
10. Du, Y., Wan, J., Zhao, Y., Zhang, B., Tong, Z., Dong, J.: Giaotracker: A comprehensive framework for mcmot with global information and optimizing strategies in visdrone 2021 pp. 2809–2819 (2021)
11. Evangelidis, G.D., Psarakis, E.Z.: Parametric image alignment using enhanced correlation coefficient maximization. IEEE transactions on pattern analysis and machine intelligence **30**(10), 1858–1865 (2008)
12. Gao, M., Jin, L., Jiang, Y., Guo, B.: Manifold siamese network: A novel visual tracking convnet for autonomous vehicles. IEEE Transactions on Intelligent Transportation Systems **21**(4), 1612–1623 (2019)
13. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The International Journal of Robotics Research **32**(11), 1231–1237 (2013)
14. Gong, X., Zhou, Y., Zhang, Y.: Siamot: An improved siamese network with online training for visual tracking. Sensors **22**(17), 6597 (2022)

15. Luo, H., Gu, Y., Liao, X., Lai, S., Jiang, W.: Bag of tricks and a strong baseline for deep person re-identification pp. 1487–1495 (2019)
16. Manafifard, M., Ebadi, H., Moghaddam, H.A.: A survey on player tracking in soccer videos. *Computer Vision and Image Understanding* **159**, 19–46 (2017)
17. Naiel, M.A., Ahmad, M.O., Swamy, M., Lim, J., Yang, M.H.: Online multi-object tracking via robust collaborative model and sample selection. *Computer Vision and Image Understanding* **154**, 94–107 (2017)
18. Pang, Z., Li, Z., Wang, N.: Simpletrack: Understanding and rethinking 3d multi-object tracking. arXiv preprint arXiv:2111.09621 (2021)
19. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection pp. 779–788 (2016)
20. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
21. Robin, C., Lacroix, S.: Multi-robot target detection and tracking: taxonomy and survey. *Autonomous Robots* **40**(4), 729–760 (2016)
22. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks pp. 4510–4520 (2018)
23. Tak, A.S., Kim, H., Lee, S., Jung, S.K.: Towards improving the multi-object tracking evaluation metric pp. 176–176 (2022)
24. Tak, A.S., Sultana, M., Rahman, M.M., Kim, H., Lee, S., Jung, S.K.: Visual object tracking: Datasets and related information pp. 241–244 (2022)
25. Wang, Z., Zheng, L., Liu, Y., Li, Y., Wang, S.: Towards real-time multi-object tracking pp. 107–122 (2020)
26. Weng, X., Wang, J., Held, D., Kitani, K.: 3d multi-object tracking: A baseline and new evaluation metrics pp. 10359–10366 (2020)
27. Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric pp. 3645–3649 (2017)
28. Zhang, X., Zhou, X., Lin, M., Sun, J.: Shufflenet: An extremely efficient convolutional neural network for mobile devices pp. 6848–6856 (2018)
29. Zhou, K., Yang, Y., Cavallaro, A., Xiang, T.: Omni-scale feature learning for person re-identification pp. 3702–3712 (2019)
30. Zhou, X., Koltun, V., Krähenbühl, P.: Tracking objects as points pp. 474–490 (2020)

A Study on Tracking Moving Objects: Pig counting with YOLOv5 and StrongSORT

Seunggwan Lee¹, Wonhaeng Lee², Junghoon Park^{3*}

^{*}Corresponding author: Junghoon Park, Professor, College of Computing and Informatics,
Applied Artificial Intelligence

Ajou University, Republic of Korea
206, World cup-ro, Yeongtong-gu, Suwon-si, Gyeonggi-do
{gwan7801, bc005007, stevejobs}@ajou.ac.kr

Abstract. Counting the number of pigs is a significant issue in pig farms for efficient farm management. However, in pig farms in Korea due to the small land size, pigs are crowded together in a confined space. It makes it hard to count the number of pig stocks on the farm and it gets much worse when the farm scale grows larger. To solve this problem, we suggest an automatic method for counting the number of pigs, using an AI model with CCTV that is already equipped in farms. By doing this, we can count the accurate number of pig stock which can systemize internal and external environment management and be helpful to sales and distribution. We used YOLOv5 for object detection and StrongSORT for tracking. We optimized the model by tuning epoch and batch-size and got a 0.97832 mAP score. Using the detection model with the tracking model we could assign an ID to each pig in the video, which leads to a more accurate result. By this paper, we could be close to building a smart pig farm system at a low cost.

Keywords: computer vision, pig counting, object detection, object tracking
YOLOv5, StrongSORT, smart pig farm.

1 Introduction

The annual per capita pork consumption in Korea (2020) was 26 kg, similar to the sum of beef (13 kg) and chicken (14.7 kg). From statistics since 1995, it is easy to see that pork consumption has been overwhelmingly high compared with other meat consumption [1]. However, the number of pig farms dropped below 6,000 for the first time in 2021, due to a lack of manpower and high feed costs [2]. If this situation continues, when consumption exceeds supply pork prices would skyrocket.

In this paper, we suggest deep learning based real-time pig farm monitoring module using images from CCTV on the farm. The previous paper used only YOLO in object detection and only DeepSORT in object tracking [3]. For higher accuracy and faster computation, we compare YOLO and R-CNN in object detection and Strong-SORT in object tracking. This would perform accurate pig counting and assign an ID to each pig without an additional physical device to facilitate management. Therefore,

it is possible to establish a Smart Pig Farm system that can accurately count the number of pigs and manage them systematically even with a little manpower. Through this Smart Pig Farm system, it is possible to lower the entry barrier of Korean pig farms, which are being reduced. Moreover, this AI model is not limited to the pig farming business and domestic market, it could be applied to other industries that need an accurate count of the object or global market.

2 Background

The final goal of this research is to count the accurate number of pigs for the Smart Pig Farm system. In this paper, object detection and object tracking techniques are used. We compared Fast R-CNN and YOLOv5, and for object tracking, the Strong-SORT algorithm is used.

2.1 Object Detection Module

1. Faster R-CNN

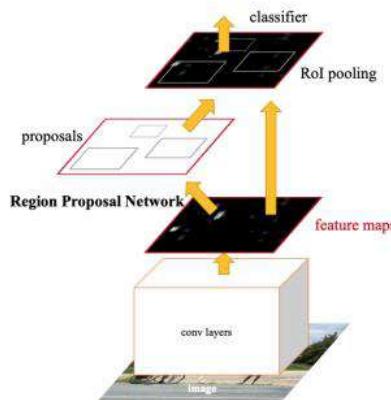


Figure 1. Process of Faster R-CNN. Convolution layers extract the Feature Map and pass it to the RPN to compute the RoI. The calculated RoI value conducts Pooling, then conducts classification for object detection [4].

Faster R-CNN inherits Fast R-CNN structure. Faster R-CNN calculates RoI through RPN that has features using Anchor boxes while Fast R-CNN uses selective search. RPN shows higher accuracy by calculating about 800 RoI, while Selective Search calculates 2,000 RoIs. Moreover, Faster R-CNN operates on GPU. This raises accuracy and speed by calculating little RoI using GPU.

2. YOLOv5

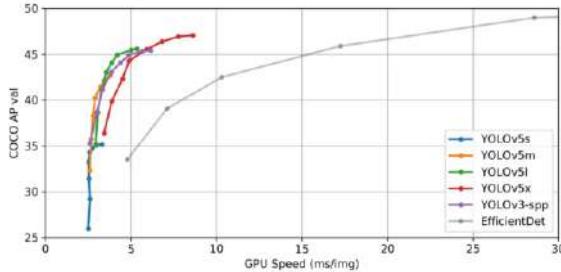


Figure 2. YOLOv5 type, GPU Speed, COCO AP Validation [5].

Previous research used YOLOv4 in object detection [3], but this paper uses an updated version, YOLOv5. YOLOv5 has a different Backbone and Head compared to other YOLO models [6]. The Backbone part extracts the Feature Map from the image, which is similar to YOLOv4, but works on lower capacity and the computing speed is faster. Backbone uses BottleneckCSP(CSPNet-based) and the Head part finds the object's position based on the Feature Map. After setting the Anchor Box (Default Box), uses it to generate the final Bounding Box. While YOLOv3 has a high FPS but mAP was relatively low. YOLOv5 however, performs well in both FPS and mAP.

3. Comparing Faster R-CNN with YOLOv5

We compared it with Faster R-CNN, which has high performance among R-CNN. By comparing the result of YOLOv5 and Faster R-CNN, Faster R-CNN AP was 0.7373, and YOLOv5 AP was 0.9837. As a result, YOLOv5 showed higher accuracy in our dataset.

Moreover, YOLOv5 inference speed was faster than Faster R-CNN. Also, unlike Faster R-CNN, YOLOv5 detects small or far away objects and has very few overlapping boxes. So, we chose YOLOv5 for Object detection [7].

2.2 Object Tracking Module

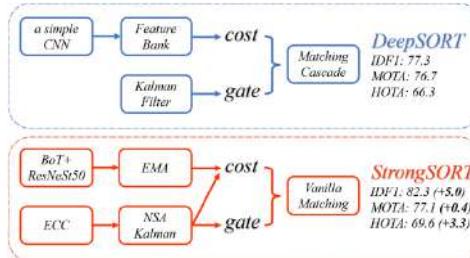


Figure 3. Framework and performance comparison between DeepSORT and StrongSORT. Performance is evaluated on the MOT17 validation set based on detections predicted by YOLOX [8].

StrongSORT improvement from DeepSORT lies mainly in the two branches, as shown in the bottom half of Figure 3. For the appearance branch, a stronger appearance feature extractor, BoT [9], is applied to replace the original simple CNN. By taking ResNeSt50 as the backbone [10], it can extract much more discriminative features.

For the motion branch, StrongSORT adopted ECC for camera motion compensation. Furthermore, StrongSORT replaced the vanilla Kalman filter with the NSA Kalman algorithm. Since it is vulnerable w.r.t. low-quality detections and ignores the information on the scales of detection noise [11].

Furthermore, to solve the assignment problem, instead of employing only the appearance feature distance during matching, StrongSORT adopted both appearance and motion information. The matching cascade algorithm limits the performance as the tracker becomes more powerful and it would limit the matching accuracy. StrongSORT replaced the matching cascade with the vanilla global linear assignment.

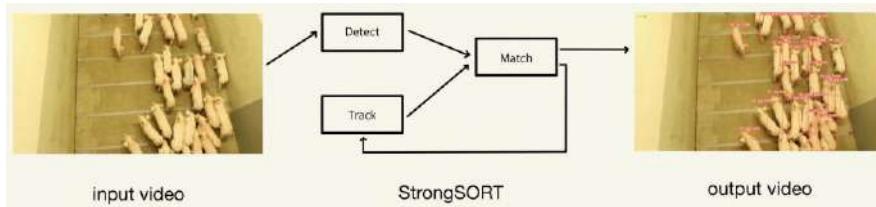


Figure 4. StrongSORT algorithm process

The main principle of StrongSORT is to predict the trajectory of the current frame from the trajectory after the Kalman filtering algorithm and make a judgment on whether to confirm it or not. After that, this detects the current frame, then correlates the data between the detection result and the predicted trajectory. Then update it after matching is completed.

After that, this continues to predict the current frame, observes the next frame, updates it, and so on cycle. If the track does not match, it will be deleted after exceeding the maximum age, and if the detection does not match, a new track is created, and the prediction is continued by Kalman filtering and repeated.

3 Experimental Results

3.1 Experimental Environment and Dataset

In this paper, the pig detecting and tracking model was trained on the KISTI National Supercomputing Center supercomputer system. For further information, the system model is Lenovo nx360-m4 with Intel Xeon Ivy Bridge (E5-2670) / 2.50GHz (10-core) / 2 socket (CPU), two V100(GPU), and 128GB DDR3(RAM). OS installed in the model is CentOS 7.9 (Linux, 64-bit).



Figure 5. Sample images from three different pig farms [12].

The dataset is a manually annotated open dataset provided by AIHub. AIDKOREA built this dataset by using CCTV in three pig farms. This dataset consisting of image file(jpg) and annotation file(json) has 2,700 images and one json file that has labels and bbox informations of the pigs in the images. As shown in Figure 5, captured image files were directly collected from the pig farm with CCTV which was already installed. Since it is CCTV, captured images are a top-down view, showing only pigs.

As a result, for the detection module, a total of 2,700 images that contain 79,326 pigs were used. 1,890 images (70%) were used for training data, 540 images (20%) were used for validation data, and 270 images (10%) were used for test data. For our final model, we used all data for training which is 2,700 images.

3.2 Results with Object Detection Module

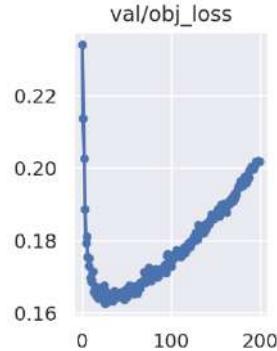


Figure 6. val/obj loss of Epoch 200, batch-size 16 model.

Table 1. Results for different epoch and batch-size.

epoch	batch-size	metrics/mAP 0.5	metrics/mAP 0.5:0.95	val /box loss	val/obj loss
30	16	0.96392	0.69898	0.029694	0.16369
30	8	0.97210	0.68231	0.026231	0.16034
30	4	0.97092	0.72676	0.027831	0.16216
30	2	0.97132	0.71402	0.026895	0.97779
50	16	0.97533	0.70647	0.028947	0.16665
50	8	0.97832	0.73375	0.026913	0.16552
50	4	0.97269	0.73252	0.027197	0.16694
50	2	0.97319	0.71294	0.027839	0.16832
100	16	0.97256	0.72234	0.027868	0.18307
100	8	0.97123	0.72429	0.027371	0.19229
100	4	0.97239	0.72301	0.027316	0.19319
100	2	0.97301	0.73125	0.026982	0.18507
200	16	0.96843	0.73142	0.027487	0.20177
200	8	0.97125	0.73372	0.026646	0.19872
200	4	0.97012	0.73192	0.027192	0.19983
200	2	0.97492	0.72915	0.027842	0.20091

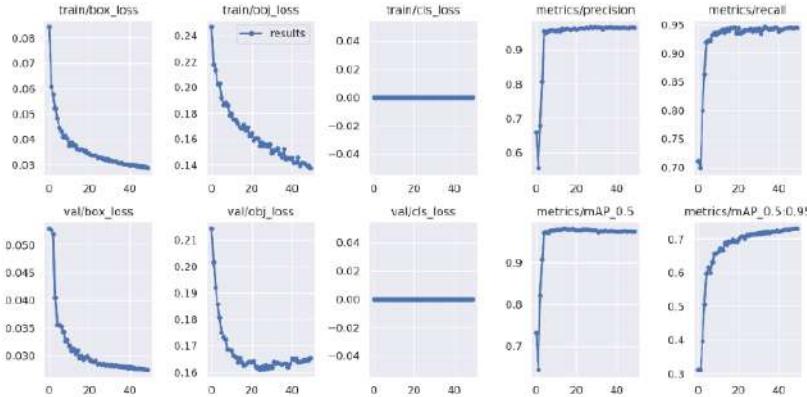
**Figure 7.** Evaluating performance of 50 epoch, 8 batch-size model YOLOv5

Figure 6. is obj_loss of the model tuned with 200 epoch and 16 batch-size which shows the overfitting problem. From the Figure 6. we can see that loss value rebounds at epoch 30~50. So, we conducted experiments with epoch sizes of 30, 50, 100, 200 and batch sizes of 2, 4, 8, 16 to find the best model. By the result shows at Table 1., for our final model we tuned parameter as batch-size 8 at epoch 50 which had the highest metrics/mAP 0.5:0.95.

Unlike Figure 6., which proceeded with 200 epochs, Figure 7. shows decrease of box_loss and obj_loss values as train and validation epochs progress. In addition, precision, recall, and mAP increase as progress.



Figure 8. The results of pig detection with YOLOv5

We tested our final detection model which is YOLOv5 tuned by 200 epoch and 8 batch-size with our test dataset. And we compared the number of pigs that our final model detected with labeling data from json file (annotation file). We calculated RMSE and the result was 0.215.

3.3 Results with Object Tacking Module

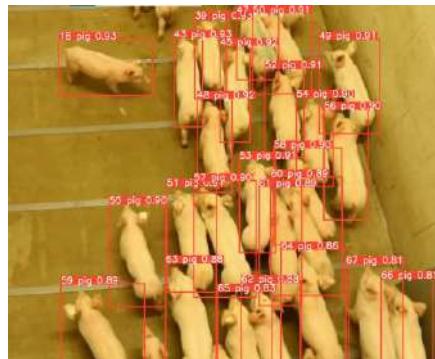


Figure 9. The results of pig tracking with StrongSORT

We use a 7-second-long video to check if our final models work well. The pig with ID 18, goes out of the frame and come back with same ID number. This shows that our final model performed better than using DeepSORT. The result of the full video is uploaded in the YouTube link below.

(<https://www.youtube.com/watch?v=BOOLM5Bl4OM>)

4 Conclusions

From the result of this paper, we compared YOLOv5 and Faster R-CNN in object detection and chose a better model which was YOLOv5. In addition, by adjusting the epoch and batch-size of YOLO in a total 16 combinations, we find the optimal model with high accuracy. In object tracking, we got more sophisticated results by using StrongSORT instead of DeepSORT [11].

Each farm can use this AI model by putting real-time CCTV video to calculate the accurate number of pigs right away for farm management and low-cost pig individual management without an additional physical device. It is possible to create a systematic Smart Pig Farm.

Moreover, the system is not limited to pigs, it could be also used for other various livestock such as cows, sheep, horses, and chickens. Furthermore, it makes it easier to manage each livestock individually and figure out the number of objects. So instead of rearing them in cages, they can be raised free at pasture.

For future research, we will create a tail that can track the movement path of the object by connecting the bboxes center points of the object. Also, not only we could identify the past path but also predict the future path. Moreover, we could study an algorithm that tracks objects moving fast.

References

1. KMTA Consumption Status Available online:
http://www.kmta.or.kr/kr/data/stats_spend.php (accessed on 25 January 2021).
2. KOSTAT Livestock trend survey dataset. Available online:
[https://aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&aihubDataSe=r%20%ealm&dataSetSn=145](http://kostat.go.kr/portal/korea/kor_nw/1/8/1/index.board?bmode=read&bSeq=&aSeq=416443&pageNo=1&rowNum=10&navCount=10&currPg=&searchInfo=&sTarget=title&sTxt=(accessed on 20 January 2022).
3. Jonggwan Kim.; Yooil Suh.; Junhee Lee. EmbeddedPigCount: Pig Counting with Video Object Detection and Tracking on an Embedded Board. Sensors. 2022.
4. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Trans. Pattern Anal. Mach. Intell. 2017.
5. Junfan WANG1.; Yi CHEN1.; Mingyu GAO. Improved YOLOv5 network for real-time multi-scale traffic sign detection. arXiv 2021.
6. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. arXiv 2016.
7. Yang, Q.; Xiao, D.; Lin, S. Feeding Behavior Recognition for Group-Housed Pigs with the Faster R-CNN. Comput. Electron. Agric. 2018.
8. Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J.: Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430 (2021)
9. Luo, H., Jiang, W., Gu, Y., Liu, F., Liao, X., Lai, S., Gu, J.: A strong baseline and batch normalization neck for deep person re-identification. IEEE Transactions on Multimedia 22(10), 2597–2609 (2019)
10. Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Lin, H., Zhang, Z., Sun, Y., He, T., Mueller, J., Manmatha, R., et al.: Resnest: Split-attention networks. arXiv preprint arXiv:2004.08955 (2020)
11. Yunhao Du.; Yang Song.; Bo Yang.; Yanyun Zhao. StrongSORT: Make DeepSORT Great Again. arXiv 2022.
12. AI-Hub Livestock behavior video dataset. Available online:
<a href=) (accessed on 13 October 2022)

BRDF Measurement with TDCRA

Atsushi Kimura¹, Ryo Kawahara^{1[0000-0002-9819-3634]}, and
Takahiro Okabe^{1[0000-0002-2183-7112]}

Department of Artificial Intelligence,
Kyushu Institute of Technology
680-4 Kawazu, Iizuka, Fukuoka 820-8502, Japan
okabe@ai.kyutech.ac.jp

Abstract. In this extended summary, we introduce our proposed method for BRDF measurement with a transmissive dihedral corner reflector array (TDCRA); a double-layer orthogonal micro mirror array. Our method combines a TDCRA and a projector-camera system, and achieves efficient BRDF measurement by controlling the directions of incident light rays without mechanical rotation, and capturing the outgoing light rays to various directions at once. We demonstrate that our method with the nonlinear interpolation of the captured BRDF values is useful for photo-realistic image synthesis.

Keywords: BRDF · TDCRA · projector-camera system.

1 Introduction

The reflectance properties of opaque surfaces are described by Bidirectional Reflectance Distribution Functions (BRDFs). BRDFs are useful for CV and CG applications such as visual inspection and photo-realistic image synthesis. Efficient BRDF measurement is an important issue to be addressed, because a BRDF is a four-dimensional function depending on both the incident and outgoing light directions, and therefore its measurement is time consuming in general.

In this extended summary, we introduce our proposed method for BRDF measurement with a Transmissive Dihedral Corner Reflector Array (TDCRA). The TDCRA consists of a double-layer orthogonal micro mirror array, and has the property that the light rays emitted from a point at one side bounce twice in the TDCRA, and then intersect at the symmetric point at the other side as shown in Fig. 1 (a). Our method combines a TDCRA and a projector-camera system, and achieves efficient BRDF measurement by controlling the directions of incident light rays without mechanical rotation, and capturing the outgoing light rays to various directions at once. We demonstrate that our method with the nonlinear interpolation of the captured BRDF values is useful for photo-realistic image synthesis.

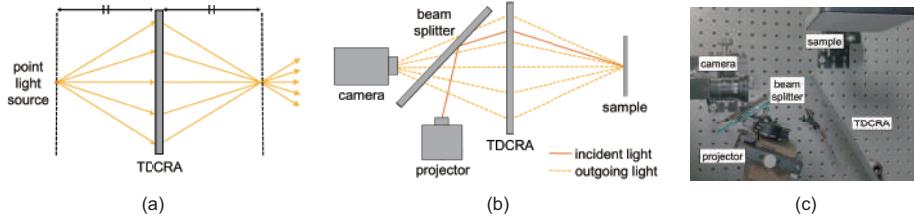


Fig. 1. Our setup: (a) a TDCRA, (b) the measurement principle, and (c) the prototype.

2 Related Work

BRDF measurement: Conventionally, gonioreflectometers [4] are used for BRDF measurement, but the measurement using them is time consuming due to its mechanical rotation and low sampling efficiency. In contrast to such straightforward measurement, image-based techniques are proposed by using spherical/cylindrical targets with uniform BRDF [5] and by using an ellipsoidal mirror [6]. Compared with those techniques, our proposed method makes use of only off-the-shelf devices and does not require such custom-order/self-built samples and devices.

TDCRA and DCRA: Recently, the TDCRA and DCRA are used for AR and MR applications. The TDCRA, originally developed for mid-air display [2], is utilized also for projection mapping with shadow suppression [3]. In addition, the DCRA is utilized for optical cloaking display [1]. Our study is a novel application of the TDCRA; we make use of it for a CV and CG application of BRDF measurement.

3 Proposed Method

Setup: Fig. 1 (b) shows the principle of BRDF measurement by using a TDCRA and a co-located projector-camera system¹. We can illuminate a point on a sample from various directions with a projector and capture the reflected light rays from the point to various directions with a camera. Fig. 1 (c) shows our prototype setup; the TDCRA is slanted 45° in order to prevent us from projecting/observing single-bounce and no-bounce light rays.

Calibration: Our setup requires geometric and photometric calibration. First, in addition to the conventional geometric calibration of a projector and a camera, we estimate the surface normal of (the planar holder of) a sample. Specifically, we replace the sample with a mirror, and then estimate the normal on the basis of mirror reflection. Second, we correct the intensities of the reflected light rays from the point on the sample. Specifically, we use a diffuse reflectance standard

¹ In theory, a pair of Fresnel lenses can be used instead of a TDCRA. We tested the Fresnel lens pair, but it did not work well due to limited image quality.

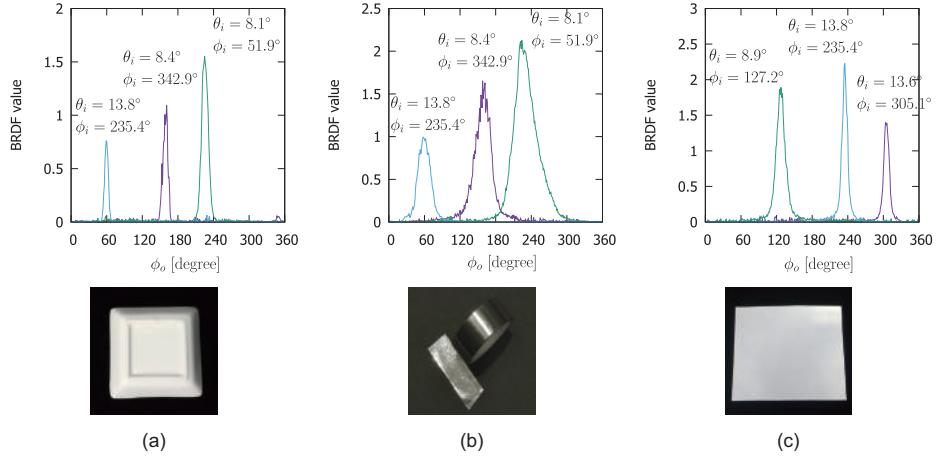


Fig. 2. The slices of the captured BRDFs: (a) a ceramic plate, (b) a metallic tape, and (c) a lens-based retroreflector.

as a sample, and correct the intensities so that they obey the cosine law of illumination and the Lambert's cosine law.

Interpolation: Because the resolution of light source directions is relatively low, the linear interpolation of the acquired BRDF values with respect to light source directions does not work well for sharp lobes due to specular and retro reflections. Accordingly, we nonlinearly interpolate the captured images with respect to light source directions. Specifically, we fit a scaled two-dimensional Gaussian distribution to a specular/retro reflection lobe, and then interpolate the parameters of the distribution.

4 Experiments

Results: We used a TDCRA from Asukanet [2]. Fig. 2 shows the slices of the captured BRDFs: (a) a ceramic plate, (b) a metallic tape, and (c) a lens-based retroreflector. Here, we denote the zenith and azimuth angles of incident and outgoing light rays by (θ_i, ϕ_i) and (θ_o, ϕ_o) , and show how the BRDF values behave with respect to ϕ_o when $\theta_i = \theta_o$ and ϕ_i are fixed. We can see that the specular and retro reflections have peaks when $\phi_o = (\phi_i \pm \pi)$ and ϕ_i respectively as expected. In addition, we can see that the difference of surface roughness values between the ceramic plate and the metallic tape are captured; the wider the distribution is, the larger the surface roughness is.

Applications: Fig. 3 shows the images synthesized from the captured BRDF of the ceramic plate. To demonstrate the effectiveness of our proposed method, in particular the nonlinear interpolation of BRDF values, we synthesized the images under a single light source direction from the BRDF values acquired



Fig. 3. The synthesized images of the ceramic plate: (a) the nonlinear interpolation and (b) the linear interpolation of the acquired BRDF values.

under three different light source directions. We can see that (a) the nonlinear interpolation works well, but (b) the linear interpolation fails; we can see three specular peaks instead of one.

5 Conclusion and Future Work

In this extended summary, we introduced our proposed method for efficient BRDF measurement with a TDCRA. Our method combines a TDCRA and a projector-camera system, and achieves efficient BRDF measurement by controlling the directions of incident light rays without mechanical rotation, and capturing the outgoing light rays to various directions at once. We demonstrated that our method with the nonlinear interpolation of the captured BRDF values is useful for photo-realistic image synthesis. Our future study includes measuring a wider range of angles of incident and outgoing light rays by rotating a sample.

Acknowledgement: This work was supported by JSPS KAKENHI Grant Number JP20H00612.

References

1. T. Aoto, Y. Itoh, K. Otao, K. Takazawa, and Y. Ochiai, “A design for optical cloaking display”, In Proc. ACM SIGGRAPH2019 Emerging Technologies, Article No.3, pp.1–2, 2019.
2. ASKA3D, <https://aska3d.com>
3. K. Hiratani, D. Iwai, P. Punpongsanon, and K. Sato, “Shadowless projector: suppressing shadows in projection mapping with micro mirror array plate”, In Proc. IEEE VR2019, pp.1309–1310, 2019.
4. H. Li, S.-C. Foo, K. Torrance, and S. Westin, “Automated three-axis gonioreflectometer for computer graphics applications”, Optical Engineering, Vol.45, No.4, 043605, 2006.
5. S. Marschner, S. Westin, E. Lafourche, and K. Torrance, “Image-based bidirectional reflectance distribution function measurement”, Applied Optics, Vol.39, No.16, pp.2592–2600, 2000.
6. Y. Mukaigawa, K. Sumino, and Y. Yagi, “Rapid BRDF measurement using an ellipsoidal mirror and a projector”, IPSJ TCVA, Vol.1, pp.21–32, 2009.

Multi-scale Recurrent Residual U-Net for Anomaly Segmentation in Industrial Images

Haoyu Chen¹ and Shivani Sanjay Kolekar¹ and Kyungbaek Kim¹

¹ Dept. of Artificial Intelligence Convergence, Chonnam National University,
Gwangju, South Korea
leochy554@gmail.com
shivani.kolekar@gmail.com
kyungbaekkim@jnu.ac.kr

Abstract. In recent years, there has been a rapid growth in automatic segmentation technology utilizing industrial imaging data. However, most manufacturing industries still use manual visual inspection of potential defects in final products, which requires a great number of manpower and is immensely time consuming task. Using automatic industrial image anomaly segmentation technology can greatly alleviate this problem, as it can reduce cost and time consumption along with improved quality control. Deep learning networks are widely used in industrial image data processing and interpretation due to their powerful feature extraction capabilities and efficient feature expression capabilities. To this end, this paper proposes a multi-scale recurrent residual U-Net model named MR2U-Net. The model introduces a multi-scale recurrent residual block to enhance the model's multi-scale industrial image anomaly segmentation ability. It uses a residual path between the downsampling and upsampling paths instead of ordinary skip connections, and narrows the semantics between feature maps for stitching difference. Compared with other popular segmentation networks, the well-trained MR2U-Net model has better stability for different types of component test results. For the evaluation, we use mean IOU coefficient and Dice coefficient of images where values are 0.5350 and 0.6490 respectively. The performance both have been significantly improved, providing a reliable solution for the automatic industrial image anomaly segmentation in large-scale industrial manufacturing.

Keywords: Industrial AI, Deep learning, Image Segmentation.

1 Introduction

In large-scale industrial production, segmentation of abnormal regions of industrial images is crucial to guarantee quality standards. The purpose of industrial image anomaly segmentation is to detect anomalous regions in individual images using various artificial intelligence-based methods. However, abnormal regions of parts in industrial production are often diverse, not only extremely complex, but also easily confused with normal regions, and it is very difficult for the model to extract effective

segmentation features, especially when abnormal samples are rare. Therefore, achieving accurate anomaly segmentation is a challenging task.

In recent years, deep learning has achieved great success in many different fields, especially in the field of computer vision (CV), such as face recognition, scene text detection, target tracking and automatic driving, etc., many abnormalities based on deep learning Region detection methods are also widely used in various industrial scenarios. For example, Zou et al. [1] proposed a new self-supervised learning scheme SPot-the-difference (SPD), which can regulate and compare self-supervised pre-training to be more suitable for anomaly detection and segmentation tasks. Nakazawa et al. [2] proposed a method to detect and segment abnormal wafer map defect patterns using a deep convolutional encoder-decoder neural network architecture.

Although great progress has been made in the research of industrial image anomaly region segmentation, there are still many problems. First of all, most of the data sets used by most researches now have fewer training samples, which are prone to overfitting. The generalization ability of the research results is poor, and the obtained models cannot assist employees in judging abnormal regions of industrial images. Secondly, the abnormal regions of industrial images are very complex and easily confused with the normal regions of the image, and it is very difficult for the model to extract effective segmentation features.

To address the above issues, this paper proposes a deep learning-based Multi-scale Recurrent Residual U-Net model (MR2U-Net) for industrial image anomaly segmentation. Firstly, a multi-scale recurrent residual block is introduced to enhance the network model's layered and multi-scale industrial image anomaly segmentation capabilities, and secondly, a residual convolutional layer is incorporated on the skip connection between the downsampling and upsampling paths to compensate for the loss from the downsampling. The difference between low-level features at early stages of the path and higher-level features from the upsampling path. Finally, good results were obtained in the industrial anomaly image test set.

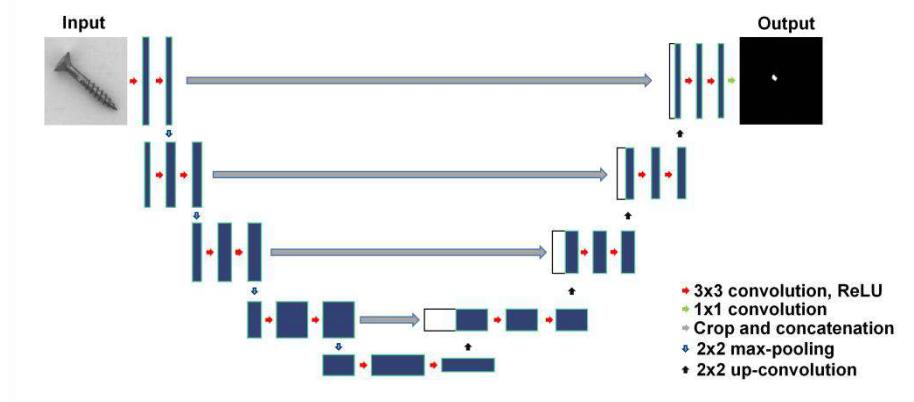


Fig. 1. U-Net model framework.

2 Related Work

U-Net Model. In 2014, Long et al. [3] used fully convolutional neural network (FCN) for end-to-end segmentation of natural images for the first time, achieving a breakthrough from traditional machine learning-based methods to deep learning methods. In 2015, Ronneberger et al. [4] proposed the U-Net network, and its structure is shown in Fig. 1. U-Net is an FCN-based network. Both encoders and decoders and skip connections are used to perform more accurate segmentation on a small number of training images. The difference between U-Net and FCN is that the U-Net network is left-right symmetric, the left side is the encoding path for capturing context information, and the right side is the decoding path for accurate positioning to restore the feature map size. After the output feature maps of each layer of the encoder are copied and cut, they are fused with the deconvoluted feature maps of the corresponding decoder, and then used as the input of the next layer to continue upsampling. The U-Net network has a large number of feature channels in the upsampling process, which enables it to pass context information to layers with higher resolution.

During the training process, since the images of abnormal areas of industrial parts are very complex, U-Net uses traditional convolution and pooling operations in the encoder to extract features. This feature extraction method can easily cause the model to fail to extract all useful feature information, and some features are lost. In addition, U-Net uses traditional convolution and deconvolution in the decoder to restore the feature map, which will cause a certain loss of feature information, making the network unable to fully restore the complex feature information of the image. Therefore, the U-Net network has become the research object of many researchers in the field of image segmentation. He et al. [5] made the network better preserve the feature maps in deeper neural networks by adding ResNet units in the U-Net network, and provided improved performance for deeper networks. Oktay et al. [6] suppress the feature responses irrelevant to the background region by adding an attention mechanism in the skip connections of U-Net, reducing the number of parameters and computational burden brought by the increase of network depth.

Recurrent Convolutional Neural Network. Recurrent Neural Network (RNN) is a special class of neural networks capable of processing data. The traditional feed-forward neural network only points to the output layer through the value of the activation function in the hidden layer, while RNN sends the result value of the activation function at the hidden layer node to the output layer and returns it to the next hidden layer node. Computational inputs form a feedback loop that is the opposite of traditional feedforward networks. In the continuous development of deep learning, RNN has gradually been introduced into the convolutional neural network (CNN) to form a recurrent convolutional neural network (RCNN). RCNN uses a circular convolutional layer (RCL) instead of a convolutional layer. RCL does not output to the pooling layer after extracting the features of the input layer, but uses a changed cyclic neural network for processing, and uses the method of adding empty data to the feature layer data.

3 Methodology

In order to solve the general problem of the traditional segmentation model's ability to segment industrial abnormal images, this paper refers to the U-shaped structure in the U-Net model, improves the convolutional layer and skip connections in the U-Net model, and proposes a multi-scale recurrent residual U-Net model (MR2U-Net), whose structure is shown in Fig. 2. The MR2U-Net model refers to the concept of recurrent convolution layer[7], and uses a multi-scale recurrent residual convolution layer to replace the convolution layer in U-Net in the encoding and decoding process, which not only increases the depth of the model and effectively retains the features in the image through recurrent convolution, but also It can solve the problem of gradient disappearance caused by the deepening of network layers, and at the same time alleviate the over-fitting problem caused by the small amount of data, and efficiently extract important features of industrial abnormal images. Next, in order to alleviate the difference between the encoder-decoder features, this paper uses the residual path instead of the skip connection in U-Net, by taking the convolution operation before the corresponding feature connection in the encoder and decoder. Through the nonlinear operation of the 3×3 convolutional layer and the 1×1 residual structure, the decoding part can better restore the original image, thereby improving the segmentation accuracy. Finally, the softmax activation function is used to perform binary classification on the decoding results to realize the segmentation of abnormal regions and backgrounds.

Multi-scale Recurrent Residual Block. The Multi-scale Recurrent Residual block used in this paper is shown in Fig. 3, which replaces all the convolutional layers in the U-Net structure in order to learn multi-scale image features. Firstly, the 5×5 and 7×7 convolutional blocks are decomposed by a series of 3×3 convolutional blocks with smaller size, and then the output is obtained from each 3×3 convolutional block and concatenated to extract spatial features at different scales. The outputs of the second and third 3×3 convolutional blocks are effectively close to the outputs of the 5×5 and 7×7 convolutional blocks, respectively, thereby reducing the number of network parameters and speeding up the training speed of the network. At the same time, the recurrent convolution operation is performed on the series convolutional blocks, and when $t = 3$ time steps, a feedforward subnetwork with the maximum depth of 4 and the minimum depth of 1 is formed, including a convolution layer and a subsequence of three recurrent convolutional layers to deepen the number of layers of the network and enhance the feature extraction ability. In addition, a 1×1 convolution residual connection is added to the module, so that the network can obtain more spatial information.

Suppose that the input sample x_n in the multi-scale recurrent layer is located at the n 'th layer, and for the pixel unit (i, j) located on the n 'th feature map in the layer, its output M at the step size t is given by the following formula:

$$M_{ijk}^n(t) = (w_k^f)^T x_n^{f(i,j)}(t) + (w_k^r)^T x_n^{r(i,j)}(t-1) + b_k \quad (1)$$

Where $x_n^{f(i,j)}$ is the feedforward input, $x_n^{r(i,j)}$ is the recurrent input of the n 'th layer, w_k^f is the feedforward weight, w_k^r is the recurrent weight, and b_k is the bias of the n 'th feature map. The output M is activated by the RuLU function with the following formula:

$$f(M_{ijk}^n(t)) = \text{Max}(0, M_{ijk}^n(t)) \quad (2)$$

In MR2U-Net, the final output of the multi-scale recurrent layer is passed through the residual connection, assuming that the output of the multi-scale recurrent residual block is x_{n+1} , then it can be calculated as follows:

$$x_{n+1} = x_n + f(M_{ijk}^n(t)) \quad (3)$$

Where x_n and x_{n+1} correspond to the input versus output of the multi-scale recurrent residual block.

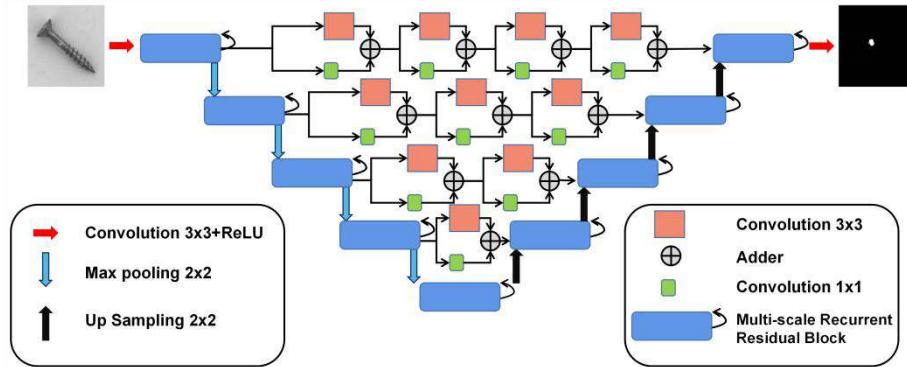


Fig. 2. MR2U-Net model framework.

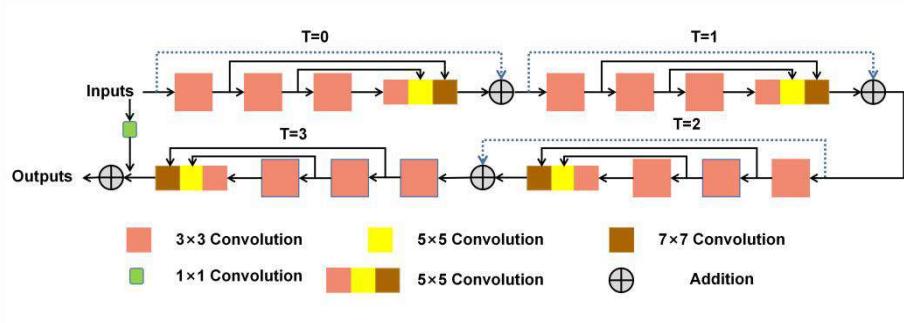


Fig. 3. Expanded multi-scale recurrent residual convolution structure for $t = 3$.

4 Experimental Results and Evaluation

4.1 Dataset

This experiment uses the MVTec AD dataset [8], which contains 5354 industrial images of different object and texture categories. For each category, it has normal (no defect) and abnormal images. We selected 1258 abnormal images of 15 categories for experiments, which contain more than 70 different types of defects. We randomly select 80% of the images as the training set and 20% of the images as the testing set. All training and test images (including ground truth) are resized to 128×128 to speed up model training.

4.2 Implementation

In this experiment, a computer with Intel Core i7-10700 2.9 GHz CPU (with 16GB memory) and NVIDIA GeForce RTX 2060 SUPER GPU (with 8GB video memory) is used for training and testing. The network model uses the Adam optimization algorithm, where the initial learning rate Set to 10-4, and the number of training epoch is set to 200.

4.3 Performance Indicators

This experiment uses Intersection-Over-Union (IOU) and Dice Coefficient (DC) to evaluate defect segmentation performance. The formulas of IOU and DC are as follows:

$$IOU = \frac{TP}{TP+FN+FP} \quad (4)$$

$$DC = \frac{2TP}{2TP+FN+FP} \quad (5)$$

where TP, FP and FN denote the number of pixels of true positive, false positive and false negative.

4.4 Experimental results and analysis

For the purpose of evaluation of MR2U-Net in the segmentation of industrial image abnormal regions, the MVTec AD dataset is compared with U-Net[4], Residual U-Net[5], Attention U-Net[6] Three models based on deep learning are used to segment and process abnormal regions of industrial images. All the methods above use the training set for training and the test set for evaluation. The results are shown in Table 1. The mean IOU and DC of the MR2U-Net model are as high as 0.5350 and 0.6490, respectively, and the results obtained are better than other schemes. Specifically, with regard to mean IOU and DC, our method achieves the best performance in 9 out of a total of 15 categories. For categories where our model did not achieve first place in segmentation performance, it still achieves comparable performance to the best

competing methods. Fig. 4 shows the segmentation results of MR2U-Net on the MVTec AD dataset. It can be seen that the segmentation results obtained by the MR2U-Net model are very close to the ground truth, which reflects that the MR2U-Net model has strong industrial image anomalies Segmentation ability.

Table 1. Performance comparison of MR2U-Net with other networks.

Category	U-Net[4]		Residual UNet[5]		Attention UNet[6]		MR2U-Net	
	IOU	DC	IOU	DC	IOU	DC	IOU	DC
Bottle	0.5976	0.7298	0.7052	0.8117	0.6476	0.7532	0.6547	0.7608
Cable	0.4676	0.5724	0.5283	0.6243	0.5400	0.6534	0.5459	0.6624
Capsule	0.1873	0.2853	0.1934	0.2711	0.2780	0.3865	0.3579	0.4615
Carpet	0.5289	0.6513	0.4928	0.6100	0.4731	0.5952	0.5063	0.6250
Grid	0.3399	0.4564	0.3944	0.5338	0.1390	0.2193	0.3283	0.4368
Hazel nut	0.7237	0.8274	0.7169	0.8273	0.7242	0.8210	0.7047	0.8132
leather	0.5824	0.7204	0.6440	0.7722	0.6410	0.7689	0.6791	0.7974
Metal nut	0.4614	0.5937	0.3582	0.4676	0.3408	0.4691	0.5517	0.6814
Pill	0.2498	0.3614	0.5699	0.6937	0.5147	0.6303	0.6330	0.7420
Screw	0.2176	0.2944	0.2178	0.2718	0.2336	0.3037	0.2662	0.3341
Tile	0.8128	0.8904	0.8080	0.8840	0.8121	0.8873	0.7787	0.8662
Toothbrush	0.3223	0.4266	0.3219	0.4489	0.3326	0.4480	0.3340	0.4682
Transistor	0.0768	0.1340	0.3800	0.5207	0.3576	0.4966	0.4816	0.6221
Wood	0.2219	0.3192	0.5221	0.6367	0.5658	0.6872	0.5820	0.7062
Zipper	0.5892	0.7309	0.6476	0.7781	0.5959	0.6510	0.6209	0.7584
Mean	0.4253	0.5330	0.5000	0.6101	0.4797	0.5847	0.5350	0.6490

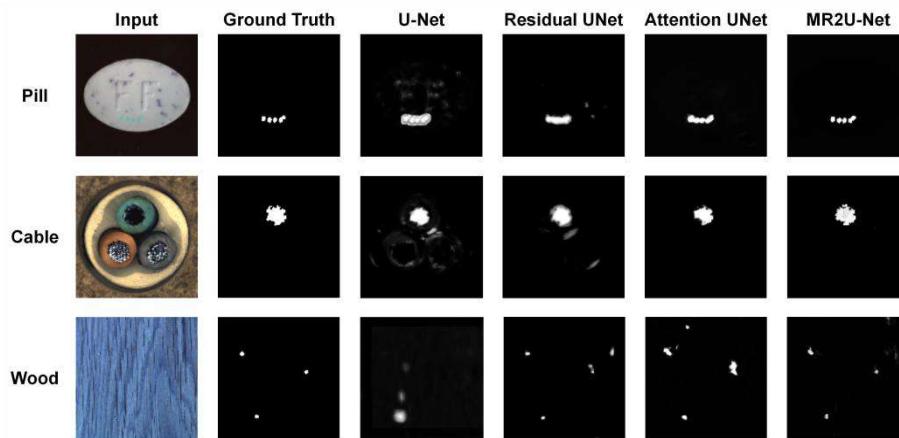


Fig. 4. Segmentation results on some test images from the MVTec AD dataset.

5 Conclusion

In this paper, Our develop and analyze a method for segmenting abnormal regions from industrial images. Based on the U-Net network structure, a multi-scale recurrent residual mechanism is introduced to enhance the feature extraction ability of the model for target regions. As the number of network layers increase, the overall robustness of the network is improved. Furthermore, our model outperforms current popular segmentation models on anomaly region segmentation in industrial images. we plan to improve our model to achieve optimized performance, in the future.

Acknowledgments

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the Innovative Human Resource Development for Local Intellectualization support program(IITP-2022-RS-2022-00156287) supervised by the IITP(Institute for Information & communications Technology Planning Evaluation). This results was supported by "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(MOE) (2022RIS-002)

References

1. Zou Y, Jeong J, Pemula L, et al. SPot-the-Difference Self-supervised Pre-training for Anomaly Detection and Segmentation[C]//European Conference on Computer Vision. Springer, Cham, 2022: 392-408..
2. Nakazawa T, Kulkarni D V. Anomaly detection and segmentation for wafer defect patterns using deep convolutional encoder–decoder neural network architectures in semiconductor manufacturing[J]. IEEE Transactions on Semiconductor Manufacturing, 2019, 32(2): 250-256.
3. Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 3431-3440.
4. Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation[C]//International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015: 234-241.
5. He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.
6. Oktay O, Schlemper J, Folgoc L L, et al. Attention u-net: Learning where to look for the pancreas[J]. arXiv preprint arXiv:1804.03999, 2018.
7. Zhou J, Hong X, Su F, et al. Recurrent convolutional neural network regression for continuous pain intensity estimation in video[C]//Proceedings of the IEEE conference on computer vision and pattern recognition workshops. 2016: 84-92.
8. Bergmann P, Batzner K, Fauser M, et al. The MVTec anomaly detection dataset: a comprehensive real-world dataset for unsupervised anomaly detection[J]. International Journal of Computer Vision, 2021, 129(4): 1038-1059.

LHFAN: Scene Text Recognition Method Based on Multi-level Feature Fusion and Enhancement of Semantic Knowledge

Ruturaj Mahadshetti, Guee-Sang Lee*, Hyung-Jeong Yang, and Soo-Hyung Kim

Department of Artificial Intelligence Convergence
 Chonnam National University, Gwangju 61186, South Korea
 ruturajm977@gmail.com, gslee@jnu.ac.kr, hjyang@jnu.ac.kr, and
 shkim@jnu.ac.kr

Abstract. In various computer vision tasks, STR has been a sensual research subject and performs a considerable utility. Modern deep-learning algorithms for scene text recognition (STR) have contrived significant advancements over the last few years. However, they aren't optimal for recognizing the text from degraded images or even when images are clear. Many STR methods employ transformers to utilize linguistic information for arduous recognition tasks rather than purely visible classification, but the recognition outcome is not superior for degraded images and confusing text fonts. Early scene text recognition approaches may even yield a substantial percentage of erroneous outputs when an image acquire in natural conditions. In this study, we proposed a novel approach called LHFAN to address the above issues by utilizing low-level features with high-level features, which improves the ability of visual features and semantic features. LHFAN employs upsampling method with ResNet and blends different scale features to enrich the feature capability of text content. We demonstrate with experimental results that our proposed method outperforms on standard text recognition datasets and also achieves state-of-the-art performance when working with blurred and perspective images.

Keywords: Scene Text Recognition · Deep Learning · Transformer · Convolutional Neural Network

1 Introduction

Scene text recognition, also known as optical character recognition (OCR) in natural scenes, is the process of automatically identifying and extracting written text from images and videos. Scene Text Recognition (STR) has emerged as a popular and intriguing study area in both academic and commercial circles as a subfield of computer vision. This technology has a wide range of practical applications, such as in self-driving cars, mobile document scanning, and augmented reality. However, the problem of scene text recognition is challenging due to the variability in text appearances, such as different fonts, colors,

and orientations, as well as the presence of noise and occlusions. Numerous text recognition techniques have recently been offered as solutions to this challenging issue, such as pictorial feature extraction methods [13, 25], attention-based mechanisms [32, 38, 39], resolution enhancement [3, 22, 26], and rectification mechanisms as pre-processing tasks [61, 1, 13].



Fig. 1. Failure results from the previous framework. (1) LevOCR [9], (29) MGP-STRF, and ground truth (gt).

In recent years, Scene text recognition approaches have made considerable progress and achieved significant performance because of their ability to learn and extract features from large amounts of data. One popular deep learning approach for scene text recognition is the use of convolutional neural networks (CNNs) and recurrent neural networks (RNNs). CNN's can be utilized to extract features from an image, while RNNs can be used to process these features and recognize the text. Another approach is to use attention-based mechanisms to focus on the most important parts of the image, like the text itself, during the recognition process. Recently, End-to-end trainable models such as the Connectionist Temporal Classification (CTC) and Attention-based methods have been proposed, which directly take an image as input and output the recognized text. These models have been shown to achieve high levels of accuracy and can be trained on large datasets to generalize well to unseen data. In addition to CNN's and RNNs, other deep learning techniques, such as Generative Adversarial Net-

works (GANs) and the Generative Pre-trained Transformer (GPT) employed to improve scene text recognition.

Recent scene text recognition techniques [9, 29] have demonstrated high performance, but their effectiveness diminishes when applied to Pixelated or partially obscured images. Many methods [9, 29, 19] take advantage of the Vision-Language Transformer and fuse the semantic feature knowledge with visual features instead of considering it separately. Da et al. [9] introduce the LevOCR framework using Levenshtein Transformer, and the refinement process uses two basic character-level operations (deletion, insertion) to enrich the accuracy of the text recognition task. These operations are learned using imitation learning, which allows for the parallel decoding of text and dynamic change in the length of text. Wang et al. [29] proposed a framework utilizing a Multi-Granularity Prediction technique to take advantage of linguistic knowledge. Wu et al. [19] employ Masked Autoencoders for a masking mechanism that allows the model to ignore the background pixels and only focus on the text pixels. [19] use the iterative correction process to improve performance. However, previous methods have a significant drawback in recognizing text from images that have partial obstructions, variations in text appearances that are seen in their curved shapes, various orientations, size variations, and fancy font styles. Fig. 1 shows the failure results of earlier work. Low-level features for scene text recognition are basic image features that are extracted from the input image and used as inputs to the text recognition system. Low-level features are used as inputs to more sophisticated text recognition algorithms, such as optical character recognition (OCR) and convolutional neural networks (CNNs). High-level features in scene text recognition capture the meaning and context of the text in an image through techniques like text semantic segmentation, confidence scores, and text-image relationships. These high-level features enhance the robustness and precision of text recognition systems, particularly in complex real-world scenarios where the text may be obscured, distorted, or written in multiple languages.

Prior approaches have acquired promising results on several benchmarks, indicating their potential effectiveness in solving a particular problem or achieving a specific task. However, recent frameworks fail to generate robust linguistic knowledge and visual features. To overcome these issues, we propose a novel approach utilizing different level features that are helpful to enhance the shallow semantic knowledge for recognition. LHFAN unite ResNet and the convolutional pyramid method to enrich extracted visual features and linguistic information about text instance. The Convolutional pyramid process upsamples features at different levels. After that, LHFAN blends low-level features with high-level features from several layers. Low-level features play a crucial role in the process of scene text recognition by providing a foundation for identifying and extracting text from an image. Combined low-level features and deep features information improve the accuracy of the text recognition process by providing additional information about the layout and structure of the text in an image. This work primarily offers the following contributions:

1. We explore an implicit method to unite low-level features and deep features to improve the robustness of extracted features and linguistic information and prove the importance of low-level features for Scene Text Recognition.
2. The proposed framework is designed to be robust and able to effectively reduce the number of false positives and increase recognition accuracy.
3. The LHFAN algorithm demonstrates exceptional results and surpasses current methodologies in the field.

2 Related work

The classical method for scene text recognition (STR) involves using a convolutional neural network (CNN) to extract visual features, an recurrent neural network (RNN) for ordering labeling, and Connectionist Temporal Classification (CTC) [10] to calculate the loss [13, 20]. Recently, a method known as GTC [4] has been developed to improve the CTC-based method by incorporating a graph convolutional network (GCN) [31] to learn local correlations of features in irregular text images. Some works have also aimed to correct these irregular text images, such as RPI [24] uses the quadratic Bezier curves, and ScRN adds symmetry constraints in addition to Thin-Plate-Spline transformation. Recent studies [9, 27, 19] have proposed innovative solutions for handling difficult situations, such as occlusion and noise, by utilizing models that incorporate semantic information.

Language-aware methods. In some approaches [5, 32], semantics are extracted using external language models. For example, SRN [32] uses the predicted text from the visual model to construct a global semantic reasoning module. In ABINet [5], the method is developed as a bidirectional cloze network, which uses an iterative correction for the language model and makes better use of bidirectional linguistic information. Another approach [14, 33, 12] involves impliedly learning semantics without using language models. Local and global mixing blocks are used in SVTR [7] to recognize both characters and their long-term dependency. Visual-semantic interaction is achieved with VST [33] by extracting semantic information from a visual feature map. ConCLR [27] framework first creates characters with varied contexts using basic image concatenation operations and then optimizes the contrastive loss on the obtained embeddings. ConCLR mitigates the side-effect of overfitting to specific contexts by gathering together clusters of identical characters within different contexts in the embedding space. Recently, various models that adopt the language model approach into account to improve performance by taking into account spatial context [], incorporating real-world images in a semi-supervised manner, or using re-ranking techniques to generate more valuable output.

Modern **Transformer-based methods** have demonstrated their effectiveness in scene text recognition (STR). For example, PIMNet [30] uses a bi-directional Transformer-based parallel decoder to iteratively gather contextual information. Also, ViTSTR [35] utilizes a Vision Transformer (ViT) encoder for recognition tasks without a decoder, and it is utilized with pre-trained parameters from DeiT. Also, ViTSTR employs a Vision Transformer (ViT) encoder for

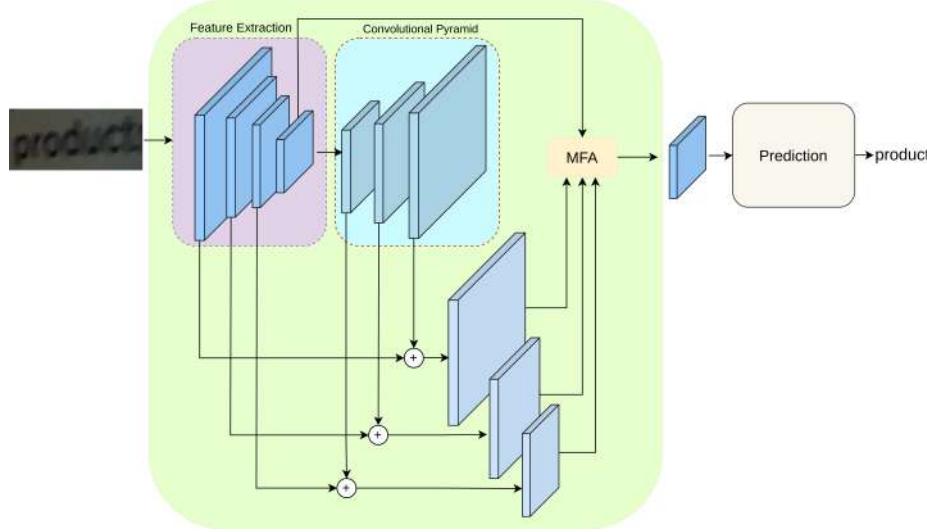


Fig. 2. The illustration of the LHFAN method, LHFAN consists of four parts: Feature Extraction, Convolution Pyramid, Multi-level Feature Aggregation (MFA), and Prediction.

recognition tasks without a decoder, and it is utilized with pre-trained parameters from DeiT. Wang et al. [29] proposed a framework using a Multi-Granularity Prediction technique to take advantage of linguistic knowledge. Wu et al. [19] employ Masked Autoencoders for a masking mechanism that allows the model to ignore the background pixels and only focus on the text pixels. In our proposed model, we employ a transformer for the prediction process.

3 Methodology

The detailed architecture of LHFAN describe in Fig. 2. Our proposed LHFAN consists of four parts: feature extraction, Convolution Pyramid, Multi-feature aggregation (MFA), and Prediction. We employ ResNet-50 as a backbone to extract features and utilize a feature aggregation approach to merge deep features and low-level features. The text images share as input to ResNet and extract features. The input image size is $R^{H \times W \times 3}$, and the output size is $R^{H \times W \times D}$. Here H is height, W is weight, and D is channel size. After extracting features (P_0, P_1, P_2, P_3), Convolutional Pyramid adopted a feature upsampling approach, which enhances the feature's representation ability (P_0^*, P_1^*, P_2^*). After upsampling, Extracted features and output of the Convolutional Pyramid merge at a different level.

Multi-level Feature Aggregation Module

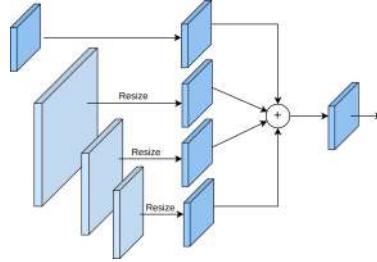


Fig. 3. Architecture of multi-level feature aggregation

The primary aim of Multi-level feature aggregation (MFA) is to unite low-level features with high-level features to enrich the visual and linguistic features from a different levels. The illustration of Multi-level feature aggregation is shown in Fig. 3. P_3 and combined features after upsampling share as input to the MFA module, which captures robust features of text content. The combined features resize to the same size as the size of P_3 using Conv2d. After resizing all layers, MFA combines these features with P_3 and shares them as input to the prediction module.

Parallel Prediction

Unlike previous approaches that separate the process of capturing visual and linguistic information into two distinct stages, we introduce a framework into a single unified architecture. LHFAN utilized a self-attention approach to the prediction task. The proposed method utilizes N transformer units, which have been shown to effectively handle long-term dependencies in recent computer vision tasks. To take into account the position of pixels, the method employs position encoding. This approach utilizes transformer units for sequence modeling instead of only for language modeling, which eliminates the effect of the length of words. The model makes predictions for multiple characters simultaneously, using the previously processed information.

4 Experimental Results

4.1 Dataset

We utilize **SynthText** and **SynthText90K** datasets for training and six datasets to evaluate the framework (three regular (IC13, SVT, IIIT5k) and three irregular datasets(IC15, SVTP, CUTE80)). SynthText [15] dataset consists of 8 million synthetic word samples. SynthText90K [8] dataset include 9 million images covering 90k English word instances. The proposed method is evaluated using the **IC-DAR 2013** (IC13), **ICDAR 2015** (IC15), **IIIT 5K-Words** (IIIT5k), **Street View Text** (SVT), **Street View Text-Perspective** (SVTP), and **CUTE80** (CUTE) datasets.

4.2 Implementation Details

Our model is built using ResNet as its backbone, and we specifically set the stride default for the initial stage and increase it for other stages. The weights are initialized using the default initialization method. In our experiments, we used an image size of 128x32 and employed a data augmentation process that included random rotation, color jittering, and perspective distortion. We conducted the experiments on an NVIDIA GTX 3090 GPU with a batch size of 192. The Adam optimizer is used to train the network completely, with a learning rate of 1e-4, in an end-to-end fashion and uses the cross-entropy loss to calculate the loss. The recognition system is designed to cover 37 characters, including letters a-z, numbers 0-9, and an end-of-sequence symbol.

Table 1. Scene text recognition accuracy compared with other STR methods on six standard benchmarks.

Method	IC13	SVT	IIT5K	IC15	SVTP	CUTE
Chu et al. [1]	95.50	95.50	96.60	84.40	89.90	90.30
Loginov rt al. [2]	96.80	94.70	93.5	80.20	89.90	-
ABINet[5]	97.30	93.50	96.20	86.00	89.30	89.20
Cheng et al. [7]	93.30	85.90	87.40	70.60	-	-
PIMNet[30]	95.4	94.70	96.70	85.90	88.20	92.70
S-GTR [16]	95.80	94.10	96.80	87.90	84.60	92.30
CornerTransformer[21]	96.40	94.60	95.90	86.30	91.50	92.00
VisionLAN [12]	95.80	95.70	91.70	83.70	86.00	88.50
SVTR-L[14]	97.20	91.70	96.30	86.60	88.40	95.10
LevOCR[9]	96.85	92.89	96.63	86.42	88.06	91.67
Zhang et al.[27]	97.70	94.30	96.50	85.40	89.30	91.30
MGP-STRF[29]	97.32	94.74	96.40	87.24	91.01	90.28
MVLT[19]	97.30	94.70	96.80	87.20	90.90	91.30
Ours*	97.00	96.01	96.81	88.40	92.28	95.20

4.3 The effectiveness of LHFAN

We evaluate the LHFAN approach to previous top-performing methods on six commonly utilized datasets, as displayed in Table 1. Generally, approaches that consider the intricacies of language tend to perform better compared to those that do not. The LHFAN approach outperforms both language-based and language-free methods by effectively utilizing both low-level features and linguistic information from different levels to enrich the recognition performance on all six benchmarks. As shown in Fig. 4, The proposed method effectively recognizes text from some challenging images, such as degraded, low-quality, and irregular images. The LHFAN approach particularly excels on regular datasets, achieving



Fig. 4. Recognition performance of the proposed method on some challenging examples. The predictions of LHFAN represent below of images.

a 0.31% and 0.01% improvement on the SVT and IIIT5K datasets. On irregular datasets, the LHFAN also shows significant improvement, with increases of 0.5%, 0.78%, and 0.1% on IC15, SVTP, and CUTE, respectively.

5 Conclusion

In this paper, we introduce a novel scene text recognition approach on multi-level feature fusion and enhancement of linguistic information, which can improve the recognition accuracy of distorted and low-quality images. The recognition accuracy is improved on indistinct and confusing text images by effectively increasing the spatial information and adding different levels of semantic information and visible features. Future work on the suggested method should focus on handling complex scenarios of erroneous text appearances.

6 Acknowledgements

This research was supported by the Bio & Medical Technology Development Program of the National Research Foundation (NRF) & funded by the Korean government (MSIT) (NRF-2019M3E5D1A02067961) and by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2018R1D1A3B05049058 & NRF-2020R1A4A1019191).

References

1. Chu, X., Wang, Y., Shen, C., Chen, J., Chu, W. (2022). Training Protocol Matters: Towards Accurate Scene Text Recognition via Training Protocol Searching. arXiv preprint arXiv:2203.06696.
2. Loginov, Vladimir. "Why You Should Try the Real Data for the Scene Text Recognition." arXiv preprint arXiv:2107.13938 (2021).

3. Chen, J., Li, B., Xue, X. (2021). Scene text telescope: Text-focused scene image super-resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 12026-12035).
4. Wenyang Hu, Xiaocong Cai, Jun Hou, Shuai Yi, and Zhiping Lin. GTC: guided training of CTC towards efficient and accurate scene text recognition. In The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, pages 11005–11012. AAAI Press, 2020.
5. Fang, Shancheng, et al. "Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.
6. Anhar Risnumawan, Palaiahankote Shivakumara, Chee Seng Chan, and Chew Lim Tan. A robust arbitrary text detection system for natural scene images. Expert Systems with Applications, 41(18):8027–8048, 2014.
7. Chen, X., Jin, L., Zhu, Y., Luo, C., Wang, T. (2021). Text recognition in the wild: A survey. ACM Computing Surveys (CSUR), 54(2), 1-35.
8. Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. NIPS, 2014.
9. Da, Cheng, Peng Wang, and Cong Yao. "Levenshtein OCR." European Conference on Computer Vision. Springer, Cham, 2022.
10. Alex Graves, Santiago Fernández, Faustino J. Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labeling unsegmented sequence data with recurrent neural networks. In William W. Cohen and Andrew W. Moore, editors, Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006, volume 148 of ACM International Conference Proceeding Series, pages 369–376. ACM, 2006.
11. Anand Mishra, Karteeck Alahari, and CV Jawahar. Scene text recognition using higher order language priors. In BMVC, 2012.
12. Wang, Yuxin, et al. "From two to one: A new scene text recognizer with visual language modeling network." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.
13. Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Trans. Pattern Anal. Mach. Intell., 39(11):2298–2304, 2017.
14. Du, Y., Chen, Z., Jia, C., Yin, X., Zheng, T., Li, C., ... Jiang, Y. G. (2022). SVTR: Scene Text Recognition with a Single Visual Model. arXiv preprint arXiv:2205.00159.
15. Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localization in natural images. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2315–2324, 2016.
16. He, Y., Chen, C., Zhang, J., Liu, J., He, F., Wang, C., Du, B. (2022, June). Visual semantics allow for textual reasoning better in scene text recognition. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 36, No. 1, pp. 888-896).
17. Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluis Pere De Las Heras. Icdar 2013 robust reading competition. In 2013 12th International Conference on Document Analysis and Recognition, pages 1484–1493. IEEE, 2013.

18. Kai Wang, Boris Babenko, and Serge Belongie. End-to-end scene text recognition. In 2011 International Conference on Computer Vision, pages 1457–1464. IEEE, 2011.
19. Wu, Jie, et al. "Masked Vision-Language Transformers for Scene Text Recognition." arXiv preprint arXiv:2211.04785 (2022).
20. Pan He, Weilin Huang, Yu Qiao, Chen Change Loy, and Xiaoou Tang. Reading scene text in deep convolutional sequences. In Dale Schuurmans and Michael P. Wellman, editors, Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA, pages 3501–3508. AAAI Press, 2016.
21. Xie, Xudong, et al. "Toward Understanding WordArt: Corner-Guided Transformer for Scene Text Recognition." European Conference on Computer Vision. Springer, Cham, 2022.
22. Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image superresolution. In IEEE Conf. Comput. Vis. Pattern Recog., pages 2472–2481, 2018.
23. Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In 2015 13th International Conference on Document Analysis and Recognition (ICDAR), pages 1156–1160. IEEE, 2015.
24. Xu, C., Wang, Y., Bai, F., Guan, J. and Zhou, S., 2022. Robustly Recognizing Irregular Scene Text by Rectifying Principle Irregularities. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 3061-3068).
25. Jiang, H., Xu, Y., Cheng, Z., Pu, S., Niu, Y., Ren, W., ... Tan, W. (2021, September). Reciprocal Feature Learning via Explicit and Implicit Tasks in Scene Text Recognition. In International Conference on Document Analysis and Recognition (pp. 287-303). Springer, Cham.
26. Zuzana B'ilkova and Michal Hradi' s. Perceptual license plate super-resolution with CTC loss. ^ J. Electron. Imaging, 2020(6):52–1, 2020.
27. Zhang, X., Zhu, B., Yao, X., Sun, Q., Li, R., Yu, B. (2022). Context-based Contrastive Learning for Scene Text Recognition. AAAI.
28. Trung Quy Phan, Palaiahnakote Shivakumara, Shangxuan Tian, and Chew Lim Tan. Recognizing text with perspective distortion in natural scenes. In Proceedings of the IEEE International Conference on Computer Vision, pages 569–576, 2013.
29. Wang, P., Da, C., Yao, C. (2022). Multi-granularity Prediction for Scene Text Recognition. In European Conference on Computer Vision (pp. 339-355). Springer, Cham. Rowel Atienza. Vision transformer for fast and efficient scene text recognition. In Josep Lladós, Daniel Lopresti, and Seiichi Uchida, editors, 16th International Conference on Document Analysis and Recognition, ICDAR 2021, Lausanne, Switzerland, 5-10, 2021, Proceedings, Part I, volume 12821 of Lecture Notes in Computer Science, pages 319–334. Springer, 2021.
30. Qiao, Z., Zhou, Y., Wei, J., Wang, W., Zhang, Y., Jiang, N., ... Wang, W. (2021, October). PIMNet: a parallel, iterative, and mimicking network for scene text recognition. In Proceedings of the 29th ACM International Conference on Multimedia (pp. 2046-2055).
31. Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. Open-Review.net, 2017.

32. Deli Yu, Xuan Li, Chengquan Zhang, Tao Liu, Junyu Han, Jingtuo Liu, and ErruiDing. Towards accurate scene text recognition with semantic reasoning networks. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020, pages 12110–12119. Computer Vision Foundation / IEEE, 2020.
33. Xin Tang, Yongquan Lai, Ying Liu, Yuanyuan Fu, and Rui Fang. Visual-semantic transformer for scene text recognition. arXiv preprint arXiv:2112.00948, 2021.
34. Yue He, Chen Chen, Jing Zhang, Juhua Liu, Fengxiang He, Chaoyue Wang, and Bo Du. Visual semantics allow for textual reasoning better in scene text recognition. In ThirtySixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022, pages 888–896. AAAI Press, 2022.
35. Rowel Atienza. Vision transformer for fast and efficient scene text recognition. In Josep Lladós, Daniel Lopresti, and Seiichi Uchida, editors, 16th International Conference on Document Analysis and Recognition, ICDAR 2021, Lausanne, Switzerland, 5-10, 2021, Proceedings, Part I, volume 12821 of Lecture Notes in Computer Science, pages 319–334. Springer, 2021.

Preliminary Study on Fish Tracking in Indoor Aquaculture through Deep Learning

Nguyen-Ngoc Huynh¹, Myoungjae Jun^{1,*}, Hang Thi Phuong Nguyen¹,
Choonsung Shin² and Hieyong Jeong^{1,*}

¹ Department of Artificial Intelligence Convergence, Chonnam National University

77 Yongbong-ro, Buk-gu, Gwangju 61186, Republic of Korea

² Graduate School of Culture, Chonnam National University

77 Yongbong-ro, Buk-gu, Gwangju 61186, Republic of Korea

{217019, 208258, cshin, h.jeong}@jnu.ac.kr, 21cnehemiah@gmail.com

Abstract. Due to food shortages, areas such as aquaculture intelligent farms are getting a lot of attention. However, many technologies still need to be solved to automatically control the growth process of randomly and infinitely moving marine organisms. Therefore, the economics of aquaculture requires the creation of systems that can do this automatic tracking of fish behavior. This paper demonstrates a software approach to fish tracking with deep learning. The result of experiments have nevertheless shown that our process is simply efficient and above all easy to implement in fish farms, although there were some restrictions.

Keywords: YOLO · DeepSORT · object detection · multi-object tracking · fish tracking.

1 Introduction

An important area of computer vision is target tracking. It is extensively employed in other industries, such as video monitoring, and aquaculture may soon follow suit. Fish tracking technology is a crucial tool for behavior observation in aquaculture. Monitoring fish behavior and growth through tracking enables the aquaculture sector to use its resources most. Fish tracking and behavior analysis can also be used to keep an eye on the environment that supports fish growth. Additionally, it can more effectively manage water quality, evaluate the health of cultured fish, and promptly address abnormal behavior. Additionally, precise feeding, disease detection, counting, and tracking of fish can all be done using fish-tracking techniques.

With the increasing demand for seafood resources and the rapid development of deep learning technology, the scale of aquaculture is constantly expanding. Therefore, the application of technology in practice is very necessary. The development of an application with the ability to detect and track underwater targets in real time greatly supports the aquaculture industry. However, in reality, it is almost difficult for humans to distinguish the fish due to its small size, transparent body, easy deformation, motion blur, and similar shapes [14]. Since then,

analyzing fish behavior from videos collected from fish farms is difficult even it can be called “impossible” in some ways.

In addition, the other three complex problems in fish tracking are occlusion, background interference, and morphology [7]. The first two problems frequently lead to disrupted trajectories by conflicting with the fish’s identification [15], [13]. The third circumstance results in poor tracking precision. (shown in Figure. 5).

Technologies for manual tracking are time-consuming and ineffective. To describe the actions of a single experimental object, extensive manual observations and labeling of image features are required. With improvements in computer performance and the effective use of convolutional neural networks in computer vision, learning-based computer vision technology offers a promising means of automating manual tracking [8]. The technique has most recently been used to automate various item-tracking systems.

Numerous manual observations are needed to completely characterize the actions of a single experimental object, labeling image features, and image features as well. Due to improvements in computer performance and the successful use of convolutional neural networks in computer vision, learning-based computer vision technology gives a workable method to automate manual tracking. The approach was most recently used to automate various item-tracking processes [3], [12], [2]. As object detection technology has advanced, tracking-by-detection technology has become the industry standard for multiple objects online monitoring because of its simple design and robust implementation. In the present study, we simulate a deep learning-based fish monitoring system. Finally, we assign the unique id to each fish during observation using deepSORT [11].

2 Related Works

Fish have been detected using a variety of techniques in order to keep track of their number and size. Various object detection techniques have been used to detect fish [1]. And to track fish size and count image processing and computer vision systems are considered. In order to track fish movement, tracking algorithms like optical flow and frame subtraction are done.

Many methods have been reported to detect fish to maintain tabs on their abundance and size. Fish detection was performed using a variety of object detection methods [1]. Technologies based on computer vision were considered to recognize fish size and number. In addition, tracking technologies like optical flow and frame subtraction were used to monitor the movement of fish.

Youssed Wageeh [10] proposed using euclidean tracking in fish farms to detect fish. Firstly, he used the Image Enhancement algorithm to improve unclear images. Then, the object detection algorithm and euclidean for tracking were used. However, pond fish videos had many disadvantages, such as blurred, low-quality videos resulting in poor recognition and switched ID (identification). Like Weiran Li [5], he introduced four sub-branches for object detection and Re-ID object extraction, extending ResNet-101 as the framework for object map ex-

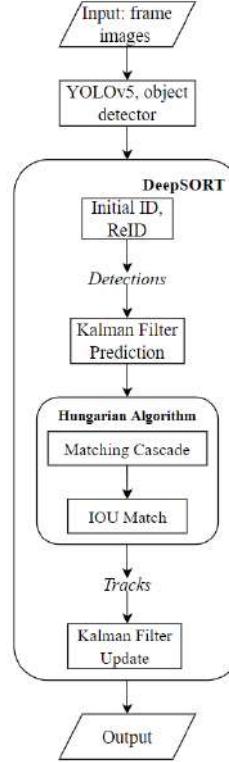


Fig. 1. The mechanism YOLOv5-DeepSORT operates. To find the target, YOLOv5 is provided input images (predict the bounding box of objects). Then, from the output of YOLOv5, REID requires a well-distinguishing feature embedding. Finally, the Kalman filter predicts the trajectory of each target. For the target's trajectory, the outputs of the YOLOv5 and Kalman filters are compared with the Hungarian algorithm for matching.

traction. However, long-term tracking performance still needed to be improved while swimming toward bright spots or pond shores.

According to our survey, improving the method of tracking multiple fish simultaneously through the software approach is necessary [6].

3 Methods

This section describes the general organization of the suggested fish-tracking method based on deep learning. First, the YOLO model is used for fish detection, and the DeepSort model is applied for fish tracking. Then, we mainly concentrate on presenting the DeepSORT tracking component, which includes using the ID appearance feature model, cascade matching, and IoU matching.



Fig. 2. Fish data set samples.

The difficulty of detection and tracking caused by dense targets and the re-identification of targets in fish farm chores led to the development of an intelligent fish recognition and tracking system. In Figure 1, the overall frame diagram is displayed. First, the location information (i.e., the center's x, y, width, and height), categorization details (class), and confidence of each target frame are acquired using YOLOv5 as the detector to extract feature information. Next, the detected results are entered into DeepSORT, where a previously predicted trajectory T_{pre} is obtained using the Kalman Filter predict module. The Hungarian Algorithm is then used to determine the degree to which the detected result of the current frame (D_t) matches the predicted track (T_{pre}). Finally, incorrect tracks are eliminated to complete fish tracking during tracking, and corrected tracks (D_{match}/T_{match}) are updated via the Kalman Filter update module.

4 Results

In this experiment, the Goldfish images are collected from YouTube videos. First, it has split frame by frame, and then we select high-quality images manually, selecting from blurry photos that do not show the subject well. Finally, the automatic decision algorithm helps us choose images from some candidates.

The 2,835 images that compose this research's data set were collected. The 595 images among the total images were chosen (ratio: 8:2) to make the test set, and the remaining images served as the training set. In Figure 2, an example of the dataset is shown.

Whole images with a resolution of 640×640 pixels are annotated by the Make-sense [9], the annotation tool, which is a free online tool for labeling photos in Computer Vision projects. The TXT annotation file in YOLO format is made. Moreover, it is used to feed into the YOLOv5 for training object detection and DeepSORT for tracking. A sample example of this tool is shown in Figure 3.

Table 1 truly configures the experiment as below: It is described in terms of hardware configuration software and libraries version.

The videos are tested with the trained model and show the effectiveness of YOLOv5 [4] and DeepSORT [11] for detection and tracking. The results are



Fig. 3. Labelled ground truth.

shown in Figure 4. The object prediction process is continuous in each frame. Furthermore, it helps DeepSORT get the coordinate information of the object, which is very useful in initializing the object's id because when the object detector misses, it will initialize multiple ids for one object.

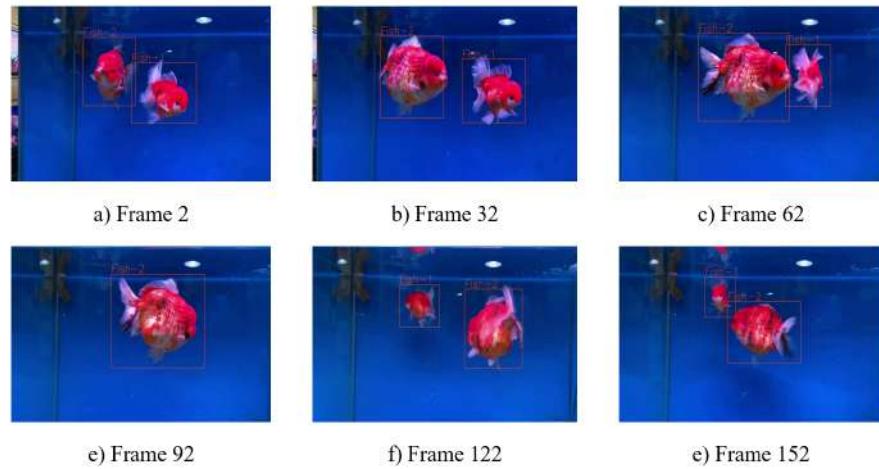
As the results show, the unique id for each object remains constant every 30 frames (the number of frames that DeepSORT tracks the object, if it disappears, more than 30 frames continue to be tracked and initialize a new id when it appears).

This study successfully recognized and tracked fish with unique identifiers, although the suggested method still had certain drawbacks.

1. Compared to genuine footage from the aquaculture farm, the data set used in this study is less varied in terms of posture, background, and image quality.
2. In the future, software that incorporates our algorithm for live video viewing of farms should be developed. Fish farmers have benefited immensely from the software's ability to tell them when fish are acting abnormally and to monitor the situation.
3. The number of fish in the video is limited to less than three fish to achieve good results with only one unique id during tracking.

Table 1. Experimental configuration.

Configuration	Parameters
Processor	Intel(R) Core(TM) i9-10900K
GPU	NVIDIA GeForce RTX 2070 SUPER
Operating system	Windows 10
Accelerated environment	CUDA 11.3 & CUDNN 8.2.1
Code editor	Visual Studio Code 1.72.1
Libraries	OpenCV 4.6.0, torch 1.12.1, numpy 1.21.6

**Fig. 4.** The efficiency of DeepSORT in different frames.

5 Consclusion and future works

To predict as well as follow fish in this study, we could put forward for consideration in deep learning technological solution. YOLOv5 was an object detector to predict the coordinate of fish in the frame by frame, and DeepSORT, as a tracker, gave the unique id for each fish. We used the Goldfish data set to demonstrate the effectiveness of our method. This study motivated us to do future research with videos from fish farms. The proposed method was simple and easy to deploy on the farm. Through this method, fish detection and tracking could be monitored conveniently.

Acknowledgements This research was supported by Korea Institute of Marine Science & Technology Promotion (KIMST) funded by the Ministry of Oceans and Fisheries, Korea (No. 20220596, Development of Digital Flow - through Aquaculture System).

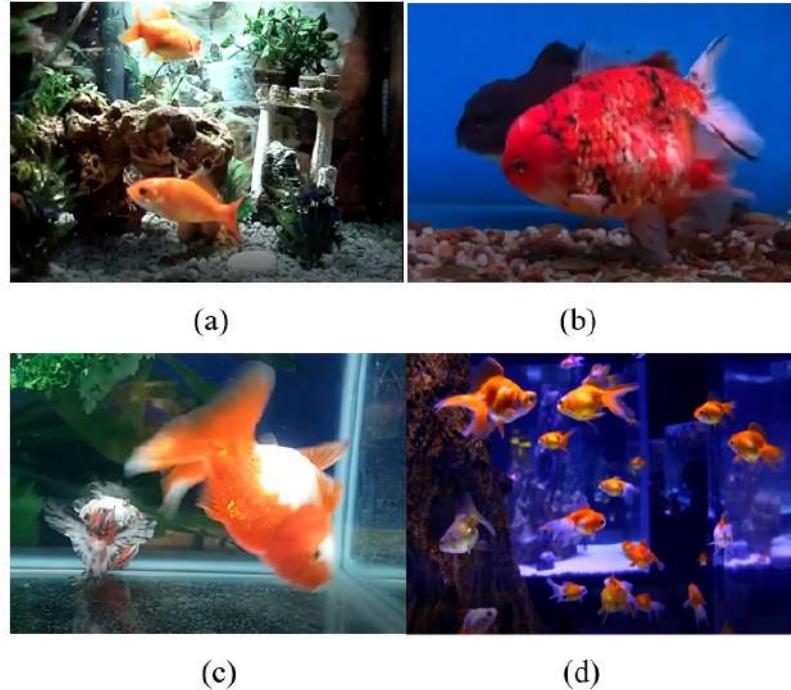


Fig. 5. Example of challenges. (a) Background complicated, (b) Overlap objects. (c) Inverted poses. (d) Similar objects.

The authors would like to express a huge thank you to the BK21 Plus program at Chonnam National University, through the National Research Foundation, funded by the Ministry of Education of Korea.

References

1. Duggal, S., Manik, S., Ghai, M.: Amalgamation of video description and multiple object localization using single deep learning model. In: Proceedings of the 9th International Conference on Signal Processing Systems. pp. 109–115 (2017)
2. Egi, Y., Hajyzadeh, M., Eyceyurt, E.: Drone-computer communication based tomato generative organ counting model using yolo v5 and deep-sort. Agriculture **12**(9), 1290 (2022)
3. Hou, X., Wang, Y., Chau, L.P.: Vehicle tracking using deep sort with low confidence track filtering. In: 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). pp. 1–6. IEEE (2019)
4. Jocher, G., Chaurasia, A., Stoken, A., Borovec, J., NanoCode012, Kwon, Y., Michael, K., TaoXie, Fang, J., imyhxy, Lorna, Yifu), Wong, C., V, A., Montes, D., Wang, Z., Fati, C., Nadar, J., Laughing, UnglvK-

- itDe, Sonck, V., tkianai, yxNONG, Skalski, P., Hogan, A., Nair, D., Strobel, M., Jain, M.: ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation (Nov 2022). <https://doi.org/10.5281/zenodo.7347926>
5. Li, W., Li, F., Li, Z.: Cmftnet: Multiple fish tracking based on counterpoised joint-net. *Computers and Electronics in Agriculture* **198**, 107018 (2022)
 6. Luo, J., Han, Y., Fan, L.: Underwater acoustic target tracking: A review. *Sensors* **18**(1), 112 (2018)
 7. Mei, Y., Sun, B., Li, D., Yu, H., Qin, H., Liu, H., Yan, N., Chen, Y.: Recent advances of target tracking applications in aquaculture with emphasis on fish. *Computers and Electronics in Agriculture* **201**, 107335 (2022)
 8. Ravoort, P.C., Sudarshan, T.: Deep learning methods for multi-species animal re-identification and tracking—a survey. *Computer Science Review* **38**, 100289 (2020)
 9. Skalski, P.: Make Sense. <https://github.com/SkalskiP/make-sense/> (2019)
 10. Wageeh, Y., Mohamed, H.E.D., Fadl, A., Anas, O., ElMasry, N., Nabil, A., Atia, A.: Yolo fish detection with euclidean tracking in fish farms. *Journal of Ambient Intelligence and Humanized Computing* **12**(1), 5–12 (2021)
 11. Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. *CoRR* **abs/1703.07402** (2017), <http://arxiv.org/abs/1703.07402>
 12. Wu, F., Jin, G., Gao, M., HE, Z., Yang, Y.: Helmet detection based on improved yolo v3 deep model. In: 2019 IEEE 16th International Conference on Networking, Sensing and Control (ICNSC). pp. 363–368 (2019). <https://doi.org/10.1109/ICNSC.2019.8743246>
 13. Xu, W., Matzner, S.: Underwater fish detection using deep learning for water power applications. In: 2018 International conference on computational science and computational intelligence (CSCI). pp. 313–318. IEEE (2018)
 14. Yang, L., Liu, Y., Yu, H., Fang, X., Song, L., Li, D., Chen, Y.: Computer vision models in intelligent aquaculture with emphasis on fish detection and behavior analysis: a review. *Archives of Computational Methods in Engineering* **28**(4), 2785–2816 (2021)
 15. Yang, X., Zhang, S., Liu, J., Gao, Q., Dong, S., Zhou, C.: Deep learning for smart fish farming: applications, opportunities and challenges. *Reviews in Aquaculture* **13**(1), 66–90 (2021)

Front Cover Image Database of Japanese Manga and Typeface Estimation of their Title

Shota Ishiyama¹, Kosuke Sakai¹, and Minoru Mori¹

Kanagawa Institute of Technology, Atsugi-shi, Kanagawa 243-0292, JAPAN
 mmori@ic.kanagawa-it.ac.jp

Abstract. Front cover design of books like Manga is one of the most important factors for appealing contents to users. Fonts used for the title in the front cover are carefully selected among a lot of ones for fitting selected fonts to the content and the design. However, this task to select fonts, that increases attractions of the front cover and are appropriate for human characters pictured in the front cover, is not easy. Few experienced designers or editors can do well. In this paper we try to estimate and recommend appropriate typefaces of fonts that seem to be appropriate for title fonts from an image of front cover of Manga and Light novels. To evaluate our framework, we gathered front cover images of Manga and Light novels, and created database that containing five kinds of images; front cover images with/without title fonts, whole-body images with/without title fonts, and face images. Each image has two types of label encoded from the count number of 5 typefaces used for title fonts. Experimental results using our database show that about 70% of typefaces are correctly estimated and suggest a strong relationship between fonts used for title and front covers.

Keywords: Front cover page · Manga · Light novels · Typeface · DNN.

1 Introduction

Lots of new contents and books of “Manga”, comics and graphic novels, and “Light novel”, Japanese young adult novels, have been continuously produced and published not even in Japan but in the world. When buying a book of Manga or Light novels, we usually select one on the base of its story, reviews, prices, character designs, and others. At least we all see a front cover page of each book. If a book of Manga or Light novels to buy is decided before, its front cover has nothing to do with sales or selections. But, if not decided, the impression and design of a front cover page seem to be a very important factor for sales on not only Manga books but other many books. Especially many Manga contents feature some human characters and the design of their character influence the popularity. Therefore, the front cover page needs to be designed for drawing attentions of users. One of important elements of the front cover design is the title font. Editors and designers carefully select kinds of fonts to bring out and appeal the charm of the Manga content. Fonts are not only medium of language

but a piece of art and impressions. From a lot of fonts, designers select fonts to fit the targeted Manga content and deliver its attraction to readers. Recently, though the number of amateur Manga and Light novels writers is increasing, they without knowledges about fonts cannot select appropriate fonts to fit their contents and design. If more appropriate fonts for the targeted Manga content or design can be easily selected, more attractive front covers with suitable fonts as its title can be produced and published.

In this paper we propose a framework that estimates suitable typefaces of fonts that fits the design of a front cover of Manga and Light novels and a database that consists of several types of images for this task. In detail, a lot of sets of the front cover image and its font information used as the title are created as the database and we evaluate our framework that estimates typefaces of fonts that fit front cover image or human characters contained in the front cover using our database. For validating relationship between fonts used as the title and human characters, we create whole-body images extracted from the front cover images and face images obtained from the whole-body image. Fig. 1 shows examples of front cover images of Manga and Light novels and fonts used as the titles in these images that we handle in this paper. Fig. 2 shows the overview of our framework that estimates typefaces included in an image using DNN. The paper is organized as follows: Section 2 provides related works about relationships between fonts and design of books or signboards. Our database and proposed method are described in Section 3. Section 4 reports evaluation experiments and discusses experimental results. Section 5 summarizes this paper and lists future works.



Fig. 1: Examples of front cover images of Manga and Light novels. (a) Typeface: Designed, Font name: Hasetoppo [for 17 Hiragana characters], Typeface: Designed, Font name: Takahand [4 Kanji characters] [1]. (b) Typeface: Mincho, Font name: A1 Mincho [for all characters] [2].

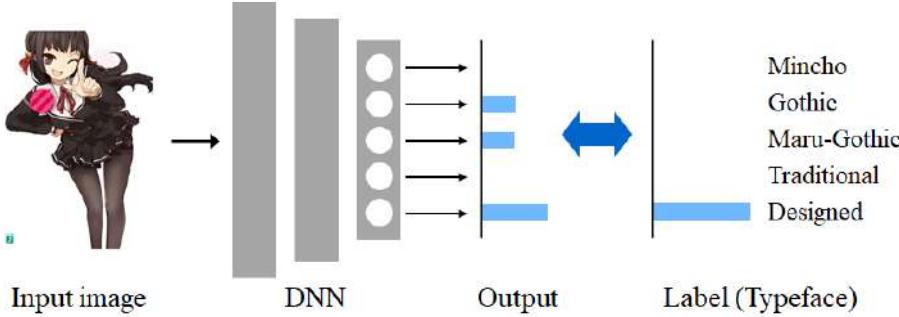


Fig. 2: Overview of our framework.

2 Related Works

Shinhara et al. analyzed that how genres of English books effect selections of colors of their front covers and fonts used as their book title [3]. They split the genres of English books into 32 kinds of groups, and font types were split into 6 typefaces. They validated the performance of genre estimation from font typefaces. Their experimental results showed that there was the relationship between the genre and font typeface. Also, books with similar colored front covers are in the same or similar genres and font type groups. In [5], they validated a relationship between fonts used in signboards and impressions of shops with these signboards. 7 signboards with each different type of fonts were prepared as test images. Users selected genres of eateries or restaurants from the impression based on appearances of each font. Experimental results showed that users often imagined different genres of eatery from different types of fonts. The tendency on the font selection on the signboard of real eateries was very similar to that obtained in their experiments. From these researches, we can say that impressions felled from designed linguistic substance like front covers of books and signboards has a strong relationship between fonts selected in them. On books of Manga and Light novels, the design of front covers seems to have some degree of relationship with fonts used in these front covers. Therefore, as one of design analyses, the estimation result of font from front cover pages enables us to easily select fonts that suit human character pictured in front cover pages for automatic or semi-automatic book design.

3 Proposed Method

This section describes our database that consists of front faces from Manga and Light novels, typefaces used in front faces, several kinds of label information for each data. And we explain a DNN model as our estimation framework used in our experiments.

3.1 Database of front cover images

For validating the typeface estimation on front cover images from Manga and Light novels, we gathered hundreds of front cover images from many books of them. Here, it's too difficult to obtain detailed information of font such as a concrete font name and a kind of typeface used in front covers only by observing font appearances. Therefore, we selected front cover images containing title fonts whose detailed information as their concrete font name and typeface information. Such books are introduced in several books that describe about designing and editing works of Manga and Light novels [6–11]. In this paper, to evaluate not only the relationship between fonts and the whole image of front cover but also that between fonts and whole-body image of human character contained in the front cover or a face of such human characters, we gathered only front cover images that involve human characters. The number of collected images is finally 581, and the number of unique titles is 227. As mentioned above, to investigate the relationship between fonts used in the front cover and whole bodies or faces too, we extracted a part of human characters and a part of faces from each front cover image. On front cover images and whole-body images from human characters, we erased font parts by using Adobe Photoshop. No Face images involve font images. Finally, we created five types of dataset that are front cover images with/without fonts, whole-body images with/without fonts, and face images. Fig. 3 shows examples of each type of images.

3.2 Typefaces of fonts used in title

This paper discusses only Japanese fonts as targets. Japanese fonts consist of more than 1,000 kinds. And designers sometimes originally adjust the shape of existing font design for suiting fonts to the design of front covers. Therefore, numbers of kinds of Japanese fonts exist. Moreover, some different kinds of fonts have very similar shape. From such reasons, estimating the concrete font name seems to be impossible. In this paper, we try to estimate not the detailed information such as concrete font name but the typeface name of each font as the general information. On the basis of Japanese font analysis in [12, 13], we classify each font into 5 typefaces; “Mincho”, “Gothic”, “Maru-Gothic”, “Traditional”, and “Designed”. And we estimate a class of typefaces of fonts used from each image such as front cover, whole body, and face. Fig. 4 shows examples of 5 typefaces mentioned above.

3.3 Label

A label information of each image contains the name of typeface used as title fonts and their number of counts. Here, some titles have multiple fonts and typefaces as shown in Fig. 1. Therefore, the label of several image has counts on multi typefaces among five typefaces mentioned above. To clarify such conditions on title fonts used in front cover of books, we investigated 581 images we gathered. Table 1 shows the number and ratio of images including only one typeface and



(a) Front cover image with title fonts (b) Front cover image with no title fonts



(c) Body image with title fonts (d) Body image with no title fonts (e) Face image (enlarged)

Fig. 3: Examples of front cover and several part images. JTC Janken font (designed typeface) is used [5].

あああ あああ あああ

(a) Mincho (b) Gothic (c) Maru-Gothic

あああ あああ

(d) Traditional (e) Designed

Fig. 4: Examples of each typeface. Each character expresses Japanese Hiragana.

that containing multiple typefaces for their labels. Table 2 provides the number and ratio of images on the main typeface of title fonts used in each image. Table 1 shows that there is a certain number of images with multiple typefaces and we cannot ignore such images. Table 2 gives that designed typefaces are often selected for the front cover of Manga and Light novels and traditional typefaces rarely are used. They seem to be common and acceptable because of characteristics of such genres. On the other hand, the count number of images containing the Mincho typeface as main title fonts is more than the sum of images including Gothic or Maru-Gothic. This is unexpected at least for us.

Table 1: Number of front cover images containing single or multiple fonts.

Num. of font typefaces	Num. of images	Ratio [%]
Single	505	86.9
Multiple	76	13.1

Table 2: Number of images for each typeface.

Typeface	Num. of images	Ratio [%]
Mincho	211	36.3
Gothic	91	15.7
Maru-Gothic	76	13.1
Traditional	20	3.4
Designed	183	31.5

On the basis of labels mentioned above, we encode each label information into two other types of label and adopt our evaluation method for each type of label. The first one is so-called one-hot encoding; a typeface with the most counts has one and the other typefaces have zero. This label has one-hot vector and this type of label are usually used for the multi-class classification task. In this paper we call this type of label “hard label”. In the evaluation step, when a typeface with the highest probability in an output is same as that with one hot value, the estimation of typeface is regarded as correct. The other type of label consists of ratio values for each typeface; each ratio is computed by dividing count number of each typeface by the total count number among all the typefaces. So, if an original label has counts for multiple fonts, this kind of label has values on multiple positions. We call this type of label “soft label” in this paper. In the evaluation, we compare a typeface of the highest probability in an output with that with the highest label value. If it’s same, the estimation is regarded as correct. Table 3 shows an original label information and 2 kinds of label that are encoded from the original one in our experiment.

Table 3: Examples of original label and 2 kinds of labels used in our data.

Typeface	Original	Hard	Soft
Mincho	0	0	0
Gothic	6	1	0.6
Maru-Gothic	0	0	0
Traditional	0	0	0
Designed	4	0	0.4

3.4 Estimation Model

As a framework for estimating typefaces used in each front cover image, we use Deep Neural Networks (DNN). In this paper, we exploit a pre-trained DNN model trained by using many data and fine-tune such a model because few training images tend to obtain a model that over-fits training images. As a pre-trained DNN model, we use the VGG16 model [14] that were trained by the ImageNet dataset that contains 14 million images of 1,000 classes. We exploit only convolution layers of this pre-trained DNN model as a feature extraction part and trained new dense layers as a classification part by the use of training data mentioned in 3.1 in the training process. Moreover, 3 convolution layers in VGG16 were re-trained in the fine-tuning process. In the training using data with hard or soft labels, dense layers with the Softmax function were used in the final output layer. Fig. 5 shows the structure of the DNN model used in our experiments.

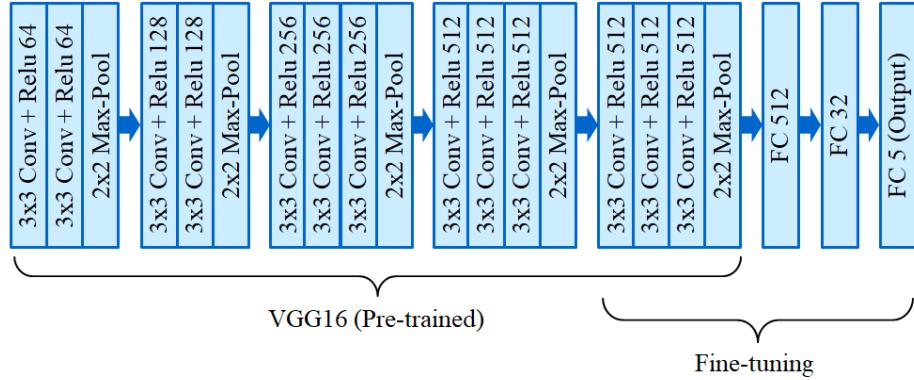


Fig. 5: The structure of our DNN model.

4 Experiments

In this section we describe evaluation experiments for validating and analyzing the performance of our approaches to estimate typefaces of title fonts in the front cover of Manga and Light novels.

4.1 Experimental Set-up

As experimental data, we split each data of front cover, whole body, and face 581 images into 4 sets. 436 images from 3 sets were used as training data and 145 images from another set were used as test one. We adopted cross-validation; therefore, each set were used as test data once in the rotation and 4 tests were totally carried out. Each result shown below was calculated as the average value among 4 tests.

Each data with hard or soft label were trained and tested as a multi-class classification problem. Thus, the cross entropy was used as their cost function for their training.

On the basis of preliminary experiment results, we set several experimental set-ups as follow. The number of epochs is 300. The batch size is 1 because of few training data. The initial learning late is 2.0e-5. Adam was used as the optimizer. The number of dense layers is 3, and the numbers of units in each dense layer are 512, 32, and 5, respectively. The cross-entropy was used as the loss function. As data augmentation, basic techniques as rotation, horizontal shift, vertical shift, and horizontal flip were adopted.

Each image is normalized into 256×256 pixels as a default size. Front cover images and body images are normalized into several sizes and details are described later.

4.2 Experimental Results

First of all, we describe experimental results using data with hard labels and default settings. Typeface classification rates within the top 1 and 2 for each image of the front covers with or without title fonts, whole body with or without title fonts, and face are shown in Table 4. Here, top 2 means the cumulative classification rate within the top 2. The reason we show the top 2 result is that some images contain multi fonts, so only one candidate output as an estimated typeface is not seemed to be so fair. Also, multiple candidates seem to be helpful and useful for designers and editors to select suitable fonts for front cover design. Table 4 gives that significant differences between images with title fonts and without title fonts. The difference of about 10% between 2 kinds of front-cover images can be regarded as the advantage that DNN models have learned the typeface information from not only the design but also fonts themselves directly. Therefore, about 63% given by front cover images without title fonts seems to be a standard estimated accuracy. Result obtained by face images is almost same as that by front cover images without title and this is a little surprise. One reasons of this result is that a facial impression may affect the selection

of fonts used for the title design in front covers. Fig. 6 shows an example that the typeface used as title font was correctly estimated from an only face image. This example seems to provide the impression of font shape is similar to that derived from the human character's face. Other reason seems that front cover images was normalized into too small ones. In the next experiment, we validate accuracies using images with another size or aspect ratio. On the other hand, the estimation using whole-body images gives lower rates than other types of images. The lower results obtained by whole-body images seem to be caused by variations of composition in whole-body images (See Appendix Fig. 9 and Fig. 10). The difference between 2 kinds of whole-body images, about 4%, is smaller than that between front cover images, about 10%. This reason is that some original whole-body images have no title fonts. Also, as shown in Table 1, several data used in the experiment have multiple typefaces. Therefore, obtaining high rates is very difficult. The fact that all the rates within the top 2 are under 90% despites of a 5 class-classification problem indicates that this task is not very easy.

Table 4: Estimation rates [%] on hard labels.

	Font cover w/title	Front cover w/o title	Body w/title	Body w/o title	Faces
Top 1	72.5	62.8	62.3	58.5	63.0
Top 2	87.9	84.3	79.9	78.8	80.7



Fig. 6: Example of a face image that a typeface of title fonts used in front cover is correctly estimated [15]. Typeface: Mincho. Font name: Marumei Old.

Next, as mentioned above, we validate other image sizes or ratios that are different from default sizes. Table 5 shows estimation rates for every type of images with several image sizes or ratios. On the basis of image sizes in our database, 1.0, 1.5 and 2.0 as ratios of an image height to an image width were used for front cover images. 1.0, 2.0, and 4.0 as ratios of an image height to an image width were assessed for whole-body images. Only face image was normalized with keeping an image ratio is 1.0. Table 5 provides that bigger images have obtained better results. In particular, the increase of rates for front cover images, about 10%, are significant. This suggests that the default size is too small for such images and often lose important information including font one. And normalized images with similar ratio to original's one have almost same rates as enlarged square images. This gives that the characteristics as crucial information for each font class are retained if an original ratio of each image is changed.

Table 5: Estimation rates [%] on hard labels in several image sizes and ratios.

Image height	Image width	Font cover w/title	Front cover w/o title	Body w/title	Body w/o title	Faces
256	256	72.5	62.8	62.3	58.5	63.0
384	256	76.9	67.1	-	-	-
384	384	78.5	65.6	64.9	63.3	62.5
512	128	-	-	60.4	58.7	-
512	256	79.0	70.6	63.3	62.3	-
512	512	81.8	70.1	65.2	62.5	65.4
768	512	80.0	68.0	-	-	63.5
768	768	80.4	72.6	68.5	65.2	63.5
1,024	256	-	-	68.3	64.2	-
1,024	512	81.4	70.4	65.9	64.4	-

Then, we compared experimental results using images with hard labels and soft ones. Table 6 shows estimation rates for 5 kinds of image with each label. Rates with soft labels are obtained using same image sizes/ratios that provided the best rates on hard labels. Compared to results with hard labels, all the results on soft labels are a little lower than those on hard labels. These results suggest that the expression by soft label encoding is not appropriate to our experimental data that have multi classes.

Table 6: Estimation rates [%] on hard and soft labels.

Label	Font cover w/title	Front cover w/o title	Body w/title	Body w/o title	Faces
Hard	81.8	72.6	68.5	65.2	65.4
Soft	78.2	72.1	66.3	63.9	64.6

Finally, we analyzed error results and causes. Main mis-estimations are that Gothic or Maru-gothic typefaces were classified into Designed one. Some kinds of Designed fonts are very similar to Gothic or Maru-Gothic fonts. To solve this kind of errors, we need to introduce another scheme or information.

5 Conclusions

In this paper we have created the database that consists of about 580 images of front cover of Manga and Light novels books, and have proposed the framework for estimating typefaces of title fonts designed in such images for recommending the selection of suitable fonts for the title design. Our database contains five types of images; two kinds of whole front cover images of each books, two kinds of whole-body images extracted from a human character pictured in front cover images, and a character face image that are a part of a human character one. The difference between two kinds for front cover images and whole-body ones is with/without title fonts. In our experiments, we have estimated a kind of typeface of title fonts used in front covers from each type of images. We have exploited the pre-trained DNN model for the font typeface estimation and criteria for each label. Experimental results have shown that the design of front cover of books and human characters contained in the front cover have strong correlation with font typefaces selected for their front cover. The top rate of estimating kinds of typefaces using images without title fonts was about 72% on front cover image and about 65% for whole-body and face images. From these results we can say that estimating typefaces of title fonts in the front cover will enable us to ease the selection of suitable fonts for front cover design by providing candidates in the near future.

Future works are to gather more samples of cover front pages of Manga and Light novels for increasing training samples, correct the imbalance of data size among typefaces, and estimate not only kinds of typeface but concrete font names.

References

1. Suzuki, D., Uru, D.: OniAni. vol.1, Media Factory (2010)
2. Haruba, N.: The Quintessential Quintuplets. vol.7, Kodansha (2019)
3. Shinhabara, Y., Karamatsu, T., Harada, D., Yamaguchi, K., Uchida, S.: Serif or sans: visual font analytic on book covers and online advertisements. In Proceedings of International Conference on Document Analysis and Recognition (2019)
4. Imakawa, S., Kodoi, T.: What kind of restaurant does the characters on the signboards conjure up?, Graduate School of Education Bulletin, Hiroshima University, vol.65, pp.249–256 (2016)
5. Akime, J., Morisawa, H.: Otome-Ge no Koryakutaishou ni Narimashita..., vol.2, Ascii Media Works (2012)
6. BNN Editorial Department.: The design of Light Novels, BNN Shin-sha (2018)
7. Sannoumaru, S., Yuzuki R.: Logotype! Ritto-sha, A5 edition (2015)

8. MdN Editorial Department.: The Graphics Design of Manga & Anime, MdN EX-TRA vol.1, MdN Corporation (2014)
9. MdN Editorial Department.: The Graphics Design of Manga & Anime, MdN EX-TRA vol.2, MdN Corporation (2015)
10. Nichibou Editorial Department.: The New Comics Design, Japan Publicaitons Inc (2020)
11. Nichibou Editorial Department.: The New Comics Design 2, Japan Publicaitons Inc (2021)
12. Morisawa + DESIGNNING Editorial Department.: Type and Font Book, Mainichi Communications (2010)
13. Date C.: Moji-no-Kihon, Graphics-sha (2020)
14. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In Proceedings of International Conference of Learning Representation (2015)
15. Natsumi, K., Shizuki, M.: Hazakura ga Kita Natsu, Ascii Media Works (2008)

Appendix

Other examples of 2 kinds of front cover images, 2 kinds of whole-body images, and face images in our database are shown below.



Fig. 7: Examples of front cover images with title fonts.



Fig. 8: Examples of front cover images with no title fonts.



Fig. 9: Examples of whole-body images with title fonts.



Fig. 10: Examples of whole-body images with no title fonts.



Fig. 11: Examples of face images (enlarged).

Robotics Education under Pandemic Lockdown Situation.

Vicente González¹, Kelvin Kung¹, Danilo Cáceres-Hernández^{1,2}, and Kang-Hyun Jo³

¹ Facultad de Ingeniería Eléctrica, Universidad Tecnológica de Panamá, Panamá, Panamá

² Sistema Nacional de Investigación (SNI), SENACYT, Panamá, Panamá

³ Intelligent Systems Laboratory, Graduate School of Electrical Engineering, University of Ulsan

{vicente.gonzalez, kelvin.kung, danilo.caceres}@utp.ac.pa,
acejo@ulsan.ac.kr

Abstract. The closure of universities due to the COVID-19 pandemic triggered or forced worldwide to change from traditional teaching and learning systems into an online learning systems. In that sense, the main goal of this project is to present ongoing results of the implemented strategy used to maintain electrical engineering education classes during the lockdown. The main concerning of this abrupt change are the high related cost that it may incur switching from one to one learning to online learning. The presented strategy relies on cameras, proved that with low-cost accessible equipment it is a viable option to provide remote control to existing electrical and/or electromechanical equipment. To ensure teaching activities outside of universities, classroom must support hybrid education, for example: by providing wireless connectivity, using modern programming language, correctly mapping functions between the hardware, software and video access. The obtained results indicated that such control may be applied to both, the academy as well as the industry sector.

Keywords: Robotics Education · Online Engineering Education · Hybrid Education · COVID-19.

1 Introduction

With COVID-19, the most productive sector was forced to remote work [1, 2]. Service focused sector successfully made a fast transition to remote work, using existing platforms such as Zoom, WhatsApp, Teams, etc. Nonetheless, many of the industries where machine control was required experienced difficulties in making the transition to working online. It happened mainly because their equipment did not have remote control capabilities. On the academics [3, 4], students did not have the opportunity to interact with experimental equipment. Tele-operated applications [5–7] exists, but mostly to be used within universities facilities, without taken into the account how expensive remote applications can

be for universities. On the other hand, during the pandemic most of the solution were focused to fight COVID-19 [8, 9] more than supporting the education sector. To solve this issue, a mobile robot that provided remote working capabilities was developed learning and teaching purposes. The article is structured as follows: Section II describes the proposed architecture and its functionalities. Section III describes the challenges of the introduced strategy (forced by COVID-19) from the student's point view. Section IV concluded the presented strategy.

2 System Overview

Figure 1 shows the robot designed during the COVID-19 to ensure learning and teaching education. Both chassis and sensors cases of the V2 robot were designed using 3D printer. The robot included the following devices and sensors: eight ultrasonic sensors (US - [HC-SR04]), one Inertial Measurement sensor with six degrees of freedom (IMU BNo055 9DoF), two speed encoders (Daoki GR-US-227), one single-board computer (SBC - Orange Pi Zero) and a micro-controller (MCU - [Teensy 3.2 [ARM Cortex M4 + 32 bits, 72 MHz, 64 KB SRAM]] & Seeeduino XIAO [ARM Cortex M0 + 32 bits, 48 MHz, 64 KB SRAM]]), two direct current (DC) motors working, a Dual H-bridge motor controller (MX1508), and a 10000mAh External Battery, see Figure 1.

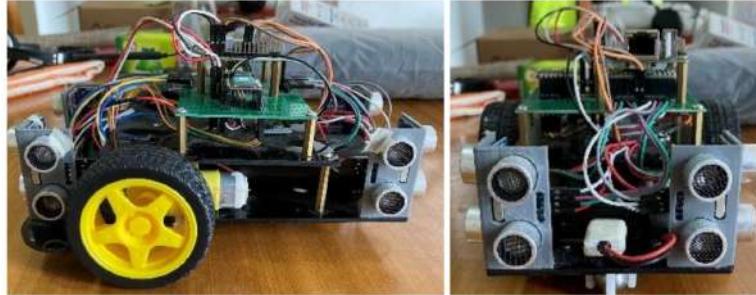


Fig. 1. V2 Robot Views. Left image shows the lateral view, Right image shows the front view.

Motors, sensors, IMU, MCU's, and SBC connections are showed in Figure 2. The Teensy MCU receives the data from the US (distance information), while the Seeeduino receive the data from the speed encoders as well as send data to the h-bridge motor controller. The SBC send and received data from the MCU's and from the IMU. As the robots include a set of multiple devices connected on the SBC, the communication is done by using I^2C connection protocol, the implementation was performed using two buses. The implementation of this setting allows the system get the data from the IMU in a faster and more efficient way, avoiding the time gap given by the communications between the master and

the slaves. The system was designed as follows: the SBC as a single master, where the MCU's and the IMU were setup as slave devices, see Figure 3.

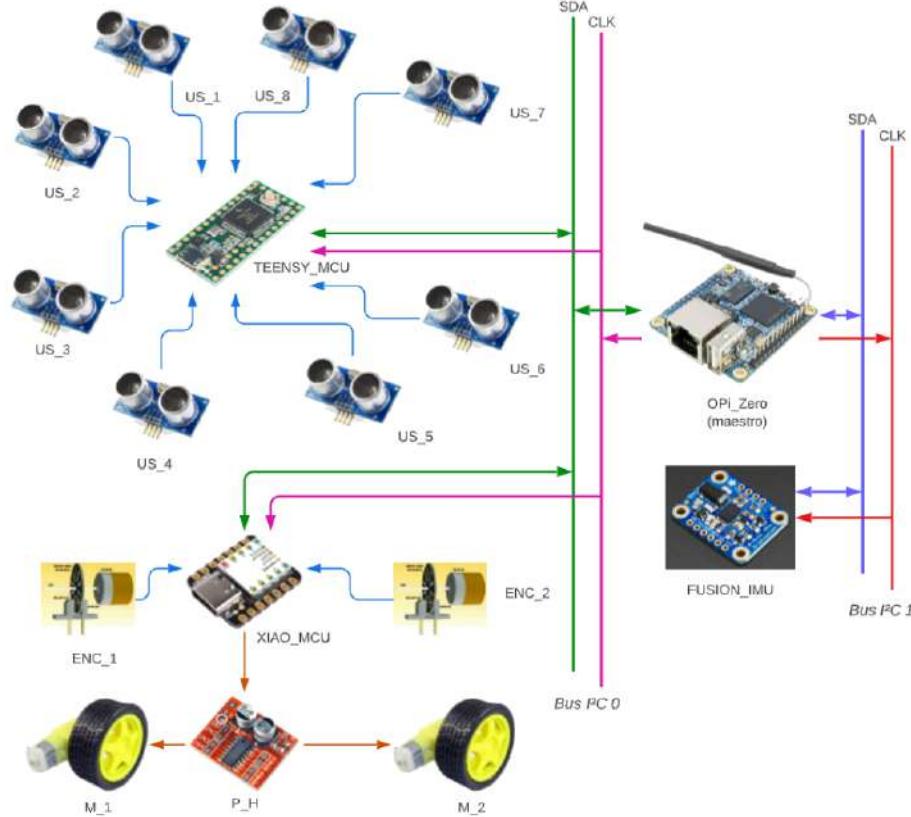


Fig. 2. V2 Robot Hardware Architecture

The operating system is Armbian based on the Debian family systems. It was chosen due to the features for the SBC and MCU's used in the V2 Robot. To get access to the system, Secure Shell (SSH) should be used. The following set of software were used: Python Interpreter, Linux (Armbian), OpenSSH, Python Toolbox to deal with: serial communication, GPIO, arithmetic operation, file handling, and time. To this end, a set of 4 applications were developed:

- Android teleoperation control with a basic login system.
- Web Django multipurpose application.
- Automated features extraction using Proportional-Integral-Differential controller (PID) model.
- Autonomous navigation algorithm

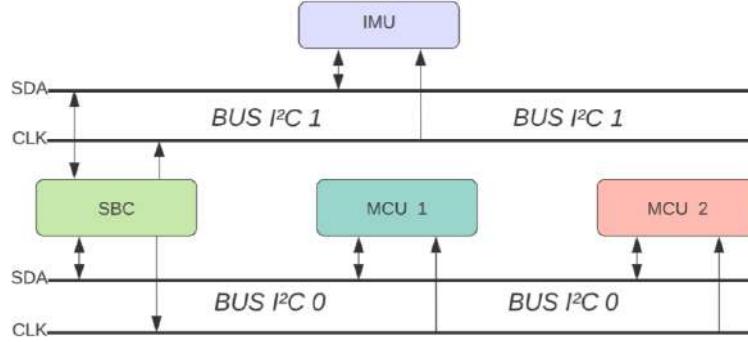


Fig. 3. V2 Robot Hardware Architecture

It should be emphasized for the rest of the document the presented idea is focused on the first application.

2.1 Android Teleoperation Control with a Basic Login System

In order to implemented this application Apache Web service was installed in the SBC. The main task was to execute the instructions given by users that were connected to the internet. The internet requests could be motion instruction given to the robot or requesting ultrasonic data information from the robot to the users. A Public Internet Protocol (IP) set in a router allows the access to the SBC, see Figure 4.

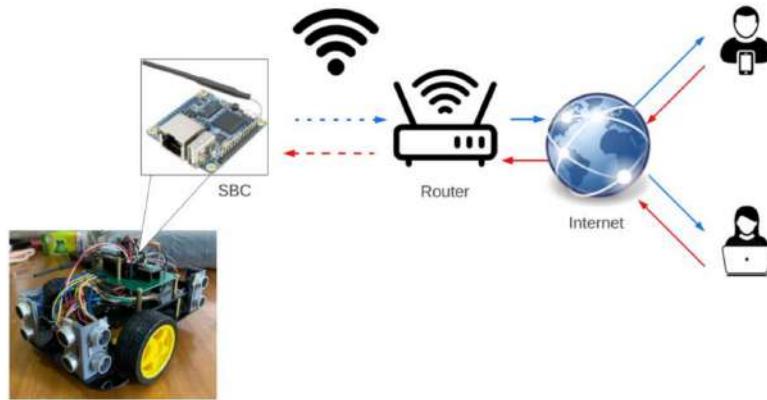


Fig. 4. Robot teleoperation system strategy via Internet

The graphical user interface (GUI) was developed under Android, see Figure 5. The interface allows users to get tele-operated or get data information from the

V2 Robot. The server request information every 0.5 seconds from the ultrasonic sensors. Then, the information is sent to the mobile device screen. To perform a motion, users should send a number of defined pulse and the period of the signal. This strategy was performed due to the latency problem.

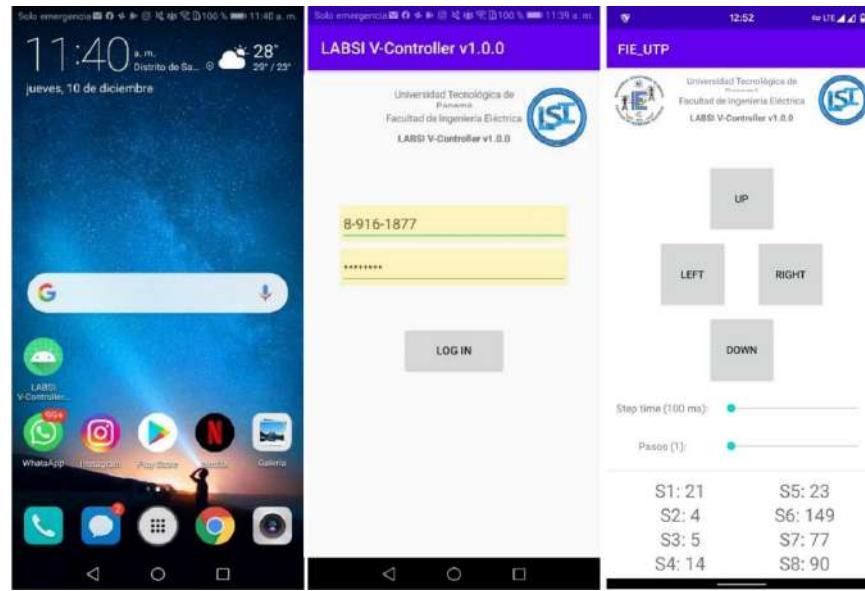


Fig. 5. Android Teleoperation Control GUI.

3 Discussion

To test the robustness of the platform, electrical engineering students of Applied Electronics course teleoperated the robot remotely. It should be noticed that at the time, in person classes were canceled by the government regulation against the COVID-19. In order to start the remote-operation; during online classes, the students were informed about the robot's hardware and software technical issues, as well as, the requirements they should know in order to use the V2 Robot web platform properly. To avoid more than one user connected to the system it was implemented a authentication method. The students should solve the maze problem, considering two regions. The first region was belonging to a controlled zone. In this regions students could see through video feed, as well as the shape and route of the maze. The second region had a cover on the top to block the view from the top camera. The idea of this task was to force students to use the sensor information available on the mobile device in order to control the robot along the maze. Figure 6 shows the region where the maze was located

as well the distribution explained in the above mentioned paragraph: the zone 1 is the visible maze, while the zone 2 is the not visible maze.

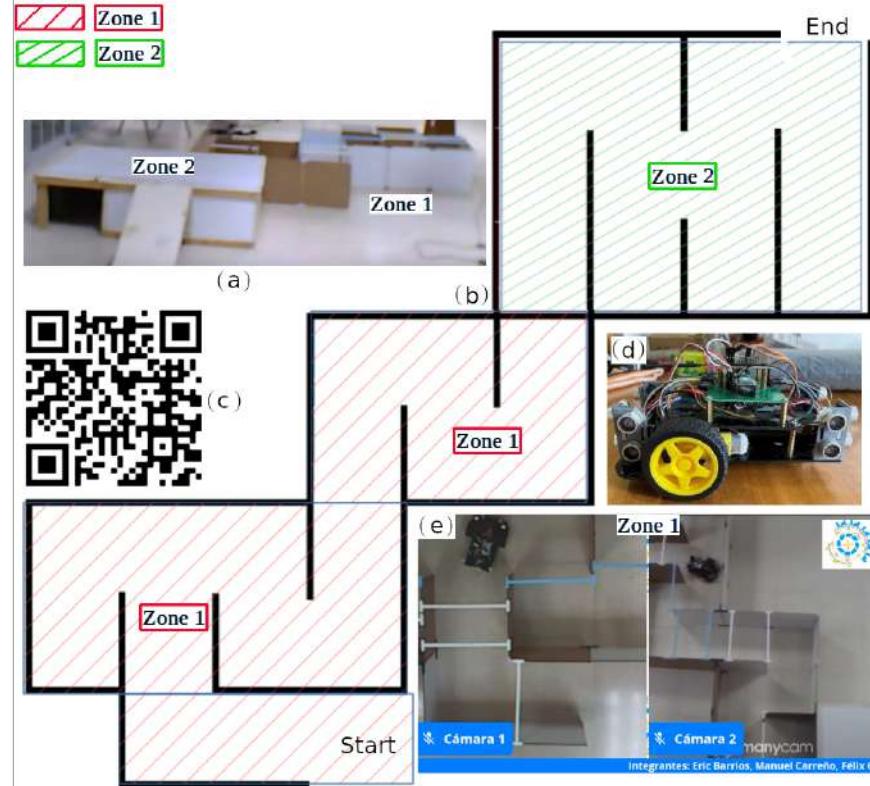


Fig. 6. Maze region. visible and not visible regions by users. (a) Lateral view of the maze region. (b) Maze design for both visible and not visible region. (c) QR code of the video evaluation. (d) Mobile robot used in the evaluation. (e) Top view of the maze that user and online audience could access by using an online channel.

3.1 Challenges: Students Evaluation Comments

A group of 46 students indicated that the experience was innovative even though classes were online. The 82 % of the student lived 20 Km around the university while the rest at that moment lived between 75 Km and 260 Km around the university. However, the review reveals that the students had issues with the internet connection (IC) at the moment to perform the online assessment task. The main problem concerning online education is related to the high quality IC. According to the IESALC [10], one of the main concernings related to higher education during the COVID-19 was the IC, due mostly to the fact that by

2018 just 45% of homes in Latinamerican countries have IC. More precisely in Panamá by the same period, there were 60% with home IC and a 120% mobile connection. Table 1 shows the relationship between the latency and the distance between the user and the robot.

Table 1. Latency and Distance Evaluation

Distance [Km]	Latency [Sec.]	Users
$D < 40$	< 10	38
≈ 75	< 30	4
≈ 260	< 45	4

Although students faced problems with the IC, the study reveals the interest in the implemented strategy used to face the COVID-19 lockdown regulations imposed in Panamá. Table 2 indicated that the 78.25% of the students affirmed the strategy was either functional or innovative, 17.9 % of the students indicated the strategy was complex, while the 4.34% did not have comments.

Table 2. Students Comments Strategy

Strategy	Users	Percents [%]
Innovative	32	69.56
Functional	4	8.69
Complex	8	17.90
Not comments	2	4.34

4 Conclusion

The lockdown and closure of universities caused abrupt changes in the learning and teaching higher education system. Although the switch to the online classes was possible in a short time, there were concerning with the quality of the IC, as well as, the strategy to motivate and engage both students and professors. The strategy used with a group of students help to identified the benefits and disadvantages of remote learning and teaching focused on robotics systems during the pandemic. The implementation of this strategy shows that online systems could be used to teach by running real life applications.

Acknowledgment

This work was supported by the Sistema Nacional de Investigaciones (SNI) of Panamá of the Secretaría Nacional de Ciencia, Tecnología e Innovación de Panamá (SENACYT) (SNI 49-2021). The authors would like to thank , the Universidad Tecnológica de Panamá, for their administrative support on the advancement of this project.

References

1. Błaszczyk, M.; Popović, M.; Zajdel, K.; Zajdel, R. The Impact of the COVID-19 Pandemic on the Organisation of Remote Work in IT Companies. *Sustainability* 2022, 14, 13373. <https://doi.org/10.3390/su142013373>
2. Organisation for Economic Co-operation and Development (2021). Teleworking in the COVID-19 Pandemic: Trends and prospects.<https://www.oecd.org/coronavirus/policy-responses/teleworking-in-the-covid-19-pandemic-trends-and-prospects-72a416b6/>. Accessed 20 Jan. 2023
3. Mok, K.H. Impact of COVID-19 on Higher Education: Critical Reflections. *High Educ Policy* 35, 563–567 (2022). <https://doi.org/10.1057/s41307-022-00285-x>
4. G. Feuerlicht, M. Beránek and V. Kovář, "Impact of COVID-19 pandemic on Higher Education," 2021 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 2021, pp. 1095-1098, doi: 10.1109/CSCI54926.2021.00231.
5. Bandala, M.; West, C.; Monk, S.; Montazeri, A.; Taylor, C.J. Vision-Based Assisted Tele-Operation of a Dual-Arm Hydraulically Actuated Robot for Pipe Cutting and Grasping in Nuclear Environments. *Robotics* 2019, 8, 42. <https://doi.org/10.3390/robotics8020042>
6. Kaarlela, T.; Arnarson, H.; Pitkäaho, T.; Shu, B.; Solvang, B.; Pieskä, S. Common Educational Teleoperation Platform for Robotics Utilizing Digital Twins. *Machines* 2022, 10, 577. <https://doi.org/10.3390/machines10070577>
7. Yamakawa, Y.; Yoshida, K. Teleoperation of High-Speed Robot Hand with High-Speed Finger Position Recognition and High-Accuracy Grasp Type Estimation. *Sensors* 2022, 22, 3777. <https://doi.org/10.3390/s22103777>
8. Wang, V. W.; Wang, L. .A literature survey of the robotic technologies during the COVID-19 pandemic. *Journal of Manufacturing Systems* 2021, 60, 823-836. <https://doi.org/10.1016/j.jmsy.2021.02.005>
9. Montes, H., Rodríguez, H., Echeverría, O., Perez, V. (2022). Semi-autonomous Mobile Robot for Environmental Surfaces Disinfections Against SARS-CoV-2. CLAWAR 2021. Lecture Notes in Networks and Systems, vol 324. Springer, Cham. https://doi.org/10.1007/978-3-030-86294-7_28
10. Instituto Iternacional para la Educación Superior en América Latina y el Caribe (2020), COVID-19 y educación superior: De los efectos inmediatos al día después. <https://www.iesalc.unesco.org/wp-content/uploads/2020/05/COVID-19-ES-130520.pdf>. Accessed 20 Jan. 2023

Lane Detection using Canny Edge Detection Algorithm for Real-time Racing Game

Sehar Shahzad Farooq^{1[0000-0002-2571-9121]}, Hameedur Rahman^{2[0000-0001-8892-9911]}, Samiya Abdul Wahid^{2[0000-0003-4322-4736]}, Iftikhar Ahmad¹, Jin Ho Lee¹, and Soon Ki Jung^{*1[0000-0003-0239-6785]}

¹ School of Computer Science and Engineering, Kyungpook National University, Daegu, South Korea

² Department of Computer Games Development, Faculty of Computing and AI, Air University, Islamabad 44000, Pakistan
 {sehar146, skjung}@knu.ac.kr

Abstract. Self-driving cars are not only a current reality, but this technology also gives us a glimpse of what we can expect from complex technology in the future. Many experts from various fields have come together to create the best autonomous vehicles possible. Lane detection is now a common feature in vehicles and requires large amounts of real-life data for training models. To create fully autonomous vehicles, this data must consider all potential weather conditions, road geographies, driver actions, and vulnerable road users. Most companies do not have the financial resources to test their cars over long distances, and it is impractical to recreate every potential scenario in the real world. One solution to train models more efficiently is to use car racing games, which offer realistic driving scenarios and can help avoid real-world security and privacy concerns. The proposed system is to develop a system for detecting road lanes in a car racing game using Python, and OpenCV, with the ultimate goal of using this system to generate data for training models in self-driving vehicles. The system will use Canny edge detection to detect the edges of the lane, along with a masking function that obscures unwanted details in images, such as trees, rocks, and power lines, will be used to allow the model to concentrate on identifying lanes. The Hough Transform is used to identify and draw the lanes in the game. The proposed system accurately detects lanes in the real-time racing car. This approach allows for the generation of a large volume of data for lane detection system in autonomous cars in a cost-effective and efficient manner, while also providing a range of driving scenarios that may be difficult or impractical to replicate in the real world.

Keywords: Lane detection · Canny edge detector algorithm · Car racing game.

1 Introduction

Human beings make numerous choices each day, and these decisions are often based on the sensory information we receive from our surroundings. Most of

this perception is visual when it comes to driving. Autonomous vehicles, also known as self-driving cars, are designed to be able to detect objects in their surroundings and make timely decisions to respond to these objects. This leads to several challenges related to computer vision in autonomous vehicles, such as detecting road signs, traffic signals, pedestrians, other vehicles, and lanes [10]. This paper focuses specifically on lane detection, which is a crucial aspect of planning the movement of a vehicle.

In this research, we propose a lane detection system using the Canny edge detection algorithm for lane detection in a real-time car racing game called Asphalt 8: Airborne. The screen of the game is accessed using the ImageGrab module of the Python Imaging Library (PIL). Canny edge detection, a widely used method in computational vision processing, is applied using OpenCV to detect the edges of the lanes, and a masking function was applied to eliminate distractions such as trees, rocks, and power lines [25]. The Hough Transform was utilized to mark and draw the lanes in the game [6]. The game environment includes realistic physics, graphics, and a variety of environments such as sharp turns, slopes, and different weather conditions which are useful in lane detection system.

The following objectives are intended to achieve through this research:

1. To utilize computer vision algorithm i.e., canny edge detection algorithm for detecting lane edges in a real-time car racing game (Asphalt 8: Airborne).
2. To develop a system which would detect lanes in racing games and develop large volume of datasets for building lane detection system in self-driving cars.
3. To contribute to minimizing the time complexity in terms of collecting real-life dataset for developing lane detection system in self-driving cars.

One of the main issues in lane detection in self-driving cars is finding enough real world data to feed into the powerful machine learning algorithms that are used to perform tasks such as lane detection [15]. Practical applications of self-driving vehicles require a large amount of data to be collected and labeled in order to train the algorithms. It is time-consuming and costly to collect and label real-world data, and it is also not feasible to test every potential scenario, such as crashing a car at high speed into a brick wall, in real life [7]. A lot of work is being done on real-life captured data of cars for building a lane detection system in self-driving cars, however, very less work has been done on lane detection in car racing games for producing large datasets which are realistic, less time-consuming and cost-effective. A lot of work can be done on detecting lanes in car racing games to produce large amounts of training datasets for building effective and efficient self-driving cars.

2 Related Work

Utilizing synthetic data for development of self-driving cars is becoming popular and is gaining a lot of attention of researchers [13]. As large amounts of real-life

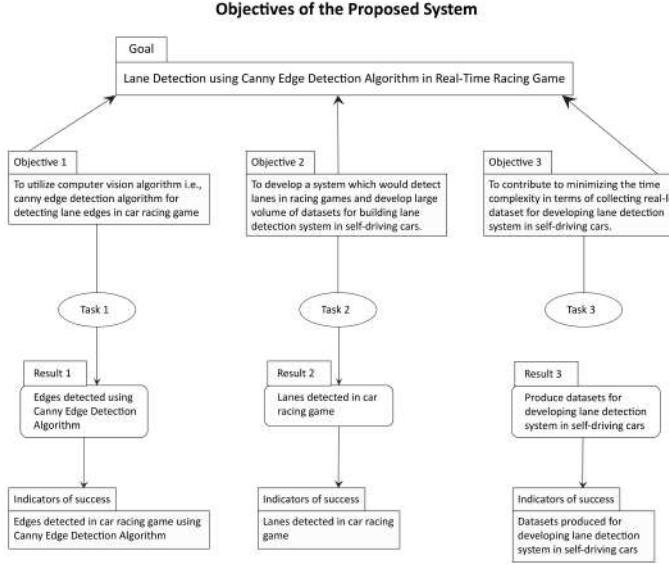


Fig. 1. Research Objective's Breakdown Structure (RBS) of our study

data for training models to create fully autonomous vehicles, this data must also consider all potential weather conditions, road geographies, driver actions, and vulnerable road users. The visual landscapes in numerous video games are so highly detailed and realistic that they can provide data that is just as accurate as real-life images [8]. Readily available computer games that feature realistic graphics and hours of gameplay can offer a more convenient way to collect large amounts of training data. A group of researchers from Intel Labs and Darmstadt University in Germany have devised an innovative method for obtaining valuable training data from the popular video game Grand Theft Auto [18]. The researchers developed a software layer that sits between the game and a computer's hardware, automatically categorizing various objects in the road scenes depicted in the game. This allows for the labeling of data to be fed into a machine learning algorithm, enabling it to identify cars, pedestrians, and other objects shown in the game or in real life. Research conducted by PhD students at the University of British Columbia demonstrates that video games can be utilized to train a computer vision system, sometimes just as effectively as real data [21]. Another research showed that video games also offer a simple way to diversify the environmental conditions present in training data. A team at Johns Hopkins University in Baltimore created a tool that can link a machine learning algorithm to any environment created using the popular game engine Unreal. This includes games like KiteRunner and Hellblade, as well as numerous impressive architectural visualizations [17]. In another research study, asynchronous learn-

ing technique was used to train an end-to-end agent for a realistic car racing game called World Rally Championship 6 (WRC6) [16]. The system does not depend on the game's score, but the agent is trained using only images and speed to determine the best actions, while taking into account real driving conditions. Moreover, in 2013 researchers created and improved a controller for a car in the TORCS open-source racing game. They added advanced collision detection, overtaking strategies, real-time evaluation, and a thorough analysis of the track using the concept of multi-agent artificial potential fields (MAPFs) [19]. In a project, researchers developed a computer program named HORCNN Network that can identify the lanes on a road, detect vehicles and buildings, and control the vehicle's direction based on these inputs, using individual frames from a video game as input [12]. Researchers have also built a system where they present a fuzzy logic-based self-driving car control system and demonstrate its use in the JavaScript Racer game. To achieve this goal, the frameworks of control theory, fuzzy logic, and computer vision were combined in the proposed intelligent driving system [9]. The above research papers and several other papers including [14, 3, 22, 5, 1, 24, 23, 4, 2] show that racing games can be used as an alternative for developing a lane detection system in self-driving cars. However, not a lot of work has been done in this aspect and as result the proposed system detects lanes in a car racing game (Asphalt 8: Airborne) using computer vision algorithms I.e., canny edge detection, Hough transform and a filtering mask.

3 Methodology

In the present system we proposed a lane detection system in a real-time car racing game named Asphalt 8: Airborne. The approach adopted in for this system is explained in the following subsection of Lane detection system.

3.1 Lane Detection System

The purpose of the lane detection system is to detect lanes in a real-time car racing game (Asphalt 8: Airborne) to provide synthetic dataset for autonomous cars to make better decisions while driving. For an instance, autonomous cars can be trained on this synthetic dataset for detecting lanes in an efficient and timely manner. As discussed in Section 1 will access the real-time screen of the game using ImageGrab module of Python Imaging Library (PIL) for detecting the lanes in game. It will then use Canny edge detection to detect the edges of the lane, along with a masking function that obscures unwanted details in images, such as trees, rocks, and power lines, and will be used to allow the model to concentrate on identifying lanes. The Hough Transform is used to identify and draw the lanes on the frame in python output window. The following section of Experimental Results gives a step by step details about the approach of the lane detection system. The main flow process of the system is demonstrated in Figure 2.

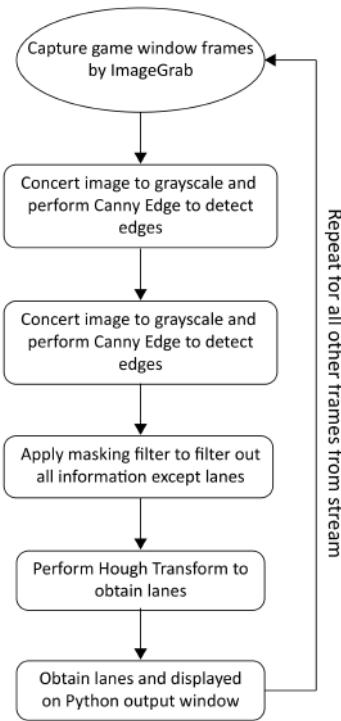


Fig. 2. Process flow chart of detecting lanes and displaying on Python output window

Figure 3 given below displays the layer by layer pipeline of detecting and displaying the lanes.

3.2 Selection of Asphalt game

We chose "Asphalt" as the game for our research because it provides a high level of realism and a range of driving scenarios that can be used to test and train the lane detection system. We believe that the realism and variety of driving scenarios provided by "Asphalt" make it an ideal platform for generating data for autonomous vehicle research.

3.3 Restricted region of interest (ROI)

The reason we chose to use a restricted ROI was to minimize the impact of other elements in the game, such as trees and rocks, on the lane detection system. However, we understand about the potential impact of the restricted ROI on the accuracy of the system, and we plan to further evaluate the impact of this choice in further experiments. we also evaluate the lane detection system under

noisy and varied situations. We plan to perform experiments in different weather conditions and with various road geographies to better understand the robustness and reliability of the system. We will also compare the results with other lane detection methods to provide a comprehensive evaluation of the system.

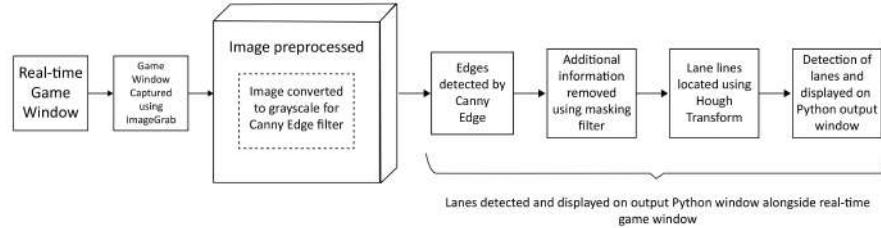


Fig. 3. Layer by layer pipeline of detecting and displaying the lanes

4 Experimental Results

4.1 Capturing the Game Screen in Real-Time

The first step in the developed system is performed by accessing and capturing the racing game screen in real-time. This is done using the ImageGrab module of PIL library [20]. The game screen is captured in real time and is illustrated in Figure 4. The image on the left is real-time game window and the image on the right shows the python window which has captured the game screen using ImageGrab module.



Fig. 4. Real-time game window running alongside game screen capture in python window

4.2 Image Processing for Edge Processing Using Canny Edge Detection Algorithm

We used a technique called Canny edge detection, which is implemented in the OpenCV library, to extract and retain useful structural information from images while significantly reducing their size [26]. The image is first converted to grayscale and then Canny edge filter is applied to obtain the edges. This process simplifies the image significantly and we obtained the edges of the image on the screen. Edges are detected which is illustrated in the Figure 5 given below.



Fig. 5. Edges detected using Canny edge detector

4.3 Applying Mask Filter to Filter all the Additional Information from the Image Other Than the Lanes

For applying masking filter, we defined a masking function. The purpose of this function was to create a polygon shape that hid all the information from the image except for the region that contains the lanes. The resulting mask allowed the algorithm to focus on the lanes in the image. The additional information which is removed from the image is the area other than the red marked area as illustrated in the Figure 6 given below.



Fig. 6. Masking all the area other than area marked with red color.

The result of the masking filter removed all the additional information other than from the image except for the lanes which are illustrated in Figure 7 given below.

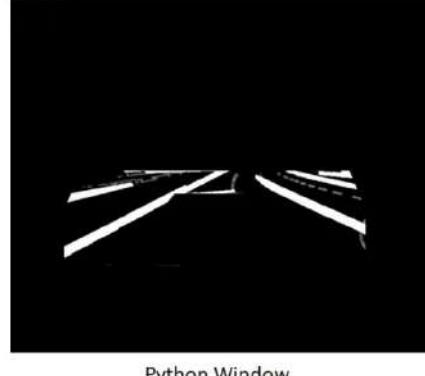


Fig. 7. All additional information filtered out using masking

4.4 Applying Hough Transform to Locate Lane Lines

To locate the actual lane lines in the image, we applied Hough Transform implemented in OpenCV to the edge-processed image [11]. To use this effectively, it is necessary to blur the image slightly before applying the Hough Transform. We used the Gaussian Blur function from OpenCV for this purpose. The lane lines

are located and displayed on the screen which is demonstrated in the Figure 8 shown below.



Python Window

Fig. 8. Hough Transform implemented for overlaying lane lines

4.5 Detecting and Drawing Lanes in Python Output Window

For obtaining the lanes and drawing them in the output window, we created a function. The desired output was obtained which detected the lanes in the output python window. The lanes are detected when game console is running in real-time. The lane detection is shown simultaneously in a python window as shown in the Figure 9 given below.



Fig. 9. Lanes detected in python window in real-time alongside the original game window running

The results of the proposed system are satisfactory as it detected the lanes in the car racing game (Asphalt 8: Airborne). We applied Canny edge detection algorithm for detecting the edges in the captured screen of the game. Canny

edge successfully detected the edges in the captured screen of the game. After the edge detection masking filter was applied for filtering out all the additional information other than the lane edges to focus on the lanes in the image. The Hough Transform located the actual lane lines in the image successfully, which was implemented in OpenCV following the Gaussian blur function. The location of the actual lane line was followed by the successful detecting and drawing of lanes in the python output window alongside the game console window running in real-time.

5 Conclusion

The proposed system successfully detected lanes in a real-time car racing game named Asphalt 8: Airborne with the help of computer vision algorithm i.e., Canny Edge Detection. The screen of the game was accessed using the Image-Grab module of the Python Imaging Library (PIL). Canny edge detection was applied using OpenCV to detect the edges of the lanes, and a masking function will be applied to eliminate distractions such as trees, rocks, and power lines. The Hough Transform was utilized to mark and draw the lanes in the game. The proposed system can be successfully deployed in generating synthetic dataset for training artificial intelligence models for the development of effective and efficient self-driving cars. It can also provide a wide range of driving scenarios that may be difficult or impractical to replicate in the real world. The generated data can be used for training models in various fields, including computer vision, machine learning, and deep learning. In future, several other training algorithms can be implemented for training autonomous vehicle lane detection by using the data generated using the proposed method. It can also be used as a benchmark study in near future for detecting lanes as well as other objects in real-time. Further studies can be done on the above mentioned idea to improve the quality of the model as well as data collection using diverse games.

6 Acknowledgement

This study was supported by the BK21 FOUR project (AI-driven Convergence Software Education Research Program) funded by the Ministry of Education, School of Computer Science and Engineering, Kyungpook National University, Korea (4199990214394).

References

1. Farooq, S.S., Aziz, A., Mukhtar, H., Fiaz, M., Baek, K.Y., Choi, N., Yun, S.B., Kim, K.J., Jung, S.K.: Multi-modality based affective video summarization for game players. In: International Workshop on Frontiers of Computer Vision. pp. 59–69. Springer (2021)

2. Farooq, S.S., Baek, J.W., Kim, K.: Interpreting behaviors of mobile game players from in-game data and context logs. In: 2015 IEEE Conference on Computational Intelligence and Games (CIG). pp. 548–549. IEEE (2015)
3. Farooq, S.S., Fiaz, M., Kim, K., Jung, S.K.: Experience modeling for candy crush game player using physiological data by means of homogeneous transfer learning
4. Farooq, S.S., Rahman, H., Raza, S.A.N., Raees, M., Jung, S.K.: Designing gamified application: An effective integration of augmented reality to support learning. *IEEE Access* **10**, 121385–121394 (2022)
5. Farooq, S., Kim, K.: Game player modeling, encyclopedia of computer graphics and games (2016)
6. Gabrielli, A., Alfonsi, F., Del Corso, F.: Simulated hough transform model optimized for straight-line recognition using frontier fpga devices. *Electronics* **11**(4), 517 (2022)
7. Givan, D., Hanshaw, N., Choi, H.: Application of lane navigation and object detection in a deep-learning self-driving car. In: Future of Information and Communication Conference. pp. 906–917. Springer (2022)
8. Knight, W.: Self-driving cars can learn a lot by playing grand theft auto (Sep 2016), <https://www.technologyreview.com/2016/09/12/157605/self-driving-cars-can-learn-a-lot-by-playing-grand-theft-auto/>
9. Korkmaz, B., Etlik, U.B., Beke, A., Kumbasar, T.: Fuzzy logic based self-driving racing car control system. In: 2018 6th International Conference on Control Engineering & Information Technology (CEIT). pp. 1–6. IEEE (2018)
10. Lee, D.H., Liu, J.L.: End-to-end deep learning of lane detection and path prediction for real-time autonomous driving. *Signal, Image and Video Processing* pp. 1–7 (2022)
11. Marengoni, M., Stringhini, D.: High level computer vision using opencv. In: 2011 24th SIBGRAPI Conference on Graphics, Patterns, and Images Tutorials. pp. 11–24. IEEE (2011)
12. Motupalli, C., Mohinth, R., Gaur, S., Mittal, V., Prakash, S.: Supervision of video game car steering implementing horcnn network. In: 2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence). pp. 291–294. IEEE (2022)
13. Nowruzi, F.E., Kapoor, P., Kolhatkar, D., Hassanat, F.A., Laganiere, R., Rebut, J.: How much real data do we actually need: Analyzing object detection performance using synthetic and real data. *arXiv preprint arXiv:1907.07061* (2019)
14. Oda, O., Lister, L.J., White, S., Feiner, S.: Developing an augmented reality racing game. In: 2nd International Conference on INtelligent TEchnologies for interactive enterTAINment (2010)
15. Pattanashetty, V.B., Mane, V., Hurkadli, S.S., Iyer, N.C., Kore, S.: Lane detection for visual assistance of self-driving vehicles for structured and unstructured roads. In: Information and Communication Technology for Competitive Strategies (ICTCS 2020), pp. 271–279. Springer (2022)
16. Perot, E., Jaritz, M., Toromanoff, M., De Charette, R.: End-to-end driving in a realistic racing game with deep reinforcement learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 3–4 (2017)
17. Qiu, W., Yuille, A.: Unrealcv: Connecting computer vision to unreal engine. In: European Conference on Computer Vision. pp. 909–916. Springer (2016)
18. Richter, S.R., Vineet, V., Roth, S., Koltun, V.: Playing for data: Ground truth from computer games. In: European conference on computer vision. pp. 102–118. Springer (2016)

19. Salman, M.: Collision detection and overtaking using artificial potential fields in car racing game torcs using multi-agent based architecture (2013)
20. Sathaye, N.: Python Multimedia. Packt Publishing Ltd (2010)
21. Shafeei, A., Little, J.J., Schmidt, M.: Play and learn: Using video games to train computer vision models. arXiv preprint arXiv:1608.01745 (2016)
22. Shahzad Farooq, S., Fiaz, M., Mehmood, I., Kashif Bashir, A., Nawaz, R., Kim, K., Ki Jung, S.: Multi-modal data analysis based game player experience modeling using lstm-dnn. Computers, Materials and Continua **68**(3), 4087–4108 (2021)
23. Togelius, J., Lucas, S.M., Nardi, R.D.: Computational intelligence in racing games. Advanced Intelligent Paradigms in Computer Games pp. 39–69 (2007)
24. Tognetti, S., Garbarino, M., Bonarini, A., Matteucci, M.: Modeling enjoyment preference from physiological responses in a car racing game. In: Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games. pp. 321–328. IEEE (2010)
25. Wu, F., Zhu, C., Xu, J., Bhatt, M.W., Sharma, A.: Research on image text recognition based on canny edge detection algorithm and k-means algorithm. International Journal of System Assurance Engineering and Management **13**(1), 72–80 (2022)
26. Xu, Z., Baojie, X., Guoxin, W.: Canny edge detection based on open cv. In: 2017 13th IEEE international conference on electronic measurement & instruments (ICEMI). pp. 53–56. IEEE (2017)

Influence Analysis of Each Facial Region on Facial Expressions Recognition

¹Minsol Park and ^{2,*}Inseop Na

¹ Dept of Computer Engineering, Chosun University, Gwangju, Korea

^{2,*} National Program of Excellence in Software Centre, Chosun University, Gwangju, Korea

*Corresponding Author : ypencil@hanmail.net

Abstract. Many researchers are conducting research to detect human emotions based on facial expressions. In this study, we tried to classify facial landmarks by face area and find out which area has the most influence on emotion recognition. To this end, we extracted 68 landmarks by receiving human face images. After that, the input image was divided into eyebrow region, eye region, nose region, and mouth region along the landmark. Seven universal emotions were evaluated for the divided images. As a result of the experiment, we confirmed that the mouth area has the most influence on emotions.

Keywords: Facial Region, Facial Expression, Facial Landmark

1 Introduction

A lot of systems are being developed that recognize emotion through existing facial information and apply them to various fields. In universal emotion is to recognize seven emotions (happy, angry, disgust, fear, neutral, sad, surprise) from the extracted data. By using this emotion information, it can be used in the fields of human behavioral psychology, security industry, and marketing. Facial recognition systems are systems that analyze and recognize changes in facial joints and muscles, and are operated by geometric information based recognition methods. However, in this paper, we analysis of influence of facial region on facial expression recognition.

2 Proposed System

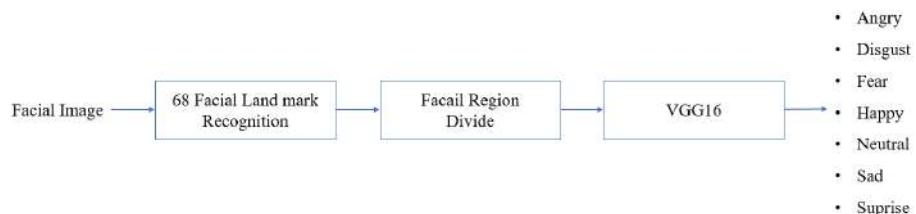


Fig. 1. A Flow char of the proposed system.

Proposed system is composed of 3 steps as shown in Fig. 1. Firstly, we recognize the 68 facial landmark from input facial image by Adrian Rosebrock's facial landmarks[1,2] as shown in Fig. 2. Secondly, we divide the facial region to eyebrow region, eye region, nose region, and mouth region along the landmark as shown in Fig. 3. Thirdly, we recognize facial expression by VGG16 as shown in Fig. 4.

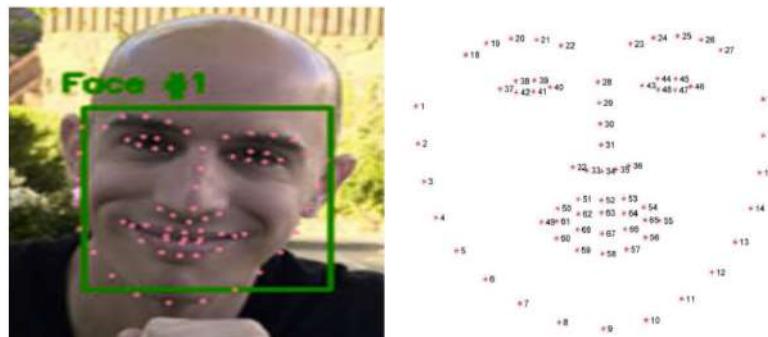


Fig. 2. Example of 68 facial landmark[1]



Fig. 3. Example of facial regions

Layer	Filter Size	filters	Output Size	
input			64x64x3	
Conv2D	3x3	32	64x64x32	
reLu				Activation
Pool	2x2		21x21x32	
Dropout				0.25

Conv2D	3x3	64	21x21x64	
reLu				Activation
Conv2D	3x3	64	21x21x64	
reLu				Activation
Pool	2x2		10x10x64	
Dropout				0.25

Conv2D	3x3	128	10x10x128	
reLu				Activation
Conv2D	3x3	128	10x10x128	
reLu				Activation
Pool	2x2		5x5x128	
Dropout				0.25
Flatten			3200	
Dense			1024	
reLu				Activation
Dropout				0.5
Dense			7	
Sigmoid				Activation

Fig. 4. Structure of VGG16

3 Experiments & Results

We use a large dataset of facial expression named FER2013 [3] as shown in Fig. 5. This dataset consist of 48x48 gray-scale images: 28711 for training, 7176 for validation and testing. In our experimentation, 7176 image used for testing.

**Fig. 5.** Samples of FER2013 from training dataset

In the case of the collected data, it is a dataset that is configured to understand human emotions in consideration of facial expressions. After recognizing the face part using dlib's shape_predictor_68_face_landmark, it was prepared by proceeding with normalization to (64,64,3) by proceeding with crop. We conduct the 300 epoch, 1e-3 loss rate, 64 batch size, Adam's optimizer and binary cross-entropy loss function. Figure 6 shows that the results of facial expression with FER 2013 on facial region. First row, image is only use the VGG16. Second row, all_landmark is use the 68 facial landmark with VGG16. Third, mouth is use the mouth region with VGG16.

	loss	accuracy	val_loss	val_accuracy
image	0.1635	0.7624	0.2204	0.6939
all_landmark	0.0791	0.7746	0.1199	0.6879
mouth	0.2136	0.6711	0.2892	0.5484
eye	0.2569	0.597	0.3607	0.4277
eyebrow	0.2859	0.5313	0.3329	0.4479
nose	0.2754	0.5619	0.3177	0.496

Fig. 6. Results of facial expression with FER 2013 on facial region

4 Conclusions

As a result of the experiment, we were able to confirm that the order of influence on facial expression recognition among facial regions was in the order of mouth, eyes, nose, and eyebrows.

References

1. Adrian Rosebrock, Facial landmarks with dlib, OpenCV, and Python, 2017.
2. K.W. Hwang, M.S. Park, I.S. Na, A comparison of Facial Landmark Based on Key Points, In Proc. of the Autumn Conference of Korea Smart Media , pp. 66-68, 2021.
3. P. Carrier and A. Courville, “Challenges in representation learning: Facial expression recognition challenge.” <https://goo.gl/kVzT48>, 2013.

Diffuse Large B-cell Lymphoma Survival Prediction using Encoding Clinical Features

Sy-Phuc Pham^{*,1}, Sae-Ryung Kang^{*,2}, Hyung-Jeong Yang^{**,1}, Deok-Hwan Yang^{**,2}, Sudarshan Pant¹, Soo-Hyung Kim¹, and Guee-Sang Lee¹

¹Chonnam National University, Gwangju, South Korea

²Chonnam National University Hwasun Hospital, Gwangju, South Korea

{phamsyphuc123,sudarshan.pant}@gmail.com,
{srkang,hjyang,drydh,shkim,gslee}@jnu.ac.kr}

Abstract. Diffuse Large B-cell Lymphoma (DLBCL) is a type of blood cancer that has a high mortality rate. Accurately predicting the survival time of DLBCL patients is crucial for guiding treatment decisions and developing new therapies. In this study, we proposed Encoding Clinical Features (ECF) and used CoxCC and DeepSurv to predict the survival time of DLBCL patients. We experimented with our dataset, which was provided by Chonnam National Hwasun Hospital. We applied ECF to the dataset using a set of dimensions for categorical variables and evaluated the performance of the model using the C-index and the Integrated Brier Score (IBS). Our results showed that the ECF technique had a high C-index and a low IBS, indicating good performance in predicting the survival time of DLBCL patients. We also found that the selected dimensions for embedding categorical variables were suitable for our dataset. Our study demonstrates the potential of ECF with CoxCC and DeepSurv for improving the prediction of DLBCL patient outcomes and for identifying important prognostic factors that can be used to guide treatment decisions.

Keywords: Diffuse Large B-cell Lymphoma · Survival analysis · Clinical information.

1 Introduction

Diffuse Large B-Cell Lymphoma (DLBCL) is a type of non-Hodgkin lymphoma, which is a cancer that affects the lymphatic system [1]. The lymphatic system is a network of vessels and organs that help to fight infection and disease in the body. DLBCL is the most common type of non-Hodgkin lymphoma and it is characterized by the rapid growth of abnormal B-cells, a type of white blood cell that is involved in the immune response [2]. The cancer cells can spread to various parts of the body, including the lymph nodes, bone marrow, liver, spleen and other organs. Symptoms of DLBCL can include swollen lymph

* These authors contributed equally to this work.

** Corresponding author.

nodes, fever, weight loss, night sweats, and fatigue. The exact cause of DLBCL is unknown, but certain risk factors have been identified such as age, certain infections, and a weakened immune system [3]. DLBCL can be treated with a combination of chemotherapy, radiation therapy, and immunotherapy. The prognosis varies depending on the stage of the cancer, the patient's overall health, and the effectiveness of the treatment. With early diagnosis and appropriate treatment, the survival rate of DLBCL can be quite high.

Survival analysis is a statistical method used to study the time to an event of interest, such as death or failure. In the context of DLBCL, survival analysis is used to estimate the probability of survival in patients with this type of lymphoma. A common method used in survival analysis is the Kaplan-Meier estimator, which is used to estimate the survival probability over time. The Kaplan-Meier [4] estimator is based on the idea of censoring, which means that some patients may not have experienced the event of interest (death) at the time of the analysis. These patients are considered "censored" and their survival time is not included in the estimate. Another method used in survival analysis is the Cox proportional hazards model [5], which is used to estimate the effect of various factors (such as age, stage of the cancer, and treatment) on the risk of death. The Cox model allows for the estimation of hazard ratios, which indicate the relative risk of death for a particular group of patients compared to another group. Overall, Survival analysis is a useful tool for assessing the prognosis of DLBCL patients and for identifying factors that may influence the risk of death. These techniques can help physicians and researchers to identify patient sub-groups that have a higher or lower risk of death, and to develop more effective treatment strategies for DLBCL.

DLBCL is a type of non-Hodgkin lymphoma (NHL), which is a cancer of the lymphatic system. DLBCL is the most common type of NHL and represents approximately 30% of all NHL cases [6]. DLBCL is typically diagnosed through a combination of clinical examination, laboratory tests, and imaging studies. A biopsy of the affected tissue is usually performed to confirm the diagnosis and to determine the stage of the disease. The treatment of DLBCL depends on the stage of the disease and the patient's overall health. The standard treatment for DLBCL is a combination of chemotherapy and radiation therapy. In some cases, immunotherapy or targeted therapy may also be used. Prognosis of DLBCL varies depending on the stage of the disease and the patient's overall health. With early detection and appropriate treatment, the overall survival rate for DLBCL is around 70%. However, the prognosis is poorer for patients with advanced-stage disease and those who do not respond well to treatment. The treatment of DLBCL typically involves a combination of chemotherapy and radiation therapy, and the prognosis varies depending on the stage of the disease and the patient's overall health. Clinical information is important to every DLBCL patient and this information is recorded by the physician throughout the course of treatment.

Artificial Intelligence (AI) has been increasingly used in survival analysis, which is a statistical method used to study the time to an event of interest, such as death. AI-based methods have been developed to improve the prediction of

patient outcomes and to identify important prognostic factors that can be used to guide treatment decisions. AI-based methods have also been used to analyze large datasets and to identify patterns and trends that can help to advance biomedical research and to develop new drugs and therapies. For example, AI-based methods have been used to analyze clinical trial data to identify important predictors of patient outcomes and to identify potential new treatments for diseases such as cancer. Despite the advancements, AI in survival analysis is still in its early stages and there are many challenges that need to be addressed. One of the main challenges is the lack of standardization and regulation in the development and deployment of AI-based survival analysis tools, which can lead to inconsistency in the quality of the tools and in the results obtained. Additionally, there is a need for more clinical validation of the AI-based tools to ensure their safety and effectiveness before they are widely adopted in the healthcare system. Overall, AI has the potential to revolutionize survival analysis and to improve patient outcomes, but it is important to continue to invest in research and development to address the challenges and to ensure that the benefits of AI are realized in the healthcare system.

In this paper, we perform survival analysis tasks based on recent methods. In the first section, we introduce overall the DLBCL, the survival task, the clinical information in DLBCL, and the AI in the survival task. We present the recent method for survival tasks in the second section. In the third section, we introduce our approach for survival task. In the next section, we summarize the dataset and experiment detail for this work. The results of our experiment are presented in the next section. We include the conclusion in the last section.

2 Related work

2.1 Cox Proportional Hazards Model

The Cox Proportional Hazards Model (CoxPH) [7] is a statistical model used to estimate the effect of various factors on the risk of an event, such as death. The model is named after its creator, Sir David Cox. The model is widely used in survival analysis, which is a statistical method used to study the time to an event of interest, such as death. The CoxPH model is a semi-parametric model, which means that it makes some assumptions about the shape of the hazard function, but it does not specify the exact form of the function. The model assumes that the hazard ratio is constant over time. This assumption is known as the proportional hazards assumption. The CoxPH model estimates the effect of various factors on the hazard ratio. The model estimates the hazard ratio as a function of the values of the predictor variables. The hazard ratio is a measure of the relative risk of an event for a particular group of patients compared to another group. The CoxPH model can be used to estimate the effect of various factors on the risk of death in DLBCL patients, allowing to identify patient subgroups that have a higher or lower risk of death and to develop more effective treatment strategies for DLBCL.

2.2 DeepSurv

DeepSurv [8] is a deep learning-based survival analysis method that uses artificial neural networks to predict the risk of an event, such as death. It is an adaptation of a traditional Cox proportional hazards model, which is a widely used statistical model in survival analysis. DeepSurv uses a neural network to model the hazard function and to estimate the effect of various factors on the hazard ratio. The neural network is trained on a dataset of patients with the event of interest, and the model learns the complex non-linear relationships between the predictor variables and the hazard ratio. One of the main advantages of DeepSurv is that it is able to handle high-dimensional and non-linear data, which can be difficult to analyze using traditional survival analysis methods. Additionally, DeepSurv can handle missing data, which is a common problem in survival analysis. DeepSurv has been applied to various medical fields, such as oncology, and has shown to improve the prediction of patient outcomes and to identify important prognostic factors that can be used to guide treatment decisions.

3 Proposed method

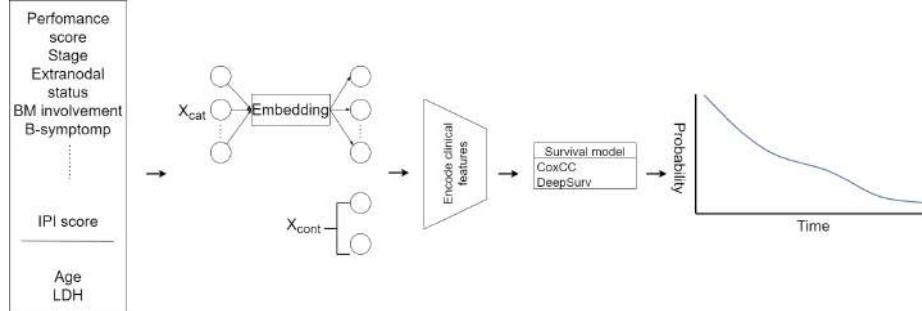


Fig. 1. Perspective of the proposed survival based encoding clinical feature for DLBCL survival prediction.

In this section, we present our method to deal with our dataset for use in the survival task. Our method has two main parts: encoding clinical features and survival prediction. The overall architecture has been shown in Figure 1. First, we introduce the encoding clinical features part. Linear analysis, machine learning, and deep learning are the core components of most approaches, however these can only be used with numerical data. Since that the data may be in categorical form, it is important to process the input data to transform the non-numerical fields into numerical fields before feeding the data into the deep network. It has been demonstrated in some research that the embedding method is superior to the one-hot vector approach, and it has also been utilized in some studies to

accomplish the encoding [9,10]. For categorized information, Mikolov et al. [11] presented an embedding method. Equation 1 shows our definition of clinical input, which incorporates both continuous and categorical data.

$$\chi_{\text{clinical}} = \chi_{\text{continuous}} \oplus \chi_{\text{category}} \quad (1)$$

We set the dimensions to represent the embedding in the continuous fields using a variable from the categorical fields $\chi_i \in \chi_{\text{category}} \rightarrow \nu_i \in \mathbb{R}^{k_i}$. The k_i dimension of space is a function of the characteristics of the variables in the category. The clinical features were obtained by appending the vectors representing the continuous variables and the embeddings together.

$$Z_{\text{clinical}} = f(\chi_{\text{continuous}}, f(g(x_1, \dots, x_n))) \quad (2)$$

where

- Z_{clinical} is the clinical feature after after feature extraction part.
- $\chi_{\text{continuous}}$ is the numeric fields.
- $x_1, \dots, x_n \in \chi_{\text{category}}$ with n is the number of categories in the clinical data.
- f is the concatenate operator.
- g is the embedding operator.

Second, we apply methods based on CoxCC [12] and DeepSurv to predict the survival hazard rate. For the purpose of determining whether or not encoding clinical information is effective, we used two standard models in the survival task.

4 Experiments

4.1 Dataset

Clinical information is essential for the management of DLBCL patients as it helps to determine the patient's diagnosis, stage of the disease, treatment options and prognosis. The diagnosis of DLBCL is made on the basis of clinical data. This includes symptoms, physical examination, laboratory test results, and imaging studies. The extent to which cancer has spread throughout the body is quantified by the disease's stage, which is in turn determined by the available clinical data. This is important in determining the appropriate treatment plan and in estimating the patient's prognosis. Clinical data are analyzed to establish the best course of treatment for the patient. This includes the type of chemotherapy, radiation therapy, and immunotherapy that may be appropriate for the patient, as well as the potential side effects of these treatments. Clinical data is analyzed to determine the patient's prognosis, which is a prediction of the patient's chance of remission. This includes factors such as age, overall health, and the stage of the disease, as well as the patient's response to treatment. Clinical data is tracked over time to assess a patient's response to treatment and make any necessary adjustments. This can include regular blood tests, imaging

Table 1. Statistics of each clinical feature of 602 DLBCL patients in the data set of CNUHH.

Data field Characteristics		Value	CNUHH (n=602)
numeric Age		min-max 17-92	
numeric LDH		min-max 144-8402	
categories Performance	1	201 (33.39%)	
	2	322(53.49%)	
	3	65 (10.8%)	
	4	14 (2.33%)	
categories B symptom	0	504 (83.72%)	
	1	98 (16.28%)	
categories Extranodal status	0	456 (75.75%)	
	1	146 (24.25%)	
categories Stage	1	118 (19.6%)	
	2	195 (32.39%)	
	3	137 (22.76%)	
	4	152 (25.25%)	
categories Spleen involvement	0	579 (96.18%)	
	1	23 (3.82%)	
categories Bone marrow involvement	0	554 (92.03%)	
	1	48 (7.975%)	
categories IPI score	0	81 (13.46%)	
	1	152 (25.25%)	
	2	134 (22.26%)	
	3	131 (21.76%)	
	4	76 (12.62%)	
	5	28 (4.65%)	
categories IPI risk	1	233 (38.7%)	
	2	134 (22.26%)	
	3	131 (21.76%)	
	4	104 (17.28%)	
categories R-IPI	1	81 (13.46%)	
	2	286 (47.5%)	
	3	235 (39.04%)	

studies, and physical examinations. The medical expert at Chonnam National University Hwasun Hospital kindly provided the dataset that was used for the experiments that were conducted for this work. In this dataset, we divided the features into two types: numeric and categorical. We detailed the clinical feature in the **Table 1**.

4.2 Experiment set up

We carried out the experiment using the CoxCC, DeepSurv models as primary methods. The dataset includes 602 patients. We separated the dataset into 5-fold for the training process. To evaluate the performance of the model, we performed on independent test data sets. In total, we used 481 patients for the training process, and we used 121 patients for testing and evaluating the performance of the model. In each of the methods, the model parameter was configured in a similar way.

5 Experimental results

Our work was implemented with the Pytorch 1.11 library. We utilized two distinct metrics in order to evaluate the performance of the models. The first was the concordance index [13], or C-index for short. This evaluation approach is the one that is employed the most commonly. It measures the ability of a model to correctly rank the event times of a population, with higher values indicating better performance. We also utilized the Integrated Brier Score (IBS) [14] as a secondary metric. IBS is a measure of the accuracy of a model's predictions over time. Encoding clinical features was performed to improve CoxCC and DeepSurv. Experimental results in Table 2 show that our proposed model suitable for clinical tabular. Although our proposed method gives better results than only using CoxCC and DeepSurv, it lacks outstanding at C-index. That is something we need to improve in the future.

Table 2. Comparison results of the encoding clinical features and without encoding clinical features

Method	Survival model	IBS	C-index
W/o encoding clinical features	CoxCC	0.164	0.545
	DeepSurv	0.146	0.700
Encoding clinical features	CoxCC	0.140	0.654
	DeepSurv	0.143	0.728

6 Conclusions

This paper presented Encoding clinical features for using in traditional architecture in survival task such as CoxCC and DeepSurv. We defined the dimension for embedding categorical features by experiment and selected the best dimension for the CNUHH dataset. DeepSurv deep learning network architecture has

somewhat better results than CoxCC statistical model. Clinical data with categorical data types are efficiently fed into deep learning network architectures and statistical models. We all know that patient data includes not just clinical information but also imaging data like PET scans and CT scans. In future work, we will integrate medical imaging into the model to conduct a survival analysis experiment of DLBCL patients.

Acknowledgments

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT). (NRF-2020R1A2B5B01002085)

References

1. Sehn, Laurie H., and Gilles Salles. "Diffuse large B-cell lymphoma." *New England Journal of Medicine* 384.9 (2021): 842-858.
2. Pileri, Stefano A., et al. "Predictive and prognostic molecular factors in diffuse large B-cell lymphomas." *Cells* 10.3 (2021): 675.
3. Li, Shaoying, Ken H. Young, and L. Jeffrey Medeiros. "Diffuse large B-cell lymphoma." *Pathology* 50.1 (2018): 74-87.
4. Goel, Manish Kumar, Pardeep Khanna, and Jugal Kishore. "Understanding survival analysis: Kaplan-Meier estimate." *International journal of Ayurveda research* 1.4 (2010): 274.
5. Cox, David R. "Regression models and life-tables." *Journal of the Royal Statistical Society: Series B (Methodological)* 34.2 (1972): 187-202.
6. Armitage, James O., et al. "Non-hodgkin lymphoma." *The lancet* 390.10091 (2017): 298-310.
7. Fisher, Lloyd D., and Danyu Y. Lin. "Time-dependent covariates in the Cox proportional-hazards regression model." *Annual review of public health* 20.1 (1999): 145-157.
8. Katzman, Jared L., et al. "DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network." *BMC medical research methodology* 18.1 (2018): 1-12.
9. Guo, Cheng, and Felix Berkhahn. "Entity embeddings of categorical variables." *arXiv preprint arXiv:1604.06737* (2016).
10. Wang, Peng, et al. "Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification." *Neurocomputing* 174 (2016): 806-814.
11. Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781* (2013).
12. Kvamme, Håvard, Ørnulf Borgan, and Ida Scheel. "Time-to-event prediction with neural networks and Cox regression." *arXiv preprint arXiv:1907.00825* (2019).
13. Harrell, Frank E., et al. "Evaluating the yield of medical tests." *Jama* 247.18 (1982): 2543-2546.
14. Gerds, Thomas A., and Martin Schumacher. "Consistent estimation of the expected Brier score in general survival models with right-censored event times." *Biometrical Journal* 48.6 (2006): 1029-1040.

Robust Data Augmentation for Accurate Human Pose Estimator

Tien-Dat Tran, Xuan-Thuy Vo, Adri Priadana, and Kang-Hyun Jo*

School of Electrical Engineering, University of Ulsan, Ulsan 44610, South Korea
 {tdat,xthuy}@islab.ulsan.ac.kr; priadana3202@mail.ulsan.ac.kr;
 acejo@ulsan.ac.kr

Abstract. Accurate occluded key point identification is a challenge and hot topic for human pose estimation. To make the occluded or invisible keypoint better, data augmentation play an important role which makes the network overcome complex case. In this paper, we want to apply cut-out technique which is a powerful method to tackle the problem. Furthermore, data augmentation demonstrates its superiority over other methods without enlarging the computational cost. Correspondingly, the proposed work focuses on powerful data augmentation for occluded keypoints. First, following a human detection in the detector network, feed the human proposal region into the data augmentation, which makes the network can learn more about the occluded cases. The data after data augmentation then apply to train for the pose estimator. The estimator collects more information in occluded keypoints, illustrating higher precision efficiency. The outputs of our experiments would also demonstrate a distinction between the use of cut-out data augmentation and existing approaches. The predicted joint heatmaps are more accurate than the baseline technique despite using the same amount of parameters due to the transition to a high-resolution network (HRNet) for the pose estimator. Regarding AP, the suggested design outperforms the baseline-HRNet by 0.2 points, but in the occluded case, the pose estimator performs much better. Additionally, the COCO 2017 benchmarks, now accessible as an open and the most popular dataset for pose estimation, were used to train the proposed network.

Keywords: Deep Learning · Occlusion Keypoint · Data Augmentation · Human pose estimation.

1 Introduction

In the modern world, 2D human pose estimation plays a crucial role but challenging function in computer vision, which can serve numerous objectives such as human robotics [23, 3], activity recognition [6, 8], human re-identification [25, 11], or film industry[2, 10]. Human pose estimation has a primary mission which is to identify body parts for human body joints.

* Corresponding author



Fig. 1. Occlusion Keypoint in the testing on the MPII dataset. The red dot is the occluded keypoint which is one of the big challenges nowadays for human pose estimation

In human pose estimation, there are many challenges that attack the network performance. Among the challenges, the occluded keypoint shown in Fig.1 is one of the biggest challenges for the network training to get better performance. To solve this kind of problem many researchers used another network such as a graph neural network[17] or Generative adversarial network[21] to generate a new structure for the human pose to train. However, utilizing a new network for the occluded problem is costly. To solve the problem, data augmentation is a potential candidate that can remedy the challenges, which is not consumed much more resources than using another network. Data augmentation does not only enhance the value of information from the image but also not consume more parameters in the training process. In more detail, the data augmentation performs a global transformation for the images. The transformation gives the network many extra points of view about images, which show a lot of improvement[4]. Besides all of the advantages, data augmentation also brings extra unimportant data, which makes the data redundant. On the other hand, many kinds of data augmentation such as crop makes the data much more margin or rotate can make the data lost information. Hence, choosing the suitable for data augmentation is really important to make the network can get better performance.

In the proposed work, we make a deep investigation into data augmentation which compares the original method and a new one. The original data augmentation[19] apply flip, rotate, scale, and half body transform. This kind of method can enhance the accuracy of keypoint however for the occluded keypoint, it shows their disadvantages which can not significantly improve the accuracy of the occluded keypoint. Hence, the proposed research applies a new kind of data augmentation which call cut-out. By using cut-out method, the whole architecture can gain more accuracy, especially in the case of occluded keypoint which can check at the experiment result.

In particular, the proposed study was based on a simple framework [4], which applies the top-down method for human pose estimation. Without taking the data much more different from the original but more occlude cases appear, the proposed network can be easy to learn the invisible keypoint. For instance, with the extra training data, the network may learn to connect the keypoint for the visible part such as occlude wrist or ankle keypoint. Furthermore, the number

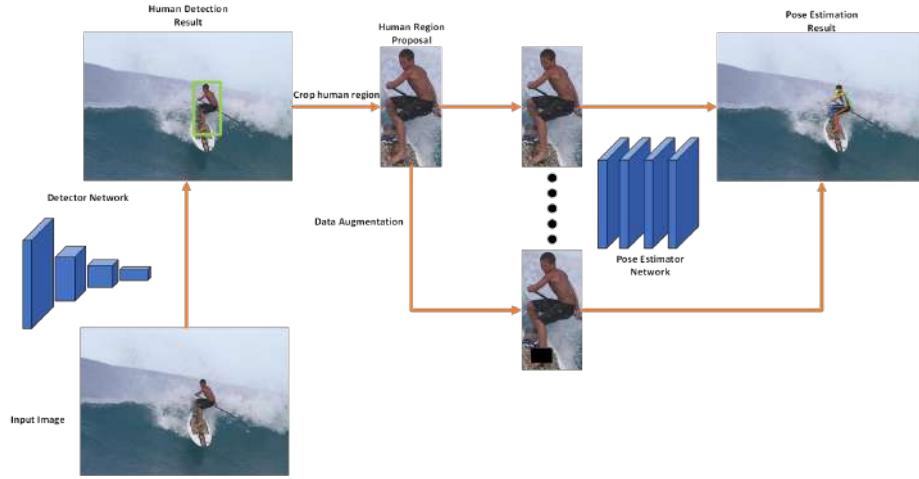


Fig. 2. Full system of 2D human pose estimation from input to pose estimation. The proposed approach split the system into 2 stages, the first stage is the human detector and the second stage is the pose estimator

of parameters was not changed, resulting in network speed not changing while the accuracy for the occluded keypoint improved much more.

To make clear about the cut-out augmentation, the transformation can apply for all of the pose estimators apply the data augmentation. Also, this method is easy to apply not only for estimator but also detector

In summary, the main contribution of the paper describes in two-fold:

- We design and apply a data augmentation called the cut-out that makes the data more information about the occluded problem.
- We comprehensively evaluate and compare the proposed method with the original method on the COCO benchmark dataset, which is the most popular dataset for keypoint.

2 Related work

2D-Human Pose Estimation The most important aspect of human pose estimate is joint detection and its interaction with spatial space, as seen in Fig.3. There are two main methods applied for human pose estimation, which is the bottom-up and top-down method. For the bottom-up method, Deeppose[22], Simple baseline makes use of joint prediction using an end-to-end network with a larger parameter. Later, Newell with the Stacked hourglass network [13] reduces the number of settings while maintaining great accuracy. To represent local joints, all of the approaches employed Gaussian distributions. After that, a convolution neural network was utilized to estimate human posture estimation. For the top-down method, first, we apply a detector for the human proposal region, and after that using the crop region for pose estimation. Because the

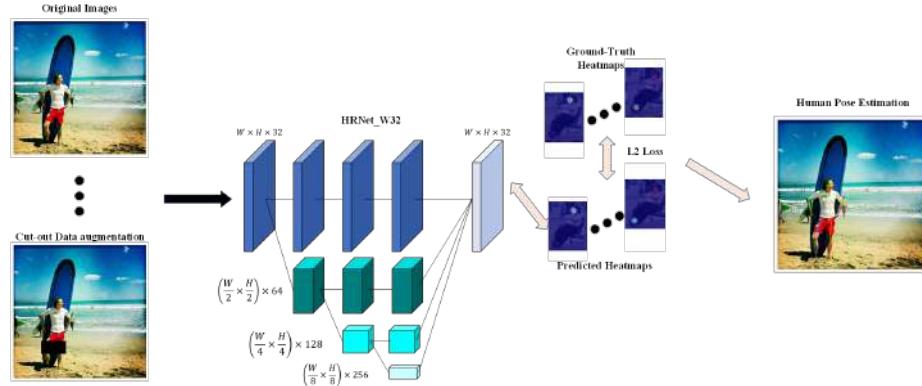


Fig. 3. Illustrating the architecture of the proposed 2D-human-pose estimator. The proposed network training with the original images and transformation images

top-down method is using the detector so the accuracy can get better than the bottom-up. And bottom-up is an end-to-end method so the inference time can be better than the top-down.

In the proposed paper, we apply the top-down method for the whole architecture which illustrates in Fig.2, From the input images, the model utilizes the existing detector for human detection. YOLO[20] is one of diversity kind detector, which has many versions for different cases such as real-time, high accuracy, or for mobile devices. To balance everything, the proposed method utilizes the YOLO-V3. After applying the detector to the human region, the whole network utilizes the pose estimator to perform training tasks in the human region. Additionally, data augmentation will apply in this stage. In comparison, the top-down method uses sufficient perspective for network design, with a limited number of parameters and high speed or a larger number of parameters and lower speed.

Data Augmentation: Data augmentation play an important role in computer vision task which compensate for the lack of data in real life. From the original input image, data augmentation makes more data for the network can learn from many perspective views. Notice that the more diversity in the dataset, the more accuracy for human detection. Moreover, data augmentation did not increase the number of parameters so the computational cost will not increase. However, data augmentation also can drive the detector worth[16] which make the detector hard to learn the feature of images, especially for occluded keypoint.

Most detector networks used the same data augmentation such as Flip, Rotation, Scale zoom in and zoom out or half body transforms with a probability of 0.3. However, this kind of data augmentation does not work well with occluded keypoint. Hence, we apply the cut-out augmentation to show the real case to build the network can learn more about the occluded keypoint. Furthermore, the occluded and invisible keypoint appear more in the data so that the

network learns better. To improve accuracy, the cut-out method shows better performance in the data augmentation tasks.

3 Methodology

3.1 Network architecture

Detector The human detector plays an important role in the whole system. First, input image matrix $\alpha(\mathbf{X})$ feed to the human detector. After that, the detector gives the result of the human region $\beta(\mathbf{X}')$ which is the subset of $\alpha(\mathbf{X})$.

$$D\{\alpha(\mathbf{X})\} = \beta(\mathbf{X}') \quad (1)$$

Following resize function make $\beta(\mathbf{X}')$ into 256×192 images which can call $\gamma(\mathbf{X}')$

$$\gamma(\mathbf{X}') = \text{Resize}(256 \times 192, \beta(\mathbf{X}')) \quad (2)$$

The proposed study utilizes YOLO-V3[18] for the main detector in the whole architecture. The YOLO-V3 is the medium detector that can balance the computational cost and accuracy.

Data Augmentation In the proposed paper, we apply one more data augmentation for the bounding box $\gamma(\mathbf{X}')$ after the detector stage. Besides the original data augmentation which includes Flip, Rotate, Scale, and Half Body Transform, additional cut-out augmentation is applied. First, the cut-out function can be understood

$$C\{\gamma(\mathbf{X}')\} = \text{Cutout}(\gamma(\mathbf{X}'), n, p) \quad (3)$$

with p is the padding fraction for cut out, which is the number of pixels applied for cut out. n is the number of cutout pads in the human region. In the training process, we set n equal to 1. For more detail, the padding p is set random base on the size of $\gamma(\mathbf{X}')$ which is 256×192 . The human region $\gamma(\mathbf{X}')$ have the coordinate of x_{min}, y_{min} and x_{max}, y_{max} which is the coordinate of the human region in the images. the padding P will take random with the condition $x_{min} \leq P_x \leq x_{max}$ and $y_{min} \leq P_y \leq y_{max}$. This research applies the Clamp function in Pytorch[15] to make the border for the cut-out pad inside the human region $\gamma(\mathbf{X}')$. After having the pad for cut-out we use the replace function to apply the pad to the human region. Finally, the cut-out image and the original pad will apply for the training part in the pose estimator

Pose estimator The pose estimator use backbone mainly HRNet-W32 and HRNet-W48 [19]. Fig.3 shows our proposed architecture for the estimator which is based on the backbone. The estimator HRNet includes 4 stages, which consist of residual blocks and connections. The input is the human region proposal from the detector resize the size to 256×192 for both HRNet. After that, each residual block is traversed by the feature maps, and each stage's $W \times H$ resolution is reduced twice. The size of the output tensor is finally reduced to $\frac{W}{16} \times \frac{H}{16}$ with 256 channels at the last bottom layer of the network after traveling down the spine. The first subnetwork, whose size is $W \times H$, is the only one that the backbone

network will employ during the regression. Additionally, each stage would see a doubling of the channel size. After the first block, it increases from 32 to 256 in the last layer. In order for the Training System to predict the human joints, the backbone network must gather data and feature maps from the input image by utilizing the cross entropy loss which describes in the Loss function part.

After extracting the information using the backbone network, the upsampling network recovers the information by taking the feature map from the final layer of the backbone network and upsampling it. Following that, the feature map will be trained with Ground-truth Heat Maps, as shown in Fig.4. The default heat map size is a quarter with the original images 256×192 for HRNet-W32 and 384×288 for HRNet-W48. However, we resize the input image for HRNet-W48 into 256×192 to save the parameter and time for training. For regression, the proposed study will use these heat maps and the ground truth heatmap to create the predicted keypoint. This article employs HRNet so the feature maps are kept to the shape with the original input(in Figure 3). The residual block contains both batch normalization and ReLU[7].

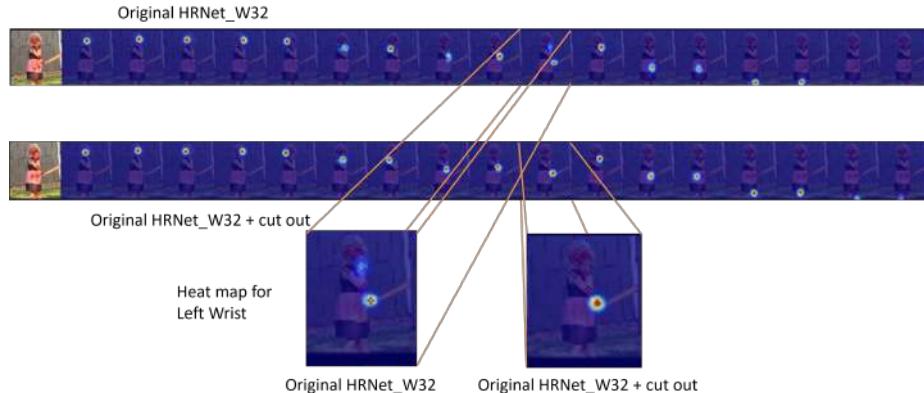


Fig. 4. Heat-map generates in pose estimator before and after applying the cut-out data augmentation. In comparison, the visible keypoint is almost the same in both cases. The occluded keypoint are much more different can show for the left wrist

3.2 Loss Function

Heat maps are used in this work to illustrate body joint locations for the loss function. As the ground-truth position in Fig. 4 by $a = \{a_k\}_{k=1}^K$, where $a_k = (x_k, y_k)$ is the spatial coordinate of the k th body joint in the image. The ground-truth heat map value H_k is then constructed using the Gaussian

distribution with the mean a_k and variance \sum as shown below.

$$H_k(p) \sim N(a_k, \sum) \quad (4)$$

where $\mathbf{p} \in \mathbf{R}^2$ represents the coordinate, and \sum is experimentally defined as an identity matrix \mathbf{I} . The last layer of the neural network predicts K heat maps, i.e., $\hat{S} = \{\hat{S}_k\}_{k=1}^K$ for K body joints. A loss function is defined by the mean square error, which is calculated as follows:

$$L = \frac{1}{NK} \sum_{n=1}^N \sum_{k=1}^K \|S_k - \hat{S}_k\|^2 \quad (5)$$

N denotes the number of samples in the training session. Using information from the backbone network's last layer, the network generated prediction heat maps using ground-truth heat maps.

4 Experiments

4.1 Experiment Setup

Dataset. The Microsoft COCO 2017 dataset [14] is used throughout the studies in the proposed network. The data collection contains 250K human samples and 200K images, each human identity has 17 keypoint labels. Three folders, labeled train set, validation set, and test-dev set, contain training, validation, and testing photographs respectively. In addition, the original is available to view, and the validation and training annotations are as well.

This study also made use of a commercial dataset that records footage of individuals working in a commercial laboratory setting. The dataset consists of 4 films with frame rates ranging from 4000 to 6000. There are several difficulties in the video, including overlapped people, crowded at scenes, and little people. Therefore, it is possible to test how effective the suggested strategy is at tracking.

Evaluation metrics. In the proposed study COCO[12], we utilized Object Keypoint Similarity (OKS) using $OKS = \frac{\sum_i \exp(-d_i^2/2s^2k_i^2)\delta(v_i>0)}{\sum_i \delta(v_i>0)}$. Specifically, d_i

represents the Euclidean distance between the ground truth and the predicted joint, v_i represents the target's visibility flag, s represents the object scale, and k_i represents a keypoint for each joint. Next, the average accuracy and recall scores are calculated. Table II shows the AR and AP averages from OKS=0.5 to OKS=0.95, with AP^M standing for medium objects and AP^L for large objects.

Implementation details All experiments are carried out using the codebase called AlphaPose[4] and tested on two datasets. The picture input resolution was reduced to 256×192 . The model was trained using CUDA 10.2 and CuDNN 7.3 on a single NVIDIA GTX 1080Ti GPU.

The method included data augmentation in model training, including flip, rotation at 40 degrees by design, and scale with the factor was set at 0.3. Set

the batch size to 4 and use the shuffle function when using training photos. In our experiment, there are 210 total epochs, and the base learning rate is set at 0.001 before being multiplied by 0.1 (learning rate factor) at the 170th and 200th epochs. The Adam optimizer, [9], was used, and the momentum is 0.9.

4.2 Experiment Result

COCO datasets result The proposed method compares each circumstance while adding different kinds of data augmentation for the pose estimator, as shown in Table 1. The Average Precision (AP) demonstrates that using the proposed method for cut-out gains 0.6 in mAP, which boosts accuracy by 1 percent. Furthermore, this study also investigates again another data augmentation[12], which is set up in almost a training process for pose estimator. The default data augmentation including Flip, Rotation, Scale, and Half body transform is trained again separately and shown in Tab.1. In total, when combining all of the data augmentations and applying the cut-out the AP slightly increases with 0.2 AP but in the case of the occluded key point in Fig.5 must better.

Table 1. The result for applying different kinds of data augmentation in HRNet

Backbone	Data augmentation	mAP
HRNet-W32	Without	72.9
HRNet-W32	Flip	73.6
HRNet-W32	Rotate	73.4
HRNet-W32	Scale	73.3
HRNet-W32	Half body transform	73.7
HRNet-W32	Cut-out (our)	73.5
HRNet-W32	All with out Cut-out	74.4
HRNet-W32	All with Cut-out	74.6

Table 2. Comparison on COCO Validation Dataset

Method	Backbone	Input size	#Params	AP	AP ⁵⁰	AP ⁷⁵	AP ^M	AP ^L	AR
8-Stage Hourglass[13]	8-Stage Hourglass	256×192	25.1M	66.9	-	-	-	-	-
Mask-RCNN[5]	ResNet-50-FPN	256×192	-	63.1	87.3	68.7	57.8	71.4	-
OpenPose[1]	-	-	-	61.8	84.9	67.5	57.1	68.2	66.5
PersonLab[14]	-	-	-	78.7	89.0	75.4	64.1	75.5	75.4
SimpleBaseline[24]	ResNet-50	256×192	34.0M	70.4	88.6	78.3	67.1	77.2	76.3
SimpleBaseline[24]	ResNet-101	256×192	53.0M	71.4	89.3	79.3	68.1	78.1	77.1
SimpleBaseline[24]	ResNet-152	256×192	68.6M	73.7	91.9	81.1	70.3	80.0	79.0
HRNetBaseline[19]	HRNet-W32	256×192	28.5M	74.4	90.5	81.9	70.8	81.0	79.8
HRNetBaseline[19]	HRNet-W48	256×192	63.6M	75.1	90.6	82.2	71.5	81.8	80.4
HRNet + our cut-out	HRNet-W32	256×192	28.5M	74.6	90.7	82.1	71.1	81.2	80.0
HRNet + our cut-out	HRNet-W48	256×192	63.6M	75.3	90.7	82.4	71.8	81.9	80.7

In Table 2, the proposed result was estimated on the COCO validation dataset. The AP in the suggested approach is greater than the Basic High-Resolution benchmark in all situations of 0.2 AP in both backbone HRNet-32 and HRNet-W48. Furthermore, the average recall (AR) is 0.3 points higher in the case of HRNet-W32 and 0.2 points higher in the case of HRNet-W48. In total, the experiment results slightly increased in both AP and AR but it significantly improve in the case of occluded keypoint. The visual result for heatmap detail can see in Fig.4 which shows that applying the proposed research makes the predicted heat map get more accurate. Fig.5 shows the qualitative result for the COCO 2017 dataset with 2 same images as the original pose estimator and proposed technique. For more detail, the green box means more occluded keypoint got detection while the red box means the wrong keypoint predicted.

Industrial datasets result: The proposed research will focus on the industrial environment in future work. Hence testing the pose estimation with diversity environment is necessary. This study tests the industrial dataset that contains 200 images for the occluded challenge. The result is shown in Fig.6 for the original and the improvement after applying the data augmentation.

5 Conclusion

This research shows the effect of the data augmentation on CNNs especially for occluded human keypoint, focusing on cut-out for human proposals. Furthermore, our work demonstrates that not increasing the computation cost, the data augmentation utilized has a more considerable effect. Moreover, the cut-out focused more on the essential feature map than the other element. The network will become more effective as a consequence, particularly for various computer vision-related tasks.

Besides, human pose estimation has several problems that need to be solved. First, the occluded joints were challenging to train and predict for the architecture. Second, human key points appear in the low-resolution images. The next issue is crowd situations which it is usually challenging to pinpoint where each participant's joints are located. Last but not least, there is a lack of data on photos with missing pieces for assessing human postures. The proposed method tries to solve the first problem is also the most complex case compared to all of the issues. Hence, future research will try to focus on the remaining problem and also try to apply the technique to other state-of-the-art pose estimators.

Acknowledgement

This results was supported by "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(MOE) (2021RIS-003)



Fig. 5. Qualitative images for human pose estimation in COCO2017 test-dev set. In two similar images, the Right side is the result of the original human pose estimator HRNet-W32, and Left side is our approach. Greenbox means better keypoint detection than normal. Redbox means inaccurate keypoint detection.

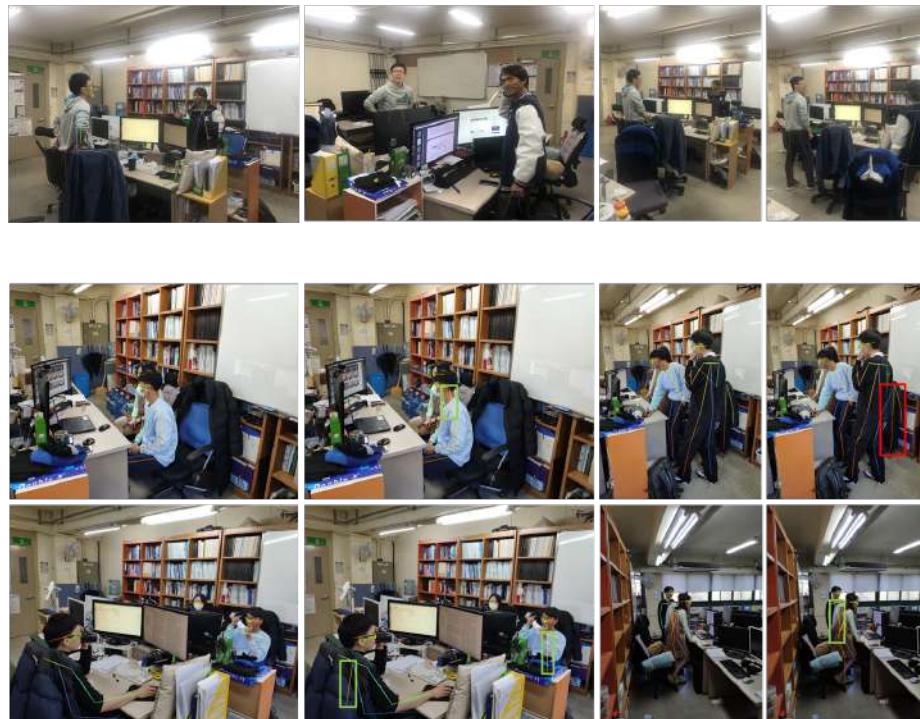


Fig. 6. Testing on the industrial dataset

References

1. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields (2016). <https://doi.org/10.48550/ARXIV.1611.08050>, <https://arxiv.org/abs/1611.08050>
2. Chen, C., Ramanan, D.: 3d human pose estimation = 2d pose estimation + matching. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5759–5767 (July 2017). <https://doi.org/10.1109/CVPR.2017.610>
3. Chou, C.J., Chien, J.T., Chen, H.T.: Self adversarial training for human pose estimation (2017)
4. Fang, H.S., Li, J., Tang, H., Xu, C., Zhu, H., Xiu, Y., Li, Y.L., Lu, C.: Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time (2022). <https://doi.org/10.48550/ARXIV.2211.03375>, <https://arxiv.org/abs/2211.03375>
5. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn (2017)
6. Hussain, Z., Sheng, M., Zhang, W.E.: Different approaches for human activity recognition: A survey (2019)
7. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift (2015)
8. Kim, E., Helal, S., Cook, D.: Human activity recognition and pattern discovery. IEEE Pervasive Computing **9**(1), 48–53 (Jan 2010). <https://doi.org/10.1109/MPRV.2010.7>
9. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. International Conference on Learning Representations (12 2014)
10. Li, S., Ke, L., Pratama, K., Tai, Y., Tang, C., Cheng, K.: Cascaded deep monocular 3d human pose estimation with evolutionary training data. CoRR **abs/2006.07778** (2020), <https://arxiv.org/abs/2006.07778>
11. Li, W., Zhao, R., Wang, X.: Human reidentification with transferred metric learning. In: Asian Conference on Computer Vision (ACCV). pp. 31–44 (11 2012)
12. Lin, T., Maire, M., Belongie, S.J., Bourdev, L.D., Girshick, R.B., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. CoRR **abs/1405.0312** (2014), <http://arxiv.org/abs/1405.0312>
13. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. CoRR **abs/1603.06937** (2016), <http://arxiv.org/abs/1603.06937>
14. Papandreou, G., Zhu, T., Chen, L.C., Gidaris, S., Tompson, J., Murphy, K.: Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model (2018). <https://doi.org/10.48550/ARXIV.1803.08225>, <https://arxiv.org/abs/1803.08225>
15. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019), <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>
16. Pytel, R., Kayhan, O.S., van Gemert, J.C.: Tilting at windmills: Data augmentation for deep pose estimation does not help with occlusions (2020). <https://doi.org/10.48550/ARXIV.2010.10451>, <https://arxiv.org/abs/2010.10451>
17. Reddy, N.D., Vo, M., Narasimhan, S.G.: Occlusion-net: 2d/3d occluded keypoint localization using graph networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7326–7335 (2019)

12 T.-D. Tran et al.

18. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement (2018).
<https://doi.org/10.48550/ARXIV.1804.02767>, <https://arxiv.org/abs/1804.02767>
19. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation (2019)
20. Tan, M., Pang, R., Le, Q.V.: Efficientdet: Scalable and efficient object detection. CoRR **abs/1911.09070** (2019), <http://arxiv.org/abs/1911.09070>
21. Tian, L., Liang, G., Wang, P., Shen, C.: An adversarial human pose estimation network injected with graph structure (2021).
<https://doi.org/10.48550/ARXIV.2103.15534>, <https://arxiv.org/abs/2103.15534>
22. Toshev, A., Szegedy, C.: Deeppose: Human pose estimation via deep neural networks. CoRR **abs/1312.4659** (2013), <http://arxiv.org/abs/1312.4659>
23. Wei, S.E., Ramakrishna, V., Kanade, T., Sheikh, Y.: Convolutional pose machines (2016)
24. Xiao, B., Wu, H., Wei, Y.: Simple baselines for human pose estimation and tracking. CoRR **abs/1804.06208** (2018), <http://arxiv.org/abs/1804.06208>
25. Yang, X., Wang, M., Tao, D.: Person re-identification with metric learning using privileged information. CoRR **abs/1904.05005** (2019), <http://arxiv.org/abs/1904.05005>

Multi-task model for glioma segmentation and isocitrate dehydrogenase status prediction using segmentation boundary

Xiaoyu Shi¹, Yinhao Li¹, Yen-Wei Chen^{*1}, Jingliang Cheng², Jie Bai², and Guohua Zhao²

1. Graduate School of Information Science and Engineering, Ritsumeikan University, Shiga, Japan,
2. The Affiliated Hospital of Zhengzhou University, Zhengzhou, China,
 is0490sr@ed.ritsumeい.ac.jp,
 yinli@fc.ritsumeい.ac.jp,
 chen@is.ritsumeい.ac.jp,
 fccchengjl@zzu.edu.cn,
 baijie113783501377@126.com,
 ghzhao@ha.edu.cn

Abstract. According to the 2021 World Health Organization IDH status prediction scheme for gliomas, isocitrate dehydrogenase (IDH) is a particularly important basis for diagnosis. In general, 3D multimodal brain MRI is an effective diagnostic tool for glioma. However, it is difficult for experienced doctors to only use brain MRI to predict the IDH status. Surgery is necessary to be performed for confirming the IDH. Past studies have confirmed that there is information about IDH in the tumor region on brain MRI images. These studies usually need to mark the glioma area in advance to complete the prediction of IDH status, which takes a lot of time and is costly. In this study, we proposed a multi-task deep learning model using 3D multimodal brain MRI images to achieve glioma segmentation and IDH status prediction simultaneously, which improved the accuracy of both tasks effectively. First, we used a segmentation model to segment the tumor region. Second, the segmentation results were then applied to the original input to obtain the glioma image, and another feature extractor was used to complete the prediction of IDH status. The effectiveness of the proposed method was validated via a public glioma dataset from the BraTS2020. Our experimental results show that our proposed method outperformed state-of-the-art methods with an accuracy of 84.2%, average dice of 79.3%. Thus, we predicted the IDH mutation status for glioma treatment with a 9% increase in accuracy and 2% increase in average dice for glioma treatment.

Keywords: Multi-task learning, Isocitrate Dehydrogenase, glioma segmentation, Computer diagnosis.

1 Introduction

1.1 Background

Brain tumors are classified as either primary or secondary, and gliomas are the most prevalent primary brain tumors [20]. Glioblastoma (GBM) is the most aggressive type of glioma worldwide. Less than 5% of glioblastoma patients survive for five years after diagnosis [26]. According to the 2016 and 2021 World Health Organization (WHO) IDH status prediction schemes for gliomas [8,18], isocitrate dehydrogenase (IDH) status is a particularly important basis for diagnosis. In addition, IDH mutation status has a strong correlation with the prognosis of glioma, and in low-grade gliomas, IDH mutant gliomas have a similar prognosis to IDH wild-type gliomas; however, IDH mutant glioblastomas have a better prognosis than IDH wild-type glioblastomas [8]. A follow-up survey showed that the presence of an IDH mutation predicted a good disease outcome and extended the median survival period of glioblastoma (IDH wild-type, 15 months; IDH mutant, 31 months) [9]. Therefore, it is necessary to predict the IDH mutation status for the treatment of glioma. 3D Magnetic resonance imaging (MRI) is commonly used to diagnose gliomas; see Fig. 1. Generally, 3D brain MRI produces four types of images: T1, T2, T1CE and FLAIR. Each modal has unique characteristics that are very useful for IDH status prediction. Moreover, the spatial information of 3D images is also very effective for IDH status prediction. Although it is difficult for experienced doctors to predict IDH using only MRI images, it can be predicted IDH using machine learning. For high-cost surgical diagnosis, the use of brain MRI to diagnose IDH status using a computer-aided diagnosis system is less costly and easier to perform. Therefore, the development of computer-aided diagnosis system based on machine learning is hot research.

With the development of machine learning, particularly deep learning, many improvements have achieved recently in computer-aided diagnosis. For the prediction of IDH status, Choi et al. [4] proposed a deep learning method using convolutional neural networks (CNNs) and glioma MRI images. Zhang et al. proposed a self-attention algorithm with a squeeze excitation network (SE-SA Net) [33]. However, owing to the limited amount of medical image data, the deep learning method often ignores the importance of medical guidance of doctors and faces the problem of overfitting. Therefore, Yan et al. [31] proposed a machine learning method based on radiomics features and medical knowledge, which achieved more satisfactory results. Furthermore, Zhang et al. [32] improved their SE-SA net with Radiomics features.

However, a general problem with these studies is the need to use glioma regions for IDH status prediction. Deep learning-based tumor IDH status prediction and gene prediction tasks usually use tumor regions as region of interest (ROI) to cut images for IDH status prediction, whereas radiomics based methods directly use tumor boundary masks to obtain internal tumor features. In practical application scenarios, it is high cost to obtain tumor region markers in advance. In addition, the segmentation and labeling of tumor regions can also assist doctors in the diagnosis of glioma. Fur

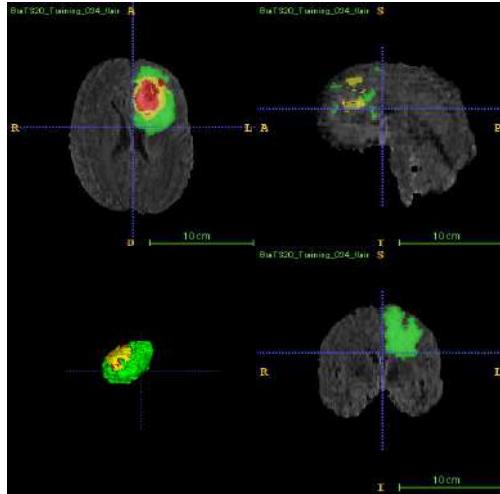


Fig. 1. Example of 3D brain MRI with glioma

–thermore, the IDH status also contains information about the glioma region owing to the existing relationship between the glioma region and the IDH status. There were several related works about multi-task learning for IDH status prediction and glioma segmentation [3,28,29]. Nevertheless, these works used entire glioma representation features for both segmentation and IDH status prediction, which ignored the importance of glioma ROI for IDH status prediction.

1.2 Contributions

To solve the above problems, we proposed a multi-task model based on deep learning, which used 3D multimodal brain MRI images to segment tumor regions simultaneously and use the segmentation results to predict the IDH genotypes. This model contains a segmentation part as well as a IDH prediction part for IDH prediction. Our key contributions are as follows:

A multi-task model for both segmentation and IDH status prediction tasks:

In this research, an integrated model is proposed to simultaneously segment glioma regions and predict IDH status. First, we use a segmentation model to segment the tumor region of the 3D MRI image. Second, we apply the obtained segmentation results to the original input to obtain the image of glioma region. Finally, we used a IDH status prediction model to extract high-level features of the tumor region and use them for IDH status prediction. Compared with the existing methods, we effectively improve the accuracy of segmentation task and prediction task using a multi-task joint model.

Effective ablation study and contrast experiments:

To prove the effectiveness of multi-task model, we conducted an effective ablation study on each part based on public dataset from the multimodal brain tumor segmentation (BraTS2020) [24] challenge. We also conducted contrast experiments with several state-of-the-art methods for brain tumor segmentation and IDH status predic

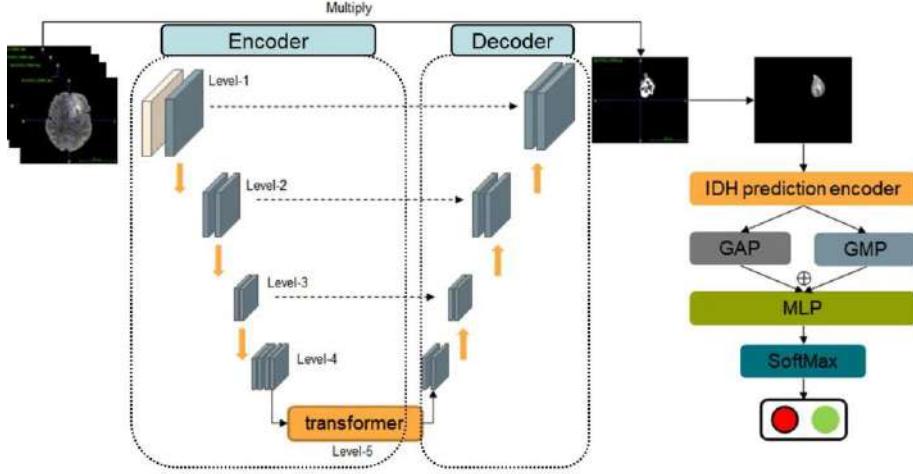


Fig. 2. Overview of multi-task model (In the multi-task network, input of 3D multimodal brain MRI generates the glioma segmentation results firstly. Then the segmentation prediction multiplied with input to get tumor area image, which is processed by IDH prediction network to predict the IDH status.)

-tion. When we compared our method with the current state-of-the-art methods, our proposed performed better both segmentation and prediction results.

The remainder of this paper is organized as follows. The proposed method using multi-task model is described in Section II. Ablation studies and comparative experiments are presented in Section III. Finally, we summarize and conclude our work in Section IV.

2 Methods

2.1 Overview

The overview of the proposed method is illustrated in the Fig. 2. In this research, we proposed a multi-task model, which contained a segmentation part for glioma region segmentation and a IDH status prediction part for IDH status prediction. This model first segmented glioma boundary using a TransBTS [27] as a segmentation network with 3D U-Net [11] encoder, decoder, skip-connection structure and a high-level vision transformer [6] layer. Then, the masks generated by the prediction results were multiplied with the original input to obtain the part containing only the tumor region. Finally, we selected to adopt the same encoder structure and transformer layer for high-level features as IDH status prediction network to extract tumor region features and used them for IDH status prediction.

In the training stage, the two models are trained with the labels of segmentation and IDH status respectively, and the loss function of the entire model is obtained by the weighted average of the loss functions of the two parts. The way the weighted average

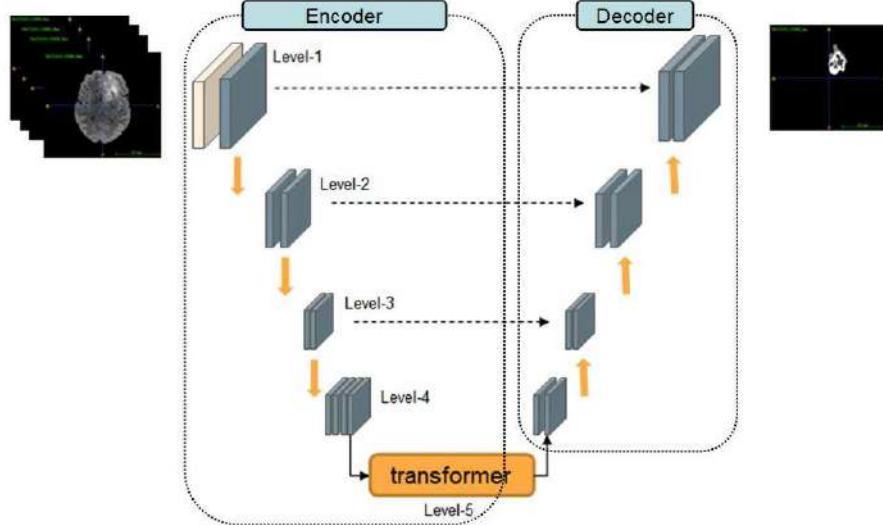


Fig. 3. Segmentation network

calculated will be introduced in the following sections. In the testing stage, only 3D multimodal brain MRI data is input to obtain the predicted tumor boundary mask and IDH status information. In the proposed method, the multi-task model using only one model to complete the requirements of two tasks at the same time and obtain better results. Details of the segmentation model, IDH status prediction model and multi-task learning loss function are introduced in the following.

2.2 Glioma segmentation

For the study of glioma, the automatic segmentation of tumor regions in multimodal brain MRI images by computer can effectively help doctors to make auxiliary diagnosis. In recent years, with the development of deep learning in computer vision, especially in the field of image processing, more and more tumor segmentation methods based on deep learning have appeared. Among them, various studies [15,30] are mainly based on U-Net [23], which proposed skip-connection structure. Compared with the past methods, these deep learning-based studies have greatly improved the accuracy of segmentation. However, limited by the size of the data set and network, most of these methods intercept MRI slices as data for training and testing, which makes the model lack of spatial relevant information about the tumor region. With the progress of equipment, many studies have recently used 3D convolutional neural networks to directly process 3D images, which retained the spatial information of the tumor as much as possible, and further improved the segmentation results, such as the use of 3D U-Net for glioma segmentation [11]. Furthermore, with the development of Transformers using self-attention mechanism in the field of computer vision in the past two years, transformers [6] are applied to more and more medical image segmentation research [19,22,23]. For the segmentation of 3D multimodal brain MRI images,

Wang et al. proposed a TransBTS [27] method based on 3D U-Net and transformer encoder to further improve the segmentation results of Glioma. In our proposed multi-task model, we selected this method as our segmentation network due to the high accuracy of glioma segmentation.

The structure of segmentation network is shown in Fig. 3.

3D CNN-transformer encoders:

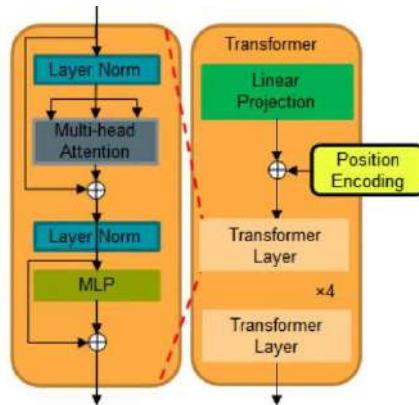


Fig. 4. Transformer unit

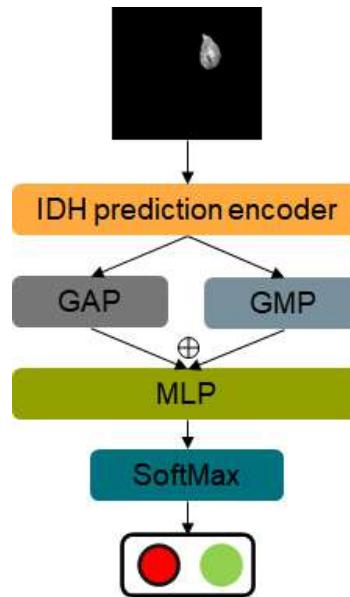


Fig. 5. IDH status prediction network

The dimension of the input $x \in \mathbb{R}^{C \times H \times W \times D}$ can be expressed as a 3D resolution of $H \times W \times D$ and C channels. The input was first operated by an initial convolution

with a kernel size of $3 \times 3 \times 3$ to generate feature maps with 16 channels. In order to capture the higher-level features and the spatial relationship of the images, we down sampled the feature maps T times by using residual convolutions. For each down-sampling operation, we used a $3 \times 3 \times 3$ convolution with a stride of 2 the output channel was set to twice the input channel. Each residual convolution block (Resnet Block) consists of two convolution blocks and a residual connection was applied to between the input and output of the residual block. Each convolution block was composed of a batch normalization, a Leaky ReLU function, and a $3 \times 3 \times 3$ convolution layer. The output of encoder E , feature maps $F_\lambda^{C' \times H' \times W' \times D'}$ in level λ can be denoted by:

$$F_\lambda = E(\theta; x; \lambda) \in R^{2^{\lambda+3} \times \frac{H}{2^{\lambda-1}} \times \frac{W}{2^{\lambda-1}} \times \frac{D}{2^{\lambda-1}}}, \quad (1)$$

where θ is the parameters of encoder, x is the input.

The architecture of transformer unit is shown in Fig. 4. Inspired by the vision transformer, TransBTS enhances the long-term connections of images by adding a transformer layer based on self-attention mechanism at the highest-level features (level-4 → level-5). One problem that needs to be solved is that for 3D images, the method of directly cutting the image into several patches and generating tokens is difficult to implement because of the large computational cost. Therefore, in this transformer, we focus more on high-level features. First, we select the features for the last layer, $\lambda=4$ and $F_4 \in R^{128 \times \frac{H}{8} \times \frac{W}{8} \times \frac{D}{8}}$. We then embedded these features to 512 channels by $3 \times 3 \times 3$ convolutions, keeping the size constant and reshaping them into $N \times d$, where the $N = \frac{H}{8} \times \frac{W}{8} \times \frac{D}{8}$ and is the dimension of feature maps. To learn the positional information, a learnable position embedding $E_{pos} \in R^{N \times d}$ was fused into the patch embedding $E_{pat} \in R^{N \times d}$ by an addition operation, which is denoted as follows:

$$Z_0 = E_{pos} + E_{pat}, \quad (2)$$

where Z_0 denotes the feature of transformer encoder input. Like the regular transformer encoder, the self-attention mechanism triplet (Q , K , V) at l-layer transformer encoder can be denoted as:

$$Q = z^{L-1} w_Q, K = z^{L-1} w_K, V = z^{L-1} w_V, \quad (3)$$

where the $w_Q, w_K, w_V \in R^{d \times d}$ are learnable parameters of the three linear projection layers. Then the self-attention SA can be calculated as:

$$SA(z^{L-1}) = z^{L-1} \left(\frac{Qk^T}{\sqrt{d}} \right) V. \quad (4)$$

Multi-head self-attention (MSA) is a very important part in transformer. It can complete the learning of representations within different subspaces by multiple heads.

Specifically, it divides the input of transformer layer into an independent parts, processes each part in parallel using SA operation, and then projects these concatenated results using a linear projection layer. Therefore, MSA can be denoted as:

$$MSA(z^{L-1}) = \text{Concat}(\text{SA}_1(z^{L-1}), \dots, \text{SA}_n(z^{L-1}))w_o, \quad (5)$$

where $w_o \in \mathbb{R}^{d \times d}$ are learnable parameters of the linear projection layer. Then, the output of MSA was converted by an MLP block with a residual skip as the layer output z^L . The transformer encoder can be denoted as follow:

$$z_k^L = z^{L-1} + MSA(z^{L-1}), \quad (6)$$

$$z^L = z_k^L + \text{MLP}(z_k^L). \quad (7)$$

Segmentation decoder:

In order to obtain voxel-level segmentation results, we need to pass high-level features through decoder to generate regions for glioma segmentation. In this process, we select $3 \times 3 \times 3$ 3D convolutions with stride 2 to generate segmentation results as $H \times W \times D$ from feature maps $d \times \frac{H}{8} \times \frac{W}{8} \times \frac{D}{8}$. Skip-connection was used to further improve the segmentation effect. In this process, we used the features of the corresponding level in the encoder to perform skip-connection. Finally, a $1 \times 1 \times 1$ convolution followed by a SoftMax function as the segmentation layer of the decoder was applied to generate the segmentation result.

2.3 IDH status prediction

It has been confirmed in past studies that the information extracted from the glioma region image has the information of IDH status. Some of these studies used deep learning methods, and some used radiomics to extract the features of the tumor region to predict the IDH status of patients. In this study, we selected to use the 3D CNN-transformer encoder with the same structure as the segmentation network to extract high-level features from tumor regions and use MLP as a classifier to predict IDH categories.

As illustrated in Fig. 5, the IDH status prediction network used the same structure as the segmentation encoder. We selected to adopt level 4 and level 5 feature maps as IDH prediction features to perform the prediction. The sizes of feature maps are $128 \times \frac{H}{8} \times \frac{W}{8} \times \frac{D}{8}$ and $512 \times \frac{H}{8} \times \frac{W}{8} \times \frac{D}{8}$. First, we reshaped feature maps into $128 \times N$ and $512 \times N$. Second, in order to highlight important features, we computed a global Max pooling on the same feature map in addition to the usual global average pooling. Using pooling also helps to save computation because pooling does not involve the computation of parameters. After performing two kinds of pooling on the two feature maps, we obtained the following feature maps: $F_{\text{GAP}}^4 \in \mathbb{R}^{128}$, $F_{\text{GMP}}^4 \in \mathbb{R}^{128}$, $F_{\text{GAP}}^5 \in \mathbb{R}^{512}$, $F_{\text{GMP}}^5 \in \mathbb{R}^{512}$. To predict the IDH status using these feature maps, an MLP consisting of multiple linear projection layers was used as a classifier. Finally, we combined these feature maps by concatenation, and used MLP to reduce the

features. A SoftMax converts the features into the probability value of IDH mutation and non-mutation prediction and calculated the loss function using the prediction probability for training. This process can be denoted as:

$$P = \text{SoftMax}(\text{MLP}(\text{Concat}(F_{\text{GAP}}^4, F_{\text{GMP}}^4, F_{\text{GAP}}^5, F_{\text{GMP}}^5))) \quad (8)$$

2.4 Multi-task model learning

Fig .2 shows the overview of the multi-task model. We can think of the network as the following parts: (1) Segmentation encoder, E , (2) Segmentation decoder, D , (3) IDH status prediction network, P . For a multimodal 3D brain MRI as input $x \in \mathbb{R}^{C \times H \times W \times D}$, we first input it into the segmentation encoder E to obtain the high-level features of the entire MRI image, and then segmented the tumor boundaries using the decoder D and a SoftMax function to obtain the segmentation probability result g_p . Second, we multiplied the predicted tumor segmentation region by the original input x to generate the tumor region part, $\tilde{x} \in \mathbb{R}^{C \times H \times W \times D}$, and input it into the IDH state prediction network to obtain the predicted IDH result y' . The overall computational process of the model can be denoted as:

$$g_p = \text{SoftMax}(D(E(x))), \quad (9)$$

$$y' = \text{SoftMax}(P(g_p \times x)). \quad (10)$$

We got the output of the segmentation model, g_p and the prediction model, y' . The model used dice loss function to train the model for the segmentation task and cross-entropy function to train the model for the prediction task. The loss function can be denoted as:

$$L_{\text{seg}} = 1 - \frac{g_p g_t}{g_p + g_t}, \quad (11)$$

$$L_{\text{idh}} = - \sum y \log y', \quad (12)$$

where the g_t is the ground truth of tumor mask, and y is the label of IDH status.

In order to jointly train the multi-task model, it is necessary to select appropriate weights for each task to train. However, manually selected weights take a lot of time to adjust and are also not practical. Therefore, we refer to the training method based on uncertainty model [22], use maximum likelihood estimation to calculate the weight of each model, and define two learnable parameters to dynamically adjust the weight of the loss function, so as to obtain better training results. The total loss function for the multi-task model is denoted as:

$$L_{\text{joint}} = \frac{L_{\text{seg}}}{2\sigma_{\text{seg}}^2} + \frac{L_{\text{idh}}}{2\sigma_{\text{idh}}^2} + \log \sigma_{\text{seg}} \sigma_{\text{idh}}, \quad (13)$$

where the σ_{seg} and σ_{idh} are the learnable parameters for network training. These two values are initialized to 1 to ensure that the training starts with the same weight for both tasks.

3 Experiments

3.1 Dataset

We obtained 148 multimodal MRI images from the public dataset from the multimodal brain tumor segmentation (BraTS2020) [24] challenge training dataset as our training dataset, whereas 70 multimodal brain MRI images from the Brats2020 validation set as test set. Part of BraTS2020 data belongs to The Cancer Imaging Archive (TCIA) [14], which can be available from our public repository. Genomic information is provided from The Cancer Genome Atlas (TCGA) [19]. The MRI images used in this experiment included two types: IDH wild-type and mutated-type. This training dataset contained 57 patients with an IDH mutation and 91 patients with wild type IDH. The test dataset contained 32 mutant cases and 38 wild cases.

Each patient data used in the experiments had four MR image modalities (T1, T2, T1ce, and FLAIR) with IDH status information. Registration of multimodal brain MRI images is very important due to the differences between different modalities. In BraTS2020, both training set and validation set data are labeled by experts and registered on T1 modal. In our experiments, we directly use the registered data. In this dataset, there are three parts of annotations for the tumor region, which are respectively whole tumor (WT), tumor core (TC), and enhancing tumor (ET).

3.2 Experiments

To confirm the influence of multi-task learning between segmentation task and IDH status prediction task, we performed the following ablation studies. The first model (Model 1) was the IDH status prediction model. We trained the IDH network same as the multi-task learning in the same manner and validated the IDH status prediction results. The second model (Model 2) was the segmentation model. We used the same method as the proposed method to segment the tumor area. The third model (Model 3) was the proposed novel multi-task model, which used the segmentation prediction as the tumor area to predict the IDH status using IDH network. We also conduct comparative experiments with the state-of-the-art methods for single task.

We selected the area under the curve (AUC), accuracy (Acc), sensitivity (Sens), and specificity (Spec) as our IDH status prediction evaluation measures. We selected the Dice coefficient (Dice) as the segmentation evaluation measure. For the three parts of tumor area, we computed dice for the three parts independently and finally compute the three-part average.

We used the Adam optimizer with a batch size of 1 and a learning rate of 0.001, without data shuffling in our deep learning model. All the training was conducted on two Nvidia GeForce RTX 3090 24GB GPU. The proposed method achieves the best performance. In order to ensure the effectiveness of the segmentation results, we first pre-train the segmentation network and then add the IDH status prediction network for fine-tuning.

3.3 Experimental results

The results of the ablation studies are presented in Table 1. As shown in Table 1, Models 1 and 2 show the results of using only IDH status prediction or segmentation network. Model 3 shows that better results are achieved when integrating the two tasks. We thought that because the tumor region was used as the input of the IDH state prediction network, the prediction ability of the IDH state was improved compared with the method of inputting the whole image, and the information of the IDH state further improved the ability of the segmentation network to locate the tumor region.

Finally, we conducted a comparative experiment with the current state-of-the-art methods [7,12,13,23,28,29]. The results are presented in Table 2. Our proposed method achieved better IDH status prediction and segmentation performance than the state-of-the-art methods which means that our multi-task model performs better. The segmentation performance of our proposed multi-task method in TC and ET is significantly higher than other models, which is because IDH is more sensitive to the information within the internal boundaries of the tumor rather than glioma edema region.

4 Discussion and Conclusion

In this study, we proposed a multi-task model that combines both IDH status prediction task and brain tumor segmentation. From the experimental results of our proposed method, we achieved AUC of 88.2%, accuracy of 84.2%, sensitivity of 71.8%, specificity of 94.7% for IDH status prediction task and Dice of whole tumor of 86.84, tumor core of 76.33, enhancing tumor of 74.66, which achieved better performance than state-of-the-art methods. Multi-task learning is a field worth exploring in the field of medical image processing. Multi-task learning not only improves the performance of two tasks using the complementarity of multiple tasks to a certain extent, but also needs computer-aided diagnosis systems that can complete multiple tasks to assist doctors in diagnosis in the medical field. Since the IDH status prediction task needs to predict using the tumor target region, we used the tumor region generated by the segmentation task as the input of the prediction network to complete the need to achieve two tasks at the same time. Meanwhile, the IDH network further improves the performance of the segmentation network in the joint learning process.

In the future, we would like to improve our multi-task network by more efficient segmentation networks as well as IDH status prediction networks. At present, the problem we face is that for 3D medical images, the 3D network model we now design is large and has some shortcomings in the training process and practicability. We hope to explore more connections between multiple tasks and use smaller networks to achieve better results with less cost.

Table 1. Ablation study of single task and multi-task model

	AUC(%)	Acc(%)	Sens(%)	Spec(%)	WT (mean%)	TC (mean%)	ET (mean%)	Avg (mean%)
Model1	88.2	81.4	62.5	97.3	-	-	-	-
Model2	-	-	-	-	88.61	74.46	68.65	77.24
Model3	88.2	84.2	71.8	94.7	86.84	76.33	74.66	79.27

Table 2. Comparison with the state-of-the-art methods

	AUC(%)	Acc(%)	Sens(%)	Spec(%)	WT (mean%)	TC (mean%)	ET (mean%)
SENet101[12]	79.6	77.1	65.6	86.8	-	-	-
DenseNet121[25]	78.2	77.1	65.6	86.8	-	-	-
ResNet50[13]	77.5	74.2	62.5	84.2	-	-	-
U-Net[23]	-	-	-	-	88.79	71.35	71.84
V-Net[7]	-	-	-	-	88.84	70.44	69.67
UNETR[1]	-	-	-	-	87.42	68.68	70.39
SGPNet[28]	81.3	80.0	62.5	94.7	87.04	69.33	71.43
CMSVNet[29]	75.9	72.9	56.3	86.8	78.29	58.43	67.50
Proposed method	88.2	84.2	71.8	94.7	86.84	76.33	74.66

needs to predict using the tumor target region, we used the tumor region generated by the segmentation task as the input of the prediction network to complete the need to achieve two tasks at the same time. Meanwhile, the IDH network further improves the performance of the segmentation network in the joint learning process.

In the future, we would like to improve our multi-task network by more efficient segmentation networks as well as IDH status prediction networks. At present, the problem we face is that for 3D medical images, the 3D network model we now design is large and has some shortcomings in the training process and practicability. We hope to explore more connections between multiple tasks and use smaller networks to achieve better results with less cost.

References

1. A. Hatamizadeh et al., "UNETR: Transformers for 3D medical image segmentation," arXiv:2103.10504, 2021.
2. Chen, Y.-W., and Jain, L.C., (Eds): *Deep Learning in Healthcare*, Springer, 2020.
3. Cheng J, Liu J, Kuang H, Wang J. A Fully Automated Multimodal MRI-Based Multi-Task Learning for Glioma Segmentation and IDH Genotyping. IEEE Trans Med Imaging. 2022 Jun;41(6):1520-1532. doi: 10.1109/TMI.2022.3142321. Epub 2022 Jun 1. PMID: 35020590.
4. Choi, Yoon Seong, et al. "Fully automated hybrid approach to predict the IDH mutation status of gliomas via deep learning and radiomics." Neuro-oncology, 2020.
5. David N Louis, Arie Perry, Pieter Wesseling, Daniel J Brat, et al. "The 2021 WHO IDH status prediction of Tumors of the Central Nervous System: a summary", *Neuro-Oncology*, Volume 23, Issue 8, Pages 1231–1251, 2021,
6. Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

7. F. Milletari, N. Navab, and S.-A. Ahmadi, “V-Net: Fully convolutional neural networks for volumetric medical image segmentation,” in *Proc. 4th Int. Conf. 3D Vis. (3DV)*, pp. 565–571, Oct. 2016.
8. Hai Yan, D. Williams Parsons, Genglin Jin, et al. “IDH1 and IDH2 mutations in gliomas.” *N Engl J Med*, 2009.
9. Han S, Liu Y, Cai SJ, Qian M, et al. “IDH mutation in glioma: molecular mechanisms and potential therapeutic targets”. *Br J Cancer*. PMID: 32291392; PMCID: PMC7250901. Doi: 10.1038/s41416-020-0814-x, 2020.
10. Huang, H., Zheng, H., Lin, L., et. al. “Medical Image Segmentation with Deep Atlas Prior.” *IEEE Trans. Medical Imaging*, Vol.40, No.12, pp.3519-3530, 2021.
11. J. Cheng, J. Liu, L. Liu, Y. Pan, J. Wang, et al. “Multi-level glioma segmentation using 3D U-Net combined attention mechanism with atrous convolution,” in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Nov. 2019, pp. 1031–1036
12. J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 7132–7141, Jun. 2018,
13. K. Chang, H. X. Bai, H. Zhou, and C. Su, “Residual convolutional neural network for the determination of IDH status in low-and high-grade gliomas from MR imaging,” *Clin. Cancer Res.*, vol. 24, no. 5, pp. 1073–1081, 2018.
14. K. Clark et al., “The cancer imaging archive (TCIA): Maintaining and operating a public information repository,” *J. Digit. Imag.*, vol. 26, no. 6, pp. 1045–1057, 2013.
15. K. Qi et al. “X-Net: Brain stroke lesion segmentation based on depthwise separable convolution and long-range dependencies,” in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, pp. 247–255, 2019.
16. Kitrungrotsakul, T., Chen, Q., Wu, H., et. al. “Attention-RefNet: Interactive Attention Refinement Network for Infected Area Segmentation of COVID-19,” *IEEE Journal of Biomedical and Health Informatics*, Vol.25, No.7, pp. 2363-2373, 2021.
17. Liang, D., Lin, L., Hu, H., et. al. “Combining Convolutional and Recurrent Neural Networks for IDH status prediction of Focal Liver Lesions in Multi-Phase CT Images,” In: Frangi A., Schnabel J., Davatzikos C., Alberola-López C., Fichtinger G. (eds) *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018. MICCAI 2018, Lecture Notes in Computer Science*, LNCS7951, Springer, pp.666-675, 2018.
18. Louis DN, Perry A, Reifenberger G, et al. "The 2016 World Health Organization IDH status prediction of tumors of the central nervous system: a summary". *Acta Neuropathol*; 131(6):803–820, 2016.
19. M. Ceccarelli et al., “Molecular proling reveals biologically discrete subsets and pathways of progression in diffuse glioma,” *Cell*, vol. 164, no. 3, pp. 550–563, 2016.
20. Ostrom QT, Gittleman H, Xu J, et al. “Primary brain and other central nervous system tumors diagnosed in the United States in 2009–2013”. *CBTRUS statistical report*, Neuro Oncol, ;18(suppl_5): v1–v75, 2016.
21. Peng, L., Lin, L., Hu, H., et. al. “IDH status prediction and Quantification of Emphysema Using a Multi-Scale Residual Network,” *IEEE Journal of Biomedical and Health Informatics*, Vol.23, No.6, pp.2526-2536, 2019.
22. R. Cipolla, Y. Gal, and A. Kendall, et al. “Multi-task learning using uncertainty to weigh losses for scene geometry and semantics,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 7482–7491, Jun. 2018.
23. Ronneberger, Olaf, Philipp Fischer, Thomas Brox, et al. “U-Net: Convolutional networks for biomedical image segmentation,” *International Conference on Medical image computing and computer-assisted intervention*, Springer, Cham, 2015.

24. S. Bakas et al., "Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features," *Scientific data*, vol. 4, Art. no. 170117, Sep. 2017.
25. S. Liang et al., "Multimodal 3D DenseNet for IDH genotype prediction in gliomas," *Genes*, vol. 9, no. 8, p. 382, Jul. 2018.
26. Steven De Vleeschouwer S. "Glioblastoma [Internet]." *Brisbane (AU): Codon Publications*; PMID: 29251853, 2017. Doi: 10.15586/codon.glioblastoma, 2017.
27. Wang, Wenzuan, et al. "Transbts: Multimodal brain tumor segmentation using transformer," *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, Cham, 2021.
28. Y. Wang, Y. Wang, C. Guo, S. Zhang, and L. Yang, "SGPNet: A threedimensional multi-task residual framework for segmentation and IDH genotype prediction of gliomas," *Comput. Intell. Neurosci.*, vol. 2021, pp. 1–9, Apr. 2021.
29. Y. Zhou et al., "Multi-task learning for segmentation and classification of tumors in 3D automated breast ultrasound images," *Med. Image Anal.*, vol. 70, May 2021, Art. no. 101918.
30. Y. Zhou, W. Huang, P. Dong, Y. Xia, S. Wang, et al. "D-UNet: A dimension-fusion U shape network for chronic stroke lesion segmentation," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 18, no. 3, pp. 940–950, May 2021
31. Yan, J., Zhang, B., Zhang, S. et al. "Quantitative MRI-based radiomics for noninvasively predicting molecular subtypes and survival in glioma patients." *npj Precis.* Doi: 10.1038/s41698- 021-00205-z, 2021.
32. Zhang, X., Shi, X., Iwamoto, Y., et al. "IDH mutation status prediction by a radiomics associated modality attention network". *Visual Comput.* Doi: 10.1007/s00371-022-02452-y, 2022.
33. Zhang, Xinran, Iwamoto Yutaro, et al. "IDH Mutation Status Prediction by Modality Self Attention Network." *Innovation in Medicine and Healthcare*. Smart Innovation, Systems and Technologies, vol 242. Springer, Singapore, pp.51-57, 2021.

Impression Estimation of Suit Patterns Based on Style Features Using Multi-scale CNN

Eiki Tsumura¹, Kesnke Tobitani^{2[0000-0002-3898-8435]},
 Miyuki Toga^{1[0000-0002-2385-4389]}, and Noriko Nagata^{1[0000-0002-2037-1947]}

¹ Kwansei Gakuin University, 2-1 Gakuen, Sanda-shi, Hyogo 669-1337, Japan
 {E.T-tsumu17,toga.m,nagata}@kwansei.ac.jp

² University of Nagasaki, 1-1-1 Manabino, Nagayo-cho, Nishi-Sonogi-gun, Nagasaki
 851-2195 ,Japan
 tobitani@sun.ac.jp

Abstract. In the field of fashion design, impressions evoked by the texture of materials, such as “flashy” and “cool” (affective texture), attract attention. The affective texture is considered necessary in evaluating and judging the quality of an object and personal preferences. In the fashion domain, there is a high demand for personalization of designs when diversifying user needs. One example is a custom-made suit service. However, there is a problem that it is labor-intensive to find a suit that matches image from many patterns, and colors. It is, therefore, necessary to understand affective texture. Many studies evaluate aesthetics using product style, suggesting that style is highly associated with affective texture. Multi-scale CNN has attracted attention for image recognition and is more accurate than single-scale CNN. However, no model that combines style and multi-scale CNN has been developed in previous studies. In this study, we propose a method for affective texture (visual impressions) evoked by suit patterns, corresponding to different scales of the pattern. (1) Suit patterns are collected in different resolution images, and impression evaluation experiments quantify (2) affective texture. (3) Model the relationship between the affective texture and physical characteristics (style features) of the pattern images using a multi-scale CNN. Then, the correlation coefficient between the impression values of the test data and the estimated impression values of the models. The results showed that multi-scale CNN has better accuracy than single-scale CNN, confirming the effectiveness of this method.

Keywords: Fashion · Suit · Style · Multi-scale CNN · Impression Estimation Models.

1 Introduction

In the field of fashion design, impressions (including aesthetics) evoked by the texture of materials, such as “flashy” and “cool” (affective texture), attract attention. The affective texture is considered necessary in evaluating and judging the quality of an object and personal preferences. In the fashion domain, the

impression that clothes make on people plays a vital role in expressing a person's impression [1]. The pattern and color give different impressions to people. Therefore, one must dress appropriately for different occasions when wearing a suit. Therefore, technologies are required to quantify, index, and model affective texture [2].

With the improvement of information technology, users can quickly obtain various of product information through the Internet and SNS. User needs and preferences have become increasingly diverse in recent years, and demand for customization and personalization of products and designs has grown. Therefore, it is essential to accurately grasp users' affective values, such as their preferences and satisfaction [3].

The tailor-made suit service is an example of the customization and personalization of design in the fashion field. However, finding a suit that matches one's taste and image from many available materials, patterns, and colors is time-consuming and labor-intensive. The perception of a pattern's motif may significantly affect the pattern's overall impression (a unit that constitutes the subject of the patterns) [4]. The entire motif is observed at a low resolution if the motif is large. If the motif is small, the details are observed at high resolution (motifs can be combined to form a concept motif at a higher level). In this way, people form an impression of the pattern while appropriately changing the resolution according to the size of the motif.

In this study, we propose a method to construct impression estimation models for suit patterns based on affective texture by modeling the pattern's visual impression and physical characteristics. In particular, we aim to extend the model to impression estimation using multi-resolution (multi-scale) images to accommodate differences in pattern scale.

2 Previous Research

Using machine learning techniques, numerous studies have been conducted to model the impressions evoked by products and their physical characteristics. These include technologies that use a product's color, texture, and shape, as physical characteristics [5] and retrieval technologies based on impressions of images [6].

There has been much research on textures [7–9]. On the deep learning framework, Gatys et al. proposed a style transformation algorithm that content features and style features extracted from VGG19 [10] which is a CNN used for general object recognition [11]. Content features are feature maps output from middle layer of VGG19 style features is grammaticalized versions of content features. Gatys et al. have proposed that content features hold a lot of shape information necessary for general object recognition. In contrast, style features are texture features that hold many detailed appearance information, such as color and pattern, in an image.

Many studies have related to image style and kansei (sensibility) [12, 13]. These studies have built models that use style to evaluate the aesthetics of

photographs [12] and to estimate the affective texture of clothing patterns [13]. The results suggest that style information in images is a feature with a high affinity to affective texture. However, since the emotions and impressions evoked by products and textures vary depending on the object's size, an estimation model that can respond to differences in the object's scale is desirable.

Multi-scale CNN, a deep learning model that considers differences in image scale, has attracted attention in image recognition. Wetteland et al. proposed a multi-scale CNN with pre-trained VGG16 concatenated in parallel [14]. It is a classification network that takes individual input images of different resolutions, and concatenates the features of each resolution image. Global Average Pooling (GAP) and Dropout layer are added to prevent over-learning. The results showed that multi-scale CNN improved classification accuracy over single-scale CNN.

In this study, we construct impression estimation style features with high affinity to affective texture and a multi-scale CNN framework to deal with differences in the scale of suit patterns (motifs). No model that combines style and multi-scale has been developed in previous studies.

3 Proposed Method

In this study, we propose a method for estimating affective texture (visual impressions) that the suit patterns evoke in response to differences in pattern scale. Fig. 1 presents an overview of the proposed method. (1) First, a scanner captured the suit's fabric which is collected as pattern images at various resolutions. (2) Next, we conducted an impression evaluation experiment on the pattern image of the suit to quantify the affective texture. (3) We then used the style feature as the suit patterns' physical characteristics. We modeled relationship between the quantified affective texture and the physical characteristics using a multi-scale CNN. Finally, we conducted impression estimation on the test data based on the constructed models to confirm the proposed method's effectiveness.

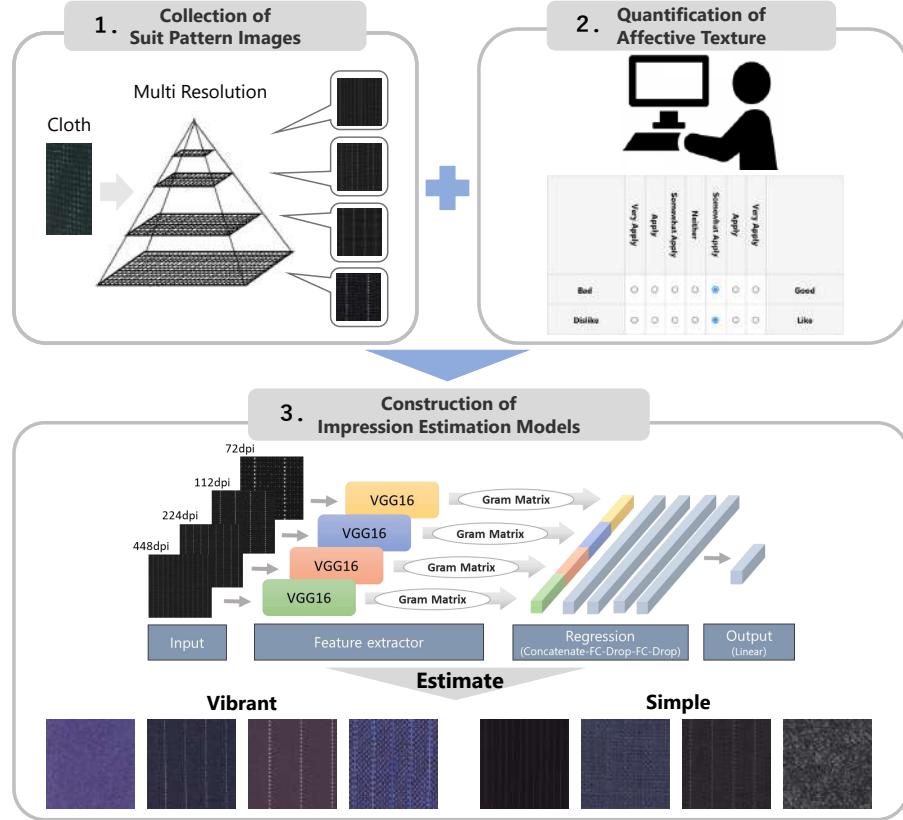


Fig. 1. Overview of our proposed method.

4 Quantification of Affective Texture

4.1 Collection of image data

The suit fabrics were photographed with high dynamic range (HDR) images at different resolutions to collect images representing different scales. We collect sample images at different resolutions for 613 different suit cloths based on the image pyramid of Tada et al. [15]. The resolution was set to 72 dpi, 112 dpi, 224 dpi, and 448 dpi, and a total of 2,452 pattern images were collected. Without loss of generality, capturing with HDR images preserves tone and light regions, resulting in a more transparent representation of woven patterns such as stripes and checks. However, HDR images cannot be viewed on a standard display, so they must be converted to low dynamic range (LDR) images, which can be viewed by compressing their dynamic range. In this study, we used “Photographic Tone Reproduction” by Reinhard et al. [16], a representative tone mapping method,

and converted the images to LDR images. We normalized all image sizes to 224 × 224pixel. Examples of suit patterns collected from the process are shown in Fig. 2. A scanner (Epson GT -X830) and software SilverFast 8 were used to perform the analysis. The suit fabrics used for the photography dealt with some from the 2016-2019 SS and AW.

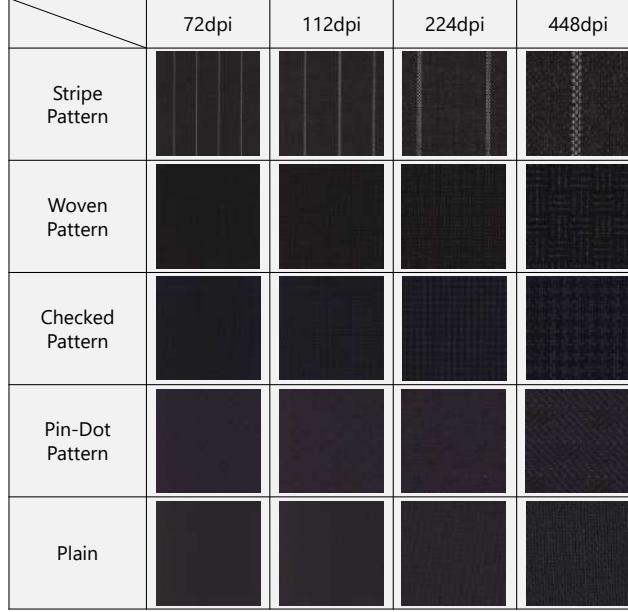


Fig. 2. Pattern images of suits with different resolutions.

4.2 Impression evaluation experiment

To quantify the affective texture evoked by suit patterns, we conducted an impression evaluation experiment using crowd-sourcing on a total of 2,452 pattern images of 613 types and four resolutions. We used Lancers as a crowd-sourcing service. In kansei (sensibility including aesthetics) research, models of kansei are often expressed in a hierarchical structure [17] consisting of three layers: emotion, impression, and physical element (Fig. 3). By quantifying the impression layer, we assume that we can clarify the basis (causal relationship) for the formation of emotional values in the correspondence between "people" (emotions) and "objects" (physical characteristics). In the experiment, to select evaluation words to be used in the experiment, more than 60 words were listed through discussions with the suit experts, and we asked the suit experts to evaluate their impressions based on these evaluation words. We conducted factor analysis on the evaluation data obtained, and we adopted words with high factor loadings as

suitable words for evaluating suit patterns. Table 1 lists the selected evaluation terms. We selected 44 words, 4 for the emotional layer and 40 for the impression layer, corresponding to the hierarchical structure of kansei (Fig. 3). The experiment participants were 3,080 of gender and age and were asked to rate a single stimulus for 20 people.

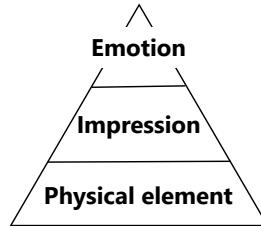


Fig. 3. Hierarchical structure of kansei (sensibility).

Table 1. Forty-four evaluation words were used in the impression evaluation experiment.

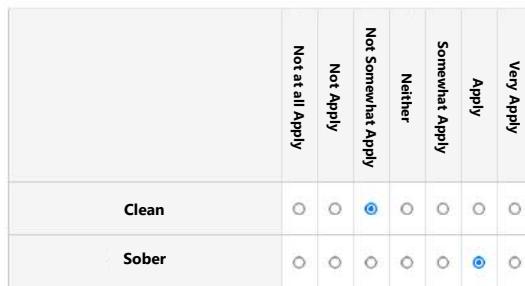
crisp	calm	clear	formal
plain	unique	vibrant	suitable for everyone
sharp	fashionable	simple	sober
well-tailored	adult	standard	youthful
refreshingly cool	dressy	massive	trendy
elegant	powerful	modern	sexy
classical	kind	neat	smart
supple	prominent	squeezed	stylish
gorgeous	good impression	clean	serious
light feeling	cool	humility	warm
deactivation - activation	displeasure - pleasure	dislike-like	bad-good

The experimental procedure was conducted using a Likert scale and the Affect-Grid method of Russell et al. [17]. Participants were asked to observe the pattern images of the suits displayed on the screen. They were asked to respond to the degree to which each evaluation term applied to them using a seven-point scale consisting of “Not at all Apply,” “Not Apply,” “Not Somewhat Apply,” “Neither,” “Somewhat Apply,” “Apply,” and “Very Apply.” The scores for each rating scale were divided into 1-point increments, with 1 point representing “Not at all Apply” and 7 points representing “Very Apply.” Two words, “displeasure - pleasure” and “deactivation - activation” was evaluated using the Affect-Grid method. In this study, we used an affect grid of two dimensions, with the horizontal axis indicating “displeasure - pleasure” and the vertical axis indicating “deactivation - activation.” We asked respondents to respond on a nine-point scale from 1 to 9 for each dimension. The experimental screen is shown in Fig. 4.

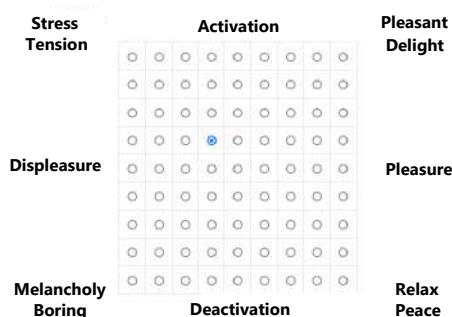
The above process was used to obtain the evaluation data of 44 evaluation words for 20 participants for each stimulus.

After that, the obtained evaluation data was cleaned for respondents who were dishonest or only completed the survey for a short amount of time. When assessing the stimuli, dummy images are prepared, and those who have not performed the evaluations instructed beforehand on dummy images are considered dishonest respondents. Among the dishonest respondents, the peak (mode) response time was calculated, and those whose response time was shorter than the peak were defined as the short time of the respondents. Due to the cleaning process, 90% of the 3,080 total respondents were valid.

Finally, a discrete probability distribution with the scored evaluation values as the random variable was used as the impression distribution of the suit pattern images. In this case, the impression distribution was normalized by the number of raters for each stimulus because the number differed between stimuli based on cleaning the evaluation data. The obtained impression distribution is converted to expected values and used as the teacher data for the impression estimation models to be constructed later.



(a) 7-point Likert scale.



(b) Affect-Grid.

Fig. 4. Experiment screen.

5 Modeling the Relationships Between Affective Texture and Physical Characteristics

5.1 Construction of Multi-scale CNN

In this study, we construct a multi-scale CNN based on the deep-learning model of Wetteland et al. [13]. As a detail of the model to be constructed, the input images are images of four resolutions: 72dpi, 112dpi, 224dpi, and 448dpi. We use the pre-trained VGG16 as a feature extractor. VGG16 is characterized by its emphasis on texture information [19, 20]. We use it as the basis for this model. Multi-scale CNN places four VGG16s in parallel to generate multiple feature vectors. These feature vectors are concatenated before being input to the FC layer. The FC layer has the same size as the original VGG16: 4,096 neurons.

Style features in Gatys et al.'s style transfer [11] are used as physical characteristics to represent the pattern image of the suit. In this case, style features are a Gram matrix of feature maps from the middle layer of CNN and have been extracted from the second pool layer of VGG16. However, the dataset is small and multi-scale CNN has many parameters, making them prone to overfitting during training. Therefore, a Dropout layer was added after each FC layer to prevent overfitting. Both Dropout rates were set to 0.5. Since VGG16 is a model that solves a classification problem, a regression problem replaces it when estimating visual impressions. Therefore, we focused on the activation functions in each layer of VGG16, replacing the ReLU function in the middle layer with the hyperbolic tangent function, and the SoftMax function in the output layer with the identity function. The VGG16 architecture handled by multi-scale CNN is shown in Fig. 5. The architecture of all four VGG16s is the same as in Fig. 5.

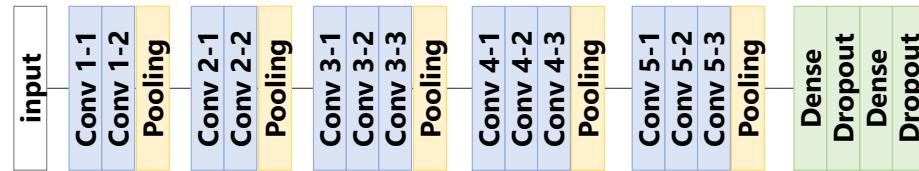


Fig. 5. VGG16 architecture.

5.2 Training

To estimate affective texture of suit patterns, we used the framework described in Section 5.1. In doing so, we solved a regression problem in which the expected value of the impression distribution (impression value) created in Section 4.2 was the objective variable, and the style features extracted from the suit pattern images were the explanatory variables. Multi-scale learning and evaluation were performed using impression values of 72 dpi, 112 dpi, 224 dpi, and 448 dpi images as ground truth for a dataset of 2,452 images. Cross-validation was used to train

and evaluate the models, and 10-fold cross-validation was used to divide the dataset into Train: Validation: Test = 9: 1: 1.

The Adam optimizer was used during training. We used Mean Squared Error (MSE) for the loss and evaluation functions. The learning rate was set to 0.001, the batch size to 64, and the number of epochs to 100. To avoid overfitting, we used Early Stopping, which terminates training when no accuracy improvement is expected. Learning is stopped when there has been no improvement in accuracy over the past ten epochs.

6 Results and Discussions

6.1 Accuracy comparison between Single-scale CNN and Multi-scale CNN

To validate the effectiveness of the proposed method, we compared the estimation accuracy of the model by 10-fold cross-validation between single-scale CNN and multi-scale CNN. We use the correlation coefficient between the impression value for the test data and the impression value estimated by the models as a measure of estimation accuracy. Table 2 shows the correlation coefficients for the impression values of 72 dpi, 112 dpi, 224 dpi, and 448 dpi images as ground truth for each evaluation term. Table 2 shows the correlation results for the single-scale CNN in the left column and the multi-scale CNN in the right column. As a result, moderate or higher positive correlations were found for most evaluation terms. This confirms that the suit pattern's style features have a high affinity with people's affective texture. There were 37 evaluation terms an exceptionally high correlation (0.4 or higher), with an average coefficient of 0.57. Among them, “deactivation - activation,” “vibrant,” “calm,” and “sober” showed high correlations at all resolutions. We believe that these evaluation words tend to have high correlations because they are low-order impressions associated with the color and characteristics of the pattern. The average correlation coefficients for all evaluation words were more accurate with the multi-scale CNN than with the single-scale CNN, confirming the effectiveness of the multi-scale CNN.

6.2 Human Visual angle and Resolution

We discuss visual processing and resolution when humans look at the pattern of a suit. The visual angle is the angle the object projected onto the eye makes and is the angle at which the object is viewed. In this study we compare the relationship between the range of resolution at which the human looks the suit patterns and the resolution in the constructed impression estimation models. The viewing angle is determined by the size of the visual object and the viewing distance when observing it and is calculated from Eq. (1).

$$v = 360/\pi \times \arctan(s/2d) \quad (1)$$

Where v is the viewing angle, s is the object's size, and d is the viewing distance. We then calculated the resolution at which a person sees the suit pattern from

Table 2. Correlation coefficients for each evaluation term.

(Left column: single-scale CNN, Right column: multi-scale CNN.)

evaluation word	72dpi		112dpi		224dpi		448dpi	
	single-scale	multi-scale	single-scale	multi-scale	single-scale	multi-scale	single-scale	multi-scale
deactivation - activation	0.79	0.76	0.71	0.69	0.63	0.77	0.70	0.68
displeasure - pleasure	0.30	0.40	0.19	0.25	0.39	0.37	0.30	0.43
dislike-like	0.59	0.63	0.42	0.44	0.34	0.41	0.41	0.47
bad-good	0.72	0.68	0.38	0.24	0.43	0.47	0.35	0.26
crisp	0.64	0.57	0.35	0.46	0.36	0.47	0.18	0.32
clear	0.62	0.70	0.30	0.49	0.57	0.61	0.31	0.43
plain	0.43	0.68	0.62	0.64	0.49	0.59	0.46	0.50
vibrant	0.76	0.80	0.66	0.74	0.68	0.71	0.77	0.66
sharp	0.54	0.52	0.29	0.53	0.25	0.41	0.15	0.27
simple	0.68	0.74	0.66	0.73	0.55	0.67	0.64	0.61
well-tailored	0.16	0.45	0.07	0.20	0.18	0.07	0.26	0.22
standard	0.65	0.78	0.60	0.68	0.63	0.66	0.52	0.43
refreshingly cool	0.51	0.63	0.49	0.63	0.36	0.53	0.48	0.55
warm	0.42	0.47	0.27	0.35	0.38	0.47	0.38	0.47
Massive	0.73	0.70	0.64	0.67	0.63	0.64	0.22	0.36
elegant	0.36	0.39	0.34	0.39	0.20	0.21	0.23	0.31
modern	0.16	0.23	0.22	0.30	0.27	0.23	0.29	0.13
classical	0.72	0.68	0.44	0.58	0.29	0.49	0.29	0.30
neat	0.78	0.75	0.67	0.70	0.63	0.69	0.58	0.60
supple	0.16	0.32	0.13	-0.06	0.38	0.29	0.39	0.40
squeezed	0.57	0.68	0.51	0.54	0.44	0.57	0.35	0.37
gorgeous	0.62	0.72	0.44	0.50	0.25	0.46	0.55	0.49
clean	0.40	0.55	0.37	0.45	0.34	0.40	0.30	0.45
light feeling	0.67	0.69	0.60	0.64	0.35	0.53	0.40	0.27
calm	0.78	0.82	0.67	0.76	0.71	0.76	0.71	0.70
formal	0.75	0.80	0.70	0.74	0.63	0.68	0.53	0.44
unique	0.67	0.75	0.68	0.70	0.59	0.67	0.75	0.69
0.74	0.79	0.70	0.71	0.64	0.69	0.64	0.57	
fashionable	0.51	0.50	0.41	0.42	0.33	0.45	0.46	0.43
sober	0.77	0.79	0.70	0.75	0.57	0.75	0.68	0.75
adult	0.67	0.67	0.59	0.65	0.45	0.68	0.27	0.54
youthful	0.59	0.58	0.43	0.52	0.32	0.53	0.27	0.45
dressy	0.54	0.61	0.49	0.51	0.39	0.38	0.48	0.42
trendy	0.29	0.45	0.49	0.48	0.22	0.39	0.10	0.31
powerful	0.68	0.70	0.55	0.60	0.49	0.58	0.15	0.50
sexy	0.61	0.59	0.39	0.53	0.22	0.46	0.04	0.36
kind	0.40	0.52	0.45	0.35	0.26	0.22	0.34	0.34
smart	0.39	0.58	0.31	0.50	0.50	0.55	0.30	0.29
prominent	0.69	0.63	0.38	0.48	0.48	0.56	0.24	0.34
stylish	0.49	0.52	0.21	0.33	0.48	0.39	-0.03	0.25
good impression	0.41	0.67	0.46	0.43	0.46	0.38	0.37	0.39
serious	0.74	0.77	0.66	0.74	0.63	0.68	0.67	0.66
cool	0.31	0.61	0.28	0.37	0.28	0.51	0.43	0.39
humility	0.73	0.79	0.76	0.78	0.64	0.74	0.69	0.67
average	0.56	0.63	0.47	0.53	0.44	0.52	0.40	0.44

the viewing distance and the minimum resolution of the human eye. Suppose a person's visual acuity is 1.0. In that case, the minimum resolution of the human eye is approximately 1/60 degree in terms of visual angle, and the resolution at which a human can look is calculated from Eq. (2).

$$dpi = \frac{2.54}{2 \times d \times \tan(v_1/2)} \quad (2)$$

Where d is the viewing distance and v_1 is the viewing angle (1/60 degree). In this study, we use Eq. (1) and Eq. (2) to compute the range of resolution over which a person sees a suit pattern. One obtains the finest resolution is when looking at the fabric up close. At this time, assuming the normally quoted viewing distance of 57cm, the resolution was about 153dpi, calculated from Eq. (2). Next, one obtains the coarsest resolution when looking at the entire suit jacket. Here, the effective viewing angle for a person is assumed to be 30 degrees, which is generally defined. The visual distance for observing the suit pattern was obtained from Eq. (1), used the actual size of the suit as 75 cm (average value). As a result, the viewing distance for the suit was 140 cm. Calculations using Eq. (2) based on the calculated viewing distance indicate a resolution of approximately 62dpi.

Substituting this into the constructed impression estimation models, we can expect the highest estimation accuracy when the impression value of the 72dpi or 112dpi images is used as ground truth. In this study, the actual size of the suit fabric was small, making it difficult to capture images at a resolution of 72dpi or lower. Table 2 shows the estimation accuracy at each resolution. The average correlation coefficient for all evaluation words is highest when 72dpi is used as ground truth, and accuracy tends to decrease as resolution increases. These results suggest that the relationship between the resolution at which one look at the suit patterns and the optimal resolution in the impression estimation models is close. This confirms that the relationship between the range of resolution at which people actually see the pattern of the suit and the resolution in the impression estimation model is a close result. Looking at the evaluation terms that differ in accuracy between 72dpi and 448dpi, "massive" and "light feeling" are listed. Both impressions expressed "weight-lightness," suggesting a more macroscopic observation of the pattern in these evaluation terms. Furthermore, it was suggested that the affective texture may change depending on the pattern's scale.

6.3 Confirmation of estimation results

Visually confirm the relationship between the estimation results by the constructed impression estimation models and the pattern image of the suit. We show the top and bottom two suit pattern images with the highest estimation accuracy (Fig. 6) for each resolution. The evaluation words for the estimation results are four impression words, "simple," "unique," "vibrant," and "calm," and two emotion words, "deactivation - activation" and "dislike-like. The models with the highest accuracy in cross-validation were used for estimation.

In the “simple” category, the top patterns primarily were solid black or gray and pinstripe patterns (thin stripes), while the bottom patterns were often loudly colored stripes and staggered plaid patterns. In the “unique” category, the top patterns were often loudly colored stripes and staggered plaid, while the bottom patterns were standard patterns such as plain black or gray and stripes. In the “vibrant” category, the top patterns were often brightly colored solid colors and stripes with blue as the base color, while the lower patterns were often plain colors and stripes of subdued colors. In the “calm” category, the top patterns for “calmness” were plain black or gray and shadow stripe patterns, while the bottom patterns were brightly colored stripes and staggered lattices. In the “deactivation - activation” category, most activation patterns are brightly colored stripes or staggered plaid, while deactivation patterns are calmly colored, such as plain gray or black. In the “dislike-like” category, standard patterns dominated responses, such as pinstripes and solid colors for the “like” response. In contrast, patterns with gaudy colors and geometric patterns were selected for the “dislike” response. These results confirm that the estimation results are close to the human image.

6.4 Relationship between resolution and pattern motifs

Based on the constructed impression estimation model, we discuss the relationship between resolution and pattern motifs. Using the actual impression values and the model-estimated impression values, we calculate the correlation coefficient for each pattern, and visually confirm the pattern with the highest correlation. We do so at each resolution. Fig. 7 shows some of the patterns with high correlation coefficients (estimation accuracy) at each resolution.

From Fig. 7, patterns with large motifs, such as alternate stripes and solid colors, are more common at 72 dpi. Alternate stripes are stripes with alternating lines of different colors and are believed to have a larger motif than a typical stripe pattern. At 112 dpi, there are medium-patterned stripes with medium-sized motifs, such as chalk stripes. On the other hand, patterns with large plain-like motifs are also mixed. Similarly, 224dpi has many medium-sized stripes with medium-sized motifs, such as pinstripes. At 448dpi, patterns with small motifs, such as staggered plaid and dot patterns, were observed. These results suggests that people change their resolution (viewing angle) according to the motif’s size and that the affective texture obtained from the pattern may depend on the motif’s size.

7 Conclusion

In this study, we proposed a method for estimating affective texture that the suit patterns evoke in response to differences in pattern scale and conducted the following procedure. (1) collect pattern images of suits at different resolutions, and (2) quantify the affective texture of 44 evaluation words through an impression evaluation experiment. (3) Style features extracted from an image are used as physical characteristics, and the relationship between affective texture and

evaluation word		72dpi	112dpi	224dpi	448dpi	
simple	higher					
	lower					
unique	higher					
	lower					
vibrant	higher					
	lower					
calm	higher					
	lower					
deactivation - activation	higher					
	lower					
dislike-like	higher					
	lower					

Fig. 6. Estimation results from impression estimation models.

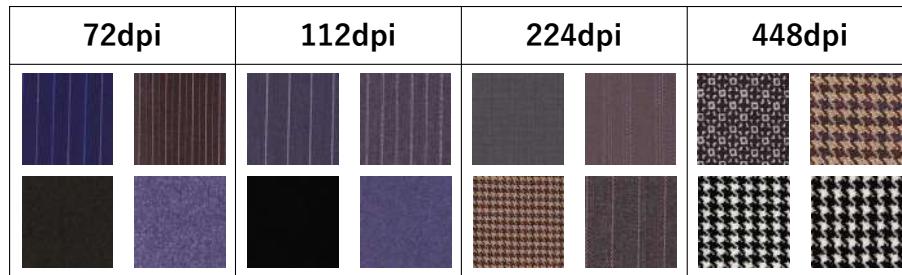


Fig. 7. Suit patterns with high estimation accuracy at each resolution

physical characteristics is modeled using a multi-scale CNN. A 10-fraction cross-validation was performed, and the correlation coefficients between the impression values of the test data and those estimated by the models were calculated. Results showed moderate or higher positive correlations for most evaluation terms. We found that multi-scale CNN improves the accuracy compared to single-scale CNN, confirming the effectiveness of this method. The resolution at which the suit patterns are viewed from the human visual angle was calculated and compared with the results of the constructed impression estimation models. They confirmed that the estimation accuracy was highest for images at 72dpi, close to the human resolution.

Future work, we expect to improve the estimation accuracy given the small dataset in this training by increasing the training data using GAN [18].

Acknowledgements This work was supported by JST COI Grant Number JPMJCE1314.

References

- Yamamoto, M., Onisawa, T.: Interactive Fashion Design and Coordinate System Considering User's KANSEI. *Transactions of Japan Society of Kansei Engineering* 15(1), (2016).
- Yamazaki, Y., Imura, M., Tobitani, K., Tani, Y., Nagata, N.: Development of measurement and simulation scheme for digitalization of tactile perception. In: Asia International Symposium on Mechatronics (AISM), pp. 981-986 (2019).
- Tobitani, K., Shiraiwa, A., Katahira, K., Nagata, N., Nikata, K., Arakawa, K.: Modeling of "high-class feeling" on a cosmetic package design. *Journal of the Japan Society of Precision Engineering* 87(1), 134–139 (2021).
- Tani, S., Matsunashi, K., Shimazaki, K.: A Study on the Configuration of Curtains (Part 5) —The Influence of Polka-dot Patterns on the Apparent Configuration of Curtains—. *Journal of the Japan Research Association for Textile End-Uses* 54(7), 646-655 (2013).
- Niwa, S., Aoyama, Y., Sudo, K., Taniguchi, Y., Kato, T.: Modeling relationship between visual impression of commodities and their graphical features. In: IPSJ SIG Technical Reports 2013-HCI-152, pp. 1–4 (2013).

6. Chen, Y.W., Huang, X., Chen, D., Han, X.H.: Generic and specific impressions estimation and their application to KANSEI based clothing fabric image retrieval. *J. Pattern Recognit. and Artif. Intell.* 32(10), 1854024 (2018).
7. Tani, Y., Nagai, T., Koida, K., Kitazaki, M., Nakauchi, S.: Experts and novices use the same factors-but differently-to evaluate pearl quality. *PLOS ONE* 9(1), 1-7 (2014).
8. Tobitani, K., Matsumoto, T., Tani, Y., Fujii, H., Nagata, N.: Modeling of the relation between impression and physical characteristics on representation of skin surface quality. *The Journal of The Institute of Image Information and Television Engineers* 71(11), 259-268 (2017).
9. Doizaki, R., Iiba, Saki., Okatani, T., Sakamoto, M.: Possibility to Use Product Image and Review Text Based on the Association between Onomatopoeia and Texture. *Transactions of the Japanese Society for Artificial Intelligence : AI*, 30(1), 124-137 (2015).
10. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: *The 3rd International Conference on Learning Representations (ICLR)*, pp. 1-14 (2015).
11. Gatys, L. A., Ecker, A. S., Bethge, M.: Image style transfer using convolutional neural networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2414-2423 (2016).
12. Gao, F., Li, Z., Yu, Jun., Yu, Junze., Huang, Q., Tian, Q.: Style-adaptive photo aesthetic rating via convolutional neural networks and multi-task learning. *Neurocomputing* 395, 247-254 (2020).
13. Sunda, N., Tobitani, K., Tani, I., Tani, Y., Nagata, N., Morita, N.: Impression estimation model for clothing patterns using neural style features. In: Stephanidis, C., Antona, M. (eds.) *HCI International 2020 - Posters*. HCII 2020, Communications in Computer and Information Science, vol. 1226, pp. 689-697. Springer, Cham (2020).
14. Wetteland, R., Engan, K., Eftestøl, T., Kvikstad, V., Emiel A., Janssen, M.: A Multiscale Approach for Whole-Slide Image Segmentation of five Tissue Classes in Urothelial Carcinoma Slides. *Technology in Cancer Research & Treatment* 19 (2020).
15. Tada, M., Kato, T.: Similarity Image Retrieval System Using Step-by-Step Hierarchical Classification. *IPSJ Transactions on Databases (TOD)* 44(8), 37-45 (2003).
16. Reinhard, E., et al.: Photographic tone reproduction for digital images. *ACM Transactions on Graphics* 21(3), 267-276 (2002).
17. Miyai, S., Katahira, K., Sugimoto, M., Nagata, N., Nikata, K., Kawasaki, K.: Hierarchical structuring of the impressions of 3D shapes targeting for art and non-art university students. In: Stephanidis, C., (ed.) *HCI International 2019 - Posters*. HCII 2019, Communications in Computer and Information Science, vol. 1032, pp. 385-393. Springer, Cham (2019).
18. Russell, W., Mendelsohn, G.A.: Affect-Grid: A Single-Item Scale of Pleasure and Arousal. *Journal of Personality and Social Psychology* 57(3), 493-502 (1989).
19. Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., Brendel, W.: ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231* (2018).
20. Tuli, S., Dasgupta, I., Grant, E., Griffiths, T. L.: Are convolutional neural networks or transformers more like human vision?. *arXiv preprint arXiv:2105.07197* (2021).
21. Tsumura, E., Tani, I., Tobitani, K., Nagata, N.: Textile-GAN: Generation of Texture for Woven Pattern Using Generative Adversarial Networks. In: *The Institute of Electronics, Information and Communication Engineers (IEICE)*, pp.19-20 (2021).

A Cascaded Structure of pre-trained Convolutional Neural Network for Weed Classification

GwangHyun Yu¹, Dang Thanh Vu¹, JaeCheol Jeong², ChilWoo Lee³ and JinYoung Kim¹

¹ Department of ICT Convergence System Engineering, Chonnam National University, 77, Yongbong-ro, Buk-gu, Gwangju 61186, Korea

² Department of Biomedical Engineering, Chonnam National University Hospital, Korea

³ Department of Computer Information and Communication, Chonnam National University, 77, Yongbong-ro, Buk-gu, Gwangju 500757, Korea

Abstract. This study proposes a cascaded structure of a pre-trained convolutional neural network (CNN) for the classification of weeds. The method, known as transfer learning, involves dividing pre-trained CNN models that have been successful in image classification into models that classify families and species of plants. To evaluate this technique, the authors collected 60,506 datasets for 40 species from 5 selected families of exotic weeds and used domestic plant and weed experts to classify them. In their cascaded structure experiment, they then used DenseNet and EfficientNet models, which had previously shown promising results in single CNN transfer learning experiments. The results showed that the DenseNet-based cascaded structure model had 96.057% accuracy in classifying exotic weeds, which was 0.671% higher than the DenseNet transfer learning model and also reduced the model size by 18.7%. Similarly, the EfficientNet-based cascaded structure model had 96.24% accuracy and reduced model size by 18.3%. The authors suggest that this cascaded structure method can be effective for hierarchical datasets such as exotic weeds.

Keywords: Image Classification, Deep Learning, Lightweight model.

1 Introduction

Exotic weeds refer to plants that have intentionally or unintentionally left their place of origin, spread to other lands, and settled there, mainly due to the movement of humans, animals, and means of transportation. Recently, interest in the ecological characteristics and management methods of foreign weeds has increased as the spread of foreign plants and their impact on the ecosystem has become serious. Compared to other native weeds, foreign weeds expand their habitat faster and have the characteristics of adapting well to very diverse environmental conditions. Foreign weeds invading a certain area inevitably lead to changes in the entire ecosystem, mainly in the structure of vegetation communities. It causes changes in flora, changes in flora, etc., and ultimately leads to a

This research was supported by the BK21 FOUR Program(Fostering Outstanding Universities for Research, 5199991714138) funded by the Ministry of Education(MOE, Korea) and National Research Foundation of Korea(NRF)

decrease in species diversity. Foreign weeds through various routes, foreign weeds enter agricultural lands such as rice fields, fields, and orchards; they rapidly adapt to the agricultural land environment, which is rich in nutrients and richer in nutrients than the original ecosystem, and is less competitive, and reproduce in large quantities to fight native weeds through various routes. It has a superior position compared to [1].

There are many cases in which there is no special method for controlling foreign weeds, which cause more damage than native weeds, and in some cases, the situation is aggravated by incorrect control. In the case of weeds that grow naturally in Korea, it is possible to identify them accurately, and commercially available herbicides can be sufficiently effective, but in the case of foreign weeds, it is difficult to identify them accurately. There is no clear effect after treatment [2]. Therefore, accurately identifying foreign weeds is crucial to present accurate control methods. To this end, this paper proposes a CNN cascaded structure for classifying foreign weeds in a hierarchical structure through an effective pre-trained CNN model.

2 Related works

This study focuses on utilizing deep learning techniques to classify weeds. The aim is to extract plant features using convolutional neural networks (CNN) and develop a smartphone application system [3]. The study utilized a technique of inputting overlapping patches without image segmentation. Previous research [4, 5] in this field has yielded high accuracy rates, such as 93.8% using a deep learning-based weed classification model and 86.2% using a ResNet model with 10,413 plants of 22 species. However, these models cannot incorporate actual weed data or a comprehensive array of weed species present in cultivated fields. This study aims to address these limitations by utilizing a different approach for weed classification.

Previous studies have utilized convolutional neural network (CNN) model ensemble methods for weed classification, such as AgroAVNET, a CNN model that combines the strengths of AlexNet and VGG. This model achieved 93.64% accuracy in a study that classified 4,200 plants of 12 species [6]. Another study using five foreign weed datasets from Chonnam National University achieved a maximum accuracy of 98.77% through a Late Fusion method, which ensembles up to 5 models for 21 species [7]. However, it is noted that the use of multiple deep learning models may require a large number of parameters and significant training time.

Previous studies have employed a Hierarchical Approach with a Convolutional Neural Network (CNN) to classify foreign weeds [8]. This approach utilizes a cascaded structure of a recently advanced pre-trained CNN model, which enables the classification of foreign weeds with improved accuracy and fewer parameters. This approach also suggests methods to decrease the learning time.

3 Pre-trained CNN cascaded structure for exotic weed classification

3.1 CNN cascaded structure

This paper proposes a CNN cascaded structure model for classifying foreign weeds. This model employs a hierarchical approach that classifies data based on plant taxonomic characteristics, such as families and species. The proposed model has a cascaded structure, which includes a CNN model for classifying families and separate CNN models for classifying species. This approach allows for more accurate classification using a smaller number of parameters than a single CNN model. This is achieved by tailoring small CNN models to specific families and species rather than using a single, general model.

The image recognition problem can be mathematically viewed as conditional Bayesian Probability, where an input (a given foreign weed data image or a feature vector obtained from foreign weed data) X_i , a correct answer class Y_i , and a correct answer F_i be a family, $i = 1, \dots, N$ denotes a training data sample. All correct classes are composed of one-hot encoding vectors, $Y_i \in \{0, 1\}^M$ and $F_i \in \{0, 1\}^K$ is vector of correct answer classes and families, respectively. With this basic notation, we can define three kinds of learning models:

$$P_w(\hat{Y} | X) \quad (1)$$

$$P_w(\hat{Y}, \hat{F} | X) \quad (2)$$

$$P_w(\hat{Y} | \hat{F}, X) \quad (3)$$

The learning model of equation (1) is a conventional classifier that predicts a class, the learning model of equation (2) is a global classifier that predicts a family and a class by joint probability, and equation (3) is a cascaded local classifier that predicts classes conditionally on the family. In the present study, we employ a learning model in which the weights are updated by a training algorithm such as back-propagation. Given an input feature vector, the model predicts a class and the family of the given input. The objective function is the categorical cross-entropy for both maximizing the likelihood of family and class prediction.

As illustrated in Figure 1, the proposed method for classifying foreign weed species employs a pre-trained convolutional neural network (CNN) model that first categorizes the samples into five families and then subsequently classifies them into 40 species using the results of the initial family classification. The architecture of this cascaded CNN model is designed to improve the classification performance by reducing the number of species to be classified. Specifically, it is demonstrated that by utilizing a cascaded pre-trained CNN structure that classifies an average of 8 classes, superior classification results are obtained compared to classifying all 40 species with a single pre-trained CNN model. Furthermore, it is worth noting that in previous studies, constructing a CNN cascaded structure required the design of models optimized for each family and species, which resulted in varying classification performance depending on the

researcher's knowledge and experience. However, in this study, a pre-trained CNN model is utilized, enabling the selection of an appropriate model regardless of the researcher's expertise, leading to higher classification accuracy with fewer parameters and less training time.

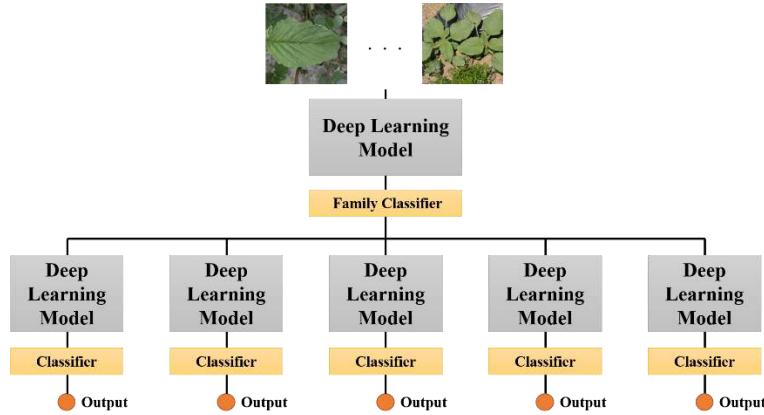


Fig. 1. Simple CNN cascaded structure.

3.2 Chonnam National University exotic weed dataset



Fig. 2. Samples after cropping by parts using a cropping tool.

Table 1. 5 families and 40 classes of original datasets

Dataset	Training	Validation	Testing	Total
Amount	50,802	16,940	16,958	84,700

The Chonnam National University foreign weed dataset was developed by a collaborative effort between Chonnam National University located in Gwangju, Korea, and six institutions supervised by the Rural Development Administration, namely the National Institute of Agricultural Sciences, Chungnam National University, Shinyeong

University, Hankyong University, and Sungkyunkwan University. The dataset comprises a total of 40 species, including seven species of Amaraceae from the National Institute of Crop Science, six species of Ginsengaceae from Chungnam National University, seven species of Meongajuaceae from Shinyeong University, six species of Convolvulaceae from Hankyong University, and 14 species of Asteraceae from Sungkyunkwan University. The dataset was collected by photographing foreign weeds with a smartphone camera and a high-resolution digital camera while traveling all over the country, and the final data was established through inspection of the collected dataset in collaboration with the National Institute of Crop Science.

The Chonnam National University foreign weed dataset, as depicted in Fig. 2, has been developed to identify the characteristics of foreign weeds. The dataset was compiled by extracting samples of leaves, flowers, fruits, and outposts, and the composition of the entire dataset is presented in Table 1.

4 Experiments and Results

In this experiment, the deep learning model was implemented with PyTorch version 1.12.0 of the Python language, and the experiment was performed using Intel(R) Core(TM) i9-7900X CPU 3.30GHz CPU, 128GByte DDR4 RAM, NVIDIA TITAN V with 12GByte built-in RAM.

4.1 CNN Transfer Learning for exotic weed classification

To apply a single CNN model to the proposed cascaded pre-trained CNN structure for foreign weed classification, a selection of pre-trained CNN models was evaluated, including AlexNet, VGGNet, GoogleNet, ResNet, DenseNet, MobileNetV2, SqueezeNet, ShuffleNet2, and EfficientNet, all of which were trained on the ImageNet dataset. These pre-trained CNN models were then evaluated through transfer learning using the 40 foreign weed datasets from Chonnam National University.

Table 2. Performance and model complexity of the pre-trained single model

pretraining model	Model Size (MB)	classification accuracy
AlexNet	57.2	0.86720
VGG	128	0.88303
GoogLeNet	5.6	0.92003
ResNet	11.2	0.92141
DenseNet	7.0	0.94432
MobileNetV2	2.3	0.92555
SqueezeNet	0.743	0.84935
ShuffleNetV2	1.3	0.88966
EfficientNet	20.2	0.95711

Table 2 presents transfer learning results using a single pre-trained CNN model for foreign weed classification. It is observed that the EfficientNet and DenseNet models,

which were pre-trained with the ImageNet dataset, demonstrated superior classification performance with an accuracy of 95.71% and 94.43%, respectively. The hyperparameters utilized for transfer learning in these two CNN models include a batch size of 64, an Adam optimizer, a learning rate of 0.0001, and 20 training epochs.

4.2 Pre-trained CNN cascaded structure for exotic weed classification

In light of the superior performance of EfficientNet and DenseNet models in foreign weed classification, as determined through transfer learning in a single CNN model, an experiment was conducted to evaluate the effectiveness of a cascaded pre-trained CNN structure for foreign weed classification. The proposed architecture is designed to prioritize ease of model design by utilizing the largest and smallest pre-trained CNN models instead of custom-designing CNN model structures for each family and species.

Table 3. Performance of DenseNet cascaded structure.

	large model	small model	classification	Asteraceae	bind-weed	Pig-weed	Amaranthaceae	Ginsengaceae
Model Size	18.2	7	7	7	7	7	7	7
						42.0		
accuracy	0.9538	0.9443	0.9795	0.9631	0.9349	0.9691	0.9097	0.98
						0.9605		
batch size	32	64	64	64	64	64	64	64
learning time (s)	12				06:04			

As shown in Table 3, a comparison experiment was conducted between DenseNet201, with a model size of 18.2MB, and DenseNet121, with a model size of 7MB, and it can be observed that the size difference between the single largest transfer-learned CNN model and the single smallest transfer-learned CNN model is approximately threefold or more. The results demonstrate that the foreign weed classification accuracy is 95.386% for the single largest transfer-learned model and 96.057% for the CNN cascaded structure model. Furthermore, the CNN cascaded structure, which learns by dividing several small models into family and species models, is more suitable for a parallel server architecture. Additionally, it can be noted that the learning time is reduced by about two times, even when learning sequentially on a single server, at 6 minutes and 4 seconds.

Table 4 illustrates the results of a comparative experiment conducted using EfficientNet_b5, with a model size of 28.4MB, and EfficientNet_b0, with a model size of 4MB, for the classification of foreign weeds. The classification accuracy for the single giant transfer learning CNN model was 94.617%, and for the CNN model was 94.617% while reducing the model size by 18.3%. The results confirm that the cascaded structure model demonstrated superior performance, with an accuracy of 96.24%. Furthermore, due to the ability of the CNN cascaded structure to learn by dividing several small models into family and species models, it is well-suited for parallel server architecture. Even when learned sequentially on a single server, it resulted in a reduction of learning

time by a factor of approximately 3, and on a parallel server architecture, it is possible to train six small models simultaneously, resulting in a reduction of learning time by a factor of approximately 20.

Table 4. Performance of EfficientNet cascaded structure.

	large model	small model	classification	Asteraceae	bind-weed	Pig-weed	Amaranthaceae	Ginsengaceae
Model Size	28.4	4	4	4	4	4	4	4
						24		
accuracy	0.9461	0.9523	0.9936	0.9859	0.9791	0.9943	0.9710	0.9961
						0.9624		
batch size	4	64	64	64	64	64	64	64
learning time (s)	59				03:34			

Table 5. Comparison with the previous study

	Model Size (MB)	Accuracy (ACC)
ResNet-based CNN cascaded structure [8]	10.4	0.9561
Single DenseNet	18.2	0.9443
Single EfficientNet	28.4	0.9461
DenseNet-based CNN cascaded structure	42(7)	0.9538
EfficientNet-based CNN cascaded structure	24(4)	0.9624

The results of this study have confirmed that the proposed pre-trained CNN cascaded structure for foreign weed classification can achieve higher classification accuracy with fewer parameters than a single large transfer-learned CNN model. Additionally, as demonstrated in the ResNet-based CNN cascaded structure study presented in Table 5, when the CNN cascaded structure is tailored to the foreign weed dataset and based on the researcher's experience and knowledge, it is possible to achieve a reduction in overall CNN cascaded structure model size while simultaneously improving classification accuracy when compared to a single large CNN model. Furthermore, it was found that not only is it possible to achieve a reduction in model size but also high accuracy can be obtained if the researcher constructs the CNN cascaded structure in the order of the transfer learning results of the pre-trained CNN model.

5 Conclusions

In this study, we investigate using a pre-trained convolutional neural network (CNN) cascaded structure for classifying foreign weeds. To this end, a dataset was constructed for 40 species of foreign weeds identified as ecosystem disruptors and selected in consultation with domestic plant and weed experts. A comparative experiment was conducted with EfficientNet_b5, a model size of 28.4MB, and EfficientNet_b0, a model size of 4MB. The results showed that the accuracy of the classification of foreign weeds was reduced by 18.3% for the single giant transfer learning CNN model, with an

accuracy of 94.617%. However, the CNN cascaded structure model demonstrated superior performance, with an accuracy of 96.24%. Additionally, the deep learning CNN cascaded structure is well-suited for parallel server architecture, as it learns by dividing several small models into family and species models. This not only results in a reduction of learning time by approximately 3 when learned sequentially on a single server but also by approximately 20 when trained on a parallel server architecture.

References

1. Saber, M., Lee, W.S., Burks, T.F., Schueller, J.K., Chase, C.A., MacDonald, G.E., Salvador, G.A., "Performance and Evaluation of Intra-Row Weeder Ultrasonic Plant Detection System and Pinch-Roller Weeding Mechanism for Vegetable Crops", Proceedings of International Meeting. American Society of Agricultural and Biological Engineers, pp. 1, 2015
2. K. Thorp and L. Tian., "A review on remote sensing of weeds in agriculture", Precision Agriculture, Vol. 5, No. 5, pp. 477–508, Oct, 2004.
3. N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J. Kress, I. C. Lopez, and J. V. Soares. Leafsnap: A computer vision system for automatic plant species identification. In Computer Vision–ECCV 2012, pages 502–516. Springer, 2012.
4. HAUG, Sebastian, et al. Plant classification system for crop/weed discrimination without segmentation. In: IEEE winter conference on applications of computer vision. IEEE, 2014. p. 1142-1149.
5. M. Dyrmann, H. Karstoft and H. S. Midtiby, "Plant species classification using deep convolutional neural network," Biosystems Engineering, vol. 151, pp. 72-80, 2016.
6. Shah MP, Singha S, Awate SP. Leaf classification using marginalized shape context and shape+ texture dual-path deep convolutional neural network. In2017 IEEE International Conference on Image Processing (ICIP) 2017 Sep 17 (pp. 860-864). IEEE.
7. Trong, V. H., Gwang-hyun, Y., Vu, D. T., & Jin-young, K. (2020). Late fusion of multi-modal deep neural networks for weeds classification. Computers and Electronics in Agriculture, 175, 105506.
8. Yu, G., Lee, J., Trong, V. H., Vu, D. T., Nguyen, H. T., Lee, J., ... & Kim, J. Exotic Weeds Classification : Hierarchical Approach with Convolutional Neural Network. The Journal of Korean Institute of Information Technology, 17(12), 81-92.

This paper is an extended version of the paper “Exotic Weeds Classification: Hierarchical Approach with Convolutional Neural Network” which has been published in “The Journal of Korean Institute of Information Technology.”

Two-stream Network for Moving Object Detection

Dhammadorn Wisan, Naoshi Kaneko, Seiya Ito, and Kazuhiko Sumi

Aoyama Gakuin University, Japan

Abstract. Object detection is one of the fundamental challenges in computer vision. In the real world, however, problems such as occlusion by other objects, background or foreground variations, and low contrast of the target object arise. Attempting to detect the targets with low similarity to the object model increases the probability of false detection. If the target appears in front of the background, the detection threshold of the target can be lowered in the foreground region while keeping the probability of false detection low. Using this idea, we propose a new method that, in addition to frame-by-frame object detection, can detect missed target objects by performing object classification on newly appearing objects in regions where no objects are detected. In our proposed network, the first stream is standard object detection and the second stream performs background subtraction for non-object regions detected by the first stream. Then the second stream performs object classification for the detected foreground regions. Finally, those two streams are merged and object class and regions are output. We applied this method to the detection of vehicles on the road and were able to detect even low-contrast vehicles that could not be detected by frame-by-frame detection.

Keywords: Object detection · Image segmentation · Vehicle detection · Background subtraction · deep neural network.

1 Introduction

Object detection is a fundamental problem in computer vision tasks. In real environments, even state-of-the-art object detection methods suffer from a variety of conditions such as low contrast, occlusion, similar backgrounds, and uneven illumination. Especially in vehicle detection on highways, physical and economic reasons limit the conditions under which cameras can be installed. As a result, the following difficulties arise. Vehicles are hidden by other vehicles, vehicles have low contrast against the background, visibility is low, and the size of the vehicle in the image varies with distance from the camera. Until the mid-2000s, background subtraction was the primary method used for vehicle detection. However, background subtraction does not work well when the background changes significantly. Subsequently, object modeling such as HOG [2] and SIFT [9] features, which are robust to background changes, have been preferred

over background subtraction methods. In particular, object modeling using neural networks has become popular in recent years. However, state-of-the-art object detection methods are unable to detect low-contrast objects, which are often found in traffic scenes. Therefore, we propose to improve the detection performance of low-contrast objects by combining a state-of-the-art object detection network with a background subtraction method. In our proposal, objects are detected in two streams. The first stream performs frame-by-frame object detection. The output of the first stream includes an object mask. In the second stream, background subtraction is performed. The foreground region is then masked by the object mask of the first stream. The image then contains regions where some moving object that was not detected in the first stream appears. These regions are segmented and object classification is applied to each region. This classification lowers the detection threshold while keeping the false positive rate low.

In summary, our contributions in this work are:

1. A method to separate areas for object detection and classification later.
We modified an existing method [5] for multi-object semantic segmentation that is robust to low contrast image. The segmented area will be used later to compare undetected foreground from next Step.
2. A method for object detection and segmentation.
In this work, we employ state-of-the-art object detection to generate reliable object bounding boxes. We modified some methods of YOLOv7 [10] and combined them with the object classification method to improve model efficiency under low contrast.
3. A method for object classification.
We apply a pre-trained a model for object classification using GluonCV trained with the CIFAR-10 dataset. GluonCV [14] is a toolkit which provides various implementations of the state-of-the-art based on using deep learning models in computer vision.

2 Related Work

Before performing object detection and segmentation, one of the most challenging tasks is to segment foreground and background from video sequences. Although many methods have now been proposed that do not require background subtraction methods in object detection and tracking, this pre-processing step is still important under heavy occlusion cases. Currently, applying deep convolutional neural networks (CNNs) shows apparent achievements in the field of computer vision. Its outstanding performance provides an efficient ability to extract object features and object masks that can be used in multiple related tasks including our task in vehicle detection. We describe some details of each related component in the following sections.

2.1 Object Detection and segmentation

Object detection is one of the classic and interesting challenges in the field of computer vision. It is divided into two methods, traditional neural network methods, and recent complex deep learning methods. Generally, object detection methods can be divided into two categories, One-stage detection and Two-stage detection. You Only Look Once (YOLO) [7] and Single Shot Multibox Detector(SSD) [12] are well-known in one-stage methods. Faster R-CNN [8] and Mask R-CNN [3] are frequently referred to two-stage methods. However, when the some object parts are occluded by other obstructions, these methods are not able to detect some of those objects. A number of approaches have recently been proposed to address occlusion problem. Yuan et al. [13] presented a method for multi-object instance segmentation which can locate occluders and classify objects based on non-occluded parts in order to detect an object that is occluded by other objects.

2.2 Background Subtraction

So far, background subtraction is a significant processing part of object detection and object tracking performed with video sequences. In the past, the most widely known method was a classical background subtraction method that used a previous frame or a static image to be a background for performing subtraction. This technique can be used to a certain degree, but it is very sensitive to various conditions, such as dynamic changes, noise, illumination, and so on. Due to those issues, many approaches have been proposed to accurately segment between foreground object, and background. At present, the use of deep learning shows amazing success and have been approved its efficiency by many researchers. Long and Hacer [4] proposed a method that creates a mask to separate the object from the background from an input image trained from a video sequence. The advantage of this method is that it doesn't need to use a lot of data for training, but it provides good results. However, this method still has a limitation: the input image that will be tested for segmentation must be an image with the same background as the data used.

3 Methodology

In this section, we introduce our main structure for object detection using vehicle model under various conditions in traffic videos. An algorithm is explained by following:

- Step 1: We first obtained the image input by an RGB camera and defined it as I^{xy} when the (x, y) is coordinate.
- Step 2: Masks of sub ROI [5] areas are extracted from background subtraction approach based on deep learning technique in order to classify undetected objects later. Foreground F can be obtained after performing an

input image I processed by GluonCV:

$$F(x, y) = \begin{cases} 1, & \text{if } I(x, y) \text{ is foreground} \\ 0, & \text{if } I(x, y) \text{ is background} \end{cases} \quad (1)$$

Step 3: The YOLOv7 [10] with pre-trained weight is used to detect the vehicles of input image.

Step 4: Semantic segmentation of each vehicle area is applied to create masks for comparing with masks extracted from previous background subtraction in Step 2. Foreground G can be obtained after performing an input image I processed by semantic segmentation:

$$G(x, y) = \begin{cases} 1, & \text{if } I(x, y) \text{ is segmented area} \\ 0, & \text{if } I(x, y) \text{ is unsegmented area} \end{cases} \quad (2)$$

Step 5: We compare different regions, between the mask from Step 2 and the mask from Step 4 then the objects in different regions are classified as to whether they become a vehicle category. The salient is undetected area and need to be classified the category which can be explained by:

$$S(x, y) = F(x, y) - G(x, y) \quad (3)$$

where S is area for classification.

Step 6: S is combined with RGB image input. Then, to classify whether the object in the area S is a vehicle, we used a pre-trained model named CIFAR_ResNet110_v1 of image classification network from the GluonCV [14] framework.

To summarize all methods, a flowchart of the proposed framework is shown in Fig. 1.

4 Experiment and Discussion

In this section, we evaluate the effectiveness of our proposed method by using two different datasets, CDnet2014 [11] and a custom dataset constructed from the CARLA simulator¹. Each of the datasets contains challenging conditions such as dynamic background movement, shadow, noise, and so on to evaluate our method. An example of output result is shown in Fig. 2.

4.1 Vehicle Dataset

Video sequences from the CDnet2014 dataset [11] are used for training with spatial resolution 320×240 that contain various scenarios such as dynamic background motion, illumination changes, multiple objects, contrast, etc. Moreover,

¹ <https://carla.org>

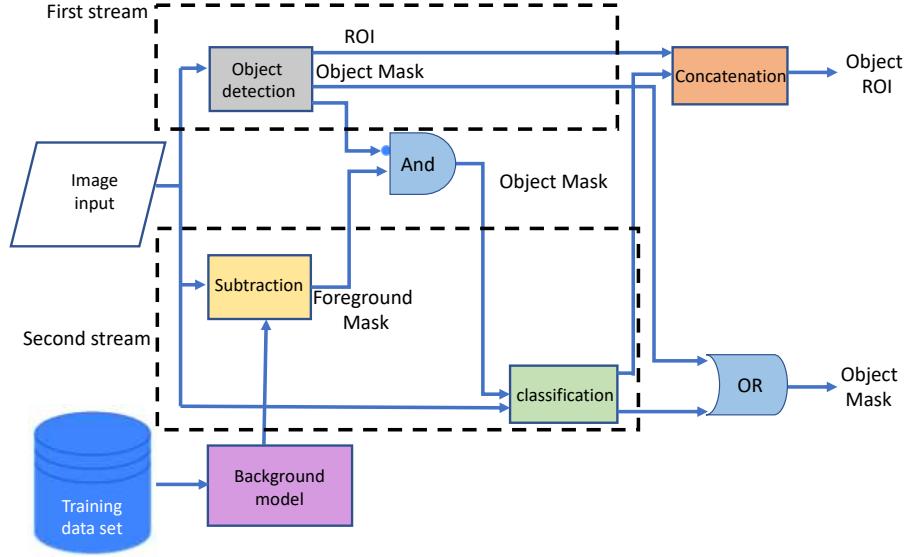


Fig. 1. Overall flow of the proposed method.

in order to increase the flexibility of the datasets and various aspect adjustments not available in the downloaded datasets, the CARLA simulator is applied to construct datasets under various challenging conditions. The CARLA is an open-source simulator widely used for urban driving research.

4.2 Evaluation Measure

To ensure that our proposed framework can be certainly used in various conditions, we have trained a model using different image sizes of each dataset. However, for a fair comparison, test images used for assessment of the performance of the system are performed with the same object detection and classification methods.

We used F-Measure, precision, and recall value to assess the performance of our model. The formulas of accuracy, precision, and recall we use to evaluate and describe the quality of our model are as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (4)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (5)$$

$$\text{F-Measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (6)$$

where TP, FP, and FN mean true positive, false positive, and false negative, respectively. In our experiment, we randomly select images for testing from CDnet2014 20 images and randomly select from our custom datasets 10 images. The result of experiment is shown in Table 1.

Table 1. Accuracy of the network model.

Dataset	Precision	Recall	F-Measure
CDnet2014	0.9917	0.9523	0.9716
Our test set	1.0000	0.7586	0.8627

5 Conclusion and Future Work

In this paper, we presented an efficient method for object detection using vehicle model under several conditions from the perspective of surveillance cameras for various surveillance video scenes with the pipeline shown in Fig. 1. First, image or video sequence is used as input of system. Next, vehicle detection is performed by YOLOv7 with pre-trained weight. To address the problem of the small or occluded vehicles, which cannot be detected. A more effective sub-area was obtained by extracting mask of foreground segmentation of each vehicle shape from road surface areas. Then, each object in sub-ROI areas of each frame detected to obtain better vehicle detection results from object classification method. Finally, bounding boxes are constructed in case object's category is classified as vehicle. If not, nothing performed. In addition, we also conducted some pre-processing and post-processing of traditional methods to enhance the performance and results of our framework further. Our proposed method clearly shows better results in vehicle detection compared with traditional methods. For future work, key-point detection and tracking from the optical flow method can be aggregated in the pipeline for improving efficiency of framework.

Acknowledgements

Part of this research was operated as the project of Center for Advanced Information technology Research (CAIR), Aoyama Gakuin University.

References

1. Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple online and realtime tracking. In: 2016 IEEE International Conference on Image Processing (ICIP), pp. 3464–3468 (2016). <https://doi.org/10.1109/ICIP.2016.7533003>

2. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) **1**, 886–893 (2005)
3. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. arXiv (2017). <https://doi.org/10.48550/ARXIV.1703.06870>
4. Long, A. L., Hacer, Y. K.: Foreground segmentation using convolutional neural networks for multiscale feature encoding, vol. 112, pp. 256–262. Elsevier (2018). <https://doi.org/10.1016/j.patrec.2018.08.002>
5. Long, A. L., Hacer Y. K.: Learning multi-scale features for foreground segmentation. Pattern Analysis and Applications, vol. 23, pp. 1369–1380. Springer Science and Business Media (2019). <https://doi.org/10.1007/s10044-019-00845-9>
6. Nicolai, W., Alex B., Dietrich P.: Simple Online and Realtime Tracking with a Deep Association Metric. arXiv (2017) <https://doi.org/10.48550/ARXIV.1703.07402>
7. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You Only Look Once: Unified, Real-Time Object Detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788 (2016)
8. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. arXiv (2015). <https://doi.org/10.48550/ARXIV.1506.01497>
9. Lindeberg, T.: Scale Invariant Feature Transform. Scholarpedia **7**, 10491 (2012)
10. Wang, C., Bochkovskiy, A., Liao, H.: YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv (2022) <https://doi.org/10.48550/ARXIV.2207.02696>
11. Wang, Y., Jodoin, P., Porikli, F., Konrad, J., Benezeth, Y., Ishwar, P.: CDnet 2014: An Expanded Change Detection Benchmark Dataset. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 393–400 (2014). <https://doi.org/10.1109/CVPRW.2014.126>
12. Wei, L., Dragomir, A., Dumitru, E., Christian, S., Scott, R., Cheng, Y., Alexander, C. B.: SSD: Single Shot MultiBox Detector: Computer Vision – ECCV 2016, pp. 21–37. Springer International Publishing (2016)
13. Xiaoding, Y., Adam, K., Yihong, S., Alan, Y.: Robust Instance Segmentation through Reasoning about Multi-Object Occlusion. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11136–11145 (2021)
14. GluonCV Homepage, <http://cv.gluon.ai/>. Last accessed 16 Jan 2023

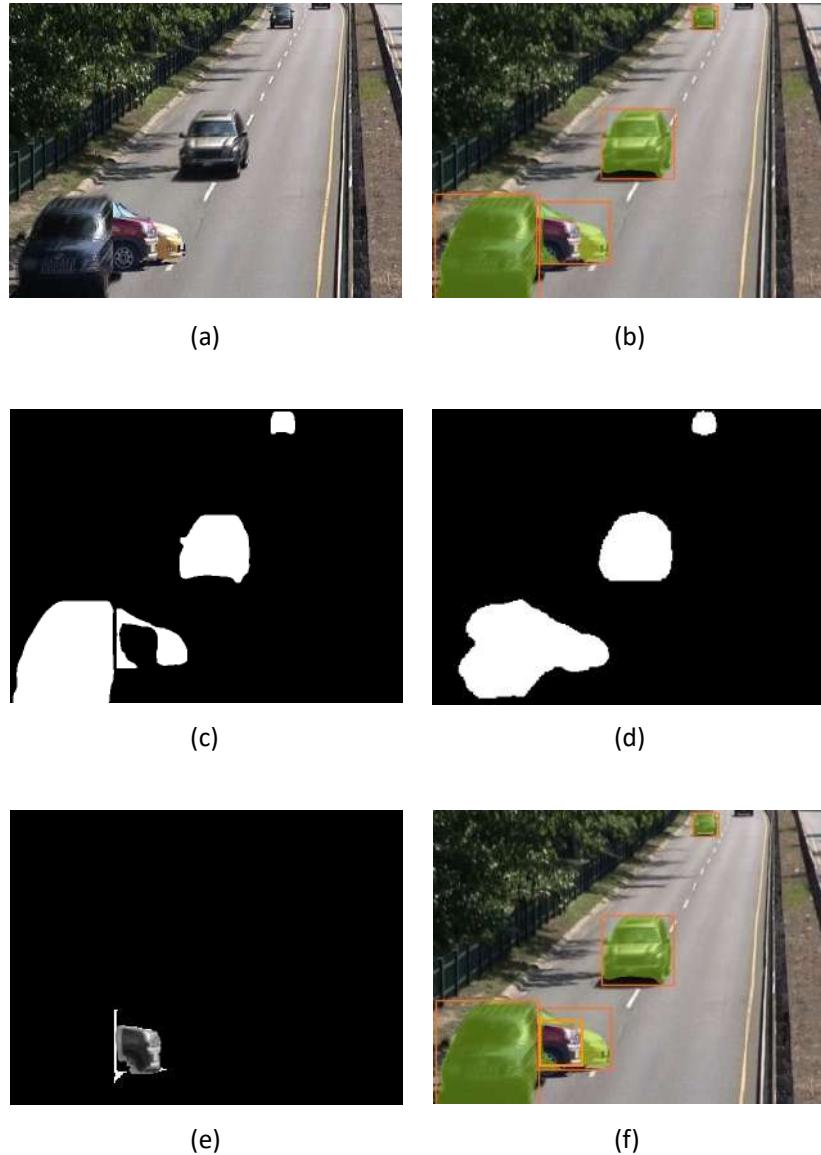


Fig. 2. Single-frame video object detection result. (a) RGB image input; (b) object detection and segmentation from YOLOv7; (c) mask segmentation from YOLOv7; (d) mask segmentation from [4]; (e) different region between (c) and (d); (f) result of full image detection method.

Multimodal Transformer for Automatic Depression Estimation System

Dang-Khanh Nguyen¹, Guee-Sang Lee¹, Soo-Hyung Kim¹,
Hyung-Jeong Yang^{1,4}, Seung-Won Kim¹, Min Jhon², and Joo-Wan Kim³

¹ Department of AI Convergence, Chonnam National University, Gwangju, Republic of Korea

² Department of Psychiatry, Chonnam National University Hwasun Hospital, Gwangju, Republic of Korea

³ Department of Psychiatry, Chonnam National University Hospital, Gwangju, Republic of Korea

⁴ Corresponding Author: hjyang@jnu.ac.kr

Abstract. People are under increasing pressure as the pace of work and life quickens, increasing their chances of developing depression. Due to its severe consequence, a specific effort is spent to diagnose this mental disorder as soon as possible. Machine learning and deep learning are expected to support clinicians in detecting depressed subjects via visual, text, and audio data. In this paper, we exploit the useful information from multiple modalities by utilizing transformer-based fusion to handle depression diagnosis. We conduct the experiment on a depression dataset, D-Vlog, in order to examine our deep-learning model. The promising results create a foundation for the further development of the Automatic Depression Estimation System.

Keywords: Depression Recognition · Transformer fusion · Multimodal fusion

1 Introduction

World Health Organization predicts depression can be the most widespread mental disorder by 2030 [1]. It affects seriously the quality of life and probably leads to some physical or mental illness. There is a certain probability that the severely depressed subject can commit suicide [2]. Consequently, detecting depression is a crucial topic that requires thorough knowledge and experience. However, it is also subjective and time-consuming [3].

To support clinicians in detecting depressed subjects, an automatic depression estimation (ADE) system is used to collect audiovisual data from the target and diagnose the level of depression. This discriminative system is expected to improve the speed and accuracy of professionals in early detecting depression. Recently, there has been a noticeable amount of research developing machine learning models to exploit the audiovisual clues in the ADE systems.

In this research, we would like to leverage the power of deep learning, particularly, multimodal transformer, in order to diagnose depression via subjects'

information, such as acoustic clues, visual data, and interview transcripts. Moreover, we contribute one more architecture in the taxonomy of [6], named hierarchical cross-attention to concatenation. This idea is expected to fully exploit the information of multimodal to accomplish higher performance compared to the traditional approaches. To evaluate our method, we use a recent depression detection benchmark, D-Vlog, Depression V-log from the Youtube platform.

2 Related works

There is a significant effort on developing the transformer [12] as a convolution-free machine learning model to achieve various tasks. Vision Transformer [13] is devised to handle image classification tasks and other computer vision problems while Audio Spectrogram Transformer [14] works with audio signal input. On the other hand, Tsai [7] uses a transformer as a mechanism to fuse the embeddings from multiple modalities. Shvetsova [8] combines multimodal transformers with contrastive loss to attain an impressive result in the video retrieval task.

Yoon et. al. [4] collect an audiovisual dataset for depression classification and introduce a baseline model using multiple levels of transformer encoder layers. Initially, they apply self-attention for each modality followed by cross-attention transformer layers between audio and visual features. Afterward, these features are concatenated in temporal dimension and fed into multimodal transformer layers. A global average pooling is used to get the representation of the whole sample and a fully-connected layer is employed to generate the final prediction.

EATD-Corpus [5] is a Chinese depression database recording interviews with some depressed and non-depressed subjects. The audio and transcript are collected and an SDS score [17] of each subject is recorded. The authors also provide a recurrent neural network baseline model exploiting the sequence information of each modality. ELMo [15] extracts sentence embeddings from raw transcript while NetVLAD [16] is used to obtain acoustic features from Mel spectrograms. A simple concatenation followed by a fully-connected layer is applied to fuse the attributes of text and audio modalities.

3 Transformer-based Fusion Methods

Our exploration of multimodal fusion is inspired by the taxonomy of Xu in [6]. Based on the research, we implement three versions of the transformer-based networks: early concatenation, hierarchical attention, and cross-attention to concatenation. Additionally, we contribute a combined interaction, which is hierarchical cross-attention to concatenation. The illustration of these architectures is shown in Fig. [1].

Early concatenation is a naive approach where the embeddings from two modalities are concatenated in the temporal dimension. A classification token is prepended then the sequence is fed into multiple transformer encoder layers to jointly learn the sequence information and the interaction between two modalities. After the multimodal transformer, the classification token is considered as

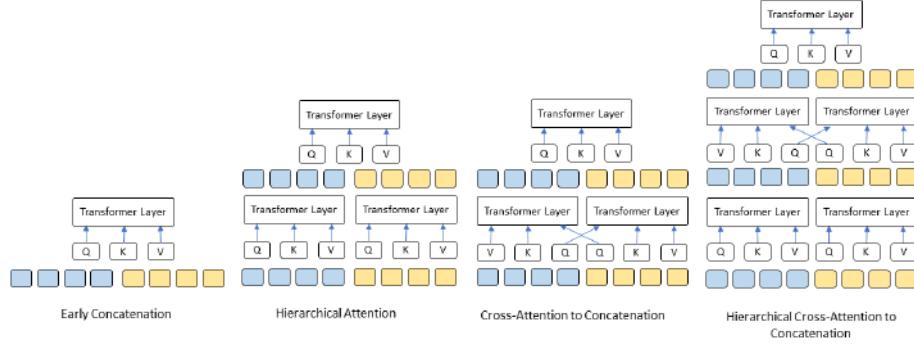


Fig. 1. Multimodal Transformers. "Q", "K", and "V" stand for query, key, and value embedding, respectively

the representation of the whole sample and a multilayer perceptron is used to generate the output for a specific task.

Hierarchical Attention and Cross-attention to Concatenation are two extended versions of the early concatenation, where self-attention and cross-attention are devised, respectively. In hierarchical attention, intra-modality information is learned by the attention mechanism via encoder layers. On the other hand, cross-attention exploits inter-modality knowledge. In both approaches, the multimodal transformer is utilized similar to the early concatenation concept.

The final approach, hierarchical cross-attention to concatenation, is our proposal for this paper. Firstly, we apply self-attention for acoustic and visual features separately. After learning the intra-modality knowledge, the model applies cross-attention where "query" comes from the source modality, "key" and "value" come from the target modality. After finishing exchanging inter-modality information, the features of the two modalities are concatenated in temporal dimension and fused by the multimodal transformer. Instead of using global average pooling, we use a classification token to acquire the representation of the video. A multilayer perceptron is adapted to generate the prediction logit.

4 Experiments and results

4.1 Dataset

Yoon et. al. introduce an audiovisual dataset for depression classification. The samples are collected from Youtube videos by searching for vlog-related keywords, such as: ‘depression vlog’, ‘daily vlog’, ‘depression journey’, etc. The videos are then extracted into acoustic and visual features. To obtain audio modality, OpenSmile [10] toolkit and eGeMAPS [11] are employed. The sound is re-sampled with a frequency of 1Hz and the final acoustic features comprised of spectral flux, loudness, MFCCs (Mel-frequency cepstral coefficients), etc. Re-

garding the visual modality, dlib [9] open-source software is utilized to extract facial landmarks.

The videos are re-sampled at a frequency of 1Hz for both modalities so the two modalities are aligned in terms of time. As a result, the dimension of visual and acoustic features are $t \times 136$ and $t \times 25$, respectively, where t is the length of the video in seconds. Moreover, the number of depressed and non-depress samples in D-Vlog dataset are 555 and 406, respectively. Therefore, unlike the former depression databases, it does not suffer from the imbalance issue. The dataset is split into train, validation, and test set with a ratio of 7:1:2.

4.2 Experiment setting

Pytorch framework is used for our implementation. The embedding dimension and feed-forward dimension in transformers are both set to 256. The number of attention heads and encoder layers is equal to 8 and 4, respectively. During the training process, we chose a learning rate of $1e-5$ and a drop-out probability of 0.1. For each network configuration described in section 3 we trained the model in 10 epochs with a repetition of 10, and the average scores of 10 experiments were recorded.

For each sample in the dataset, we downsample the audio and visual features in the temporal dimension with the rate of 4 and different offsets. By this processing, we create 4 new shortened versions with a length of one-fourth compared to the original samples. The downsampling augmentation not only decreases the complexity of the training by shortening the sequence length but also creates more samples for training the machine learning models.

4.3 Results

For D-Vlog benchmarks, the weighted average F1-score, recall, and precision are used for evaluation. Unsurprisingly, the hierarchical cross-attention to concatenation achieves the highest evaluation scores on the validation set of D-Vlog. Using self-attention or cross-attention separately is not an optimized setting, compared to early concatenation only. The detailed measures of each model are listed in Table 1.

Table 1. Evaluation scores of each model on D-Vlog validation set

Model	F1-score	Recall	Precision
Early Concatenation	63.92	64.71	64.58
Hierarchical Attention	62.32	63.14	63.44
Cross-Attention to Concatenation	61.36	62.55	62.87
Hierarchical Cross-Attention to Concatenation	64.70	65.29	65.66

After evaluating these transformer-based fusions on the validation set, we use the best model of each configuration and run it with the test split of D-Vlog.

Similarly, the hierarchical cross-attention to concatenation attains the highest F1 score and precision, compared to the baseline and other network settings. However, the baseline performs the best recall measure among the fusion methods. The weighted average F1 score, recall, and precision of these models are shown in Table 2.

Table 2. Evaluation scores of each model on D-Vlog test set

Model	F1-score	Recall	Precision
Depression Detector [4]	63.50	65.57	65.40
Early Concatenation	61.97	63.21	62.43
Hierarchical Attention	62.20	62.74	62.15
Cross-Attention to Concatenation	63.08	63.21	63.00
Hierarchical Cross-Attention to Concatenation	63.86	63.68	65.83

5 Conclusion

In this paper, we revised the multimodal fusion techniques that utilize the multi-head attention mechanism, especially, the multimodal transformer. By conducting the experiments on various network settings, we concluded that the combination of self-attention, cross-attention, and multimodal transformer can boost the performance of the model on a specific task, particularly, depression recognition. However, concatenating the modality feature in the temporal dimension can lengthen the sequence length. Consequently, the resources such as computational time and memory will increase dramatically due to the quadratic complexity of the pairwise attention with token sequence length. In the future, we will resolve this bottleneck to achieve better performance of the transformer-based fusion model.

Acknowledgements This work was supported by a National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT). (NRF-2020R1A4A1019191).

References

1. Mathers, Colin D., and Dejan Loncar. "Projections of global mortality and burden of disease from 2002 to 2030." PLoS medicine 3, no. 11 (2006): e442.
2. Kessler, Ronald C., Patricia Berglund, Olga Demler, Robert Jin, Doreen Koretz, Kathleen R. Merikangas, A. John Rush, Ellen E. Walters, and Philip S. Wang. "The epidemiology of major depressive disorder: results from the National Comorbidity Survey Replication (NCS-R)." Jama 289, no. 23 (2003): 3095-3105.

3. Maj, Mario, Dan J. Stein, Gordon Parker, Mark Zimmerman, Giovanni A. Fava, Marc De Hert, Koen Demyttenaere, Roger S. McIntyre, Thomas Widiger, and Hans-Ulrich Wittchen. "The clinical characterization of the adult patient with depression aimed at personalization of management." *World Psychiatry* 19, no. 3 (2020): 269-293.
4. Yoon, Jeewoo, Chaewon Kang, Seungbae Kim, and Jinyoung Han. "D-Vlog: Multi-modal Vlog Dataset for Depression Detection." Proceedings of the AAAI Conference on Artificial Intelligence (2022).
5. Shen, Ying, Huiyu Yang, and Lin Lin. "Automatic Depression Detection: An Emotional Audio-Textual Corpus and a GRU/BiLSTM-based Model." In ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6247-6251. IEEE, 2022.
6. Xu, Peng, Xiatian Zhu, and David A. Clifton. "Multimodal learning with transformers: A survey." arXiv preprint arXiv:2206.06488 (2022).
7. Tsai, Yao-Hung Hubert, Shaojie Bai, Paul Pu Liang, J. Zico Kolter, Louis-Philippe Morency, and Ruslan Salakhutdinov. "Multimodal transformer for unaligned multimodal language sequences." In Proceedings of the conference. Association for Computational Linguistics. Meeting, vol. 2019, p. 6558. NIH Public Access, 2019.
8. Shvetsova, Nina, Brian Chen, Andrew Rouditchenko, Samuel Thomas, Brian Kingsbury, Rogerio S. Feris, David Harwath, James Glass, and Hilde Kuehne. "Everything at Once-Multi-Modal Fusion Transformer for Video Retrieval." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 20020-20029. 2022.
9. King, Davis E. "Dlib-ml: A machine learning toolkit." *The Journal of Machine Learning Research* 10 (2009): 1755-1758.
10. Eyben, Florian, Martin Wöllmer, and Björn Schuller. "Opensmile: the munich versatile and fast open-source audio feature extractor." In Proceedings of the 18th ACM international conference on Multimedia, pp. 1459-1462. 2010.
11. Eyben, Florian, Klaus R. Scherer, Björn W. Schuller, Johan Sundberg, Elisabeth André, Carlos Busso, Laurence Y. Devillers et al. "The Geneva minimalistic acoustic parameter set (GeMAPS) for voice research and affective computing." *IEEE transactions on affective computing* 7, no. 2 (2015): 190-202.
12. Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).
13. Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani et al. "An image is worth 16x16 words: Transformers for image recognition at scale." arXiv preprint arXiv:2010.11929 (2020).
14. Gong, Yuan, Yu-An Chung, and James Glass. "Ast: Audio spectrogram transformer." arXiv preprint arXiv:2104.01778 (2021).
15. Neumann, ME Peters M., M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. "Deep contextualized word representations." arXiv preprint arXiv:1802.05365 (2018).
16. Arandjelovic, Relja, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. "NetVLAD: CNN architecture for weakly supervised place recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 5297-5307. 2016.
17. Zung, William WK. "A self-rating depression scale." *Archives of general psychiatry* 12, no. 1 (1965): 63-70.

Motion synthesis for automatic animation of sign language

Jong Ho Jeong¹, Hee Jae Hwang², Hong Nyeom Sung³ and Chil Woo Lee⁴

^{1, 2, 3} Department of Computer Engineering, Chonnam National University, Korea

⁴ Department of Software Engineering, Chonnam National University, Korea

¹wkrlsk23@gmail.com, ²certificate@kakao.com, ³6590f1@naver.com,

⁴leecw@jnu.ac.kr

Abstract. In this paper, we describe an algorithm and system that automatically animates a given Korean sentence into sign language. We plan to use this technology directly in cultural facilities such as museums, exhibition, and performance halls to improve the quality of life of the hearing-impaired. To achieve this goal, we made a sign language dictionary composed of about 2,500 words by capturing the sign language motion of the hearing-impaired and develop a technology to automatically generate sign language animation based on Unity. However, since all situations cannot be directly expressed only with the words recorded in the dictionary, it is necessary to create a sign language action of new meaning by combining two or more actions. In this paper, we describe the overview of the system developed so far and the connection of continuous sign language motions for expressing various sign language sentences and the synthesis algorithm of both arm motions in detail.

Keywords: Sign Language, Unity, automatic Animation algorithm, motion synthesis

1 Introduction

As information and communication technology develops, we use many useful services in our daily lives. However, the disabled people who have unwell body or difficulties in seeing and hearing often do not benefit from such high-tech services. In particular, the inconvenience that the hearing-impaired people have in using cultural facilities such as museums, exhibition halls, and performance halls is very serious. [1] In this study, to solve the difficulties of hearing-impaired people's experience when using cultural facilities, we describe a sign language animation automation system that translates Korean sentences into sign language sentences and then converts sign language sentences into sign language.

Since sign language has a different grammatical system from Korean, [2] it is treated as a separate language, just like foreign languages. Therefore, an independent Korean sign language dictionary has been defined as a separate language. But it's currently urgent to supplement and standardize the dictionary because not only the number of words is small, but also the expression behavior of one sign language word varies depending on the region or individual.

To automatically generate Korean sign language in cultural facilities, it is necessary to solve the following problems. The first is the definition and standardization of terms used in exhibition and performance halls. The terms used in exhibition halls are highly specialized, and there are many things that have not yet been defined in standard sign language. Therefore, definition and standardization of difficult terminology must be preceded. Second, since Korean sign language has a completely different grammar system from Korean, it is necessary to develop 'Korean'-Korean sign language' automatic translation technology. As artificial intelligence technology develops, technology for automatically translating foreign languages into Korean or Korean into foreign languages has been developed and is widely used in everyday life, but the technology for effectively translate Korean to Korean sign language is still insufficient. Thirdly, technology is needed to convert translated Korean sign language sentences into Korean sign language animations. In other words, with an input data composed by Korean sentence, a technique for playing the sign language with a continuous animation suitable for the Korean sign language grammar system is required. We are trying to solve these three problems through joint research "development of intelligent exhibition commentary text/Korean sign language conversion technology for the hearing impaired" supported by the Ministry of Culture, Sports, and Tourism of Korea. In this paper, we describe the natural connection method of continuous motions which is the core of an algorithm, that converts a given sign language sentence into sign language animation, mentioned in third problem.

If one sentence is composed of several sign language words and the sign language actions corresponding to the words are accurately defined in the sign language dictionary, one sign language sentence can be completed by continuously connecting the words in the sign language dictionary. However, since the motion data for each word defined in the dictionary always includes the start and end motions for each word to express it as an independent word, simply connecting the words in sentence does not naturally create a continuous motion, so it cannot express the original meaning. In other words, connecting several independent motions into one natural motion without discontinuity becomes the most important factor in automatic sign language generation.

Sign language words have different meanings depending on the shape of the hand and fingers, the direction of the palm, the position of the hand, and the movement of the hand [3]. Therefore, if the left and right Sign language express independent words with different meanings, various Sign language can be expressed by the combination of the two Sign language. Since we cannot record all the Sign words using in our daily life in a sign language dictionary; That is, since we cannot record all Sign motions as motion capture data, it is necessary to combine independent motions and use them as new words or other expressions. In this paper, we describe an algorithm that continuously connects independent words recorded in the Korean sign language dictionary, synthesizes two words with different meanings to create a new compound word, and automatically creates a sign language animation using it. Through this algorithm, the number of sign language words can be increased, various and natural sign language animations can be displayed.

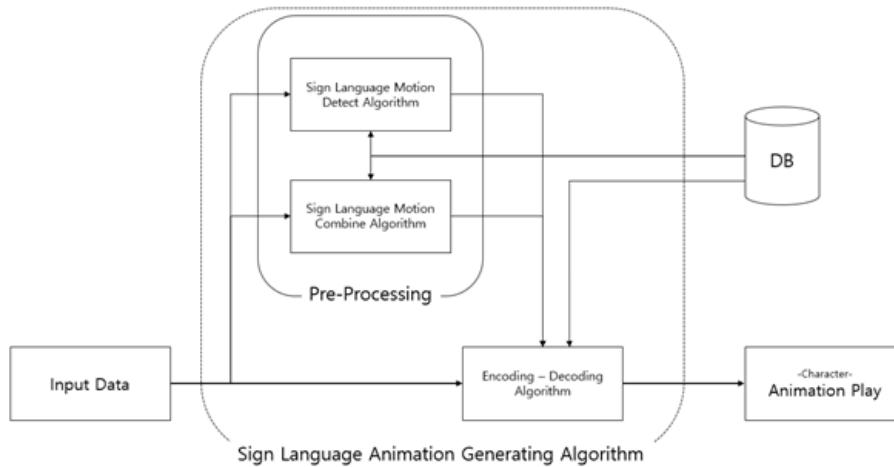


Fig. 1. Korean Sign Language automatic generation system Algorithm diagram. Before encoding-decoding algorithm performed, pre-processing is performed if necessary.

2 Sign language automatic generation system

2.1 Overall system overview

Sign language automatic generation system goes through the process shown in Figure 1. First, system analyze the input sentence to determine whether preprocessing is needed or not. If it is determined that pre-processing is necessary, a pre-processing step is performed. In the pre-processing step, system obtain the information which is necessary in final animation step by playing the sign language motion, synthesize the sign language motion which is necessary in final animation step. By synthesizing sign language motion in advance and using the motion data as it is in the final animation process, natural motion synthesis is possible. Therefore, the pre-processing process is divided into a sign language motion detection process and a sign word synthesis process.

After the pre-processing step, the animation step is executed. In the animation step, the Korean sign language sentence input using the pre-captured data is reproduced through and encoding-decoding algorithm. In this step, algorithm reproduce facial expression and sign language motion at an appropriate position input by user to make a natural sign language expression based on input sentence.

3 Representation of sign language word data

3.1 Korean sign language dictionary

To automatically generate sign language, it is first necessary to translate Korean sentences into sign language. In this study, sign language translation is performed by deep learning [4], and the result of translate is used as input to the system. As shown in Figure 2, in the entire process, we use the sign language dictionary produced through this study. This dictionary is divided into several categories to suit the characteristics of the word. In detail, There is SUJI-words (sign language word) that can be expressed with hand gestures, BISUJI-words (non-manual signal) that cannot be expressed by hand, such as facial expressions, JIWHA-words (finger language word) that written directly with finger, such as numbers, and SUHYUNG-words (Hand Style word) that used to synthesize two sign-language words. Each word has a category number to which the word belongs, a unique number of the word, and motion data in FBX format. Simply speaking, this dictionary can be said to be a database that systematically links sign language motion data to Korean words. This dictionary is used to find out if input data is a new word, or to generate an animation of input sentence. In this study, about 2500 words were divided into categories, and the sign language motion performed by the hearing-impaired for each word were captured as digital data to create a sign language dictionary. Since sign language is expressed as one independent motion that expresses meaning unlike general language, one sentence can be completed by continuously connecting motions corresponding to each word.

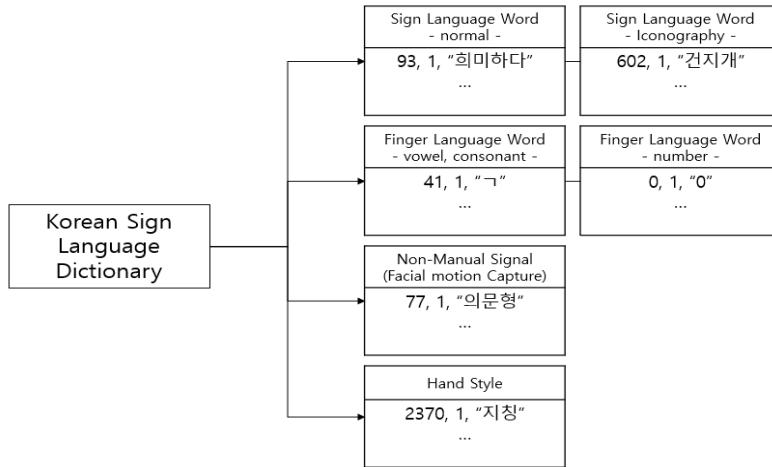


Fig. 2. Korean Sign Language Dictionary composition. The Dictionary compose 4 categories. In case of Sign language, sign language used in special situation such as museum are classified as iconography, and the others are classified as general. In case of Finger language, it is distinguished by finger language refers to vowels, consonants of Korean, and finger language refers to numbers.

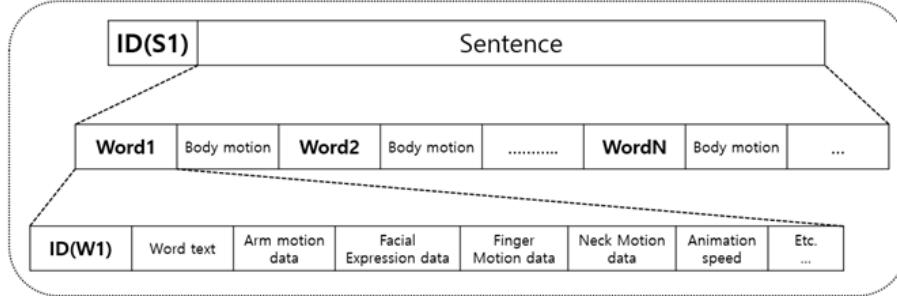


Fig. 3. Sign Language Sentence and Word Description Model. One Sentence contains several words, and one words contains various variables.

3.2 Korean sign language descriptor

The automatic sign language generation system reproduces Korean sentences translated into sign language words as sign language animations [5][6]. At this time, the sign language descriptor shown in Figure 3 is used to accurately reproduce the input sign language sentence as an animation. According to the sign language descriptor, one sentence data consists of several words, and various variables for animation control are included inside one word. In the system proposed in this paper, when playing an animation using the encoding-decoding algorithm in the animation playback stage, natural animation can be created at the correct timing by using the variables that words have.

4 An Algorithm for Automatic Sign language generation

4.1 Pre-processing for motion combination

Sign language animation automatic generation algorithm

Pre-processing for motion combining.

The sign language animation generation algorithm is preprocessed through two processes. First, in the sign language motion detection process, information necessary for connecting two motions is inspected to create one animation data, and in the second step, two words are synthesized to create a new word.

4.1.1 Connection of two consecutive sign language motions

Sign language words store motion representing the meaning of each word as digital data. When acquiring motion data using a motion capture device, motions at the start and end of words are inserted to distinguish motions. Therefore, to create a natural sign language animation, it is necessary to remove the motion used for word classification [7].

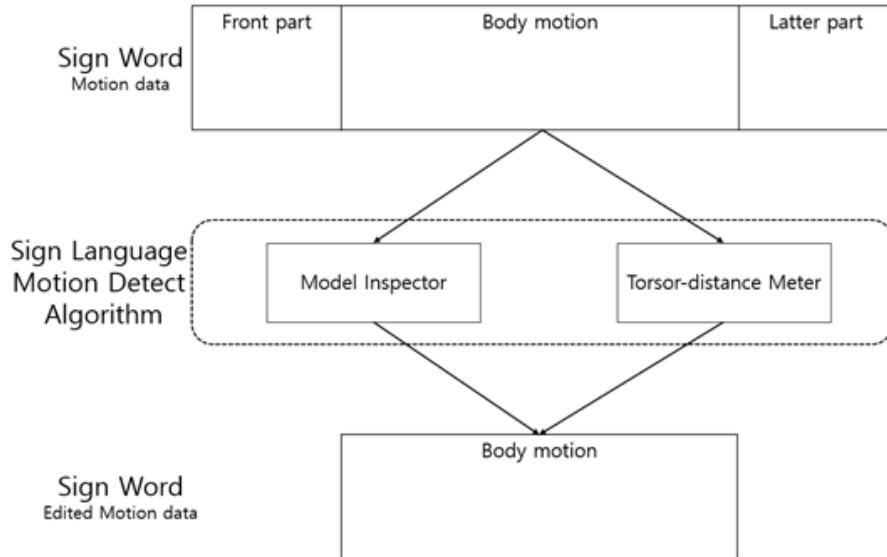


Fig. 4. Sign Language Motion Detect Algorithm diagram. 2 modules work in Sign Language Motion Detect Algorithm to inspect motion used for motion classification. After inspect Motion data, Sign language word edited.

The algorithm in Figure 4 is used to cut the back-and-forth motion of the motion data. In this algorithm, the Model Inspector, which reproduces sign language motions and detects the position of the hand and the torsor-distance Meter module, which calculates the distance to the torsor, operate. Model Inspector designates a specific location and records the current time when a hand enters the designated location and is detected. Torsor-distance Meter measures the distance from the character's torsor to the wrist, and if the value is below a certain value, it is judged as an action step and the current time is recorded. Using these two modules, the expected motion time of the sign language word motion data is specified, and the motion data is modified according to the time. The edited data is stored in the DB so that the motion can be seen naturally when Korean sign language animation is created later.

4.1.2 Hand Motion Combine (HMC) Algorithm

Sign language can express new meanings by combining the movements of the two arms in various ways. In other words, when emphasizing instructions such as ‘this one’, ‘this way’, or indicating the size of an object by one hand, two sign language are combined. Since words representing indications or emphasis are independently defined in sign language dictionaries, an algorithm that synthesizes two independent sign words into a single sign word is required for a new expression. That is, the sign word motion corresponding to the left-hand motion and the sign word motion corresponding to the right-hand motion are synthesized into one independent word, and the process proceeds through the algorithm shown in Figure 5.

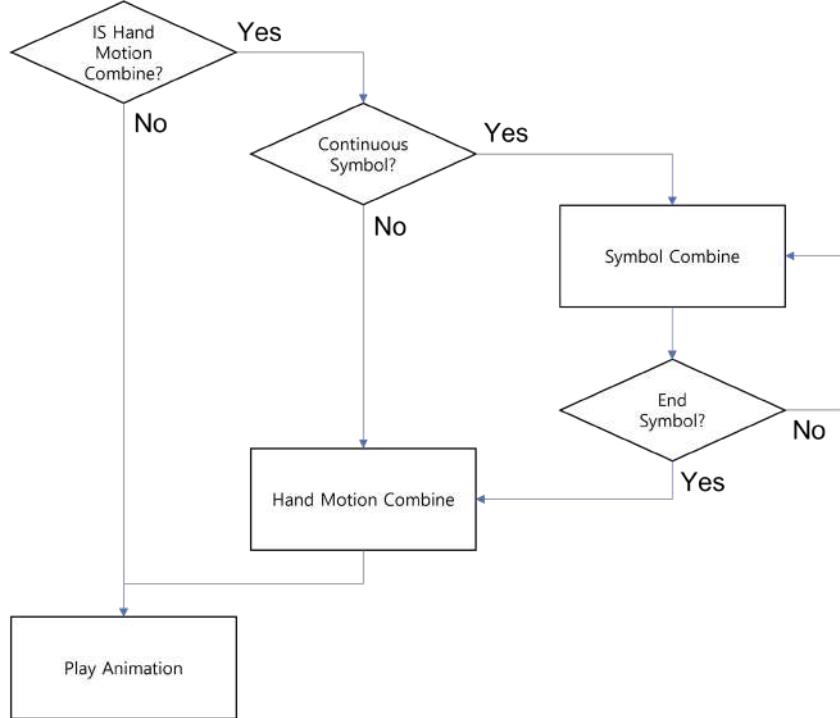


Fig. 5. Automatic Sign Language generation Algorithm diagram. The algorithm determines whether a Hand Motion Combination is necessary and, if necessary, determines whether a word is continuous symbol. If it is continuous, do the symbol combination. After then, it is used for animation playback after Hand Motion Combination.

First, the entire input data is scanned to review whether synthesis of left-hand and right-hand motions is necessary. As a result of the sentence scan, if it is determined that the synthesis of the left-right hand motion is necessary, it is determined whether the sign language word motion corresponding to the right-hand motion is a continuous motion consisting of an explanatory word. In this process, if it is not a continuous motion composed of language words, the two motions are synthesized through the HMC.

4.1.3 Symbol combine algorithm

In the automatic sign language animation generation algorithm, animation is created with data of one word unit. JIWA-words are composed of phoneme units of characters such as 'o' and '|'. In the algorithm, one phoneme corresponds to one word, so in the case of the continuous action of the right hand JIWA-words, one left hand action will be combined with the right-hand actions. In this case, since the left-hand motion is repeated whenever the right hand motion is changed, the animation

becomes unnatural, so the right hand motion must be combined into one motion and then combined with the left hand motion.

In the HMC synthesis step, if the sign word motion corresponding to the right-hand motion is a continuous motion consisting of JIWHA-word, a process of combining the continuous motion into one motion data (this is called Symbol Combine) is executed. Symbol Combine uses Unity's additional modules Recorder Module and FBX Recorder Module. [8][9] Whenever one JIWHA-word is combined, the algorithm determines whether the continuous motion has been completed, and all JIWHA-words are combined to form one sign language motion. After making it, it is synthesized into a new word by applying the HMC algorithm.

4.2 Animation play procedure

4.2.1 Encoding Decoding Algorithm

In the previous paper, sign language animation was generated through the encoding-decoding algorithm [10]. The encoding-decoding algorithm is an algorithm that analyzes and encodes input data, then decodes and animates the encoded data in a generator. The encoding and decoding process is performed as shown in Figure 6. In the encoding process, the input data is analyzed through the DB, synthesized into word descriptors, and sentence data composed of word descriptors is sent to the generator. In the decoding process, sentence data is decoded in units of word descriptors, and playback variables in word data are analyzed, and then animations are displayed at appropriate timings. Through this, a user may display a sign language animation by combining words for which data exists without complicated manipulation.

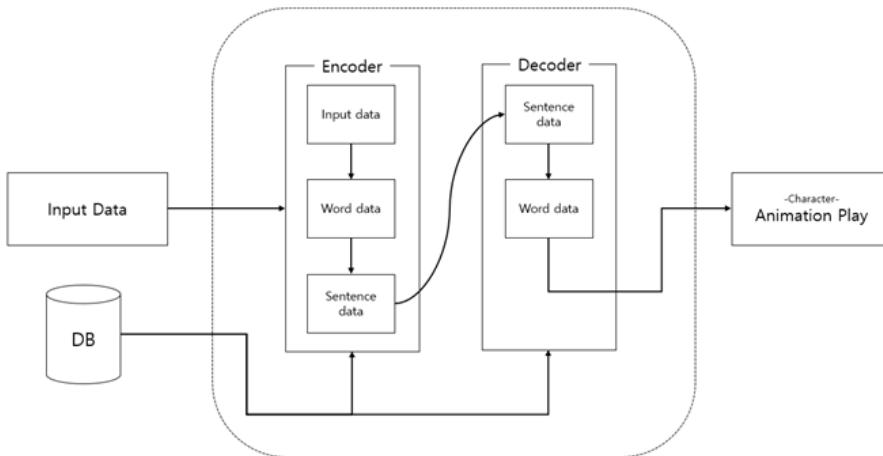


Fig. 6. Encoding Decoding Algorithm diagram. Input data encode to Sentence data in encoder. After then, decoder receive the Sentence data and decode that data to Word data. In Word data, there were various variables to control animation play.

5 Animation result

To confirm the results of the algorithm presented in this paper, motion capture data used in previous studies were used. [11] Figure 7 shows the result of HMC when the left-hand words are not continuous JIHWA-words. Figure 8 shows the result of HMC when the left-hand words are continuous JIHWA-words, and in this case, Symbol Combine is used to make the JIHWA-words into one word. Figure 9 shows the result of HMC using the motion words created in Figure 8. The sentence playback result is shown in Figure 10. The motion used at points 8 and 9 in Figure 10 is the same HMC + Symbol result as in Figure 9, and through this, the connection between the motions can be seen to be natural.

Furthermore, when the automatically generated Korean sign language animation was applied through the demonstration service conducted in joint research, it was confirmed that the understanding increased. This indicates that the animation is playing an important role in improving the senses and understanding of users and providing a better experience. Figure 11 shows the photos and app screens at the time of the demonstration service. Figure 12 shows the change in satisfaction after the demonstration service, and it indicates that the overall satisfaction increased when the animation was applied.

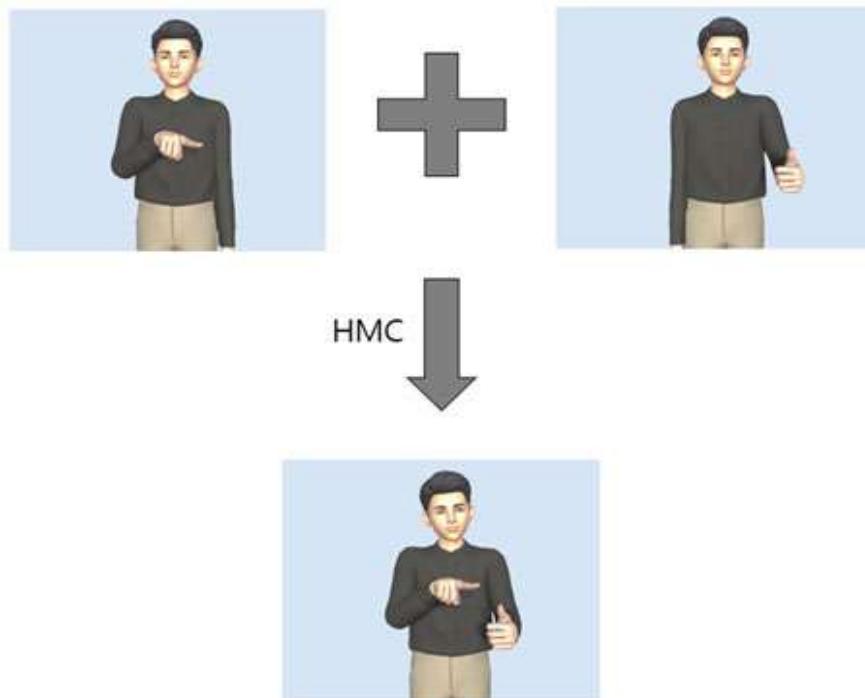


Fig. 7. Output of Hand Motion Combine Algorithm. It can be seen the left-hand motion and the right-hand motion are combined to become one new motion.

10

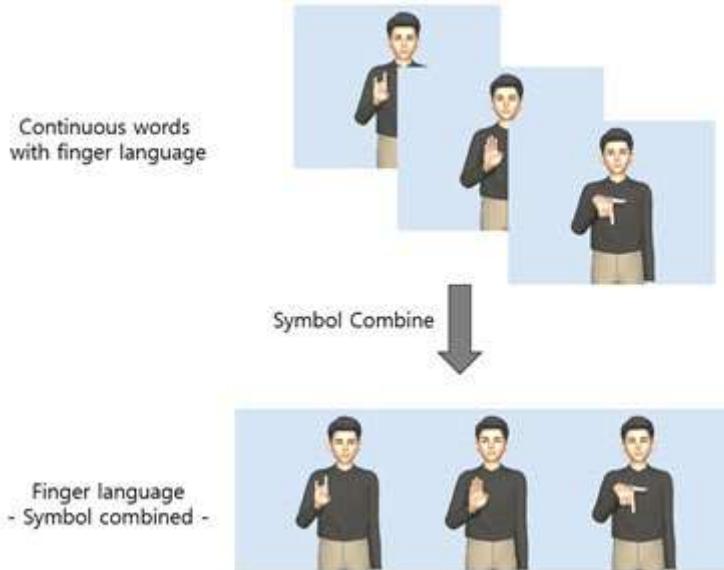


Fig. 8. Output of Symbol Combine Algorithm. Continuous words with finger language are combined to one right-hand motion.

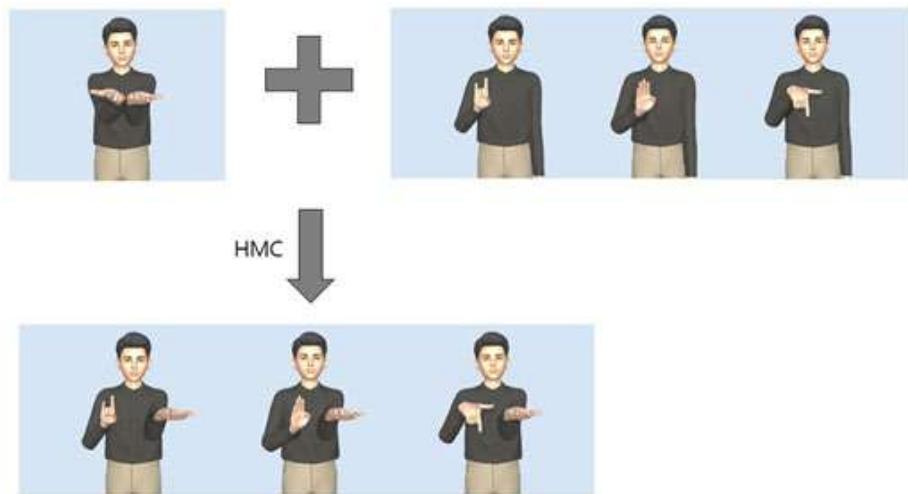


Fig. 9. Output of Hand Motion Combine Algorithm after Symbol Combine. The result of Figure 8 is combined on the right-hand motion. The right-hand motion (Continuous finger language words) is reproduced while maintaining the left-hand motion

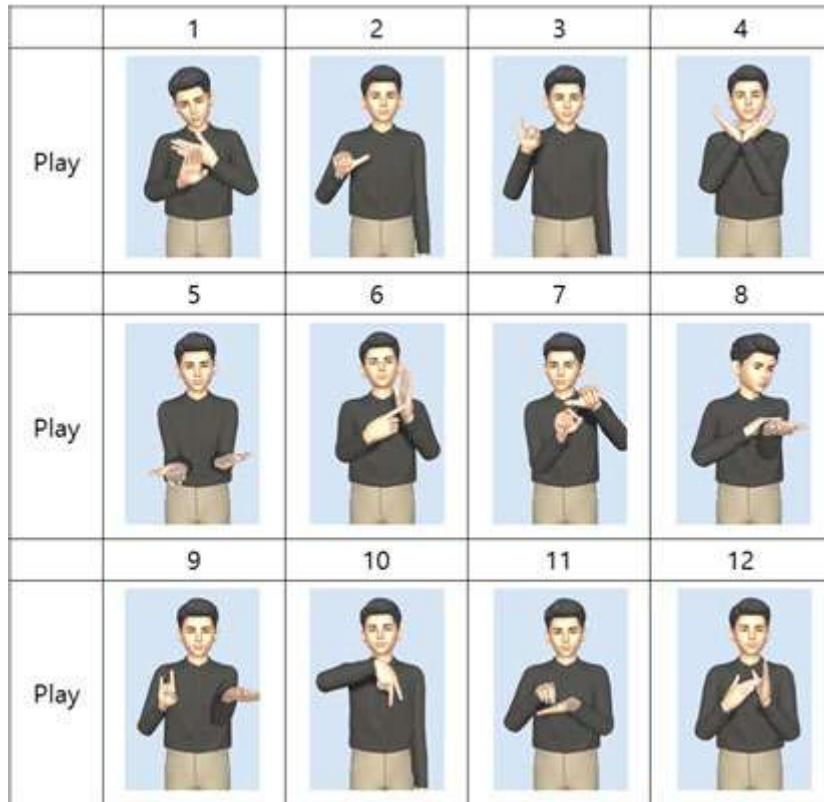


Fig. 10. Output of playing Sign Language Sentence. It is a reproduced sentence including the Figure 9. In frame 9, The HMC worked fine.



Fig. 11. Left shows the photo at the time of the demonstration service. Right shows the app screen at the time of the demonstration service.

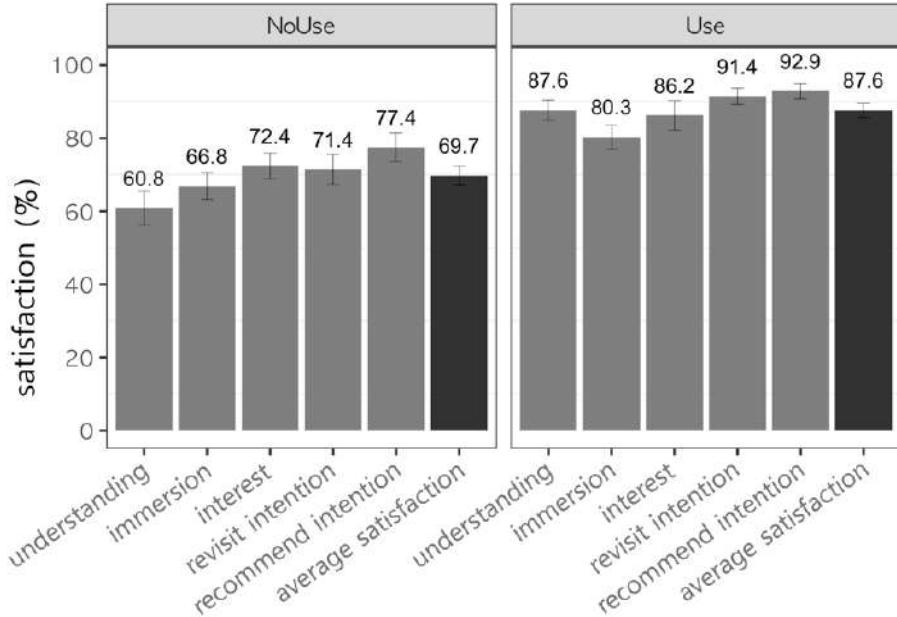


Fig. 12. The change in satisfaction before and after using application. It can be seen overall satisfaction has increased after using it.

6 Conclusion

Korean sign language is a language with a different grammar system from Korean, and technology to automatically convert Korean into Korean sign language animation is needed for the smooth cultural life of the hearing-impaired. To do this, a Korean-sign language conversion dictionary is required. If the sign language dictionary has sign language motion data corresponding to sign language words, it is possible to express a Korean sentence in sign language by continuously connecting sign language words. However, if the sign language words stored in the dictionary are connected as they are, an unnatural motion occurs, and the meaning is not properly conveyed. Since the meaning of a sign language is determined by its shape of hands, and a sign language with a new meaning can be created through a combination of two sign language actions, an innumerable number of words that can be created by combining sign words must be captured in a sign language dictionary. However, it is realistically impossible to capture an innumerable number of sign language words. To solve these problems, this paper describes a system that reproduces Korean sentences as sign language sentences after preprocessing sign language words. Motions including start and end motions were refined through pre-processing, and when a word composed of a combination of two sign language motions was needed, a synthesis algorithm was

used to synthesize them, and then sign language animation was reproduced using an encoding-decoding algorithm.

In the future, we plan to improve the completeness of the system by adding basic words necessary for motion synthesis and correcting minute discontinuities that appear after motion synthesis.

Acknowledgements

This research is supported by Ministry of Culture, Sports, and Tourism (MCST) and Korea Creative Agency (KOCCA) in the Culture Technology (CT) Research & Development Program (R2020060002) 2022.

References

1. Mi Jeong Lee, Sun Hwa Lee, Jin Hwe Kim, "A Survey on the Utilization of Welfare Services for the hearing-impaired and Their Needs", Korea Disabled People Development Institute, 2010.
2. Seong Ok Won, et al. "Korean Sign Language and Grammar Research", Korea Welfare University Industry-Academic Cooperation Group, 2019.
3. Introduction of Korean Sign language,
https://www.korean.go.kr/front/page/pageView.do?page_id=P000300&mn_id=202
4. Vu, Dang Thanh, et al. "Text Data Augmentation for the Korean Language." *Applied Sciences*, 12.7 (2022).
5. Young Min Ko, et al. "A Sign Word Description Model for Efficient Generation of Sign Language", Korea Smart media Conference, 10(1), pp.15-18, 2021.
6. Young Min Ko, et al. "The Development of Animation Technology for Intelligent Sign Language Transformation for the Hearing Impaired", MITA, 2020.
7. Jong Ho Jeong, et al. "A FBX File Combination Algorithm for Natural Sign Language Animation", Korea Smart media Conference, 2021.
8. Unity Recorder module,
<https://docs.unity3d.com/Packages/com.unity.recorder@2.2/manual/index.html>
9. Unity FBX Recorder module,
<https://docs.unity3d.com/Packages/com.unity.formats.fbx@4.0/manual/recorder.html>
10. Jong Ho Jeong, et al. "Advanced Sign Word Generation Algorithm for Natural Animation", Korea Smart media Conference, 2022.
11. Jong Woo Seo, et al. "A study on Korean sign language translation script method", a Study on the Integrated Academic Presentation of the Korean Management Association, 191-198, 2021

Cattle Action Recognition with Multi-Viewpoint Cameras based on Deep Learning

Muhammad Fahad Nasir^{1,2}, Alvaro Fuentes^{1,2[0000-0001-8847-1541]}, Shujie Han^{1,2}, Jongbin Park^{1,2}, Sook Yoon³, and Dong Sun Park^{1,2}

¹ Department of Electronics Engineering, Jeonbuk National University, Jeonju, 54896 South Korea

² Core Research Institute of Intelligent Robots, Jeonbuk National University, Jeonju, 54896 South Korea

³ Department of Computer Engineering, Mokpo National University, Muan, 58554 South Korea

dspark@jbnu.ac.kr; syoon@mokpo.ac.kr

Abstract. Cattle activity recognition is an essential element in monitoring cattle health. Cattle behavior is a primary indicator of its well-being, any anomalous behavior displayed by cattle is the earliest indication of illness, and if treated immediately it can prevent aggravation and development of diseases, which is detrimental to cattle's health. To study cattle activity, embedded devices have been used but they can be a cause of discomfort, and stress. This paper focuses on training a deep learning model to detect and localize multiple cattle in video frames, captured from multiple cameras at multiple angles during day and night with an overall precision of 95.3%. Our system provides information about cattle mobility, which can detect any inconsistent behavior exhibited by cattle, leading to early detection and prevention of disease. Furthermore, we share an in-depth analysis of the model's performance on our raw dataset and its effectiveness in recognizing individual, group, and part cattle behavior, and facilitating cattle health monitoring.

Keywords: deep learning, animal welfare, action recognition, video, camera.

1 Introduction

The growing population of the human race is challenging the cattle industry, with sheer intensity of demand for cattle products [3]. The agriculture industry needs to optimize itself to keep up, and this opened the doors for innovation, technology, and automation [5,12]. Hence, cattle welfare is of the utmost importance for maintaining ideal cattle produce. Cattle are susceptible to disease and illness, in general, diseases are detected

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF), funded by the Ministry of Education (No. 2019R1A6A1A09031717); Korea Institute of Planning and Evaluation for Technology in Food, Agriculture and Forestry (IPEF) and Korea Smart Farm R&D Foundation(KosFarm) through Smart Farm Innovation Technology Development Program, funded by Ministry of Agriculture, Food and Rural Affairs (MAFRA) and Ministry of Science and ICT (MSIT), Rural Development Administration(RDA)(421044-04); and National Research Foundation of Korea(NRF) grant funded by the Korea government (MSIT) (2020R1A2C2013060).

too late when the spread has continued for too long causing severe illness, and often death [4]. However, the intensity of disease is preventable by monitoring cattle behavior meticulously [2]. Cattle tend to depict anomalous behavior when they are under duress, anxiety, or the influence of disease [6]. There are key behavioral indicators depicted which can help in the early detection and prevention of diseases [7].

Traditionally, cattle welfare has been carried out via manual checkups conducted by a farmer or physician [2]. This approach is laborious and time-consuming, and the quality of results is mainly dependent on the intuition, and expertise of the observer [3]. To improve the monitoring and observations of cattle behavior, a better alternative has been using the assistance of technology [12]. In this regard, the term Precision Livestock Farming (PLF) has been coined, for monitoring of every aspect of cattle activity using technology [1]. For instance, portable and wearable devices have been extensively adapted in cattle farming, but these technologies have been also called into question as they tend to be intrusive, stressful, and uncomfortable for cattle, causing cattle to deviate from normal behavior [11]. To overcome these challenges, our approach is to utilize a non-intrusive, highly efficient deep learning model, using RGB cameras to detect animal behaviors and contribute to animal welfare.

Our work extends the work of [1], in categorizing cattle activities into 15 respective hierarchical behaviors. The behaviors are sub-categorized into three categories, namely: i) Individual cow actions, ii) Group cow actions, and iii) Part cow actions. These respective categories are illustrated in Fig 1.

For this research work, we installed RGB cameras in the farms to gather videos for our dataset. In our specific application, we faced significant challenges such as 1) Camera viewpoint, when the entire cowshed is not visible, and some cows may move to blind spots [16]; 2) Illumination: at night time, the cow shed is not as well-lit and affects the visibility [16]; 3) Deformation: cow poses from some viewpoints are deformed and it is difficult to interpret the specific action [8]; 4) Occlusion: this is one of the biggest challenges, as either cattle itself, cow shed structure or blind-spots can hide considerable part of cattle's body, stressing the visibility for camera and challenging the model robustness [17]; 5) Background Clutter: combined with occlusion, a model can overlook some cattle and generate false negatives [8]. Also, during nighttime cattle standing far away cannot be differentiated from the background.

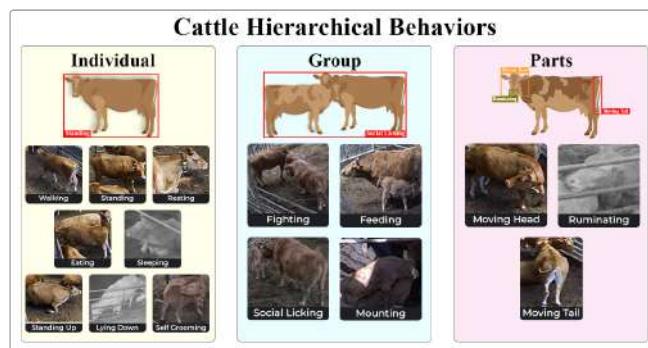


Fig. 1. Hierarchical behaviors depicted by Hanwoo cattle.

Our proposed solution is to use a deep learning model, on our dataset collected from multiple cowsheds, inhabiting 18 cows and some calves, from multi-viewpoint cameras, during day and night. The video recordings collected are converted to image frames and annotated using bounding boxes and classified into 15 different behavioral categories. For the robustness of our model, our dataset contains real-world challenges as stated above, this allows us to widen the applicability of our work in different farms. Our model performance for 90 percent of actions is more than 97% and above, and an overall accuracy of 95.3%.

Our main contributions in this paper are: i) A highly accurate deep learning-based model to detect cattle behavioral characteristics; ii) A diverse dataset, collected from different farms, with multiple cattle, from different RGB camera viewpoints and during the day and nighttime; and iii) An analysis and discussion on how these actions help detect a pattern for early detection of illness.

The remainder of the paper is structured as follows. We present the dataset and methodology in Section 2. Details of our experimentation and analysis are described in Section 3. We then conclude our findings in Section 4.

2 Methodology

Our work on cattle action recognition is aimed at studying Hanwoo, a cattle native to Korea, raised for its meat. Due to the mountainous geography of Korea, open cattle farming is not an option, so Hanwoo are raised indoors in cow sheds. Most research on Hanwoo are specific to just a few hierarchical behaviors such as ruminating [3], eating, and drinking [14], some are disease-oriented [15] or research conducted via labor work manually [2]. We propose a system that performs a comprehensive study of Hanwoo actions and discusses their impacts. Our system is composed of four stages namely, data collection, data preparation, annotations, and deep learning model as shown in Fig 2.

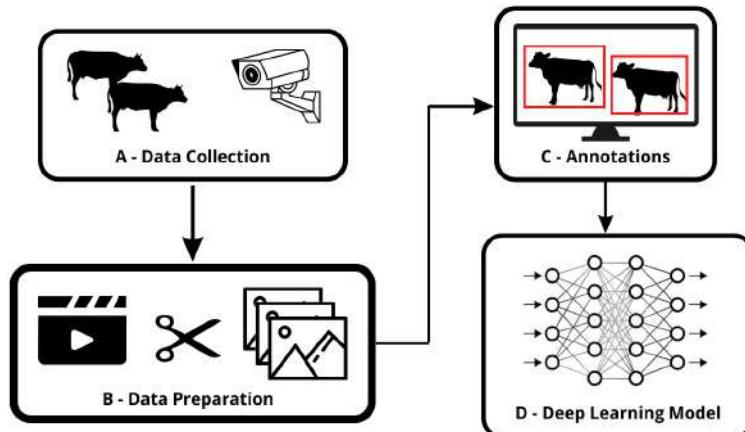


Fig. 2. Overview of the proposed system for Hanwoo cattle behavioral recognition.

2.1 Model Pipeline

For this study, the objective was to develop a framework that would aid in cattle welfare and thereby early detection and prevention of disease; another priority was non-intrusiveness. For these purposes, we installed RGB cameras in a cowshed to collect a diverse set of data, annotated it, and then proceeded to utilize that data in our model for training it. We share each step in detail below.



Fig. 3. Cattle sheds camera viewpoint for the dataset, of Hanwoo cattle.

A) Data Collection. To compose a dataset, we visited indoor Hanwoo cattle farms in South Korea. To get a realistic idea of the environment in Hanwoo farms, we shortlisted a cow shed with 18 cows and 3 calves of varying sizes and growth stages. We proceeded to install RGB cameras in the shed, at multiple angles. The dataset consists of RGB video frames of the Hanwoo cattle enclosure. We choose recording footage from three different cameras, some details are available in Fig 3.

B) Data Preparation. After the collection of data from the three cameras, we chose recording clips and form a unique dataset. The goal was to incorporate almost all types of actions performed by cattle as suggested in Fig 1. We compiled snippets of recordings from all three cameras and converted these recordings into image-level frames. We then proceeded to send the prepared data to the annotation stage.

C) Data Annotation. Image frames obtained were then labeled with each cow's actions. The categorization of these actions is 1) Individual Actions: it specifies action being performed by a single cow. It considers the entire cow and classifies it to a certain action as shown in Fig 1. 2) Group Actions: it specifies actions being performed by more than one cow. These actions represent cattle social interactivity with neighboring cattle, as shown in Fig 1. 3) Part Actions: they refer to the activity and movement of specific body parts of an individual cattle, including head movement, tail wagging, and ruminating, as shown in Fig 1.

The ground truth annotations were done manually by labeling every cow in the visible area. Table 1 showcases the instances of each of the 15 hierarchical actions being performed in our dataset.

D) Deep Learning Model. Our proposed approach is to perform action recognition of cattle farms. We faced numerous challenges as mentioned in Section 1, and multiple

cattle (objects) to be detected. Since we aimed to create a system capable of detecting cattle activity in real time of multiple objects, we required a model operating in real-time, with high accuracy. We achieved this by using YOLOv5 [9], a single-stage object detector, capable of operating in real-time.

Table 1. Hanwoo cattle action instances. The values reflect the annotations.

Hierarchical category	Actions	Dataset A	Dataset B	Dataset C	Total
Individual	Walking	3,305	1,560	1,562	6,427
	Standing	25,585	10,853	25,044	61,482
	Resting	18,768	7,080	456	26,304
	Eating	-	-	-	-
	Sleeping	-	-	-	-
	Standing Up	243	284	-	527
	Lying Down	-	-	-	-
	Self Grooming	247	224	-	471
Group	Fighting	-	-	-	-
	Feeding	228	-	-	228
	Social Licking	-	-	-	-
	Mounting	-	-	-	-
Part	Moving Head	308	231	660	1,199
	Ruminating	5,430	1,241	4,161	10,832
	Moving Tail	3,505	1,998	16	5,519

A detailed overview of the proposed framework is depicted in Fig 4. We provide our annotated dataset to the model, it passes through the model backbone, where features are extracted from the image and a reduced spatial resolution of the respective feature maps is obtained. The feature maps are forwarded to the model neck which performs further processing by increasing the depth and reducing the spatial resolution. The results are sent to the model head where the final predictions for cattle actions are conducted.

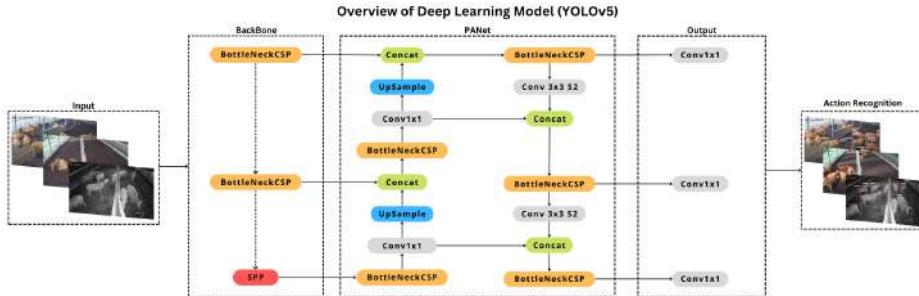


Fig. 4. The overall structure of the model and detailed insight into YOLOv5 architecture.

Our model provides us with the class of action detected and the confidence score of the accuracy of that prediction. During training, the loss function continues to optimize the model to overcome the discrepancy between predicted output and ground truth. In object detection, another factor of vital importance is the Intersection over

Union (IoU) of bounding boxes, as the model aims to find the exact coordinates of the bounding boxes, so the loss function equation can be written as:

$$\text{Loss} = L_{\text{class}} + L_{\text{conf}} + L_{\text{CIOU}} \quad (1)$$

where L_{class} , L_{conf} , and L_{CIOU} are the classification loss function, confidence loss function, and bounding box regression loss function, respectively. Details about each function are provided in [9].

3 Experimental Results

This section shares our model specifications, performance, and experiments on the cattle farm during different times of the day and from multiple viewpoints. We share detailed insights about the overall performance of our model, performance with independent actions, and how our system aids in cattle welfare.

3.1 Implementation

We utilized the YOLOv5x model and train it on our dataset. To challenge our model, we made sure not to select temporally adjacent frames but rather built our test dataset by selecting random frames. These frames sometimes introduce part actions at slightly occlusive situations as well, this confronted the robustness of our model. We achieved the best results using a 70/30 train-test split of our dataset, and we trained the model for 300 epochs. Our server consisted of 2 NVIDIA TITAN V GPUs with 12 GBs each. The training time for the model was 34.7 hours with a batch size of 16. We also utilized several data augmentation techniques such as mosaic, copy-paste, albumentations, and random horizontal flip. Allowing our model to be able to deal with viewpoint, occlusion, and deformation challenges to increase our dataset for better performance of the model. These allow us to surge the dataset size and generate a robust model.

3.2 Quantitative Results

The metric for measuring the performance of object detectors is the mean average precision (mAP). For our model, we evaluated the performance with the mAP at an IoU threshold of 0.5, meaning the mapping of prediction and ground truth is computed at a 0.5 threshold. We adjusted the learning rate of the model at 0.01, whilst setting the momentum at 0.937 and a weight decay of 5×10^{-4} . We achieved an overall mAP of 95.3%. Since our model was trained on the detection of 15 behavioral categories of cattle actions, individual accuracies can be seen observed in Fig 5a.

The confusion matrix provides an in-depth understanding of where the model makes incorrect detections, these are generally due to the vision challenges mentioned in the introduction section, and mainly due to transitional actions. As shown in Fig 5b, an example of that is walking and standing, in video, these actions are quite clear due to the visible temporal information, but in images, when a cow starts walking there are instances where all of its limbs are touching the ground, but the cow is still in a walking

instance and the model interprets to have stopped and incorrectly states it to be standing. Similarly, part actions such as moving head, ruminating, and moving tail have a few False Negatives (FN) and False Positives (FP) because of the non-availability of temporal information.

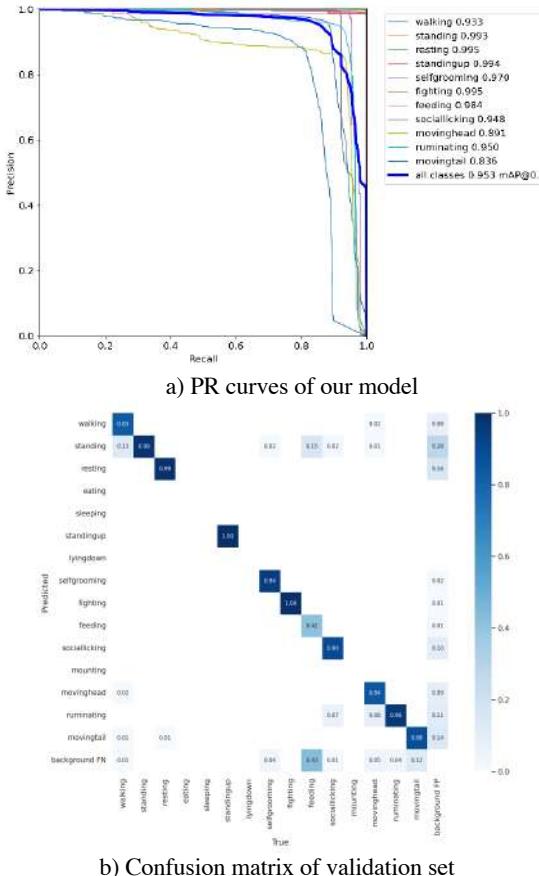


Fig. 5. Detailed analysis of model performance on our dataset.

3.3 Qualitative Results

We implemented our trained model on the video recordings of cattle farms and the results are showcased in Fig 6, along with the respective time duration. Our model provides bounding boxes along with action classification and a confidence score, with a threshold of 45% confidence during inference.

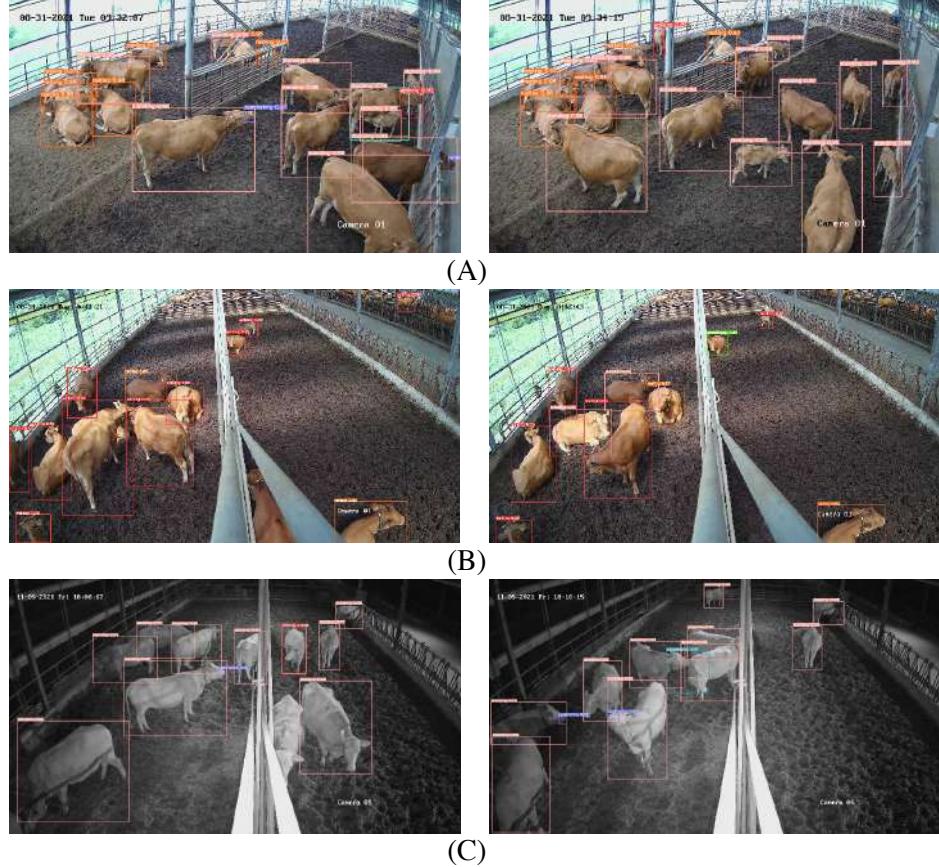


Fig. 6. The implementation of our deep learning model on video recordings from cattle farms during the day and nighttime. (A) Detection at a time interval of 09:00 – 10:00; (B) Detection during 10:00 – 11:00; (C) Results during nighttime 18:00 – 19:00. The bounding boxes inform us of the action being performed with a confidence score.

3.4 Cattle Welfare Analysis

For the results of our model, we observed that actions were performed with high accuracy. Using this, we can monitor and analyze individual cow actions and detect any anomalous behavioral patterns. Cow actions reflect animal health such as rumination time, as farmers monitor the rumination time to predict the calving period and oversee the process to avoid any complications [3]. Similarly, early indications of lameness can be observed by walking patterns and head movements [6,10]. Cows socializing less, spending more time resting or lying down, spending less time walking, and reduced eating and ruminating behavior are indications of a cow under stress or suffering from a disease [13]. In general, healthy cows ruminate more than cows suffering from disease [14]. If suddenly a cow's rumination time decreases, it is a sign of ailment [3]. Any significant change in the daily activity routine of cows provides information about their

health. These indicators are incredibly important as early detection of any disease can lead to timely medical treatment and a higher likelihood of recovery of the animal and less cost to the cattle owner [15].

4 Conclusion

In this paper, we proposed an effective cattle monitoring technique that utilizes cattle farm video recordings and recognizes cattle behaviors. For the execution of this study, we collected a diverse dataset from indoor cattle farms using RGB cameras placed at multiple viewpoints and collecting data during the day and nighttime. Our system is accurate and robust as it is trained on a raw dataset, with numerous visibility challenges, multiple objects for detection, 15 classes, and real-time rendering capabilities. We also provided an analysis on how using our research on cattle activity can help in detecting a pattern for cattle stress and ideally inform cattle owners of early disease development in cattle, thereby leading to early treatment or prevention of disease.

References

1. Fuentes, A., Yoon, S., Park, J., & Park, D.S. (2020). Deep learning-based hierarchical cattle behavior recognition with spatio-temporal information. *Comput. Electron. Agric.*, 177, 105627.
2. Kim, N.Y., Kim, S.J., Jang, S.Y., Oh, M.R., Tang, Y., Seong, H.J., Yun, Y.S., & Moon, S. (2017). Behavioral characteristics of Hanwoo (*Bos taurus coreanae*) steers at different growth stages and seasons. *Asian-Australasian Journal of Animal Sciences*, 30, 1486 - 1494.
3. Ayadi, S., Said, A.B., Jabbar, R., Aloulou, C., Chabbouh, A., & Achballah, A.B. (2021). Dairy Cow rumination detection: A deep learning approach. *ArXiv, abs/2101.10445*.
4. Ryan, C., G'ueret, C., Berry, D.P., Corcoran, M., Keane, M.T., & Mac Namee, B. (2021). Predicting Illness for a Sustainable Dairy Agriculture: Predicting and Explaining the Onset of Mastitis in Dairy Cows. *ArXiv, abs/2101.02188*.
5. Halachmi, I., Guarino, M., Bewley, J.M., & Pastell, M. (2019). Smart Animal Agriculture: Application of Real-Time Sensors to Improve Animal Well-Being and Production. *Annual review of animal biosciences*, 7, 403-425.
6. Arazo, E., Aly, R., & McGuinness, K. (2022). Segmentation Enhanced Lameness Detection in Dairy Cows from RGB and Depth Video. *ArXiv, abs/2206.04449*.
7. Ryan, C., Gu'eret, C., Berry, D.P., & Mac Namee, B. (2020). Can We Detect Mastitis earlier than Farmers? *ArXiv, abs/2011.03344*.
8. Nguyen, C.H., Wang, D., Richter, K.V., Valencia, P., Alvarenga, F.A., & Bishop-Hurley, G. (2021). Video-based cattle identification and action recognition. *2021 Digital Image Computing: Techniques and Applications (DICTA)*, 01-05.
9. Jocher, G.R., Stoken, A., Borovec, J., NanoCode, Chaurasia, A., TaoXie, Changyu, L., Abhiram, Laughing, tkianai, yxNONG, Hogan, A., lorenzomammana, AlexWang, Hájek, J., Diaconu, L., Marc, Kwon, Y., Oleg, wanghaoyang, Defretin, Y., Lohia, A., ah, M., Milanko, B., Fineran, B., Khromov, D.P., Yiwei, D., Doug, Durgesh, & Ingham, F. (2021). ultralytics/yolov5: v5.0 - YOLOv5-P6 1280 models, AWS, Supervise.ly and YouTube integrations.

10 M.F. Nasir et al.

10. Helwatkar, A., Riordan, D., & Walsh, J. (2014). Sensor Technology For Animal Health Monitoring. *International Journal on Smart Sensing and Intelligent Systems*, 7, 1 - 6.
11. Tuyttens, F.A., Molento, C.F., & Benaissa, S. (2022). Twelve Threats of Precision Livestock Farming (PLF) for Animal Welfare. *Frontiers in Veterinary Science*, 9.
12. Mahmud, M.S., Zahid, A., Das, A.K., Muzammil, M., & Khan, M.U. (2021). A systematic literature review on deep learning applications for precision cattle farming. *Comput. Electron. Agric.*, 187, 106313.
13. Zheng, Z., Zhang, X., Qin, L., Yue, S., & Zeng, P. (2023). Cows' legs tracking and lameness detection in dairy cattle using video analysis and Siamese neural networks. *Computers and Electronics in Agriculture*.
14. Borchers, M., Chang, Y.M., Tsai, I.C., Wadsworth, B.A., & Bewley, J.M. (2016). A validation of technologies monitoring dairy cow feeding, ruminating, and lying behaviors. *Journal of dairy science*, 99 9, 7458-7466.
15. Liang, D., Arnold, L.M., Stowe, C.J., Harmon, R.J., & Bewley, J.M. (2017). Estimating US dairy clinical disease costs with a stochastic simulation model. *Journal of dairy science*, 100 2, 1472-1486.
16. Han, S., Fuentes, A., Yoon, S., Park, J., & Park, D.S. (2022). Multi-Cattle tracking with appearance and motion models in closed barns using deep learning. *Korean Institute of Smart Media*.
17. Singh, C. (2021). Applications and Challenges of Deep Learning in Computer Vision. *Health Information Science*.

Convolutional Neural Networks with Particle Swarm Optimization: A Reliable Method for SARS-CoV-2 Detection in X-Ray Images

Waqas Ahmed

Department of Computer Science University of Engineering and Technology Taxila, Pakistan
waqas1sep@yahoo.com

Atif Ali

Research Management Centre (RMC), Multimedia University, Cyberjaya 63100 Malaysia.
dralexaly@gmail.com

Farrukh Lodhi

Department of Computer Science University of Engineering and Technology Taxila, Pakistan
rulkodhi@yahoo.com

Waqar Ahmed

Department of Computer Science University of Engineering and Technology Taxila, Pakistan
waqar_casian@yahoo.com

Naveed Baloch

Department of Computer Engineering University of Engineering and Technology Taxila, Pakistan
naveedbaloch@uettaxila.edu.pk

Abstract—The coronavirus pandemic, commonly known as COVID-19, set free in Wuhan, China, has travelled globally. This pandemic is quickly spreading all around the world with physical contact and air. The first step to prevent its spread is accurately diagnosing the presence of disease in patients. A much faster and time-saving real-time method of diagnosing this disease is required as traditional methods, including Reverse Transcription Polymerase Reaction, are inefficient and defective. To start, we would gather and organize image data through data synthesis. After the initial step, we would train the proposed 2D-CNN, known as the 2 Dimension Convolutional Neural Network model and test it accordingly. To do this, the image data will be divided into three syndicates, each containing several photos for training, testing, and validation. The features are then classified using the Particle Swarm Optimization (PSO) technique and smoothed using Principle Component Analysis. A data support vector machine (SVM) is employed to classify picture data further. This method delivers the results required to diagnose the illness in several datasets.

Keywords: 2D-Convolutional Neural Network, Swarm Optimization, Coronavirus, Principle Component Analysis, Support Vector Machine.

I. INTRODUCTION

Several persons in China's Guangdong province contracted a severe acute respiratory syndrome virus in February 2013. Eventually, SARS was found in 8000 individuals across 26 countries, and 774 SARS-related deaths were documented by the World Health Organization (WHO). A similar incident involving the Middle East respiratory syndrome virus occurred in September 2012. MERS-CoV was found in 2494 confirmed cases of illness and 858 fatalities. SARS and MERS are less significant than the most recent CoV outbreak regarding human health. Pneumonia-like illnesses with un-

known causes began to appear in Wuhan, China, in November 2019, killing hundreds of people in the first few weeks. The International Committee on the Taxonomy of Viruses designated the SARS-CoV-2 virus as the cause of Coronavirus Disease 2019 (COVID-19) in the early months of 2020. (ICTV). Coronavirus, a serious respiratory illness that spreads when an infected person coughs or sneezes, is beginning to spread in Wuhan, the capital city of China's Hubei province. Common coronavirus symptoms include cough, sore throat, fever, exhaustion, and shortness of breath.

According to a study [1], the basic cause of COVID-19 spread is physical contact and via air through droplets coming out of the mouth during breathing; thus, the primary task in preventing disease spread is to identify people infected with the virus, isolate them, and thus break the virus's spreading chain. A common laboratory test used to identify the coronavirus is the WHO-recommended reverse transcription polymerase chain reaction (RT-PCR) (WHO). However, the RT-PCR test is unreliable, and its results are time-consuming; it takes two days, and during those two days, the patient can be a source of disease propagation. When a person's test results are awaited, he may spread the virus since he is unsure of the outcome, increasing the chance of disease transmission. As a result, a more rapid and dependable method of COVID-19 detection is necessary to replace standard RT-PCR. Until an appropriate medicine is developed, the only option to control the pandemic is to isolate the individual who is ill and provide him with proper medication so that he can recover in isolation.

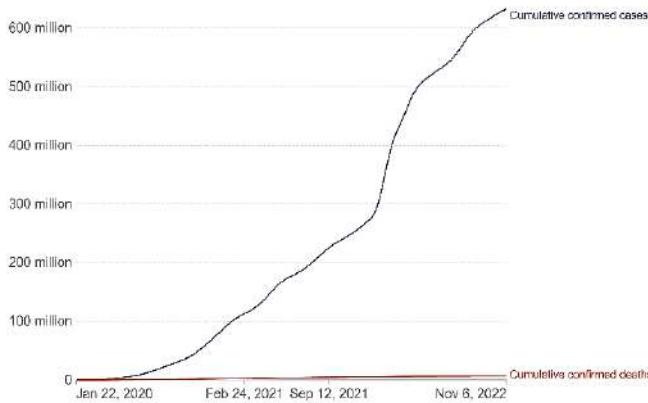


Fig. 1. Corona Virus Cases vs Death Rate

The Pandemic affected millions of people across the world. The COVID-19 pandemic has affected the entire world, with over 113.13 Million people and 2.50 Million fatalities so far (as per reports of 25th February 2021). According to the death rate in western countries and Europe, countries are more than compared to other countries. The mortality rate of coronavirus across the world is approx. 2%. The death rate in western countries is comparatively high, especially in America, Brazil and European countries. Out of 2.5 million casualties, 0.5 million people died in America, which is very alarming, especially for developed countries like America and Brazil. Isolation / Quarantine and taking care of patients are key to being safe from this pandemic disease. China has overcome and stopped the spreading of coronavirus by lockdown and social distancing, especially by isolating suspicious patients. This disease is dangerous because it is easily transmissible by contacting the affected people. COVID-19 is a dangerous disease for mostly those with diabetes, heart disease, lung disease and kidney/liver disease and also people who do not have a good immune system to fight against this pandemic disease. Many countries have imposed border restrictions, flight restrictions, social distancing and awareness programs for social distancing and forced lockdowns to prevent the spreading of coronavirus and decrease the rate of positive patients.

Main Contributions

A brief summary of our research is as follows: we would propose a method which can replace the inefficient RT-PCR test worldwide.

This paper proposes using the 2-D CNN method for feature extraction and using data pre-processing to arrange the data by grouping/clustering data that will help further diagnose the disease in several clusters of data. Then features are chosen by using Particle Swarm Optimization and other analysis approaches. To detect a disease from a large set of data, Support Vector Machine (SVM) is further used for classification. In compari-

son to the RT-PCR test, this method enforces effective and accurate results to detect the disease quickly.

To determine the relative effectiveness of this approach, the accuracy rates of genetic research and particle swarm optimization are compared to those of the earlier experiments. Our study would also concentrate on the proper model evaluation through dataset organization, using various techniques to arrange and correlate data to assess the accuracy of both the test and hybrid models.

II. LITERATURE REVIEW

To deal with epidemic disease, the researchers [3] have designed many techniques to detect for COVID-19 to get a better system. We have reviewed some of the systems to detect COVID-19 through X-Ray images. Early detection of COVID-19 through CT scan images using Harmony Search and Otsu threshold method. The region of interest is extracted from binary images to check the level of infection or disease. The CNN technique with VGG19 for image modalities and classification uses X-Ray, Ultrasound and CT-scan images. To deal with the challenge, the datasets [4] available for COVID-19 are used for pre-processing to test and develop deep learning techniques. Deep learning technology is used in healthcare applications, i.e., lung classification and thyroid diagnosis. Deep learning can also be used to diagnose the COVID-19 pandemic to reduce the spreading of COVID-19 worldwide. Deep architecture, also called COVID-Net model [5], is designed to detect COVID-19 with utmost taxonomy. In this method, the two datasets, which include positive COVID-19/pneumonia confirmed and normal X-Ray images, are used to identify the COVID-19 detected patients using deep learning techniques, i.e. CNN model for classification. This tested model gives results with an accuracy of up to 96.78%. In binary classification, this process has an accuracy of 98.08%, and in multi-class classification, it has an accuracy of 87.02%. In this method, feature extraction is done using a combined neural network, and disease detection from X-ray pictures is done using a long short-term memory (LSTM). The method uses deep neural networks [6] to detect coronavirus from X-Ray images. CNN is used to extract features, and SVM is an approach which classifies the virus-affected x-ray image. This method first uses pre-trained models for feature extraction. Then it uses SVM to classify and detect the virus from extracted features' x-ray images and give results with an accuracy of 95.38%. This tested model gives results by using ResNet50 with an SVM classifier. Using radiological images, i.e. X-Ray images, the coronavirus is detected using CNN and LSTM models and the Residual attention network approach; the accuracy for testing of COVID-19 is 98%. The model is designed using Residual Attention Network with feature extraction by using various models, i.e. VGG, mobilNET, and DenseNET.

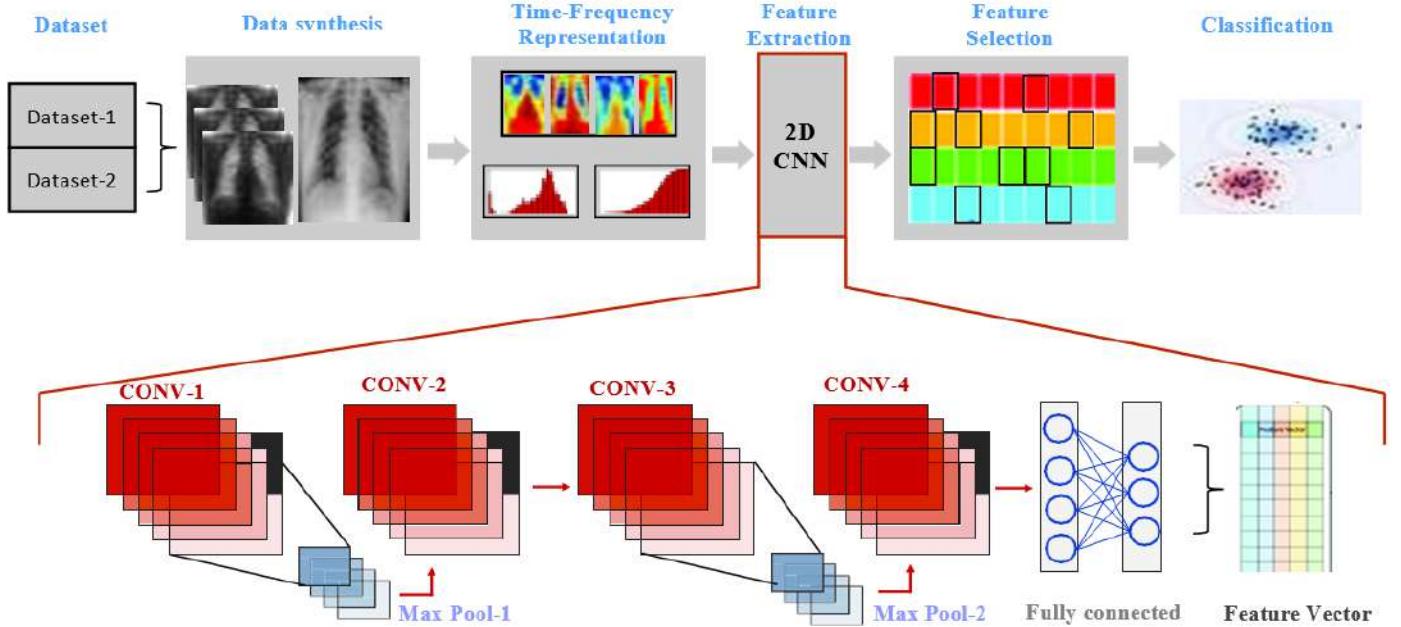


Figure 2. Framework that is suggested and 2D-CNN architecture

Using the convolutional CapsNet approach [1] to diagnose COVID-19 utilizing X-Ray pictures and capsule networks is suggested. The suggested method is used to identify coronavirus using binary classification (COVID-19 vs Not detected) and multiclass classification, with an accuracy of 97.24% for binary classification and 84.22% for multiclass classification (COVID-19 vs No disease found vs Pneumonia).

III. MATERIALS AND METHODS

A. Data Synthesis

Our work includes a critical process known as data synthesis. This task acts as a continual flow of information from the neural network. To begin, the dataset is partitioned and preprocessed to fit the model. This involves scaling, cropping, increasing quality, reducing noise, segmentation, and presenting the photos. The significance of this procedure is that raw photos can have a significant impact on biomedical image processing. One potential concern is that producers may "fabricate" the material to trick the discriminator into thinking their data belongs to the desired distribution. Any incorrect measurements are removed before image processing begins to avoid inaccuracies that could lead to misclassification.

B. Time-Frequency Representation that is Not Separable

The Non-separable Wavelet Transform (nSWT) [10] converts an image into a two-dimensional time-frequency spectrogram. This technique is frequently used in medical imaging to detect minor differences in pictures, identify changes in an X-ray image, and transform it into a spectral image. The X-ray image, initially in raw form,

is first divided into a series of features created from the primary (prototype) wavelet utilizing dilation and translation properties. The image is then analyzed using the Separable Wavelet Transform (SWT) with a manifold resolution scale factor, resulting in greater dimensions [10]. We use a non-focused filter and a non-separable wavelet transform (nSWT) with an unseparated sample (low resolution) to reduce the image to a single resolution. This transformation creates an picture with low resolution and a non-directional wavelet sub-image. The image is transformed for binary channel output using the equation below. Figure 3 depicts the spectrogram of the non-separable transform.

C. Feature Extraction

A 2D convolutional neural network is used to extract features in our suggested method (CNN). CNN is a widely utilised biomedical and machine learning method, allowing scientists and researchers to perform rapid recognition and categorization. CNN is an artificial intelligence technology for automatically identifying objects. In this study, we create a 2D CNN model with X-ray training data and testing data. The CNN receives the TFR images from the nSWT. The CNN model is then tested on a new dataset. We have validated our suggested model with GoogleNet and EfficientNet. The multi-layered neural network with 12 layers for feature extraction comprises four convolutional layers, two max-pooling layers, two dropouts, and two fully connected layers. Table I shows all the parameters, including the output image size for each layer. This network's input layer is equipped with a 224x224 input map. Fig 2 depicts the general tiered organization of a 2D Convolutional Neural Network. To prevent overfitting, the output vector (features data) contains 1000 attributes with a 50% exclusion layer for each layer. As a result, the final feature vector is 930x1000.

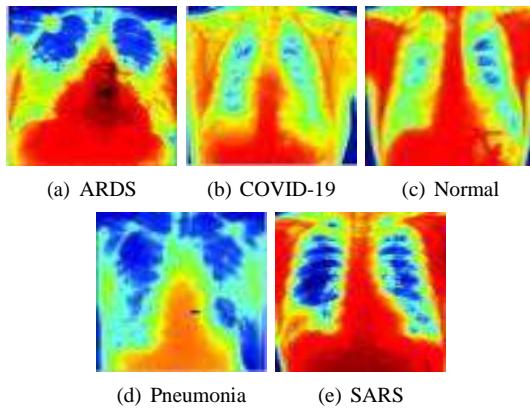


Fig. 3. Different X-ray Images Represented by Time-Frequency [1]

TABLE I
PROPOSED 2D-CNN MODEL LAYER PARAMETERS

Layer	Name	Size	Ker-nel	Pa-rameter	Stride
Layer1	Input	224 × 224	-	-	1
Layer2	Conv-1	32 × 128	3	576	2
Layer3	MaxPool-1	-	-	-	1
Layer4	Conv-2	32 × 128	128	73,728	2
Layer5	Conv-3	64 × 64	128	294,192	1
Layer6	MaxPool-2	-	-	-	2
Layer7	Conv-4	16 × 16	512	1,179,648	1
Layer8	FC-1	1000	1000	2,097,152	-
Layer9	Dropout-1	-	50%	-	-
Layer10	FC-2	1000	1000	1,362,712	-
Layer11	Dropout-2	-	50%	-	-
Layer12	Output	1 × 1000	-	42,000	-

D. Selection and Reduction of Features

1) *Particle Swarm Optimization (PSO)*: This technique performs intelligent function grouping [12]. Jams Kennedy developed PSO in 1995, and [13] offered an upgraded binary PSO (iB-PSO) used for cancer staging. In our paper, we apply PSO in a similar mode to classify COVID-19. PSO is used to lower the size/dimensions of the output by selecting higher-quality and less computationally intensive functions. PSO is an algorithm for collective intelligent search. This search is carried out using a set of randomly generated probable solutions. A swarm is a collection of potential solutions; each potential solution is referred to as a particle. The function's quality is dictated by the particles' social and cognitive learning rates. These velocities are used to assess the function's quality. As a result, the rest of the feature values are ignored, yielding a low-size dimensional feature vector.

2) *Principle Component Analysis (PCA)*: While the Particle Swarm Optimization (PSO) method intelligently clusters functions, the Principal Component Analysis (PCA) [14] technique is used to smooth and minimize characteristics. A sequence of data collected for a correlation variable, such as one with fluctuating numerical values, is transformed into a collection of mono values, which are non-correlated variables, through the analytical process known as PCA. The term "main components" refers to these values. The goal of this

conversion is to increase the likelihood that the limit will change with each succeeding component. Therefore, the key component that was found first would have the most probability of altering. A resulting vector, or set of non-correlated or orthogonal criteria, is what this method produces. Furthermore, the PCA method is sensitive to the initial input variable's magnitude.

E. Classification

1) *Linear Support Vector Machine (LSVM)*: A support vector machine (SVM) classifier with linear kernel functions is used in this classification approach [15]. LSVM is more effective when the number of classes is limited, like in our class 3 and 5 data. The LSVM classifier works well with the 2D CNN functions. Because fewer calculations are required, the selected feature vector may be detected quickly. The classifier's results outperform the usual COVID-19 detection technique.

2) *k- Nearest Neighbor (k-NN)*: The second classifier employed in our work for COVID-19 classification is the k-Nearest Neighbor (k-NN) method [15]. Similar to how SVM uses a linear kernel to classify data, k-NN does the same. The distance between the feature values of each class is calculated using the selected features, and the feature value establishes the feature class. Figure 5 displays the k-NN classifier's classification performance. At k=2, classification performance is at its best.

3) *Naive Bayes*: A Naive Bayes classification [16] is a strategy that integrates numerous predictive models to build a new one. Many names know Bayes models, e.g. simple Bayes, naïve Bayes and independent Bayes. It is extensively employed in machine learning models for pattern identification. It employs a probability-based ranking to determine the most powerful assumptions of dependency between attribute values. In our work, we validated the proposed strategy using a set and obtained better results than a single model.

IV. EXPERIMENTS AND RESULTS

A. Dataset

Datasets Da-taset-1 and Dataset-2 from Kaggle [17] are the publicly accessible datasets used in this research. Approximately 6,000 radiographs of healthy volunteers, patients with bacterial pneumonia, and patients with viral pneumonia are included in Dataset-1. The photos are divided 80/20 into training, testing, and validation folders, with dataset class subfolders. The three categories of pneumonia in Dataset-1 are typical bacterial pneumonia, viral pneumonia, and mixed pneumonia. 930 5-degree radiographs of patients with Covid19, SARS, typical pneumonia, and ARDS are included in Dataset 2. Unlike the RT-PCR test, both data sets are recognized or labelled initially.

TABLE II
PERFORMANCE METRICS USING VARIOUS MODELS

Model	Feature Dimension	Sensitivity	Parameters	No of Layers	F-1 score	Training time	Recall
AlexNet	930 × 4096	0.89	61M	8	0.90	22.08	0.81
2D-CNN (Proposed)	930 × 1000	0.92	42K	12	0.99	3.41	0.84
2D-CNN-PSO (Proposed)	618 × 1000	0.92	42K	12	0.91	2.82	0.87
GoogleNet	930 × 4096	0.83	7M	22	0.86	12.64	0.79
2D-CNN PSO PCA (Proposed)	250 × 412	0.92	42K	12	0.98	1.30	0.88

B. Network Modeling and Training

The suggested 2D CNN model is trained on a device with an AMD GTX 1080 graphics processing unit using Windows 10. Network formation time is sped up through feature selection and reduction. Thus, the selected feature can perform learning processing using a straightforward corei3 procedure. The suggested network's error rate during validation is shown in Figure 4. It has been shown that the error decreases as the number of epochs increases.

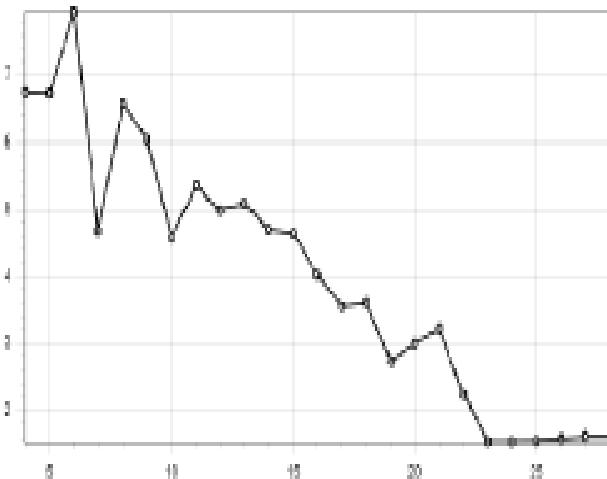


Fig. 4. Testing Error Rate of 2D-CNN

C. Classification Performance

1) *Evaluation Metrics:* To prevent the over-fitting issue, the dataset is partitioned into M randomly chosen portions, each of equal volume. The technique is tested on the remaining part after training on M-1 pieces. The total metric is determined by adding up the measurements from M training cycles. The effectiveness of the training dataset and the training times obtained through cross-validation are shown in Table II.,

Using a confusion matrix, we calculated precision, accuracy, recall, specificity and f1-score. The above parameters are computed using the keywords listed below. We calculate the above parameters using the keywords listed below

$K_{positive}$ means disease is predictable and is present.

$P_{positive}$ means disease is predictable and is not present.

$K_{negative}$ means disease is not predictable and is not present.

$P_{negative}$ means disease is not predictable and is present.

The accuracy of the training set is given by

$$Acc = \frac{K_{negative} + K_{positive}}{negative + positive}$$

The precision of the training set is given by

$$Prec = \frac{K_{positive}}{F_{positive} + T_{positive}}$$

Specificity is taken from

$$Spec = \frac{K_{negative}}{K_{positive} + K_{negative}}$$

Recall of the training set is estimated by

$$Recall = \frac{K_{positive}}{P_{negative} + K_{positive}}$$

f1-score is provided by

$$f1\text{-score} = 2 \times \frac{Prec - Recall}{Prec + Recall}$$

The parameters above can be used to assess the model's performance. LSVM, NB and k-NN perform grouping by cross-examining the extracted features' results with characteristics from already proposed groupings and classification techniques, primarily using cross-validation because it is highly dependable.

D. Comparison with the existing method

To confirm the validity of our proposed network, we compare it to existing COVID-19 detection techniques and a trained neural network for benchmarking and validation. As shown in table II, our proposed feature selection combined with the CNN method for identifying COVID-19 surpasses recent corona detection methods. When comparing the proposed method to a pre-trained neural network, namely the AlexNet image classification algorithm, it is discovered that GoogleNet, which has the ability of training a large - scale dataset and is capable of grouping images into 1000 objects, provides us with the accuracy shown in table II.

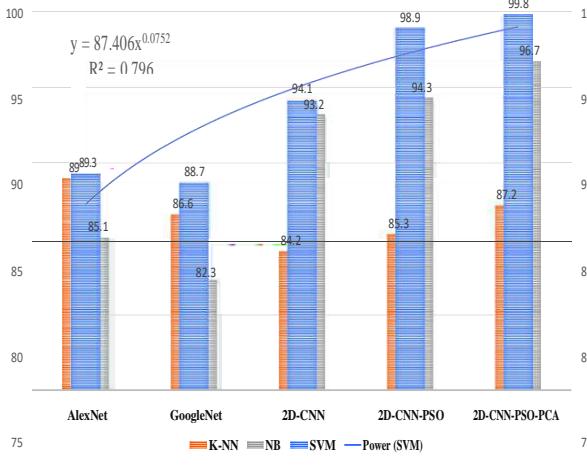


Fig. 5. Classification Performance

TABLE III

COMPARISON WITH RECENT COVID-19 DETECTION METHODS

Technique	Classification	Accuracy %
Binary Classification [3]	DNN	87.02
LSTM [4]	CNN	99.4
Transfer Learning [5]	CNN	96.78
Convolutional CapsNet [6]	NANN	95.38
ResNet50 [8]	DNN	95.33
2D-CNN-PSO-PCA [Proposed]	SVM	99.80

Different techniques are used with different classification methods for detecting the coronavirus in the body using X-rays images. The accuracies of the models are shown in table III

V. CONCLUSION

The WHO RT-PCR test for detection is time-consuming and unreliable, therefore, a faster and more suitable substitute for particle swarm optimization is one technique for collecting and detecting data with the help of genetic research. As the recent Corona Virus (COVID-19) epidemic is causing alarm worldwide owing to its rapid spread, the only way to manage the pandemic until a solution is discovered is to segregate those who have the sickness. This study proposes a more accurate and efficient COVID-19 diagnosis than already in use methods. The data is grouped, extracted, and put into the precise model chosen; the model is evaluated, tested, and confirmed using three separate sets of data before being used for feature selection utilizing PSO methods. This data is analysed by classification, using SVM and ensemble. In contrast to the RT-PCR test, which can take up to two days and raises the risk of disease transmission and infection among health-care workers, this process is extremely trustworthy and has a 99.80% overall accuracy.

REFERENCES

- [1] Altan, Aytaç, and Seçkin Karasu. "Recognition of COVID-19 disease from X-ray images by hybrid model consisting of 2D curvelet transform, chaotic salp swarm algorithm and deep learning technique." Chaos, Solitons & Fractals 140 (2020): 110071.

- [1] Novel, Coronavirus Pneumonia Emergency Response Epidemiology. "The epidemiological characteristics of an outbreak of 2019 novel coronavirus diseases (COVID-19) in China." Zhonghua liu xing bing xue za zhi= Zhonghua liuxingbingxue zazhi 41, no. 2 (2020): 145.
- [2] Ozturk, Tulin, Muhammed Talo, Eylem Azra Yildirim, Ulas Baran Baloglu, Ozal Yildirim, and U. Rajendra Acharya. "Automated detection of COVID-19 cases using deep neural networks with X-ray images." Computers in Biology and Medicine (2020): 103792.
- [3] Islam, Md Zabirul, Md Milon Islam, and Amanullah Asraf. "A combined deep cnn-lstm network for the detection of novel coronavirus (covid-19) using x-ray images." Informatics in Medicine Unlocked (2020): 100412.
- [4] Apostolopoulos, Ioannis D., and Tzani A. Mpesiana. "Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks." Physical and Engineering Sciences in Medicine (2020): 1.
- [5] Toraman, Suat, Talha Burak Alakus, and Ibrahim Turkoglu. "Convolutional capsnet: A novel artificial neural network approach to detect COVID-19 disease from X-ray images using capsule networks." Chaos, Solitons & Fractals 140 (2020): 110122.
- [6] Sethy, Prabira Kumar, and Santi Kumari Behera. "Detection of coronavirus disease (covid-19) based on deep features." Preprints 2020030300 (2020): 2020.
- [7] Sethy, Prabira Kumar, and Santi Kumari Behera. "Detection of coronavirus disease (covid-19) based on deep features." Preprints 2020030300 (2020): 2020.
- [8] Lalmuanawma, Samuel, Jamal Hussain, and Lalrinfela Chhakchhuak. "Applications of machine learning and artificial intelligence for Covid-19 (SARS-CoV-2) pandemic: A review." Chaos, Solitons & Fractals (2020): 110059.
- [9] Ali, A., Hafeez, Y., Ali, S., Hussain, S., Yang, S., Malik, A. J., & Abbasi, A. A. (2021). A Data Mining Technique to Improve Configuration Prioritization Framework for Component-Based Systems: An Empirical Study. Information Technology and Control, 50(3), 424-442.S. Mallat, "A theory for multiresolution signal decomposition: The Wavelet representation," IEEE Trans. Pattern Anal. Machine Intell., vol. 11, pp. 674–693, July 1989.
- [10] Ali, A., Jadoon, Y. K., Dilawar, M. U., Qasim, M., Rehman, S. U., & Nazir, M. U. (2021, April). Robotics: Biological Hypercomputation and Bio-Inspired Swarms Intelligence. In 2021 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA) (pp. 158–163). IEEE..
- [11] Ali, A., Hafeez, Y., Hussain, S. M., & Nazir, M. U. (2020, January). BIO-INSPIRED COMMUNICATION: A Review on Solution of Complex Problems for Highly Configurable Systems. In 2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET) (pp. 1-6). IEEE.
- [12] Granato, Daniel, Ja nio S. Santos, Graziela B. Escher, Bruno L. Ferreira, and Rube'n M. Maggio. "Use of principal component analysis (PCA) and hierarchical cluster analysis (HCA) for multivariate association between bioactive compounds and functional properties in foods: A critical perspective." Trends in Food Science & Technology 72 (2018): 83-90.
- [13] Palaniappan, R.; Sundaraj, K.; Sundaraj, S. A comparative study of the SVM and K-nn machine learning algorithms for the diagnosis of respiratory pathologies using pulmonary acoustic signals. BMC Bioinform. 2014, 27, 15–223.
- [14] Yoshimasa, T; Jun'ichi T. Training a naive bayes classifier via the EM algorithm with a class distribution constraint. In Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4 (CONLL '03). Association for Computational Linguistics, USA, 2003, 127–134.
- [15] Cohen, Joseph Paul, Paul Morrison, and Lan Dao. "COVID-19 image data collection." arXiv preprint arXiv:2003.11597 (2020).

Multi-region based radial GCN algorithm for real-time action recognition

Han-Byul Jang^{1[0000-0003-4815-1513]} and Chil-Woo Lee^{1[0000-0002-3391-1631]}

¹ Chonnam National University, Yongbong-ro 77, Buk-gu, Gwangju, Republic of Korea

Abstract. The MRGCN algorithm [1] uses simple optical flow and image gradient instead of skeleton data as inputs for deep learning, so that implementation of an action recognition system used in the real world can be possible. However, to be applied to various application systems, real-time processing is absolutely necessary. For example, in the case of an intelligent surveillance system that responds to crimes or accidents occurring on the street, real-time processing is essential. In this paper, we describe the parallel processing algorithm developed for the implementation of real-time behavior recognition system using MRGCN and the neural network structure that has variability according to the sampling time of input data. By analyzing the processing modules constituting the algorithm and executing the modules that can be processed simultaneously, it was possible to obtain a processing speed improved by nearly 50% compared to the existing sequential processing method.

Keywords: Human action recognition, Graph convolutional network, Real time system.

1 Introduction

If the action of a person appearing in a video can be recognized, it is able to be applied to various intelligent services. Since the purpose of such services is to provide immediate results without human intervention, real-time algorithms must be implemented. In particular, in an intelligent surveillance system, it is very important to implement a real-time behavior recognition algorithm with fast processing speed, as it is a key function to identify risky behaviors and quickly respond to them using behavioral recognition. For example, in order to find out a crime or accident situation in real time from a camera monitoring the street as shown in Figure 1, a recognition algorithm that can quickly process image data acquired through CCTV is essential.

Recent action recognition studies generally use deep learning approach. Deep learning makes it possible to effectively process complex data that is difficult to analyze through existing pattern recognition methods, so the recognition rate of algorithms and the performance of application services are further improved by using large amounts of data as inputs for deep learning. Recently, high-performance action recognition algorithms using GCN, the latest deep learning technique for learning data having a graph structure, have been announced, and they generally use skeleton joint coordinates as

learning data. However, the skeleton data has a problem in that it is difficult to obtain accurate values and so it is impossible to use in actual application services. In the previous study of this study [1], we proposed MRGCN algorithm using new input data that combines easily obtainable optical flow and image gradient instead of skeleton data. MRGCN achieved a higher recognition accuracy than previous research achievements through effective neural network learning by applying graphs tailored to the characteristics of new input data.

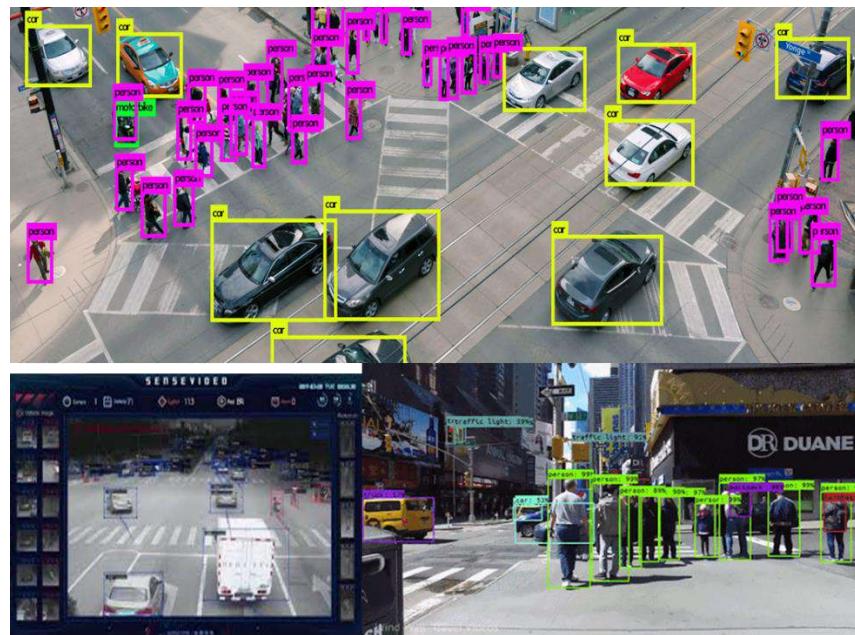


Fig. 1. Automatic surveillance system monitoring the street.

MRGCN is good to use in application services because it uses easily obtainable image information instead of skeleton data and has high performance, but it does not have a structure suitable for real-time processing. This paper describes a real-time algorithm that improves the structure of MRGCN to be suitable for application services. Action recognition application services are often not equipped with high-performance computers, but lack of processing capability can delay data acquisition timing and generate low frame rate data. Therefore, real-time action recognition algorithms should be tested so that they can be used even at low frame rates and improved to save processing time as much as possible. In this study, after re-training MRGCN according to input data of low frame rating, it was confirmed that proper recognition rate could be maintained up to 10 fps (frame per second) by experiment. In addition, in order to obtain faster processing speed, the modules of the algorithm were analyzed, and the structure was changed to parallelize the modules that could be processed simultaneously. As a result of applying the parallel processing structure, the processing speed could be improved by about 50%.

The order of this paper is as follows. First, this section introduces the outline of the paper. Section 2 examines the existing action recognition studies. Section 3 describes the structure and characteristics of MRGCN, the previous study of this paper. Section 4 proposes improvements of the MRGCN algorithm for real-time action recognition. Finally, Section 5 describes the conclusions and the factors that need improvement through future research.

2 Related Studies

Recently, most studies of human action recognition use deep learning approach. In the past, action recognition studies often used classic pattern recognition methods, but deep learning-based research has become mainstream because deep learning using large amounts of data has made it possible to analyze complex information inherent in action. Various neural network models such as CNN, LSTM, RNN, and GCN are used for action recognition. For example, [2] uses LSTM and CNN to classify feature data obtained from skeleton data with LSTM while performing CNN classification using data transformed into map images. The paper [3] defines the human body into five parts to generate input data of RNN. In each part, the skeleton data is processed to feature data, which is then input into the RNN to classify the actions. [4] uses a spatial-temporal LSTM algorithm using optimized skeleton data, and [5] uses TCN (Temporal Convolutional Network), a CNN method that processes feature information extracted from video images. In addition, [6] published by Deep Mind combined RGB image and optical flow image and used it as input data and achieved good recognition rate by using several deep learning techniques in combination.

Recent action recognition studies generally use GCN model. Since GCN can perform neural network learning using data that has a graph structure, it is very suitable for analyzing skeleton data expressed in graphs in the structure of human body joints. In [7] and [8], the authors introduced GCN into action recognition research for the first time and showed improved results than existing methods. The ST-GCN model presented in [7] achieved a high recognition rate by using a graph of skeleton data continuously connected over time. Since then, many studies have been published that improved ST-GCN. For example, [9] improves the recognition rate by dividing the skeleton data into four parts according to the composition of the human body. In [10], the authors presented AS-GCN (Actional structural GCN) that can better reflect the feature and the structure of actions. And [11] proposed a more mathematically optimized method GR-GCN (Graph regression based GCN) model.

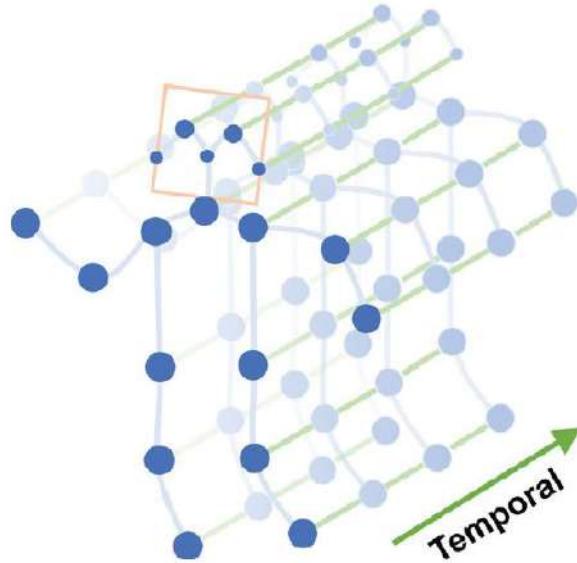


Fig. 2. A graph design same with the human joint connection form in ST-GCN [7].

The action recognition studies discussed above generally use skeleton data that records changes of coordinates of human joints as input data of the neural network. Since skeleton data contains well the context information of human body movement, it can be effectively used for action recognition and is particularly suitable when it uses the GCN algorithm with the graph shown in Figure 2. However, there is necessary conditions that special hardware such as Kinect [12] or posture recognition software such as OpenPose [13] must be used to obtain the accurate coordinate of the joints. Therefore, there is a problem in that it is difficult to acquire accurate skeleton data in a field where stable conditions are not be satisfied.

MRGCN [1], a previous study of this study, can learn and recognize using new input data that can be easily acquired from images instead of skeletons, while using the highly efficient GCN algorithm. In detail, MRGCN achieved higher recognition accuracy than that of the existing GCN-based action recognition by using a feature vector that combines optical flow and image gradient that was obtained from an input image through simple processing. This paper shows that this MRGCN algorithms can be effectively applied to action recognition application services by improving them to be suitable for real-time processing.

3 The Structure of MRGCN

3.1 The Overall Structure of MRGCN

Skeleton data, which is a stream of human 3D joint coordinates, is used as general input data for action recognition. To acquire the skeleton data, special hardware such as Kinect [12] or professional posture recognition software such as OpenPose [13] must be used. However, it is difficult to apply such special hardware or software when the data acquisition environment is poor because accurate results can be obtained only when the very restrictive environment is provided. Therefore, in [1], the previous study of this paper, we proposed MRGCN, a more effective action recognition algorithm using optical flow and image gradients, which can be simply calculated by processing 2D images instead of skeleton data.

The overall sequence of the MRGCN algorithm is shown in Figure 3. As shown in this figure, MRGCN first calculates the optical flow and image gradient in the person's area appearing in the input image at (a-b) stages, and then converts it into compressed data of HoFG (histogram of flow and gradient), code, mean, and standard deviation at (c) stage. The converted data is used for network training and recognition as an input to the neural network at (d) stage. In the recognition process, the recognition result is finally output through the classifier stage (e).

3.2 The Generation Method of Input Data

As described above, MRGCN uses optical flow and image gradient as input. The reason for combining the two data is that it is assumed that action can be described by combining human movement and shape information. Optical flow is good for expressing movement information, and image gradient is good for representing shape information.

After the two kinds of data are generated as 32-dimensional histograms based on the direction using the HoG (histogram of gradient) algorithm [14], they are reduced and converted into 3-dimensional respectively, total 6-dimensional vectors to enable fast neural network learning. The 3-dimensional vectors are the HoFG code obtained using Equations (1) and (2), and the mean and standard deviation of the histogram. In the Equation (1), the Mode value is sequentially obtained at the histogram index with the histogram value greater than the neighboring value. Next, the HoFG Code is calculated by applying the obtained Mode values to Equation (2). In this case, C is a constant using 20.

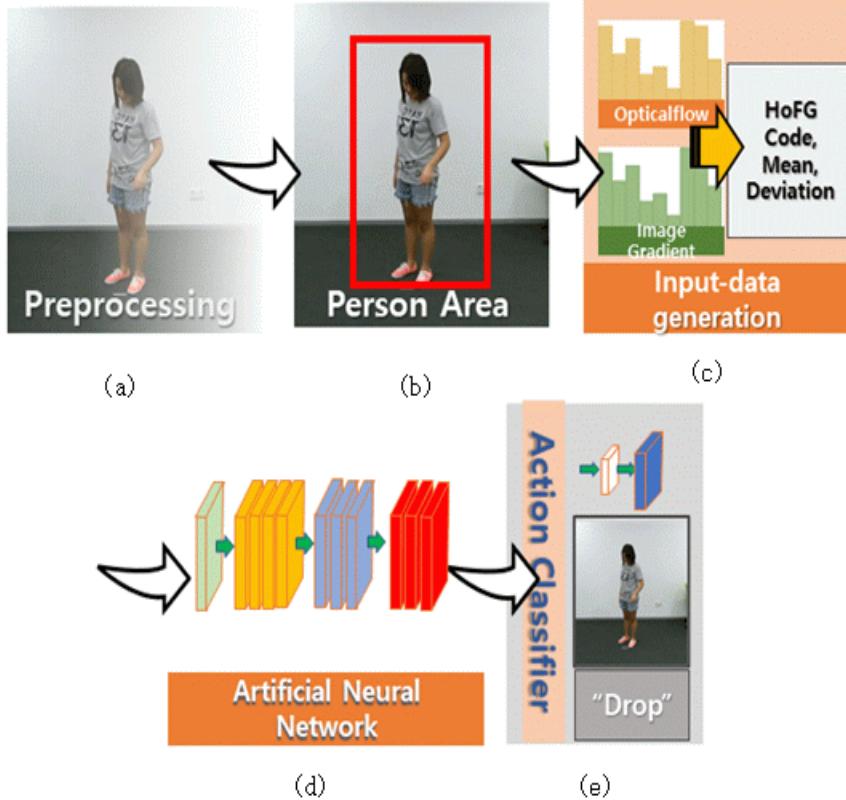


Fig. 3. Overall flowchart of MRGCN algorithm (The source of flowchart is [1] and the input image is from [15]). (a) Preprocessing of input image stage, (b) Finding “Person area” stage, (c) Generation input data stage, (d) Applying input data to the MRGCN stage and (e) Classifying the result action stage.

$$\text{Mode}(\text{order}) = \begin{cases} (\text{bin index } i) + 1: & \text{if the bin is mode} \\ 0: & \text{otherwise} \end{cases} \quad (1)$$

$$\text{HoFG Code} = \sum_{\text{order}=1}^k \text{Mode}(\text{order}) * C^{k-\text{order}} \quad (2)$$

3.3 Graph Design of MRGCN

Studies that learn GCN using skeleton data generally use the same graph as the joint structure of the human body. Since this graph can directly reflects the context of human body’s movement, is very suitable for processing skeleton data. However, MRGCN

using new input data requires new graph has the design that matches the input data. Therefore, an artificially designed graph structure is used to efficiently apply the shape and the motion changes of the human body.

Figure 4 is the plan view of the graph designed for MRGCN. As shown in Figure 5, this graph is structured to process input data systematically by arranging local data acquisition regions defined as RoM (Region of motion) using three-layers. Each RoM can generate the data of one node in the graph, and when the entire graph structure connecting all RoM nodes is drawn on a surface, it looks like it is spreads out radially, so this algorithm is named multi-region based radial GCN algorithm(MRGCN). In addition, MRGCN was able to achieve Top1 recognition accuracy of 94.28%, higher than the 93.27% Top1 recognition accuracy of the existing ST-GCN algorithm, as a result of comparative recognition experiments for 10 actions.

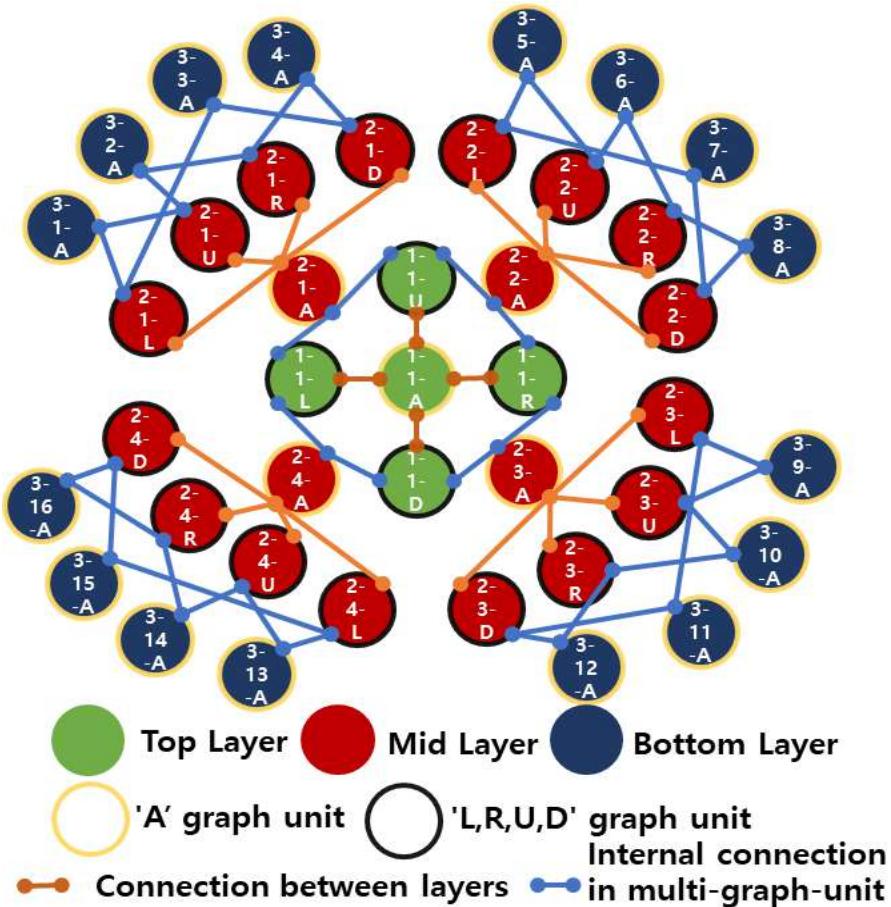


Fig. 4. Overall graph configuration of MRGCN in [1]. The graph has three-level layer structure (Top, Mid and Bottom Layer). Nodes in the graph have five types according to the characteristics of obtaining input data method (A: All, L: Left, R: Right, U: Up and D: Down).

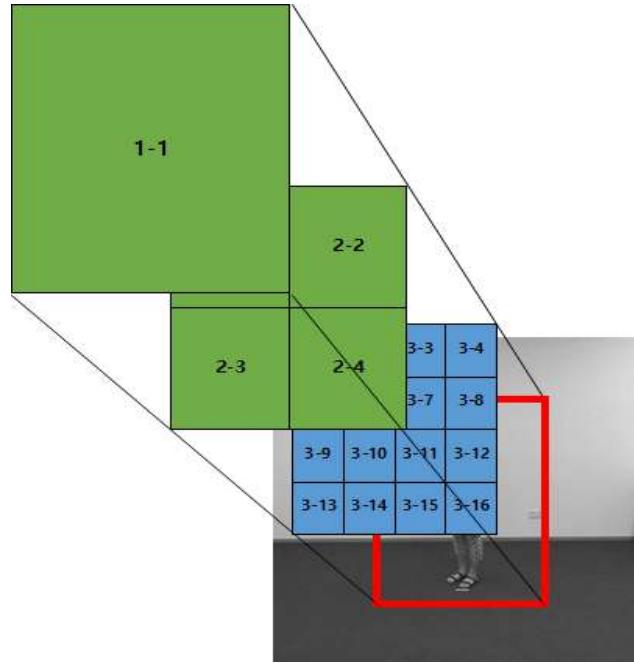


Fig. 5. Regions of data acquisition (RoM: region of motion) arranged using 3 layers in [1].

4 Improvement of MRGCN for Real-time Recognition

4.1 Improvement of Processing Speed using Multi-processing

An intelligent surveillance system, which is a representative application of action recognition, can be very important in modern society because it can automatically recognize dangerous situations such as crimes and accidents to prevent greater damage. However, in order to usefully use the action recognition algorithm in the application system, real-time processing must be possible.

As shown in Figure 3, the MRGCN algorithm has a structure in which each processing module is sequentially followed, so that the next stage can be executed only after passing through the previous stage. Therefore, faster processing speed can be obtained by processing modules that can be simultaneously processed in parallel. For example, if the “image acquisition module” is independently parallel processed, the waiting time to acquire the input image can be saved. In addition, “optical flow calculation”, “image gradient generation”, and “person area position estimation” can be simultaneously processed in the previous stage of input data generation, so a lot of processing

time can be saved. The flow chart of the improved algorithm by applying parallel processing is shown in Figure 6. As a result of improving the structure of the algorithm as shown in the flowchart, the processing speed could be increased by about 50% compared to the previous one.

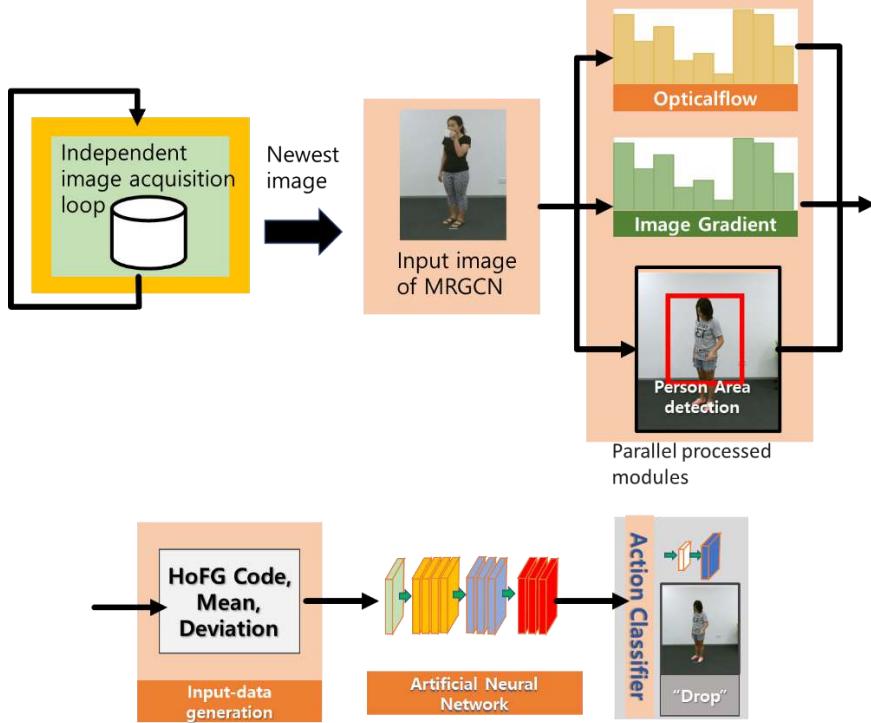


Fig. 6. Flowchart of MRGCN algorithm applying parallel processing.

4.2 Training a Neural Network with Frame Rate Change

Another issue to be considered for algorithm improvement is the quality of input data. The acquisition quality of input data may be degraded by many factors, such as the processing power of the application system, whether the using place is indoors or outdoors, or whether the lighting is bright or dark. In particular, the lack of processing power can cause lower frame rate of data acquisition because it affects the data acquisition timing. Therefore, the algorithm needs to be improved to generate accurate data even at low frame rate. Also, it is necessary to experiment whether recognition performance is maintained at low frame rate.

Since the previous MRGCN study was experimented with the data set, it used a fixed and accurate frame rate of 30 fps (frame per second). However, if the processing speed is reduced due to the large amount of calculation, the input is delayed, and recognition fails as a result. Among the input data of the MRGCN, since the optical flow is calculated between two frames before and after, data becomes unstable when the time interval between frames is too large. Therefore, even if the frame rate is low, the image

acquisition for calculating the optical flow must be adjusted to the 30 fps speed. To solve this problem, as shown in Figure 7, it is necessary to separate the image acquisition modules independently and process module in parallel. This problem can be easily solved by using parallel processing to acquire input images at 30 fps speed regularly regardless of the main processing loop.

In addition, the degradation of data quality due to the influence of the data acquisition environment has a great influence on the recognition result. In other words, a higher recognition rate can be obtained for an image with high contrast and clear human motion. In this paper, among the previously used training dataset, actions with large motion and high contrast were separately classified and used for training. As a result, the types of actions in the learning process were configured to be more clearly distinguishable, so that more distinct experimental results could be obtained.

The newly constructed experimental data includes 10 actions from NTU RGB+D dataset [15], and the types of selected actions are shown in Table 1 below. And Table 2 shows the results of experiment with frame rates of 5 fps, 10 fps, 15 fps, and 30 fps for the selected action. As shown in the table 2, the performance drop of MRGCN is larger than that of ST-GCN using skeleton data, but it maintains the same level of performance up to 10 fps or more. However, at a lower frame rate of 5 fps, there is a explicit performance degradation, and judging from this, it can be seen that the input data of MRGCN has a characteristic that the context of the action is relatively easy to miss at low frame rate.

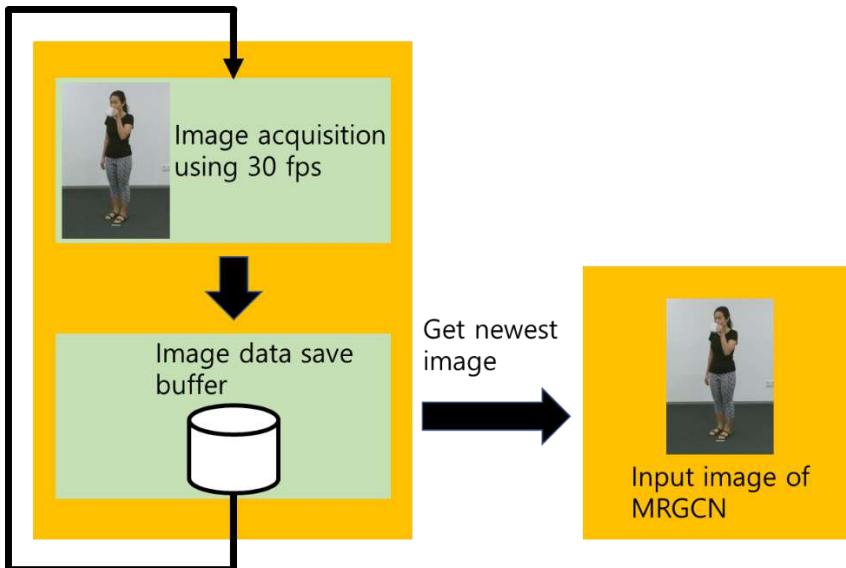


Fig. 7. Independent execution of image acquisition module.

Table 1. Selected 10 actions from NTU RGB+D dataset.

No.	Action	No	Action
1	Brush hair	6	Put on jacket
2	Throw	7	Take off jacket
3	Sit down	8	Kicking something
4	Stand up	9	Jump up
5	Reading	10	Staggering

Table 2. Comparison of recognition rates between ST-GCN and MRGCN according to frame rate

Frame rate	Top1 Accuracy of ST-GCN (Skeleton data)	Top1 Accuracy of MRGCN (6 channel data of optical flow and image gradient)
30 fps	90.31 %	93.99 %
15 fps	95.85 %	92.57 %
10 fps	93.66 %	92.86 %
5 fps	93.85 %	90.74 %

5 Conclusion

This paper describes an improved real-time MRGCN algorithm suitable for use in action recognition application systems. MRGCN is a high-performance GCN-based action recognition algorithm using new input data that can be easily acquired to solve the problem of performance degradation due to inaccurate data acquisition of existing skeleton data-based action recognition. However, to use it in an application system, it is necessary to improve the structure of the algorithm to have real-time processing capability. This is because it is possible to provide appropriate services only when the processing results can be obtained in real time in the application field of action recognition, such as an intelligent surveillance system.

To implement a real-time system, first, parallel processing was applied to the processing module of the MRGCN algorithm to achieve a speed improvement of about 50%. In addition, the input image acquisition module is independently executed in parallel so that accurate data can be calculated regardless of the variable data acquisition frame rate. To obtain clearer experimental results, the action of the experimental data set was configured to be clearly distinguishable, and as a result of verification experiments after learning at frame rates of 5 fps, 10 fps, 15 fps, and 30 fps, the recognition rate remained the same until about 10 frames.

To complete a more effective real-time action recognition algorithm in the future, more elements need to be improved. First, it is necessary to improve the training data of the neural network to better represent the action that occurred in the real environment. Currently used learning data is not optimized in a form suitable for the application system, so it is necessary to enhance the configuration of data to be more suitable for the purpose. In addition, robustness must be proven for various variables such as the

acquisition location of the input image, the intensity of lighting, and the type of background. In the input data, it is also necessary to introduce a method of adding a new type of data that can supplement information or augmenting and improving already generated data. As described above, we plan to improve MRGCN into a real-time algorithm that can be used in the real world by combining elements that can adapt to various environmental changes.

ACKNOWLEDGMENTS

This research is supported by Ministry of Culture, Sports, and Tourism (MCST) and Korea Creative Agency (KOCCA) in the Culture Technology (CT) Research & Development Program (R2020060002) 2022.

References

1. Jang, H. B., & Lee, C. W.: Multi-region Based Radial GCN Algorithm for Human Action Recognition. In *International Workshop on Frontiers of Computer Vision*, Springer, Cham, pp. 325-342. (2022).
2. Li, Chuankun, et al.: Skeleton-based action recognition using LSTM and CNN. *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, IEEE, pp. 585-590. (2017).
3. Du, Yong, Wei Wang, and Liang Wang: Hierarchical recurrent neural network for skeleton based action recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1110-1118. (2015).
4. Liu, Jun, et al.: Spatio-temporal lstm with trust gates for 3d human action recognition. *European conference on computer vision*, Springer, Cham. (2016).
5. Kim, Tae Soo, and Austin Reiter: Interpretable 3d human action analysis with temporal convolutional networks. *2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW)*, IEEE, pp. 1623-1631. (2017).
6. Carreira, Joao, and Andrew Zisserman: Quo vadis, action recognition? a new model and the kinetics dataset. *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4724-4733. (2017).
7. Yan, Sijie, Yuanjun Xiong, and Dahua Lin: Spatial temporal graph convolutional networks for skeleton-based action recognition. *Thirty-second AAAI conference on artificial intelligence*, pp. 7444-7452. (2018).
8. Li, Bin, et al.: Spatio-temporal graph routing for skeleton-based action recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, No. 01, pp. 8561-8568. (2019).
9. Thakkar, Kalpit, and P. J. Narayanan: Part-based graph convolutional network for action recognition. *arXiv preprint arXiv:1809.04983*. (2018).
10. Li, Maosen, et al.: Actional-structural graph convolutional networks for skeleton-based action recognition. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3590-3598. (2019).
11. Gao, Xiang, et al.: Optimized skeleton-based action recognition via sparsified graph regression. *Proceedings of the 27th ACM International Conference on Multimedia*, pp. 601-610. 2019.

12. Wikipedia, “<https://en.wikipedia.org/wiki/Kinect>,” Nov. 2021.
13. Cao, Zhe, et al.: OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. *IEEE transactions on pattern analysis and machine intelligence* 43.1, pp. 172-186. (2019).
14. Dalal, Navneet, and Bill Triggs: Histograms of oriented gradients for human detection. 2005 *IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, IEEE, Vol. 1. (2005).
15. Shahroudy, Amir, et al.: Ntu rgb+ d: A large scale dataset for 3d human activity analysis. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1010-1019. (2016).

ADVANCED MACHINE LEARNING TECHNIQUES TO IDENTIFY EMOTIONS IN TEXTS

Atif Ali

Research Management Centre (RMC), Multimedia University, Cyberjaya 63100 Malaysia.

dralexaly@gmail.com

Zulqarnain Farid

Dept of Criminology, University of Karachi, Pakistan

discovercrimegene@gmail.com

Abstract: This research aims to learn to classify short texts (opinions) generated on the social network Twitter according to their feelings, applying advanced machine learning techniques such as neural networks. In the first stage, text classification was explored using an LSTM neural network. In the current stage, some ways of representing texts are being analyzed to create a corpus of embedded words that will be used in other experiments.

Keywords: emotions, machine learning, neural networks, Twitter.

1. INTRODUCTION

A large amount of information in social media has led the scientific community to dedicate great efforts to analyzing, structuring, and processing this information. These media often express diverse opinions and feelings about society, products, services, politics, celebrities, etc. Companies, organizations, and governments are interested in knowing users' opinions about their activities. Digital marketing is mainly based on the opinions expressed on social networks.

Polarity detection in textual opinion is a widely researched task, especially for English. Identifying the emotion expressed in a the opinion is a less-researched task with ample research possibilities. In this regard, the frequently used approaches are supervised learning, which uses large amounts of text as input to the algorithms, and a dictionary of words associated with one or more emotions. The research group addressed these types of learning in the previous project.

II. Related Work

Deep learning approaches have demonstrated their ability to solve tasks related to natural language processing and artificial intelligence applications. Neural networks are effective when performing tasks of classifying texts [3]. [4], they apply a hybrid method with convolutional neural networks and recurrent neural networks to polarity classification in tweets. [5] uses a combination of artificial neural networks to recognize emotions in texts. The word embeddings technique consists of representing words as vectors of real numbers on which it is possible to carry out operations and obtain surprising results. [6] This is used to increase the effectiveness of classifying emotions.

Before the learning stage, it is necessary to carry out preprocessing actions to eliminate those characteristics that can produce noise in the following stages, for example:

- Tokenization and lemmatization

- Elimination of stopwords
- Elimination of images, links and references to users

Generally, social network users often use emojis to highlight what it wants to express, as a form of voice intonation or body language. In the previous project, it was shown that keeping emojis and hashtags is relevant. Therefore, they must be transformed into text. These types of elements, in general, are not considered in the works cited above.

Python is one of the most accepted programming languages by the scientific community. It is powerful and is characterized by its simplicity, open-source distribution and the possibility of integrating with multiple libraries. For text processing with Python, there are a variety of computer tools. In [7], some of them were analyzed, showing that Freeling and Stanford are the most reliable in terms of tokenization and grammatical labelling.

III. RESEARCH AND DEVELOPMENT LINE

This research project proposes to detect feelings expressed in texts, particularly textual opinions issued in a social network. The project is developed in the following stages:

- Review of the literature relevant to the problem of opinion and sentiment mining.
- Evaluation and comparison of deep learning techniques for text classification.
- Evaluation and comparison of other machine learning techniques for textual opinion classification.
- Development of a prototype for the classification of opinions.

The tweets captured for the previous project and others obtained last year totalled more than 150,000. Many of these tweets were discarded for containing only images, icons or text with little information for analysis.

IV. RESULTS OBTAINED/EXPECTED

In the first part of the project, experiments with neural networks for learning text classification began.

To combat the vanishing gradient, which occurs in Recurrent Neural Networks (RNN), LSTM (Long Short-Term Memory) networks arise, which are a special type of recurrent network [8]. The main characteristic of LSTMs is that the information can persist by introducing loops in the network diagram to

decide the next one [9]. Figure 1 shows the typical structure of this type of neural network.

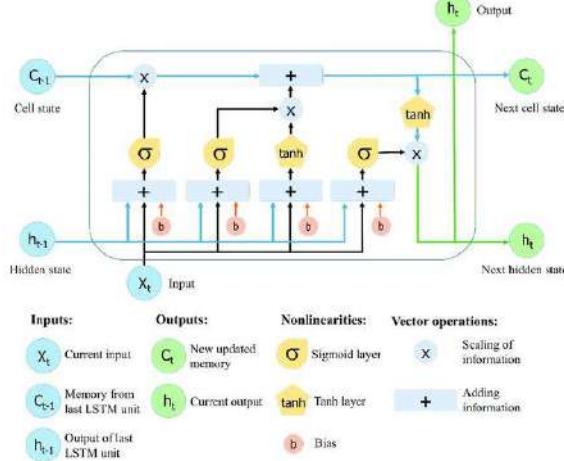


Figure 1. Long Short-Term Memory [9]

The main difference with traditional neural networks (RNT) is that they do not have the persistence (memory) of the previous data. An RNT cannot use its reasoning about previous events to decide on later ones. The dataset from the previous project [2], a set of tweets in Spanish classified by expressed emotions, was used to perform the LSTM network tests. for a first approach to the process, it was decided to work with two categories, in such a way to classify the tweets by their polarity. They were grouped as follows:

Positive = happiness and surprise. Negative = disgust, anger, sadness and fear

The valence of tweets in the EI-reg and EI-of datasets is displayed in Figure 2. Observe that, as expected, the chosen query terms resulted in the anger, fear, and sadness datasets having a majority of negative tweets and the pleasure dataset having a majority of positive tweets. The reason why the two affect dimensions are not perfectly connected is revealed by such tweets (or perfectly inversely correlated).

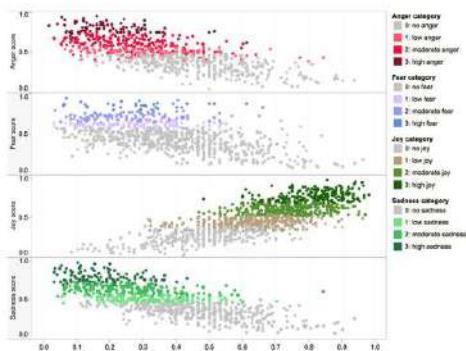


Figure 2: Valence of tweets in the EI-reg and EI-of datasets. Shows the distribution of tweets according to the emotions expressed

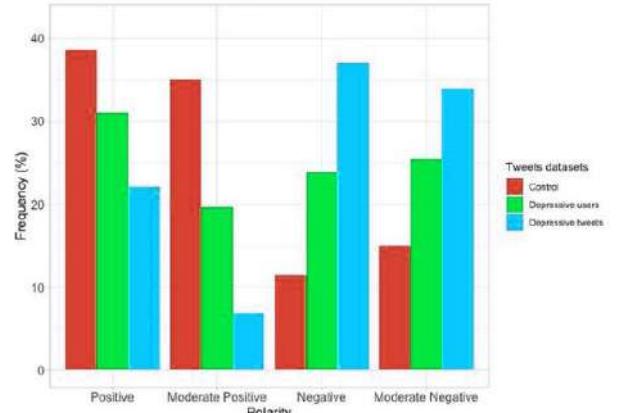


Figure 3. Polarities of the tweets according to the SentiCo Polarity tool in the three datasets.

The configuration and parameterization of the RNN were carried out using the Colaboratory tool provided by Google. Figure 3 contains part of the Python code used to configure the RNN

```
# RNN/LSTM is defined
defRNN():
    inputs =
    Input(name='inputs',shape=[max_len])
    layer =
    Embedding(max_words,120,input_length=max_1 in)(inputs)
    layer = LSTM(64)(layer)
    layer = Dense(1,name='out_layer')
    (layer)
    layer = Activation('softmax')(layer)
    model =
    Model(inputs=inputs.outputs=layer)
    returnmodel
```

Figure 3. RN configuration

The process continues with the execution of the defined RNN and the evaluation of the built model, taking the training set separated in a previous stage as reference. Experimentation with neural networks is currently continuing.

The next objectives to be achieved are:

- Create a corpus of word embeddings from the collection of available tweets
- Experience learning with the corpus of word embeddings generated and with others available for research.
- Select the algorithms that best classify opinions.

This line of research is expected to continue and broaden the understanding of natural language processing. It is intended that this project encourage interest in research and this subject in the students.

V. Conclusion

With cutting-edge machine learning techniques like neural networks, this project intends to learn how to categorise brief words (opinions) created on the social network Twitter according to their emotions. The initial step investigated text classification using an LSTM neural network. In the current stage, several representational strategies are being examined to build a

corpus of embedded words that will be utilised in additional experiments.

References

- [1]. Ali, A., Said, R. A., Rizwan, H. M. A., Shehzad, K., & Naz, I. (2022, February). Application of Computational Intelligence and Machine Learning to Conventional Operational Research Methods. In 2022 International Conference on Business Analytics for Technology and Security (ICBATS) (pp. 1-6). IEEE.
- [2]. Ali, A., Qasim, M., Dilawar, M. U., Khan, Z. F., Jadoon, Y. K., & Faiz, T. (2022, February). Nanorobotics: next level of military technology. In 2022 International Conference on Business Analytics for Technology and Security (ICBATS) (pp. 1-7). IEEE.
- [3]. A deep reinforcement learning approach to solve the vehicle routing problem with resource Constrains. (2023). <https://doi.org/10.2514/6.2023-2662.vid>
- [4]. Deep reinforcement learning. (2022). *The Science of Deep Learning*, 229-250. <https://doi.org/10.1017/9781108891530.017>
- [5]. A. Ali et al., "The Threat of Deep Fake Technology to Trusted Identity Management," 2022 International Conference on Cyber Resilience (ICCR), Dubai, United Arab Emirates, 2022, pp. 1-5, doi: 10.1109/ICCR56254.2022.9995978.
- [6]. Vanneschi, L., & Silva, S. (2023). Artificial neural networks. *Natural Computing Series*, 161-204. https://doi.org/10.1007/978-3-031-17922-8_7.
- [7]. Introduction to semantics of programming languages. (2021). *Concepts and Semantics of Programming Languages* 1, 15-33. <https://doi.org/10.1002/9781119824121.ch2>.
- [8]. Okafor, N., Delaney, D., & Mathew, U. (2022). ProxySense: A novel approach for gas concentration estimation using long short-term memory recurrent neural network (LSTM-RNN). <https://doi.org/10.36227/techrxiv.20306418.v1>.
- [9]. Larsen, K. R., & Becker, D. S. (2021). Why use automated machine learning? *Automated Machine Learning for Business*, 1-22. <https://doi.org/10.1093/oso/9780190941659.003.0001>.

Object Pose Estimation Based on Template-matching Using Attention Module and Residual Block

Gaeun Noh^{1[0000-0002-6125-8289]} and Jong-II Park^{1[0000-0003-1000-4067]}

¹ Department of Computer Science, Hanyang University, Seoul, Republic of Korea
 {shqmffl486, jipark}@hanyang.ac.kr

Abstract. This paper proposes a method to create synthetic datasets using RGB-D camera and to train a template matching-based object pose estimation network to improve the accuracy of object pose estimation for unseen objects. Previous studies were limited by a lack of adequate training data and a relatively simple network model, which resulted in overfitting. In order to enhance performance, the proposed method incorporates synthetic data of objects with symmetrical shapes or limited textures into the existing datasets. By applying Convolutional Block Attention Module to a ResNet50 based model, the intermediate features of objects were more effectively emphasized and suppressed to improve performance. Through comparative experiments with existing methods, it was confirmed that the proposed method provides higher accuracy for unseen objects compared to the existing methods.

Keywords: Object Pose Estimation · Template-Matching · Deep Learning

1 Introduction

Recently, with the development of deep learning technology, the accuracy of image recognition has increased. In the field of Augmented Reality (AR), the fundamental objective is to utilize technology to recognize objects in images and facilitate interaction between virtual and real environments. Object pose estimation is a crucial technology in Augmented Reality that determines the 3D position and orientation of an object captured in an image by analyzing its shape. Among various techniques, object pose estimation is a key technology in the field of Augmented Reality. There are difficulties in performing object pose estimation, such as occlusion, scale variations, changes in lighting, and separating objects from the background, and various methods have been proposed to overcome these difficulties. [1] proposes a 6D object pose estimation model that estimates the 3D transformation and rotation of an object from an image by determining the center of the object and predicting the distance from the camera, with the orientation being regressed as a quaternion representation. However, it is difficult to accurately estimate the pose when occlusion occurs in the camera image. On the

other hand, [2] uses deep networks to extract landmarks of the object in the image and estimate the 6DOF pose through 2D-3D correspondences. This is achieved through the precise multi-precision supervision architecture proposed by predicting landmarks, and it robustly estimates the pose even in the presence of occlusion. It showed higher overall performance than [1], but the accuracy was lower for unseen objects. To address these limitations, a template matching-based object pose estimation research [3] was proposed. This object pose estimation approach that matches the template of an object rendered from multiple views in a CAD model with the highest similarity to the real input image (query image), showing generalization and robustness to occlusion for unseen objects. However, the existing datasets lacks objects with symmetrical shapes or limited textures, making it difficult to estimate poses for complex objects.

In this paper, we suggest to achieve high accuracy in estimating the pose of unseen objects, including objects with symmetrical shapes and with limited textures, in template matching-based object pose estimation. The proposed research uses a mask to remove the background of the captured image before computing the global shape, which can significantly slow down the matching speed. Therefore, to estimate a faster and more accurate pose, the paper proposes to learn a local feature that can be used to match the captured image and synthesized template. In this paper, we create a 3D object synthetic datasets using RGB-D camera to achieve accurate pose estimation even in the case of symmetrical objects and objects with complex shapes and limited textures. Furthermore, we propose a robust model based on ResNet50 with the Convolutional Block Attention Module (CBAM) [6] applied to overcome the limitations of the previous deep learning network models and achieve even more robust object pose estimation.

2 Creating synthetic datasets using RGB-D camera

The process of creating the synthetic dataset for template-matching based object pose estimation is depicted in Fig. 1. The existing datasets used in previous studies on object pose estimation [1, 2, 3, 7] such as LINEMOD, LINEMOD-Occluded, TLESS and YCB-Video, have the limitation of object diversity. To address this, we add the data for objects with symmetrical or complex shape and limited textures. This was done by acquiring the mesh of real objects using an RGB-D camera and incorporating it into the existing CAD model to create the synthetic datasets. The objects were placed on a board printed with markers and captured at various angles using the RGB-D camera to create the synthetic datasets. Each data is composed of RGB and depth images and the extrinsic of the camera calculated through the markers are stored for each image. The point cloud is generated from the RGB and depth images, then converted into a mesh. The converted mesh is used to create a mask image and the 3D bounding box of the object.

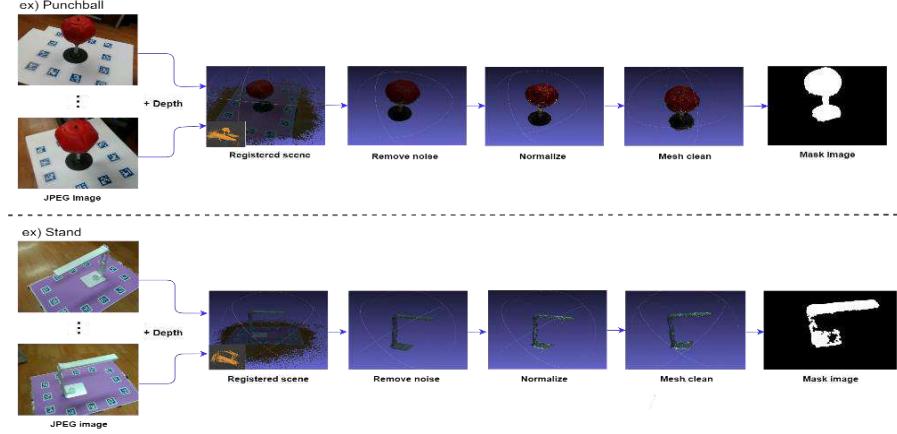


Fig. 1. Creating and processing point cloud scenes using RGB-D camera.
[3D object Synthetic datasets, ex) Punchball(symmetrical) / Stand(complex shape)]

3 Template-matching method

3.1 Framework

The method proposed in this paper is a deep learning-based learning method for estimating an object pose by matching an object template and a real image, based on the template-matching [3] framework. It is configured as shown in Fig. 2. Template matching-based object pose estimation methods generate templates rendered from multiple views and mask images inside the object, not only for seen objects but also for unseen objects and estimate poses by matching them with the most similar templates. Template generation uses BlenderProc [5] to sample synthetic templates for realistically rendered images according to the protocol in [4], and generates 3,084 templates for each object.

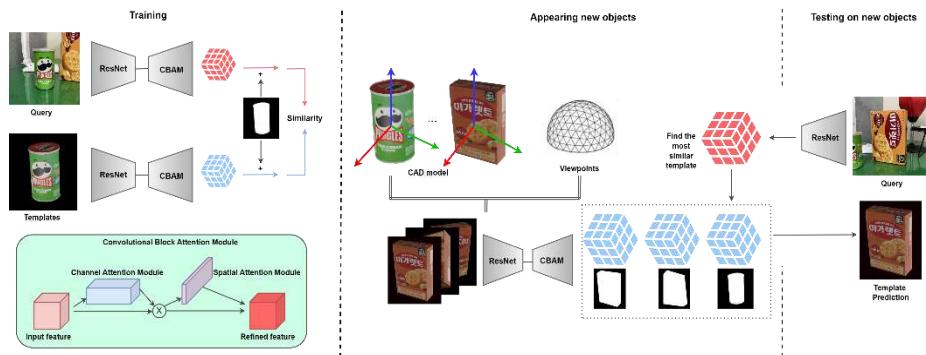


Fig. 2. Framework for Object Pose Estimation Based on Template-Matching.

Considering the template and mask, it can be robust to occlusions arising from real images because only some regions can be compared by deleting the background of real images. And it was confirmed through experiments that the unseen object is more generalized. Therefore, we compare the similarity between the template generated based on the local feature of the image and the real input image based on the local feature similarities and estimate the pose by finding and matching the template with the highest similarity in the template set. It can be partially obscured, and the pose can be stably estimated even though the background of the object is not uniform.

3.2 Network structure

Previous studies have used a CNN algorithm consisting of two convolutional layers and two fully connected layers with subsequent 2x2 max pooling layers as the "base" backbone. However, there was a problem of increasing computation due to large datasets. To solve this problem, the paper applied skip connection by applying ResNet50 to solve the vanishing gradient problem and increased performance by learning a deeper network than the CNN algorithm. In addition, all pooling, FC layers were removed and replaced with two 1x1 convolution layers that output local features, allowing the number of channels to be adjusted and the model to be configured deeper to reduce the computation.

In addition, the CNN algorithm has a problem of learning even the noise contained in the train data due to large computations and overfitting. The suggested method was to select a target object and learn the channel attention to improve the performance of the unseen object. In addition, 7x7 convolutional layer is applied to the channel compressed in the spatial attention to create an attention map. The large receptive field encodes which part of the pixel to focus on while finding spatially important regions, confirming that the performance is improved compared to the previous network with a small amount of computation.

And to reduce the loss, two loss functions were used to experiment. When InfoNCE loss was applied, the result confirmed a lower loss value than the Triplet Loss. It was judged that InfoNCE loss showed higher accuracy because it measures similarity based on mutual information when it was a large dataset (Ablation study).

4 Experiment and result

The experiment in this paper, we used RGB-D camera as well as existing datasets to generate and experiment with various synthetic datasets for additionally symmetric objects(punchball), complex shape objects(stand), and objects with limited textures (mirrors, etc.). The experimental setting was conducted at ubuntu 18.04, with epoch=100, learning rate=1e-4, batch size=8, and Adam as the optimizer. The experiment takes about 12 hours to learn from GeForce RTX 3080.

For the comparative experiment, the total dataset was classified into train:valid:test = 6:2:2 to generate a total of 88,668 data. The class of commonly set data sets was

divided into seen object (0, 1, 2, …, 8) and unseen object (9, 10, 11, …, 17). The experiment is divided into seen object and unseen object as a comparison between the proposed network and the previous network. Also, we experiment on existing datasets and synthetic datasets, respectively. The experimental results are shown in Table 1 and Table 2. Table 1 compares the performance of previous and proposed networks for seen object and unseen object using only the existing dataset. As a result of the comparison, it was confirmed that the accuracy of the seen object and the unseen object improved the performance of the proposed network compared to the previous network. Table 2 experiments on seen objects and unseen objects using only synthetic datasets containing complex objects. As a result of the comparison, the accuracy of the proposed network increased by approximately 1.2% in seen object compared to the previous network. In the unseen object, the accuracy increased by approximately 1.4%. The results of the method proposed in this paper prove higher performance not only for existing datasets, but also for synthetic datasets with symmetrical, complex shapes, and limited textures. However, we find that there is a limit to estimate poses of objects that are not in the training datasets because the unseen object has lower accuracy than the seen object.

Table 1. Comparison results using existing datasets of size 224x224

	Our method	Previous method
Seen object	0.9779	0.8386
Unseen object	0.4536	0.4418

Table 2. Comparison results using synthetic datasets of size 224x224

	Our method	Previous method
Seen object	0.9788	0.8569
Unseen object	0.5457	0.4018

5 Ablation Study

This section uses synthetic dataset for the proposed network to reduce training loss, and experiments are conducted by applying InfoNCE loss and Triplet loss, respectively. The result of applying InfoNCE loss reduced the loss by approximately 0.26 compared to the Triplet loss. This was judged to have shown higher accuracy because InfoNCE

loss measures similarity based on mutual information when learning large datasets as in experiments.

Table 3. Comparison results of loss function for seen object

	InfoNCE loss	Triplet loss
Our method	0.9743	0.7152

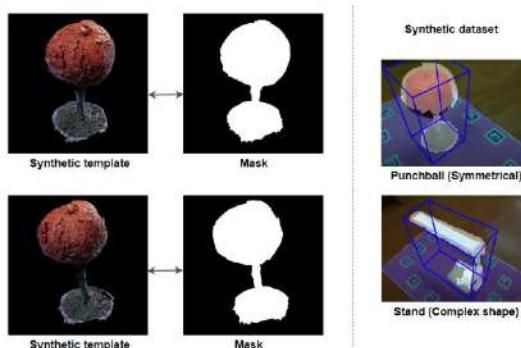


Fig. 3. Synthetic template, mask and special case (synthetic dataset).

6 Conclusion

This paper proposes a method to increase the accuracy of template matching-based object pose estimation not only for seen objects but also for unseen objects by creating a synthetic datasets of 3D objects and applying the Convolutional Block Attention Module to the proposed ResNet50 network. Specifically, the scene of the point cloud was generated and processed as a synthetic dataset and experimented with synthetic datasets of 3D objects and an existing LINEMOD datasets, respectively. As a result, the networks proposed in this paper achieve higher accuracy for complex objects for seen objects and unseen objects than previous networks. In addition, higher accuracy was confirmed when InfoNCE loss was applied than Triplet loss in ablation study. In future research, we will make it possible in real-time by focusing on higher performance object pose estimation for unseen objects.

References

1. Xiang, Yu, et al. "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes." *arXiv preprint arXiv:1711.00199* (2017).
2. Chen, Bo, Tat-Jun Chin, and Marius Klimavicius. "Occlusion-Robust Object Pose Estimation with Holistic Representation." *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2022.
3. Hu, Yinlin, et al. "Templates for 3D Object Pose Estimation Revisited: Generalization to New Objects and Robustness to Occlusions." *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022.
4. Paul Wohlhart and Vincent Lepetit. Learning Descriptors for Object Recognition and 3D Pose Estimation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
5. Maximilian Denninger, Martin Sundermeyer, Dominik Winkelbauer, Youssef Zidan, Dmitry Olefir, Mohamad Elbadrawy, Ahsan Lodhi, and Harinandan Katam. Blenderproc. *arXiv preprint arXiv:1911.01911*, 2019.
6. Woo, Sanghyun, et al. "Cbam: Convolutional block attention module." *Proceedings of the European conference on computer vision (ECCV)*. 2018.
7. Chen, Bo, Tat-Jun Chin, and Marius Klimavicius. "Occlusion-robust object pose estimation with holistic representation." *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2022.

COVID -19 Detection based on CT Scan Images using Deep Learning Methods

Tuan Le Dinh^{1[0000-0003-3672-3510]}, Jae-Hyun Kim ¹, Suk-Hwan Lee ², Ki-Ryong Kwon ¹

¹ Pukyong National University, Busan, South Korea

² Donga University, Busan, South Korea

ldt.uet@gmail.com, kjh96mine@naver.com, skylee@dau.ac.kr,
kiryongkwon@gmail.com

Abstract. Since the outbreak of COVID-19, there are many attempts to investigate techniques to detect COVID-19 cases using a quick and safe procedure. Utilizing radiology modality to identify the existence of COVID-19 was considered the prominent method for COVID-19 screening. CT scans play an important role as the novel modality for COVID-19 detection. In this paper, we study the current advanced in COVID-19 detection using CT Scan images and deep learning. In this survey paper, three state-of-the-art methods are analyzed including TeliNet, CovidCTNet, and Covid DeteCT, then we describe the advantage and disadvantages of each method and analyze the performance of these methods. We also discuss the limitations and future directions to enhance the performance of these approaches.

Keywords: Coronavirus, COVID-19, Radiology images, CT Scan

1 Introduction

The standard testing method for COVID-19 is RT-PCR, which is one variation of PCR testing that adds one step of RNA to DNA reverse transcription. The benefit of RT-PCR is its high sensitivity, reliability, and real-time. However, it still has some drawbacks, such as false-negative rates are high, giving us only the presence of the virus but not the real infected detection, and the cost to establish the laboratory and trained technicians. Meanwhile, using radiology methods to detect COVID-19 can replace RT-PCR as the prominent alternative for many reasons. First, using the radiology method, especially, CT Scan reduces the large cost to set up the facility for testing. Second, the procedure is simple, no need to take the sample, which reduces the possibility of spreading the virus. And lastly, the method is reliable and real-time since the available CT Scan images for training and the inference time of the method is just a few seconds.

2 Related Works

There are many attempts to apply deep learning and radiology for COVID-19 detection. We can name some of them, such as the work of Horry, Michael J., et al [1] which try

to detect COVID-19 using multimodal radiology images and transfer learning. Another work by Amyar, Amine, et al. [2] proposed multi-task deep learning to classify and segment COVID-19 pneumonia based on CT Scan images. In the work of Polzinelli, M, et al. [3] the authors try to enhance the performance of Convolutional Neural Network (CNN) architecture by using a lightweight CNN which is 10 times faster and better than complex CNN. CNR-IEMN is a multi-task and multi-stage deep learning approach that uses XG-Boost classifier in the second stage.

3 Methodology

In this study, we analyze three convolutional neural networks, which are TeliNet, CovidCTNet, and Covid DeteCT. The reason that the author had chosen these neural nets is because of the simplicity, robustness, and efficient performance of this network when running on the medical dataset, especially on Chest CT Scan images.

3.1 TeliNet

A simple and efficient Convolutional Neural Network dedicated to CT images to diagnose COVID-19 associated with ICCV 2021 competition [4]. The network's main idea is to try to minimize layers of the architecture to cut down the number of parameters in the network overall.

Network Architecture. The shallow network consists of four layers, including 2D convolutions, max pooling, LeakyReLU, and batch normalization layers. The total number of trainable parameters is no more than 8.5 million and 15 times lighter than the well-known VGG-16 [5] which contains over 134 million parameters. The input image size is 256 x 256, the convolutional layers use 16 filters of size 3x3, followed by LeakyReLU [6] activation functions and max pooling. After the convolution layers are two dense layers and finally is the Sigmoid for binary classification.

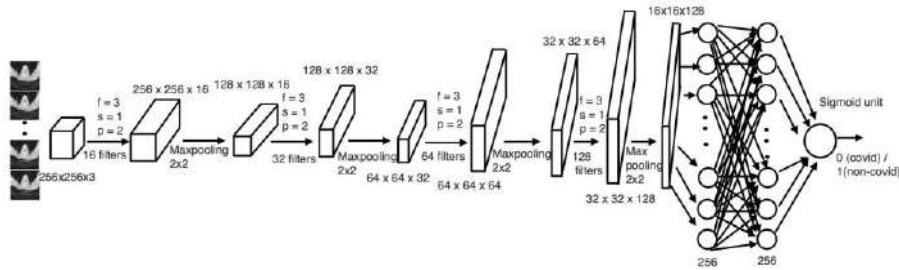


Fig. 1. TeliNet Architecture Overview.

Dataset. The dataset was downloaded from MIA-COVID which contains 609057 chests of CT scan images and divided into two COVID-19 and nonCOVID-19

categories. There is a total of 5000 CT scan series divided into three subsets, the training, validation, and test set.

Results. On F1 macro score, the TeliNet (0.81) outperform VGG-16 (0.72) and compares to the standard benchmark (0.7). The experiment was conducted on MacBook Pro with 8 GB of ram. The experiment tries a range of batch sizes from 4 to 128 and picks up the best result of 32.

Table 1. TeliNet F1 score result.

Method	Train
TeliNet	0.81
VGG-16	0.72
Benchmark	0.7

3.2 CovidCTNet

This open-source project contains algorithms and a CT scan dataset for COVID-19 detection purposes. The method aid doctor with screening and detecting COVID-19 and the open-source code is freely distributed and modified for future advancement.

Network Architecture. The architecture used in CovidCTNet [7] is BCDU-Net which plays an important role when dealing with a small dataset, especially tasks in the medical domain. The overall architecture uses multistep to train and detect COVID-19. The first step in the training pipeline is to train a subset of the dataset, then the second step is to feed all the datasets to the trained model from the first step, then the last step is to classify the CT scan images by using a customized convolutional neural network.

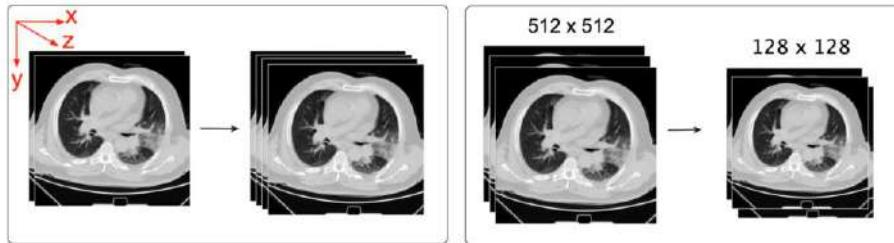


Fig. 2. Preprocessing phase in CovidCTNet training pipeline.

Dataset. The dataset was acquired from 335 patients, with a total of 16750 slices, and the second dataset of 115 patients with 5750 slices. In addition, the dataset from lung nodule classification was added to the training dataset with 70 CT scans. Overall, the dataset contains images from different institutions and countries.

Results. The CovidCTNet boosted the accuracy of COVID-19 detection with 93% of sensitivity and 95% of accuracy. Compare to other state-of-the-art methods, CovidCTNet outperforms these approaches on AUC, Sensitivity, and Specificity matrices. Besides optimizing and improving detection accuracy, CovidCTNet plays an important role in clinical treatment with simple and cost-effective methods.

3.3 Covid DeteCT

The author of Covid DeteCT [8] proposes a novel method that classifies and detects COVID-19 cases based on the entire volume of chest CT scan images. The training pipeline uses a multi-center dataset across 8 countries to train and test the neural network architecture and output reliable and accurate results for COVID-19 detection.

Network Architecture. The proposed method Deep COVID DeteCT short for DCD is a customize convolutional neural network with 27 feature extractor layers and 1 fully connected layer. The architecture uses cross-entropy as a loss function and Adam as an optimization method with 20 epochs of training in total. DCD used Inception3D as the backbone of the model and the training pipeline contains 2 tasks. The first task is the classification task, the author uses a hold-out external test set, and the validation set uses an internal dataset. And the task was evaluated based on the Area under Curve (AUC) [9] and Receiver Operating Characteristics (ROC) [10]. The second task is the prognosis task to predict clinical features such as length of hospital stay.

Dataset. In this paper, the author uses the RICORD COVID-19 dataset, which is a freely distributed multi-institute and multi-nation dataset and was labeled by an expert. The dataset contains 240 CT scan slices, and the pathway of dataset collection goes through five steps. First, the data aggregation must be shared agreement between patients and the institution, next step is deidentification using the RSNA toolkit. The third step is data transfer which transfers deidentification data to RSNA. Then radiologists join the team to annotate and segment the dataset. The final step is user access, which needs to have a user data agreement and download from the homepage site.

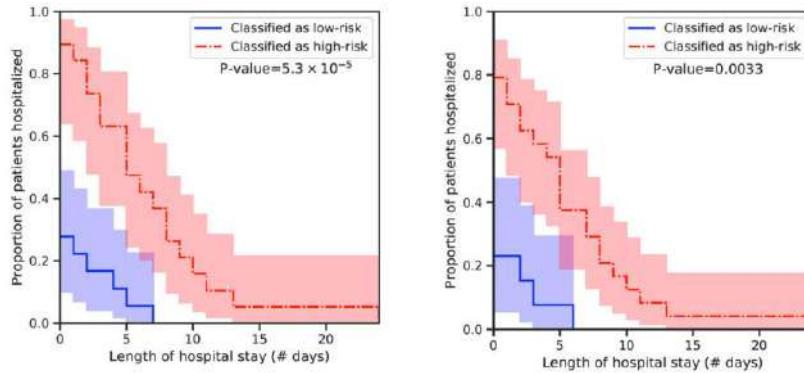


Fig. 3. Length of hospital stay prediction on the second task of DCD.

Results. In the classification task, the method outperforms ResNet-50 when training on 2D CT scan images. The DCD performance was also tested on a range of variations such as the soft tissue, the bone, and the lung with AUC above 0.8 in every test site. In the second task to predict the prognosis feature, DCD also provides us detail and precise prediction of features like the length of hospitalization and the follow-up patent over time.

4 Conclusion

In this paper, we study the application of deep learning for COVID-19 detection based on CT scan images. Three methods are subject to our investigation including TeliNet, CovidCTNet, and Covid DeteCT. These methods show that applying deep learning for CT scan image help to aid COVID-19 detection with high accuracy and reliability, in addition, radiology methods are efficient in term of cost and initial investment to build up testing facilities compare to the RT-PCR method. In future work, we will take the advantage of each method and then develop our algorithm to create a new tier of the network that is both lightweight, high accuracy, and easy to train.

Acknowledgement

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the Grand Information Technology Research Center support program (IITP-2023-2016-0-00318) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation) and by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2020R1I1A306659411, 2020R1F1A1069 124)".

References

1. Horry, Michael J., Subrata Chakraborty, Manoranjan Paul, Anwaar Ulhaq, Biswajeet Pradhan, Manas Saha, and Nagesh Shukla, "COVID-19 detection through transfer learning using multimodal imaging data," *Ieee Access*, vol. 8, pp. 149808-149824, 2020.
2. Amyar, Amine, Romain Modzelewski, Hua Li, and Su Ruan, "Multi-task deep learning based CT imaging analysis for COVID-19 pneumonia: Classification and segmentation," *Computers in Biology and Medicine*, vol. 126, p. 104037, 2020.
3. Polsinelli, Matteo, Luigi Cinque, and Giuseppe Placidi, "A light CNN for detecting COVID-19 from CT scans of the chest," *Pattern recognition letters*, vol. 140, pp. 95-100, 2020.
4. M. N. Teli, "TeliNet: Classifying CT scan images for COVID-19 diagnosis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, 2021.

5. Simonyan, Karen, and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556* , 2014.
6. Xu, Bing, Naiyan Wang, Tianqi Chen, and Mu Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853* , 2015.
7. Javaheri, Tahereh, et al., "CovidCTNet: an open-source deep learning approach to diagnose covid-19 using small cohort of CT images," *NPJ digital medicine*, vol. 4, no. 1, pp. 1-10, 2021.
8. Lee, Edward H., et al., "Deep COVID DeteCT: an international experience on COVID-19 lung detection and prognosis using chest CT," *NPJ digital medicine*, vol. 4, no. 1, pp. 1-11, 2021.
9. Bradley, Andrew P. , "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern recognition*, vol. 30, no. 7, pp. 1145-1159, 1997.
10. Metz, Charles E, "Receiver operating characteristic analysis: a tool for the quantitative evaluation of observer performance and imaging systems," *Journal of the American College of Radiology*, vol. 3, no. 6, pp. 413-422, 2006.

[Remarks]

This paper is a re-publishing (summary presentation) of the paper which has been published in *Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021* and in *NPJ digital medicine Journal* by request of the IW-FCV2023 program committee to share the research results.

CHANGE DETECTION OVER MULTISPECTRAL IMAGES USING MACHINE LEARNING TECHNIQUES: A CASE STUDY ON RUSHIKONDA

Shaik Fyzulla¹, Chitturi S Pavan Kumar², Chintakayala Pavan Veera Nagendra
Kumar³, and Punukollu Surya Prakash⁴

¹Department of Information Technology, Velagapudi Ramakrishna Siddhartha
Engineering College, Vijaywada, Andhra Pradesh, India

¹fyzullahshaik@gmail.com

²pavanchitturi@vrsiddhartha.ac.in

³pavachintakayala@gmail.com

⁴punukollusurya969@gmail.com

Corresponding Author: Dr. C S Pavan Kumar

Abstract— The ability to identify the amount of change observed in Land is needed and it is necessary to describe how human actions impact/affect the environment. The Rushikonda, which is located in the Andhra Pradesh district of Visakhapatnam, has seen changes in the land use, according to observations. The goal is to examine and comprehend how, over the course of the last six years, land use and cover has changed in the Rushikonda(2015-2021). The current study has demonstrated how changes in agricultural patterns, industry, and land use have always had an impact on the environment[11-15]. The findings of the present study have revealed that how risk to nature, industry, change in crop patterns, and land usage are already influenced by the lack of water resources. The paper's conclusion is that recent approaches help to understand how land is being used generally and that the observation of a 28% land shift helps with the planning of construction projects close to the Rushikonda.

Keywords— Remote Sensing, Principal Component Analysis, K-Means, Change Detection, Classification, Image Processing, Multi-Spectral Images, Normalized Difference Vegetation Index, Geospatial Data Abstraction Library.

1. Introduction

"Change detection" is used to identify changes that have occurred in remote sensing data over two time periods. It finds applications in a variety of fields, including surveillance videos and medical imaging. Its research goal is to separate out unwanted data and gather information on changes in the study area.

The most popular algorithms nowadays are those that involve machine learning. It will be applied in categorization, clustering, and various other applications. [16] The different types of machine learning algorithms are reinforcement, semi-supervised, unsupervised, and supervised.

Researchers believe that remote sensing is the most efficient and trusted environment for detecting changes on the earth's surface. The examination of the Land use and Land change information provides urban planners some advantage in making appropriate choices for managing land resources. To categorize the satellite images, researchers used a number of categorization algorithms..

2. Literature Review

In [1] The author discussed how to compute and assess the efficiency of both implementation options for change identification using ML/DL algorithms. [2] mentioned that initially, the photos are enhanced, and then the change detection using Fuzzy C-Means Clustering approach is used, which will result in better outcomes.[3] author employed the well-known K-Means clustering method, which improves the single-channel intensity band ratio and is difficult to determine the K value. [4] author employed the Edge Enhancement approach, which has a lower computational cost, does not require prefiltering, and gives results directly to the wavelet domain; however, the image may begin to look less natural as the overall sharpness advances. [5] author utilised the IR-MAD approach, which when using the suggested initial change mask may converge to a better no change backdrop even in the face of big changes but fails to converge when there are a lot of change pixels. [6] the author employed K-means clustering tends to produce fewer errors but results in thicker boundaries on the derived change map. [7] CNN, Sharp Mask, U-Net, and ResNet methods were used by the author to provide fast and precise image segmentation. Deeper networks are computationally expensive and require weeks to train, but they can be learned fast without increasing the error rate. [8] By analysing satellite images, the author employs image processing techniques that employ machine learning to identify changes in the pattern. Based on Convolutional Neural Networks, authors present a methodology for automatically detecting change in multispectral satellite data. [9] The author utilizes K-Means clustering and principal components Analysis, which first calculate Eigen Vector Space and then generate Future Vector Space, which is then subjected to K-Means clustering with two clusters, one stating change and the other stating no change in the Satellite Image. [10] The author uses two wavelet-based image enhancement approaches, the discrete wavelet transform (DWT) and the dual tree-complex wavelet transform (DT-CWT), to increase the image quality. Following image enhancement, Support Vector Machines are used to

classify the land use of the pictures. An evaluation of accuracy based on a confusion matrix is used to evaluate the accuracy of land-use categorization.

3. Proposed Method

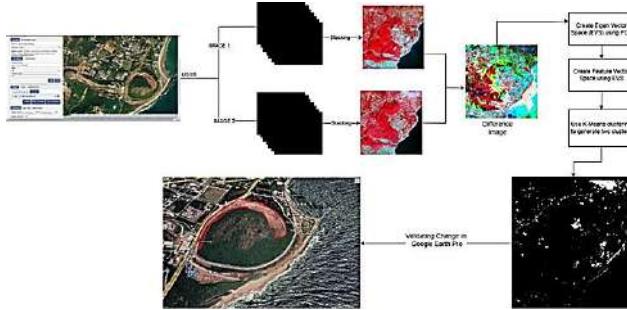


fig 1: Architecture of the proposed method

2.1 Input data collection

In this stage, we will download data from the United States Geological Survey (USGS) website for the Rushikonda area for two different time periods, 2015 and 2021, onto our local computer. The downloaded image will be divided into bands.

And then, we will perform stacking, which is the process of combining multiple images of the same resolution into a single image, where the images with higher resolution will be resampled into the target resolution.

2.2 Calculating the Difference Image

The difference image between the stacked images obtained in the previous step will be calculated in the third step

```

FOR xi=1 to length(stacked_image1)
    FOR xj=1 to length(stacked_image2)
        Diff_image=abs(xi-xj)
    END FOR
END FOR
  
```

2.3 Dimensionality reduction and feature Extraction

In this step, we will examine how to compute Eigen vectors using Principal Component Analysis and Kernel Principal Components Analysis. These are non-zero vectors that do not change direction when subjected to a linear transformation. It only varies by a single scalar factor.

The kernel PCA enables the analysis of more complex data patterns that would not be visible using only linear transformations.

$$\text{COVR} = \frac{1}{M} \times \sum_{i=1}^M I \times I^T$$

And then we use the Eigen Vector that we obtained in the previous step, sorted in decreasing order, to assemble the feature vector space when PCs are acquired.

After that the future vector obtained in the previous step is applied into the K-Means Clustering algorithm with the K value set to 2 because there are only two possibilities: change or no change.

K-Means algorithm

Step-1: K clusters will be formed (here we take 2 clusters)

Step-2: Centroids were formed based on the cluster.

Step-3: Assign points to the centroid that are close to the cluster.

Step-4: Centroids of newly formed clusters were recomputed

Step-5: Previous two steps are repeated until the value of the centroid will be constant for the next two computations.

2.4 Validating the Results

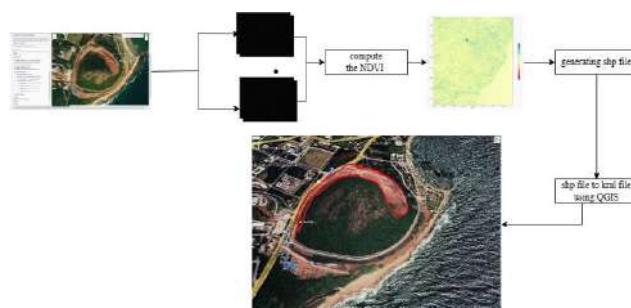


Fig 1.1 Validating the Results in Google Earth Pro

After obtaining the change map, the validation process began with the use of the Geospatial Data Abstraction Library (GDAL), which is primarily used to handle data in the raster and geospatial formats.

The inputted data will be downloaded from the USGS earlier and extracted, and this data will be used to calculate the Normalized Difference Vegetation Index (NDVI). Researchers must observe the amount of green on a given area of land is observed by looking at the various wavelengths of visible and near-infrared sunlight reflected by the plants.

Following that, a shape file will be generated and exported, allowing us to import and display the change map in Google Earth Pro.

Now launch Google Earth Pro, navigate to the menu, and select the option called open. Then you will be prompted to specify the shape file location, select the exported shape file, and click open. The change map will then be displayed in Google Maps.

4. Result Analysis

In this experiment, we obtain various outputs after performing various tasks, and each output represents a specific output in each step.

The first output is a stacked image of two timeline periods, 2015 and 2021, obtained with the QGIS tool.



Fig 2. Color Infrared Image of 2015

6



fig 3. color Infrared Image of 2021

The images above are the final images obtained after applying stacking in the QGIS tool, and they are used to generate the change map.

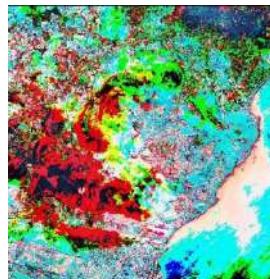


Fig 4. Difference Image of Kernel PCA

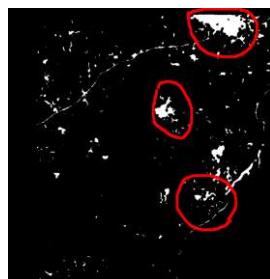


Fig 5. Change Map with K-Pca & K-means

The above images represent the difference image that we obtained and the change map obtained using Kernel PCA and K-Means.

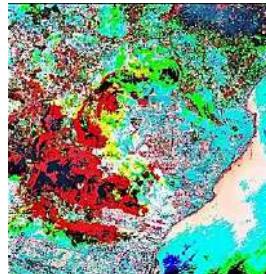


Fig 4. Difference Image of PCA

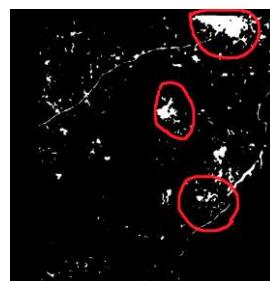


Fig 5. Change Map with Pca & K-means

The above images depict the difference image and the change map obtained using PCA and K-Means.

The images above depict the change map at two different time periods: 2015 and 2021. The normal PCA algorithm produces less accurate results than the Kernel PCA algorithm, and the results are shown above.

The outputs obtained after applying PCA and Kernel PCA are compared, with PCA producing a change in percentage of 21.06 and Kernel PCA producing a change in percentage of 18.76. According to the results, PCA produces more change than Kernel PCA.

Further to that, the obtained NDVI data reveal that there is 39.03 Vegetation Index available near Rushikonda, indicating that the green health vegetation is present.



Fig 6. Rushikonda Before 2015



Fig 7. Rushikonda after 2021

As a result, the obtained changes are validated in Google Earth Pro. The area that is automatically marked in red by Google Earth Pro near Rushikonda is as the change is obtained in the preceding process.

Algorithm	Percentage of change obtained during the year 2015 and 2021
PCA & K-Means	21.06
Kernel PCA & K-Means	18.76

Table 1. Comparing the results of PCA and Kernel PCA with K-Means

5. Conclusion

We have analysed the two algorithms PCA and Kernel PCA for better accurate results. Since the proposed method is unsupervised, there is no need for expensive training data sets catered to change detection. It is designed to detect change in land use/cover of Rushikonda. The years 2015 to 2021 were used to detect land cover changes using remote sensing, satellite imagery, and image processing techniques.

References

1. T. Vignesh; K. K. Thyagarajan; K. Ramya: Change Detection using Deep Learning and Machine Learning Techniques for Multispectral Satellite Images.
2. M. H. Kesikoglu; U. H. Atasever; C. Ozkan: Unsupervised Change Detection In Satellite Images Using Fuzzy C-Means Clustering And Principal Component Analysis.
3. Debanshu Ratha, Shaunak De, Student Member, Turgay Celik, Member, and Avik Bhattacharya, Senior Member, IEEE “Change Detection in Polarimetric SAR Images Using a Geodesic Distance Between Scattering Mechanisms”.

4. M.N. Sumaiya; R. Shantha Selva Kumari; "Unsupervised Edge Enhancement algorithm for SAR Images using Exploitation of Wavelet Transform Coefficients".
5. Prashanth Reddy Marpu; Paolo Gamba; Morton J. Canty; "Improving change detection results of IR-MAD by eliminating strong change".
6. Turgay Celik "Unsupervised Change Detection in Satellite Images Using Principal Component Analysis and k-Means Clustering".
7. de Bem, Pablo Pozzobon, Osmar Abílio de Carvalho Junior, Renato Fontes Guimarães, and Roberto Arnaldo Trancoso Gomes. 2020. "Change Detection of Deforestation in the Brazilian Amazon Using Landsat Data and Convolutional Neural Networks".
8. Greeshma Katarki; Harivijay Ranmale; Indira Bidari; Satyadhyana Chickerur; "Estimating Change Detection of Forest Area using Satellite Imagery".
9. Christopher Munyati: "Use of Principal Component Analysis (PCA) of Remote Sensing Images in Wetland Change Detection on the Kafue Flats, Zambia".
10. Karan, Shivesh Kishore, and Sukha Ranjan Samadder. "Accuracy of land use change detection using support vector machine and maximum likelihood techniques for open-cast coal mining areas".
11. Polykretis, C.; Grillakis, M.G.; Alexakis, D.D. Exploring the impact of various spectral indices on land cover change detection using change vector analysis: A case study of Crete Island, Greece.
12. Panuju, Dyah R., David J. Paull, and Amy L. Griffin. "Change detection techniques based on multispectral images for investigating land cover dynamics".
13. Minu, S., and Amba Shetty. "A comparative study of image change detection algorithms in MATLAB."
14. Rathindra Nath Biswas, Md. Nazrul Islam, M. Nazrul Islam, Md. Juel Mia, Md Nasrat Jahan, Mir Fahim Shaunak, Md. Motiur Rahman, Md. Yachin Islam. (2022) Impacts of morphological change on coastal landscape dynamics in Monpura Island in the northern Bay of Bengal, Bangladesh.
15. Saha, S.; Bovolo, F.; Bruzzone, L. Unsupervised deep change vector analysis for multiple-change detection in VHR images.
16. Abijith, Devanantham, and Subbarayan Saravanan. "Assessment of land use and land cover change detection and prediction using remote sensing and CA Markov in the northern coastal districts of Tamil Nadu, India." Environmental Science and Pollution Research (2021): 1-13.
17. Aldhshan, Shaban RS, and Helmi Zulhaidi Mohd Shafri. "Change detection on land use/land cover and land surface temperature using spatiotemporal data of Landsat: a case study of Gaza Strip." Arabian Journal of Geosciences 12, no. 14 (2019): 1-14.
18. Talukdar, Swapan, Pankaj Singha, Susanta Mahato, Swades Pal, Yuei-An Liou, and Atiqur Rahman. "Land-use land-cover classification by machine learning classifiers for satellite observations—A review." Remote Sensing 12, no. 7 (2020): 1135.
19. Vali, Ava, Sara Comai, and Matteo Matteucci. "Deep learning for land use and land cover classification based on hyperspectral and multispectral earth observation data: A review." Remote Sensing 12, no. 15 (2020): 2495.

10

20. Gong, Maoguo, Yuelei Yang, Tao Zhan, Xudong Niu, and Shuwei Li. "A generative discriminatory classified network for change detection in multispectral imagery." IEEE.

Gaussian Process based Illumination Planning for Photometric Stereo

Yuji Oyamada¹

Tottori University, Japan oyamada@tottori-u.ac.jp

Abstract. Photometric Stereo estimates surface normal from a set of shading images observed by a fixed camera under different lights. Although robust regression techniques and deep neural networks improves accuracy and robustness of normal estimation, all of the literature assumes input shading images are high quality, e.g., the corresponding light positions are spatially distributed and each image has as less shadow as possible. This is not easy to reproduce especially for less-experienced persons. Therefore, we typically increase the number of image acquisition. An interesting method was proposed by Tanikawa et al. that considers shadow and the distribution of light positions. This paper proposes a Gaussian Process based illumination planning for shading image acquisition. The method first chooses three pre-defined light positions and observes images independently and then iterates optimum light selection and shading image acquisition until some condition is satisfied. The experiment compares the proposed method, Tanikawa's method, and random selection to validate the effectiveness of the proposed method.

Keywords: Illumination planning · Photometric Stereo · Gaussian Process.

1 Photometric Stereo

This section introduces the motivation of this paper: what is photometric stereo and why we need illumination planning.

1.1 Potometric Stereo

Photometric stereo is a technique of 3D scene recovery that estimates object surface from a set of shading images observed by a fixed camera under different lightning conditions [11]. The simplest model assumes Lambertian surface and point distance light source [13]. When a point of an object of Lambertian reflectance is lit by point distant light source, the observed pixel intensity $i \in \mathbb{R}_+$ is written as

$$i = \rho \max(\mathbf{n} \cdot \mathbf{l}, 0), \quad (1)$$

where $\mathbf{n} \in \mathbb{R}^3$, $\|\mathbf{n}\| = 1$ denotes the surface normal vector, $\rho \in \mathbb{R}$ the surface albedo, and $\mathbf{l} \in \mathbb{R}^3$ the light vector. As the equation explains, the brightness of

the observation is proportional to the surface albedo ρ and the cosine similarity of the surface normal \mathbf{n} and the light vector \mathbf{l} . Note that the brightness i is 0 when the dot product $\mathbf{n} \cdot \mathbf{l}$ is negative (when the light does not arrive at the point in other words). Lambertian Photometric Stereo (LPS) recovers the surface normal vector from the observation under $F \geq 3$ non-collinear lighting conditions. By formulating the F observation as a linear equation

$$\underbrace{\begin{bmatrix} i_1 \\ \vdots \\ i_F \end{bmatrix}}_{\mathbf{i} \in \mathbb{R}^{F \times 1}} = \underbrace{\begin{bmatrix} \mathbf{l}_1^\top \\ \vdots \\ \mathbf{l}_F^\top \end{bmatrix}}_{\mathbf{L} \in \mathbb{R}^{F \times 3}} (\rho \mathbf{n}) \quad (2)$$

$$\rightarrow \mathbf{i} = \mathbf{Ls}, \quad (3)$$

LPS recovers the surface vector \mathbf{s} by least squares and then decouples it into ρ and \mathbf{n} as

$$\hat{\mathbf{s}} = \mathbf{L}^\dagger \mathbf{i} \quad (4)$$

$$\hat{\rho} = \|\mathbf{s}\| \quad (5)$$

$$\hat{\mathbf{n}} = \frac{\mathbf{s}}{\hat{\rho}}, \quad (6)$$

where \mathbf{L}^\dagger denotes the pseudo inverse matrix of the light matrix \mathbf{L} . Typically, 10-20 images are used for obtaining stable and accurate 3D reconstruction. The limitation of LPS is due to its assumption. When a target object has complex reflection property such as specular reflection and subsurface scattering, Eq. 1 does not hold. Attached and cast shadow also violates the observation model even with Lambertian objects. Furthermore, lights in daily life such as near light requires different models from Eq. 1.

State of the art Photometric Stereo methods are categorized into two types: robust least squares or deep learning methods. Robust Photometric Stereo (RPS) introduces the sense of sparse regression [14, 7, 8]. Regarding observation violating Eq. 1 as outliers, RPS recovers surface normal of non-Lambertian objects even with shadows. RPS relies on a simple assumption that inlier observation is dominant and therefore robust regression can eliminate the negative effect of outlier observations. Deep Photometric Stereo (DPS) solves the Photometric Stereo problem with deep neural network manners [10, 6, 2]. Thanks to the powerful deep neural networks, DPS can handle non-Lambertian objects and even unknown light conditions.

1.2 Good Shading Images

Aside from the aforementioned PS techniques, the quality of shading images takes very important part in PS. As least squares favors observation at distributed positions, PS does shading images with variety of lighting conditions [3].

Specifically, the spatial variance of \mathbf{L} is important. One of the simplest formulation of the variance is proposed by Drbohlav and Chantler as

$$\text{trace}(\mathbf{L}^\top \mathbf{L}). \quad (7)$$

The variance is strongly related with normal estimation uncertainty. Representing observation noise as δ_f , 1 is formulated as

$$\begin{bmatrix} i_1 \\ \vdots \\ i_F \end{bmatrix} = \begin{bmatrix} \mathbf{l}_1^\top \\ \vdots \\ \mathbf{l}_F^\top \end{bmatrix} (\rho \mathbf{n}) + \begin{bmatrix} \delta_1 \\ \vdots \\ \delta_F \end{bmatrix} \quad (8)$$

Then, the expected error given \mathbf{L} is derived as

$$\epsilon(\mathbf{L}) = E[(\mathbf{n} - \hat{\mathbf{n}})^\top (\mathbf{n} - \hat{\mathbf{n}})] \quad (9)$$

$$= \sigma^2 \text{trace}((\mathbf{L}^\top \mathbf{L})^{-1}). \quad (10)$$

1.3 How to Acquire Good Shading Images?

The way of shading image acquisition requires either expertise knowledge or computer guidance. The expertise knowledge is from intuition from experts that light positions must be distributed and as less shadow as possible, which strongly connected to the theory by Drbohlav and Chantler [3]. We have a dilemma that some people heavily rely on this kind of expertise knowledge but some avoid such indescribable knowledge.

The computer guidance method (online illumination planning) proposes next best light position to the observer [12]. The method first takes three images under pre-defined light positions and then iterates optimum light position prediction and image acquisition. Each prediction in the second step first selects a single pixel maximizing Eq. 10, namely a pixel with worst estimation uncertainty. The optimum light position is computed from the previous observations that simultaneously avoids shadow and puts the light position further from the previously observed light positions. As far as the authors know, the paper is the first work that introduces the concept of (online) view planning into PS image acquisition.

This method has two main drawbacks. The first one is in the optimum light position estimation. The effect of shadow dominates their evaluation cost and the domination tends to select pixels with same/opposite normal vectors located around object boundaries. This tendency results light positions similar to each other. The second drawback is in the pixel selection step. The method predict estimation uncertainty for all the pixels and selects the one with the largest uncertainty. We should care how many pixels a selected light shed. The worst case scenario is that a selected light sheds the pixel but not the remaining ones.

2 Proposed method

This paper proposes a Gaussian Process based illumination planning for shading image acquisition. The basic idea of the method is same as the online illumina-

Algorithm 1 The proposed method

Input: Pre-defined light positions $(\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3)$, potential light position candidates $\mathcal{L} = \{\mathbf{l}\}$, and the maximum number of observations N .

```

1:  $n = 1$ 
2:  $\mathcal{D} = \emptyset$ 
3: while  $n \leq 3$  do
4:    $\mathbf{I}_n = \text{CaptureImage}(\mathbf{l}_n)$ 
5:    $\mathcal{D} = \mathcal{D} \cup \{(\mathbf{l}_n, \mathbf{I}_n)\}$ 
6:    $n += 1$ 
7: end while
8: while  $n \leq N$  do
9:    $\mathbf{l}_n = \arg \max_{\mathbf{l} \in \mathcal{L}} \text{ComputeUncertainty}(\mathcal{D}, \mathbf{l})$ 
10:   $\mathbf{I}_n = \text{CaptureImage}(\mathbf{l}_n)$ 
11:   $\mathcal{D} = \mathcal{D} \cup \{(\mathbf{l}_n, \mathbf{I}_n)\}$ 
12:   $\mathcal{L} = \mathcal{L} \setminus \mathbf{l}_n$ 
13:   $n += 1$ 
14: end while
```

tion planning by Tanikawa et al. [12]. Algorithm 1 shows the procedure of the proposed method. The method (1) first chooses three pre-defined light positions and observes images independently, (2) selects representing pixels by considering their correlation given the observed three images, and (3) iterates optimum light selection and observation until some condition is satisfied.

The key idea is from Gaussian Process based sensor location problem [5]. By formulating target observation by a Gaussian Process [9], we can compute the uncertainty of unobserved light positions. Coupling with an approximation algorithm, the quality of light position selection is guaranteed with a constant factor. The proposed method considers the uncertainty of unobserved light positions and selects the one that decreases the uncertainty the most. Considering the uncertainty of unobserved light positions at some pixels¹, the proposed method can select as distributed light positions as possible.

2.1 Observation uncertainty of a pixel

This subsection defines the observation uncertainty of a pixel.

Assuming that shading image acquisition follows a Gaussian Process as

$$\begin{bmatrix} i_1 \\ i_2 \\ \vdots \\ i_F \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} \mu(f(\mathbf{l}_1)) \\ \mu(f(\mathbf{l}_2)) \\ \vdots \\ \mu(f(\mathbf{l}_F)) \end{bmatrix}, \begin{bmatrix} k(\mathbf{l}_1, \mathbf{l}_1) & k(\mathbf{l}_1, \mathbf{l}_2) & \dots & k(\mathbf{l}_1, \mathbf{l}_F) \\ k(\mathbf{l}_2, \mathbf{l}_1) & k(\mathbf{l}_2, \mathbf{l}_2) & \dots & k(\mathbf{l}_2, \mathbf{l}_F) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{l}_F, \mathbf{l}_1) & k(\mathbf{l}_F, \mathbf{l}_2) & \dots & k(\mathbf{l}_F, \mathbf{l}_F) \end{bmatrix} \right), \quad (11)$$

$$\mathbf{i} \sim \mathcal{N}(\mu(\mathbf{L}), \mathbf{K}) \quad (12)$$

¹ Note that the proposed method considers the limited number of representing pixels but not all pixels.

where μ denotes the mean function, $f()$ the observation function, and k kernel function represents the similarity between two light positions. Utilizing the property of Gaussian distribution, we can formulate a joint probability of the existing observation and one under a new light position \mathbf{l}_* as

$$\begin{bmatrix} \mathbf{i} \\ i_* \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} \mu(\mathbf{L}) \\ \mu(\mathbf{l}_*) \end{bmatrix}, \begin{bmatrix} \mathbf{K} & \mathbf{k}_* \\ \mathbf{k}_*^\top & \mathbf{K}_* \end{bmatrix} \right), \quad (13)$$

$$\mathbf{k}_* = (k(\mathbf{l}_*, \mathbf{l}_1), k(\mathbf{l}_*, \mathbf{l}_2), \dots, k(\mathbf{l}_*, \mathbf{l}_F))^\top \quad (14)$$

$$\mathbf{K}_* = k(\mathbf{l}_*, \mathbf{l}_*). \quad (15)$$

Thus, the mean $\mu(\mathbf{l}_*)$ and variance $\sigma(\mathbf{l}_*)$ of the new light position \mathbf{l}_* is

$$\mu(\mathbf{l}_*) = \mathbf{k}_*^\top \mathbf{K}^{-1} \mathbf{i} \quad (16)$$

$$\sigma(\mathbf{l}_*) = \mathbf{K}_* - \mathbf{k}_*^\top \mathbf{K}^{-1} \mathbf{k}_*. \quad (17)$$

Thus, we can select the next best light position that reduces the observation uncertainty.

The key factor of this model is the definition of the kernel function $k()$. With the linear kernel

$$k_{\text{linear}}(\mathbf{l}_i, \mathbf{l}_j) = \mathbf{l}_i \cdot \mathbf{l}_j \quad (18)$$

the uncertainty behaves similar to Eq. 7. The proposed method uses RBF kernel as

$$k_{\text{rbf}}(\mathbf{l}_i, \mathbf{l}_j) = \exp \left(-\frac{|\mathbf{l}_i - \mathbf{l}_j|}{\theta} \right) \quad (19)$$

2.2 Observation uncertainty of an image

This subsection extends the observation uncertainty model from a pixel to an image. The proposed method utilizes a multi-task GP model [1] that computes multiple output values, multiple pixel intensities in other words, for each input light vector. To reduce memory usage, the proposed method computes the observation uncertainty of some pixels, 16 in the experiment.

For P pixels, Eq. 12 is extended to

$$\begin{bmatrix} \mathbf{i}_1 \\ \vdots \\ \mathbf{i}_P \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} \mu_1(\mathbf{L}) \\ \vdots \\ \mu_P(\mathbf{L}) \end{bmatrix}, \mathbf{K}_l \otimes \mathbf{K}_p \right), \quad (20)$$

$$\mathbf{K}_l = \begin{bmatrix} k_l(\mathbf{l}_1, \mathbf{l}_1) & k_l(\mathbf{l}_1, \mathbf{l}_2) & \dots & k_l(\mathbf{l}_1, \mathbf{l}_F) \\ k_l(\mathbf{l}_2, \mathbf{l}_1) & k_l(\mathbf{l}_2, \mathbf{l}_2) & \dots & k_l(\mathbf{l}_2, \mathbf{l}_F) \\ \vdots & \vdots & \vdots & \vdots \\ k_l(\mathbf{l}_F, \mathbf{l}_1) & k_l(\mathbf{l}_F, \mathbf{l}_2) & \dots & k_l(\mathbf{l}_F, \mathbf{l}_F) \end{bmatrix} \quad (21)$$

$$\mathbf{K}_p = \begin{bmatrix} k_p(\mathbf{x}_1, \mathbf{x}_1) & k_p(\mathbf{x}_1, \mathbf{x}_2) & \dots & k_p(\mathbf{x}_1, \mathbf{x}_P) \\ k_p(\mathbf{x}_2, \mathbf{x}_1) & k_p(\mathbf{x}_2, \mathbf{x}_2) & \dots & k_p(\mathbf{x}_2, \mathbf{x}_P) \\ \vdots & \vdots & \vdots & \vdots \\ k_p(\mathbf{x}_P, \mathbf{x}_1) & k_p(\mathbf{x}_P, \mathbf{x}_2) & \dots & k_p(\mathbf{x}_P, \mathbf{x}_P) \end{bmatrix} \quad (22)$$

where \otimes denotes Hadamardt product of two matrices, \mathbf{K}_l and \mathbf{K}_p denote the covariance matrices of light vector and pixel locations, $k_l()$ and $k_p()$ denote the kernel function for light vectors and pixel locations, and \mathbf{x}_p denotes the pixel location of a pixel p . Similar to the single pixel case, we can derive the uncertainty of an unobserved light vector \mathbf{l}_* as

$$\Sigma(\mathbf{l}_*) = \mathbf{K}_* - (\mathbf{K}_l \otimes \mathbf{k}_*)^\top (\mathbf{K}_l \otimes \mathbf{K}_p)^{-1} (\mathbf{K}_l \otimes \mathbf{k}_*)^\top. \quad (23)$$

As mentioned at the beginning of this section, the proposed method considers the observation uncertainty of some pixels but not entire image. One critical drawback of the proposed method is its memory usage. GP computes the inverse matrix whose computational cost is proportional to $(PF)^3$. It is redundant to consider all the pixels because neighboring pixels have similar information. Therefore, the proposed method selects representing pixels from all the pixels and uses the observation uncertainty obtained from the selected pixels using Eq. 23. In this paper, we randomly select 16 pixels from entire image.

2.3 Optimum light position from observation uncertainty

Here, we describe a method selecting the next best light position from the observation uncertainty. For finite light position candidates $\mathcal{L} = \{\mathbf{l}_1, \dots, \mathbf{l}_N\}$, we select a light vector that satisfies the following condition

$$\hat{\mathbf{l}} = \arg \max_{\mathbf{l} \in \mathcal{L}} \Sigma(\mathbf{l}). \quad (24)$$

Note that the proposed method solves illumination planning as a discrete optimization. Under the selected light vector $\hat{\mathbf{l}}$, we put the light at the corresponding position and acquire a shading image. Finally, we recover surface normal by applying a Photometric Stereo method.

3 Experiment

We validated the effectiveness of the proposed method. In this experiment, we used an image acquisition simulator that renders a shading image given a light vector. We assumed that the target object has Lambertian reflection and uniform albedo object and is lit by a distant light source and no cast shadow appears. The rendering engine adds white Gaussian noise with standard deviation of 0.05 for pixel intensities ranging 0.00 to 1.00. As target object, we used Stanford bunny and Lucy. We set the number of image acquisition to 14 and ran the proposed method, Tanikawa’s method, and random selection. In the proposed method, both $k_l()$ and $k_p()$ were RBF kernels with different hyper-parameters. The hyper-parameter is optimized by Adam optimizer with learning rate 0.1 and all the implementation is done by GPyTorch [4]. In Tanikawa’s method, our implementation was the discrete optimization version to avoid local minimum. For fair comparison, first three light positions are same for all the methods. As Photometric Stereo solver, we used both L1 residual minimization and Sparse Bayesian learning method of RPS [7]².

3.1 Selected light position

Fig. 1 compares the selected light positions of all the methods, we call illumination planning map hereafter for Standard bunny case. A big gray circle represents all the potential light positions and position (x, y) in the circle represents light vector $(x, y, 1 - \sqrt{x^2 + y^2})$. Each selected light is depicted as a colored circle and the color of the circle represents the selected order.

Comparing the planning maps, our method selected light positions uniformly distributed on a sphere while Tanikawa’s method selected less distributed light positions. On the other hand, Random selection selected half distributed and half concentrated light positions. From the result, our method favors spatial distribution of light positions while Tanikawa’s method seems to put too much weight on shadow.

3.2 Photometric Stereo accuracy

We conducted quantitative evaluation among the methods. For each object, we ran each method with the maximum number of observations $N = 15$. For each observation, we computed the mean value of the angular difference in degree between the estimated normal map and its ground truth. For Random selection, we ran 20 times and computed their mean and standard deviation. Figure 2 shows the angular difference w.r.t. the number of observations for Stanford bunny and Lucy. In general, all the methods obtain smaller error for more observations. For some cases, we obtain larger error with more observations that is due to observed shadow. The reason why both the illumination planning results worse

² Note that RPS works worse for pixels dominated by shadows while one used in the evaluation of Tanikawa’s paper [12] avoid such violation from shadow.

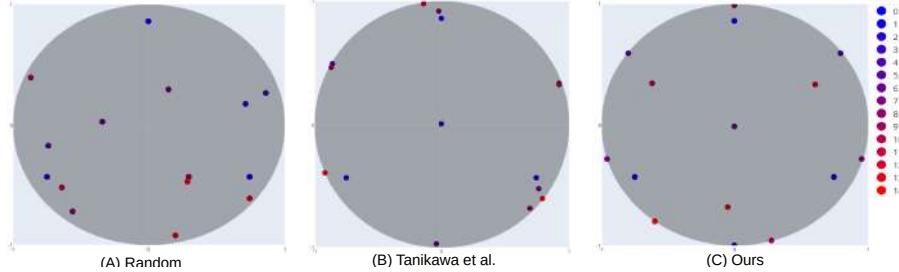


Fig. 1. Experimental result (illumination planning map): (A) Random selection, (B) Tanikawa, and (C) Ours.

error with more images is that they select light position is further to camera direction, which is the border of illumination map. When light directs further from the camera direction, attached shadow appears on more pixels. Those shadow worsen normal estimation quality.

Fig. 3 shows the ground truth and the estimated normal maps with each selected 14 lights. The similarity to the ground truth gives same impression that Tanikawa's method is less similar to the ground truth while Random selection and our method estimates closer normal map to the ground truth.

4 Conclusion

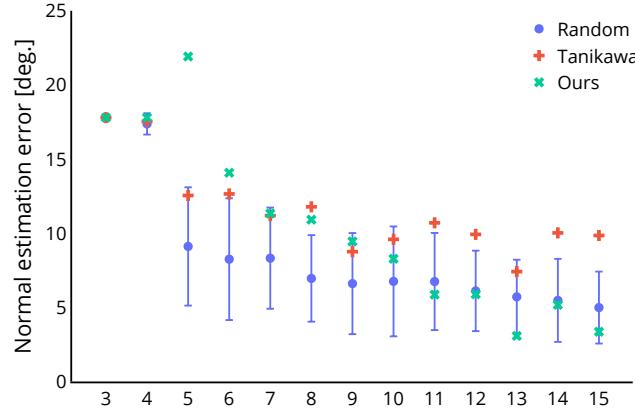
This paper tackles the illumination planning problem for Photometric Stereo. Our idea is to select light position that has the largest observation uncertainty given current observations. The positive effect of the idea is that the selected light positions are distributed more than the existing method. However, the proposed method do not outperform Random selection, Random selection outperforms the proposed method for less number of observations. This unwilling result might happen because of the following factors. First of all, we consider the observation uncertainty of very limited number of pixels to reduce memory usage and the selection is randomly done. To follow this manner, we must consider dependency among the selected pixels that well-represent entire image w.r.t. surface normal. Another factor is to consider camera direction for light position selection. As mentioned above, light position further from the camera direction worsen normal estimation quality. This expertise must be considered.

Acknowledgments

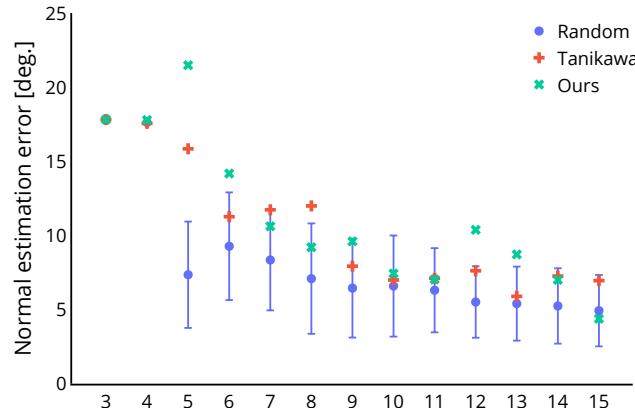
The author would like to thank the anonymous reviewers for giving advises for updating the experiments. This work was partially supported by JSPS KAKENHI Grant Number 19K20296.

References

1. Bonilla, E.V., Chai, K.M.A., Williams, C.K.I.: Multi-task gaussian process prediction. In: Proceedings of Advances in Neural Information Processing Systems (NIPS). pp. 153—160 (2007)
2. Chen, G., Han, K., Shi, B., Matsushita, Y., Wong, K.Y.K.: Deep photometric stereo for non-lambertian surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* **44**(01), 129–142 (2022)
3. Drbohlav, O., Chantler, M.: On optimal light configurations in photometric stereo. In: IEEE International Conference on Computer Vision (ICCV). vol. 2, pp. 1707–1712 (2005)
4. Gardner, J.R., Pleiss, G., Bindel, D., Weinberger, K.Q., Wilson, A.G.: Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In: Proceedings of International Conference on Neural Information Processing Systems (NIPS). pp. 7587–7597 (2018)
5. Guestrin, C., Krause, A., Singh, A.P.: Near-optimal sensor placements in gaussian processes. In: Proceedings of the International Conference on Machine Learning (ICML). pp. 265—272 (2005)
6. Ikehata, S.: Cnn-ps: Cnn-based photometric stereo for general non-convex surfaces. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Proceedings of European Conference on Computer Vision (ECCV). pp. 3–19 (2018)
7. Ikehata, S., Wipf, D., Matsushita, Y., Aizawa, K.: Robust photometric stereo using sparse regression. In: Proceedings of Computer Vision and Pattern Recognition (CVPR) (2012)
8. Ikehata, S., Wipf, D.P., Matsushita, Y., Aizawa, K.: Photometric stereo using sparse bayesian regression for general diffuse surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* **36**(9), 1078–1091 (2014)
9. Rasmussen, C.E., Williams, C.K.I.: Gaussian processes for machine learning. MIT Press (2006)
10. Santo, H., Samejima, M., Sugano, Y., Shi, B., Matsushita, Y.: Deep photometric stereo network. In: 2017 IEEE International Conference on Computer Vision Workshops (ICCVW). pp. 501–509 (2017)
11. Szeliski, R.: Computer Vision: Algorithms and Applications, chap. 3D Reconstruction. Springer (2022)
12. Tanikawa, H., Kawahara, R., Okabe, T.: Online illumination planning for shadow-robust photometric stereo. In: International Workshop on Frontiers of Computer Vision (IW-FCV). pp. 94–107 (2022)
13. Woodham, R.J.: Photometric Method For Determining Surface Orientation From Multiple Images. *Optical Engineering* **19**(1), 139–144 (1980)
14. Wu, L., Ganesh, A., Shi, B., Matsushita, Y., Wang, Y., Ma, Y.: Robust photometric stereo via low-rank matrix completion and recovery. In: Proceedings of Asian Conference on Computer Vision (ACCV) (2010)



(a) Stanford bunny



(b) Lucy

Fig. 2. The angular error in degree w.r.t. the number of observations. Blue circle with error bar represents Random selection, red cross Tanikawa's method, and green x ours.

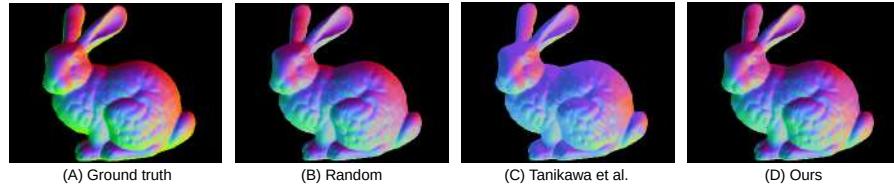


Fig. 3. Experimental result (estimated normal map): (A) Ground truth, (B) Random selection, (C) Tanikawa, and (D) Ours.

Data Generation and Deep Learning network for Micro Defect Detection

Byungjoon Kim^{1[0000-0003-0347-2776]} and Yongduek Seo^{1[0000-0002-0570-2197]}

¹ Sogang University, Seoul, 04090, South Korea
 (atbem, yndk)@sogang.ac.kr

Abstract. Object detection algorithms using deep learning have been applied in various fields. So, there are many improvements and achievements through research of vision system in each field. In order to apply deep learning to the relevant field, it is necessary to define problem and generate proper dataset. In particular, it is difficult to obtain sufficient data of OLED panel defects due to the high cost of OLED panels. This study is to define OLED panel defects, and build virtual data acquisition system for learning. In each generated image data for defect detection, defects were classified and labeled by defect types. After that, the data were applied to the RCNN-based object detection algorithm to learn the defect detection model of the OLED panel and evaluate its performance.

Keywords: Defect Detection, Data Building System, Micro Defect

1 Introduction

OLED panels have become mainstream in display devices because they are lighter and have better contrast ratios compared to LCDs. As the demand for OLED panels increases, the demand for panel inspection is also increasing. In the defect detection using the existing vision system, the corresponding panel had to be accurately positioned at specific location. Therefore, it is difficult to set up detection system for micro defect detection, and it takes a lot of money to debug errors when they occur. On the other hand, object detection based on deep learning is being applied to various fields. However, newly applied fields require a clear definition of the problem and sufficient data set for model learning.

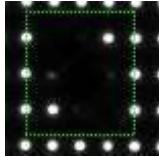
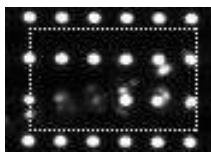
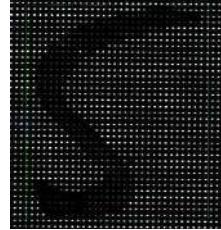
In this paper, Micro defect detection for OLED panels is performed. For detection, it is necessary to define the defects of the panel and create a dataset with the corresponding defective panels. However, due to the high cost of OLED panels, it is difficult to obtain panels with various type of defects, and it has high cost to obtain actual panels. Therefore, in this study, we implement a data generation system and propose an micro defect detection model using detection algorithm.

2 Data Acquisition System

2.1 Classification of Micro Defects

Before generating data, it is necessary to define and classify defects to be detected in OLED panels. Defects is defined to be classified into four types: Open(pixel omission), Short(pixel connection), Line-out(pixel line out), Foreign(foreign body on panel).

Table 1. Definition example of OLED micro defect

Open (74 × 81)	Short (115 × 71)	Line-out (222 × 605)	Foreign body (469 × 680)
			

Open(pixel omission) defect is an area in which a pixel is missing, and is a defect in which a specific pixel of the panel is open. So, open defect is defined as when three or more adjacent pixels within a panel are missing. Short(pixel connection) is defined as defect due to a short circuit between panel pixel, or bleeding on the panel. Line-out occurs due to a mask defect in OLED panel deposition. Lastly, Foreign is defined as a foreign object on the panel during inspection as a defect having a shape or an irregular pattern with a width of 4 pixels or more.

2.2 Data building and Preprocessing

OLED panel defect data is needed to generate data for learning. However, not only is the cost of OLED panels high, but it is also difficult to obtain a panel with various defects sufficient to learn. In order to overcome the difficulty in obtaining data, defect data is generated by outputting a virtual defect image on a panel.

A high-resolution image sensor and frame grabber are used to acquire pixel defect feature of OLED panel where virtual defects are output without loss. The panel used to generate the data has a resolution of 1080×2400 . A monotype image of 14192×10640 resolution is obtained using the corresponding panel. (see Fig. 1). The acquired raw image of the panel have fine feature of defects on the panel at high magnification.

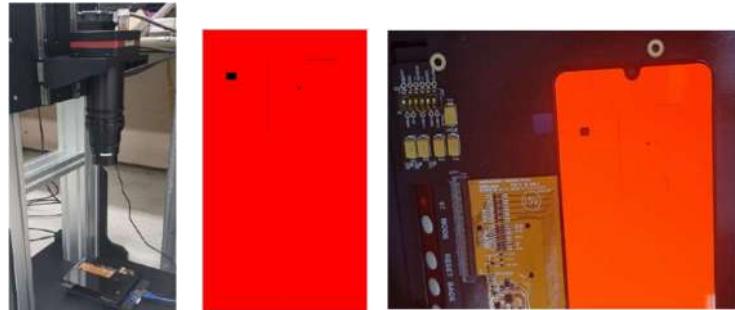


Fig. 1. Artificial Defect Device(left), Example of virtual defect image source(center), actual output on OLED panel(right).

In the generated example image, two types of defects were generated: Open and Line-out. When the panel on which the defect image is output is obtained with the device described above, a resolution of 14192×10640 is acquired.

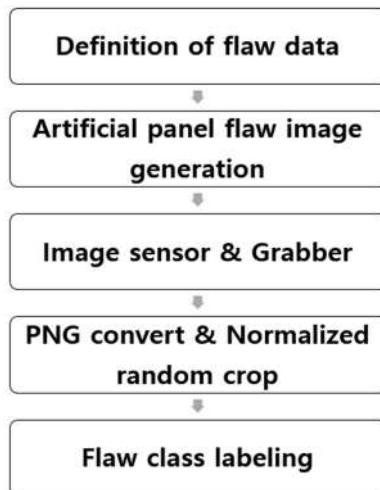


Fig. 2. Data building of OLED defect dataset

2.3 Defect Image Dataset

The acquired image has more than 150 million pixels and feature information of each panel pixel without loss. However, it is not suitable as an input for object detection model. For micro defect detection, a high magnification image that does not lose feature information of defect and an appropriate resolution that can be suitable learned by a network are required. To satisfy these two conditions, normalization is performed after converting the raw image to PNG. The converted image is cropped to 1024×1024 and

applied to the training model at the original magnification without loss of feature. (see Fig. 3)

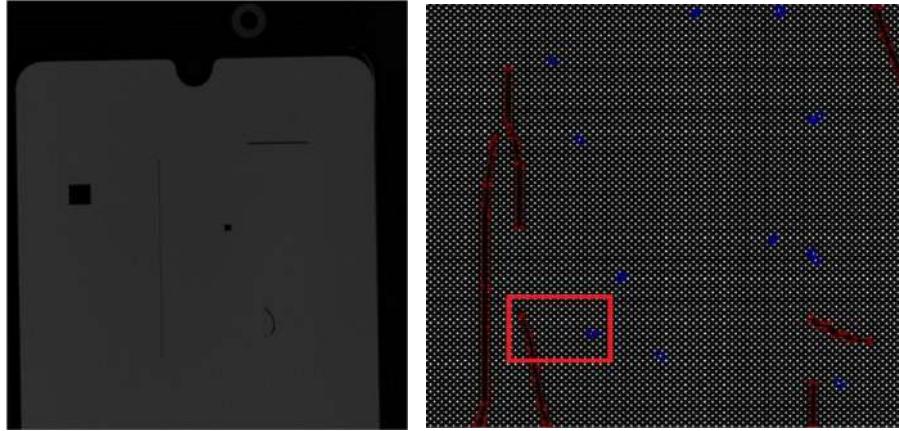


Fig. 3. High resolution (14192×10640) raw image(left), Preprocessed (1024×1024) dataset image(right)

3 Experimental Results and Analysis

The dataset consists of 240 images. 204 images are used as training and 36 images are used as test dataset. Based on Faster-RCNN, a box-based object detection algorithm, we estimate micro defect in pixel units based on area by adding a mask branch that predicts a segmentation mask to ROI recommended by RPN (Region Proposal Network). In general, ResNet-101 is used as a backbone network to improve performance, but high-resolution images with simple patterns, it was confirmed that the performance decreases as the layer deepens and the operation speed slows down neither. (see Table 2.)

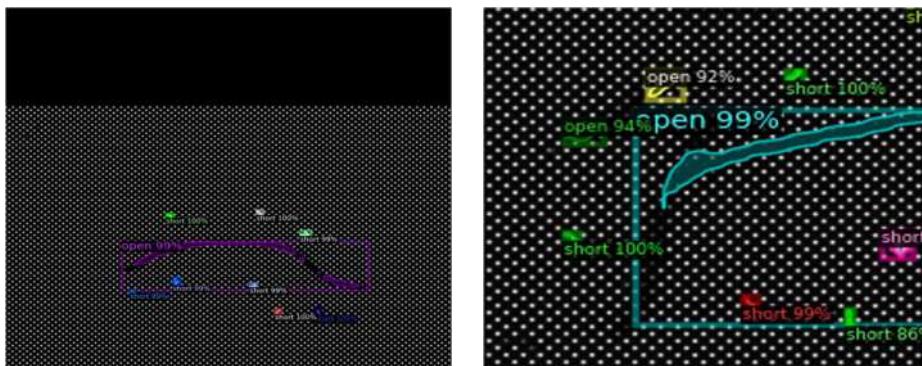
Table 2. performance comparison of backbone network

Backbone	VGG-19	ResNet-50	ResNet-101
Avg Time(ms)	92	73	109
Acc (%)	94.7	98.4	97.8

The detection rate is evaluated by the number of detections compared to the number of defective objects in the actual image. To summarize the overall results, one false positive occurred in the open defect, and most of the remaining errors were false negatives. The overall detection rate is 98.4% based on 375 total defective objects. Details are shown in the table below. (see Table 3).

Table 3. Detection rate by defect type

Defect Type	Detection(ea)	Defect(ea)	Ratio
Open	113	116	109
Short	213	215	97.8
Line-out	4	4	100
Foreign body	39	40	97.5

**Fig. 4.** Example of Test image (1024×1024) (left), Example of high-magnification(x9) detection (right)

4 Conclusion

In this study, in order to solve high cost for OLED panel, a virtual defect image is created and replaced with a system using a suitable sensor and device. In addition, to train the object detection network, a data set is designed by defining panel defects and classifying defect types. The generated defect dataset consists of images containing four types. In order to detect the defined defects, algorithm based on RCNN was applied to the generated data. The OLED panel defect detection model trained with the generated data obtained 98.4% detection ratio. In the future, it is expected to improve the OLED panel defect detection model by redefining the data and further classifying defect types.

Acknowledgement

This work was supported by the Technology development Program (1425166398) funded by Ministry of SMEs and Startups(MSS, Korea)

References

1. R. Girshich, J. Donahue, T. Darrell and J. Malik : Rich feature hierarchies for accurate object detection and semantic segmentation. CVPR 2014, <https://doi.org/10.48550/arXiv.1311.2524>.
2. J. Redmon, S. Divvala, R. Girshick and A. Farhadi: You Only Look Once: Unified, RealTime Object Detection, CVPR 2016, <https://doi.org/10.48550/arXiv.1506.02640>
3. R. Girshick: Faster R-CNN, 2015 IEEE (ICCV), pp. 1440-1448, doi: 10.1109/ICCV.2015.169.
4. J. Long, E. Shelhamer and T. Darrell : Fully Convolutional Networks for Semantic Segmentation, CVPR 2015, <https://doi.org/10.48550/arXiv.1411.4038>.
5. L. Chen, G. Papandreou, F. Schroff and H. Adam : Rethinking Atrous Convolution for Semantic Image Segmentation, CVPR 2017, <https://doi.org/10.48550/arXiv.1706.05587>.
6. K. Simonyan and A. Zisserman: Very Deep Convolutional Networks for Large-Scale Image Recognition, CVPR 2015, <https://doi.org/10.48550/arXiv.1409.1556>.
7. Vision System Technology, LLC. Products, <http://www.visionsystech.com>, last accessed 2022/10/06.

Remark

This paper is a re-publishing of the paper which has been published in “The Journal of Korean Institute of Information Technology” by request of IW-FCV2023 program committee to share the research results.

Classifying Breast Cancer Using Deep Convolutional Neural Network Method

Musfequa Rahman^{1[0000-0002-3859-8832]}, Kaushik Deb^{1[0000-0002-7345-0999]}, and Kang Hyun Jo^{2[0000-0001-8317-6092]}

¹ Department of Computer Science and Engineering, Chittagong University of Engineering & Technology (CUET), Chattogram 4349, Bangladesh

u1704050@student.cuet.ac.bd, debkaushik99@cuet.ac.bd,

² Department of Electrical, Electronic and Computer Engineering, University of Ulsan acejo@ulsan.ac.kr

*Correspondence: debkaushik99@cuet.ac.bd

Abstract. The efficacy of conventional classification systems is contingent upon the accurate representation of data and a substantial portion of the effort invested in feature engineering, which is a laborious and timeconsuming process requiring expert domain knowledge. In contrast, deep learning has the capacity to automatically identify and extract discriminative information from data without the need for manual feature creation by a domain expert. In particular, Convolutional Neural Networks (CNNs), a type of deep feedforward network, have garnered attention from researchers. This study conducts several preliminary experiments to classify breast cancer histopathology images using deep learning, given the small number and high resolution of training samples. The proposed approach is evaluated on the publicly available BreakHis dataset, utilizing both a scratch model and transfer learning pre trained models. A comparison of the proposed scratch method to alternative techniques was carried out using a suite of performance evaluation metrics. The results indicate that the scratch model, with its independent magnification factor, achieved greater accuracy, with a binary classification accuracy of 99.5% and a multiclass classification accuracy of 96.1%.

Keywords: Transfer Learning, Convolutional Neural Network, Magnification Factor, Breast Cancer Classification

1 Introduction

Currently, one of the leading causes of human death is cancer which is cell growth of type abnormal that the invading body parts have high potential. Precancerous lesions give way to malignant tumors as part of the multi phase process by which cancer cells are transformed. Alcohol and cigarette use, physical inactivity, old age, pollution, and a few additional disorders like Hepatitis C, Hepatitis B and HIV are all risk factors for the development of cancer. World Health Organization (WHO) states that according to their estimation, here will be 10.6 million cancer related deaths and 19.3 million new cases worldwide in 2020 [1].

Any part of the human body can be affected by cancer cells including liver, lungs, breast etc. Among other types of cancer, breast cancer is one of the most common for women and the mortality of breast cancer is also very high, accounting for 1 in 4 new cases and 1 in 6 cancer related deaths worldwide in [1]. Breast cancer, which accounted for 35.3% of all fresh tumors of female and contributes 20.8% to cancer deaths in total in 2012, had the highest age standardized incidence of any female cancer, according to data from the International Agency for Research on Cancer (IARC) of the WHO states [1]. In 2020, study shows of 27 million fresh cases of cancer occurred [2]. Breast tissue cells proliferate out of control and infiltrate adjacent tissues via blood and lymphatic vessels to cause breast cancer. Breast cancer can also occur in fatty tissue. Breast discomfort, skin that is pitted and red or discolored on the breast, lumps or tissue thickening that feels different from surrounding tissue, and breasts that enlarge entirely or partially are typical symptoms of breast cancer. Breast tissue, which is detected by a breast lump, develops into breast cancer, along with other modifications to the usual environment [3]. Mammography, breast cancer screening and other clinical Ultrasound, biopsies and other techniques. Only a biopsy [4] can definitively verify whether

the suspicious region is malignant in terms of diagnosis. The pathologists make their diagnoses by looking at histology slides, which is regarded as the definitive gold standards. However, the traditional method requires a heavy burden from qualified experts. Pathologists who lack sufficient diagnostic experience are more likely to make errors in diagnosis.

Deep learning algorithms have obtained results on image classification and object detection tests that are on par with those of human experts [5]. The most popular deep learning framework for learning complicated discriminative characteristics between image classes is the convolutional neural network. Over the years, many CNN architectures have delivered outstanding results on the enormous ImageNet dataset. On medical images, CNNs are being used to produce state-of-the-art results. Patch wise classification is one of the existing deep learning methods for the task of classifying breast cancer (BC) histology images [6]. By doing this, CNN typically ignores the general properties of the entire tissue and only extracts local features near the nucleus. In addition to the drawback of patching, CNN's shallow architecture does not allow for the extraction of finer and more abstract features from patient breast histopathology images.

In order to gradually and reliably categorize the Breast Cancer pathological images, we developed a CNN model from scratch to extract characteristics from images and carry out the training which is end to end in nature. Hence, the following is a list of this paper's key contributions:

1. To increase classification performance, we developed a scratch CNN model.
2. Because of the independence of our developed model's accuracy against magnification factors, it can be used with different magnification factors.
3. To improve the quality of the breast histopathology images, we developed a histopathological image enhancement method.

2 Related Works

The original goal of the image analysis system was to categorize pathological images. For more than 40 years, this concept has been investigated in the context of automatic assistance cancer detection. The complexity of image analysis, however, made it difficult to deal with the inherent complexity of histological images [7]. The workload of pathologists can be reduced by modern deep learning method [8]. The absence of extensive datasets and class disparity are the key challenges in the field of breast cancer classification research. Since images have intrinsic problems including inadequate contrast, noise, and lack of visual acuity, models have been developed to build and improve image processing.

For the recognition of breast cancer histopathology images, several researchers apply customized features. BreaKHis, dataset of breast available to public, was proposed by [9]. Six different features utilized the dataset's classification and accuracy ranged from percentage of 80 to 85. Phylogenetic diversity indexes were employed by Carvalho et al. In [10], to categorize the different forms of breast cancer. Three different features were combined by [11] for binary and eight class categorization of breast cancer histopathology images. Several researchers, notably CNNs, have become interested in deep learning as a result of its remarkable performance in image recognition in recent years.

Researchers created CNNs based CAD models based on these models and used them to diagnose cancers. There are two general categories of CNNs as pre trained CNNs [12] and CNNs which have been made from begin called scratch [13]. Additionally, deep networks can be created using transfer learning with samples which are few in number. Many researchers additionally employ CNN an extractor of features, which performs the task of extraction of features using various techniques.

For instance, in [14] employed 3-norm for feature fusion, ResNet50 for feature extraction from image patches of various sizes, and SVM for classification. On the basis of images produced from Haar wavelet decomposition, in [15] retrieved features using VGG16 and merged various properties of the next level layers for breast cancer identification. To assess the histopathology images and resolve the class imbalance issue. In [16], used ResNet50 and weighted learning machine. Using DenseNet121 [17] to extract overall features from slides. In order to diagnose breast and prostate cancer automatically, in [18] developed a 5-layered multi input CNN that took into account both RGB pictures and phase shearlet coefficients. This CNN achieved an amazing accuracy rate of 88% for a different dataset. In [19], a thorough analysis of the architecture and functioning of each network is performed and the performance of each network is then evaluated based on how accurately it diagnoses and classifies cancer causes in breast. CNN provide a little bit better precision than layer perceptron for the detection.

As a result, it can be concluded that the classification of cancer in breast has had a significant impact for a long time. Deep learning models combined with a wide range of configurations have recently exceeded current state-of-the-art methods, as well. There is a huge amount of scope for initiating innovation and development in this developing research field to overcome this.

3 Datasets

3.1 BreakHis

The images found from biopsy cancers in breast were gathered from January 2014 to December 2014 through clinical investigations and were included in the BreaKHis dataset [20]. All patients with breast cancer clinical symptoms to take part in the trial over the time period. Hematoxylin and eosin staining was used after surgical open biopsy (SOB) sample collection. Pathologists working in the PD laboratory can mark these images and use them for histological investigations. Four sub classes are further separated into each kind. Adenosis (A), Fibroadenoma (F), Phyllodes Tumor (PT) and Tubular Adenoma (TA) are examples of Benign lesions, while Ductal Carcinoma (DC), Lobular Carcinoma (LC), Mucinous Carcinoma (MC) and Papillary Carcinoma (PC) are examples of Malignant lesions. The images are 700x460 pixels in size and were created in a three channel RGB (red, green, and blue) true color space with magnifications of 40X, 100X, 200X and 400X. The distribution of images is summarized in Table 1.

Table 1: BreaKHis Dataset Summary

Magnification Factor	Benign	Malignant	Total
40X	625	1370	1995
100X	644	1437	2081
200X	623	1390	2013
400X	588	1232	1820
Total	2480	5429	7909

4 Methods

4.1 Preprocessing

Preprocessing is a necessary step in order to improve the performance for any type of breast cancer histopathological image classification model. In this work, a method was developed for improving histopathological images in low light.

Histopathological image enhancement method Due to their poor visibility, low light images are not suitable for computer vision algorithms or human inspection. Image enhancement is the

process of focusing attention to details that are obscured in an image or enhancing contrast in low contrast images. To provide an accurate contrast enhancement, we developed an image contrast enhancement algorithm. To synthesize multi exposure images and determine the best exposure ratio, a weight matrix for image fusion using illumination estimation techniques is used first, followed by a response model and help to ensure that the synthetic image is properly exposed in the areas where the original image was underexposed. To produce the enhancement result, both types of images are finally fused in accordance with the weight matrix. The preprocessing method for enhancing the histopathological images are shown in Figure 1. Illumination estimation techniques to obtain the weight matrix for image fusion. Exposure ratio is used so that the synthetic image is well exposed in the regions where the original image is under exposed. The output for enhancing the histopathological images is shown in Figure 2. So, the it can be,

$$E = I \times W + I' \times (1 - W) \quad (1)$$

Where W indicates the weight matrix and I indicates real image and E indicates the enhanced image and then I' indicates the exposure image. For breast histopathology slides, MIRNet [21] was compared with our approach of histopathological image enhancement. Using the metrics for measuring the quality of an image.

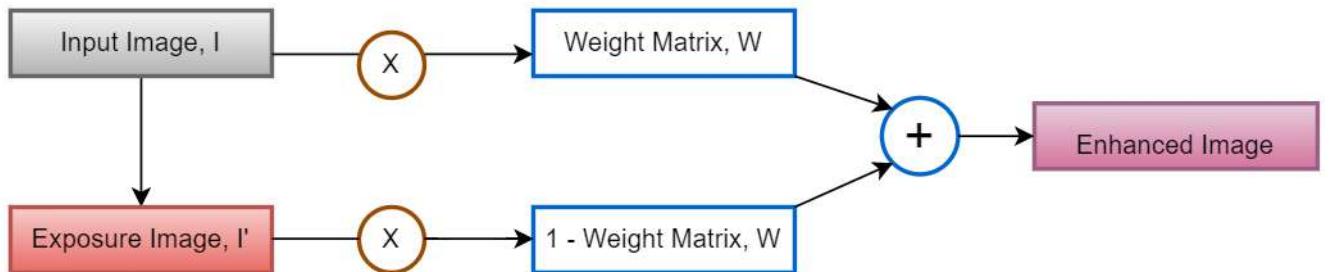


Fig. 1: Histopathological Image Enhancement Method.

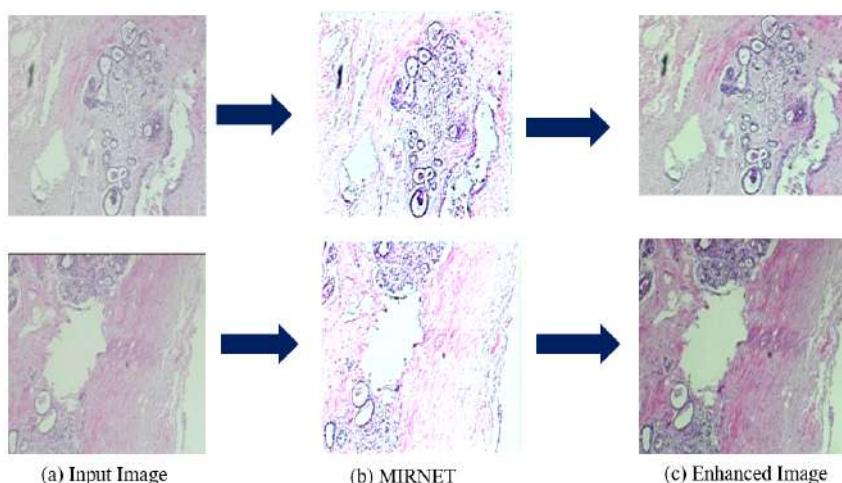


Fig. 2: Histopathological Image Enhancement Method output from. (a) Input Image. (b) MIRNET. (c) Enhanced Image.

Entropy measures an image's content and indicates how much uncertainty or randomness it has. Higher value of entropy indicates an image with higher details.

$$E(I) = - \sum_{k=0}^{L-1} p(k) \log_2(p(k)) \quad (2)$$

Peak Signal Noise Ratio (PSNR) affects how an image is represented, is the ratio between the maximum power and corrupting noise. The PSNR is frequently used to evaluate how well an image may be reconstructed. The original data in this case is the signal, while the introduced error is the noise.

$$PSNR = 10 \log_{10} \frac{\text{MAX}_i^2}{MSE} \quad (3)$$

$$PSNR = 20 \log_{10} \frac{\text{MAX}_i}{MSE} \quad (4)$$

$$PSNR = 20 \log_{10} \text{MAX}_i - 10 \log_{10} MSE \quad (5)$$

Where, MAX_i is the maximum possible pixel value of the image. MSE is Mean Square Error between the filtered image and the original image.

Similarity Index (SI) is ratio of pixels in the enhanced image that coincide with pixels in original image is known as the similarity index. When this is larger than 40 percent when represented in percent or greater than 0.4 when expressed in ones, it is suggested that an improved image be regarded as being comparable to the original image. Lower value (less than 40 percent) of SI indicates good measure. It's measurement is given by-

$$SI = \frac{m_{ab}2xy2m_a m_b}{m_a m_b x^2 + y^2 m_a^2 + m_b^2} \quad (6)$$

Image Quality Index (IQI) is a common metric for comparing the number of pixels that separate two images. The quality of the converted image is considered to be good if the IQI is less than but close to 1 (for example 0.8704). Higher value of IQI indicates good measure. It can be concluded that the proposed histopathological image enhancement method provides high percentage values than MIRNet for the quality image measurement metrics.

$$IQI = 1 - SI \quad (7)$$

Table 2: Enhancement Comparison between MIRNet and Proposed method

Performance Metrics	Input Image	MIRNet	Ours
Entropy	6.004	4.0417	6.7127
PSNR		14.1574	14.6509
SI		67.3573	29.7719
IQI		95.3723	96.7513

4.2 Proposed CNN Architecture

One of the most effective architectures for the problem of image classification is CNNs. CNNs use filtering techniques to extract the most innovative features from an image's pixels. Deep neural networks are used in images as they extract characteristics features from images, as opposed to classic ML algorithms that pick up engineered features for detection of cancer in breast. Machine learning neural networks (ML-NNs) are a type of learning and they typically require a training stage to determine the optimal weights.

CNNs are used to analyze patterns in an image. In the few early layers of CNNs, the network can identify lines and corners. However, as go deeper, these patterns may transfer via neural network and start to recognize more complicated features. CNNs are exceptionally good at identifying objects in images because of this feature. The suggested approach analyzes histopathologic images using CNNs for classification of breast cancer. The convolutional layer, pooling layer, batch normalization layer, and fully connected layer were just a few of the layers that constitute a CNN's architecture, as depicted in Figure 3.

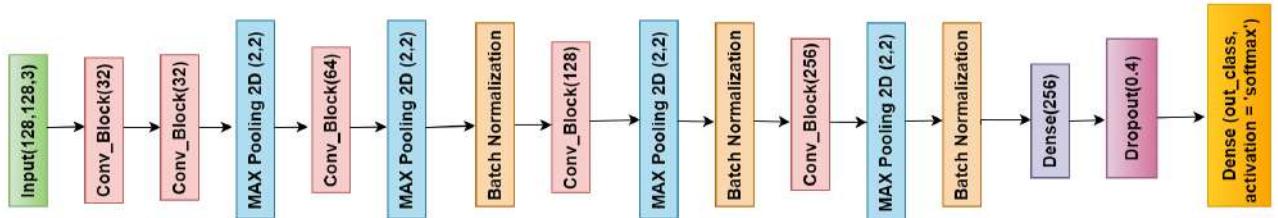


Fig. 3: Proposed Convolutional Neural Network Architecture.

The convolution layer utilizes filters that carry out convolution operations while dimensionally scanning the input image. Convolution is a linear procedure where a set of weights are multiplied and the input images are represented by metrics resembling those of conventional neural networks. All of the features were computed using the input layers and filters and are included in the output, which is known as the feature map or activation map. Here, used RGB images with a size of 128x128 pixels when using this layer. 3x3 kernel size convolutional layers with successive use of 32, 32, 64, 128 and 256 filters. As the activation function, ReLU was employed. The rectifier function was being used to increase the non linearity of the images.

The Convolved Feature's spatial size is decreased by the Pooling layer. By lowering the dimensions, this will decrease the total power required to process the slides. Max Pooling determines a pixel's maximum value from a portion of the image that the kernel has processed. By choosing the maximum value for each input channel over a pool size (2,2) input window, down sampled the input along its spatial dimensions are performed.

In Normalization, the input layers are scaled. Learning becomes more effective when batch normalization is utilized. Thus, batch normalization as regularization to prevent model overfitting is used here. A layer that is densely connected to the layer above it means that every neuron in the layer is attached to every neuron in the layer above it. A typical layer having many connections called dense layer after performing the operation on input returns the result. To do the activation (dot (input, kernel) + bias) operation, this formula is applied in it.

The regularization method used to avoid model overfitting is called dropouts. Dropouts are added to the network's neurons, which are changed at random in some proportion. The connections to the neurons' incoming and outgoing neurons are likewise broken off when they are turned off. To improve the model's learning, this is done. After that, turned off 40 percent of the neurons and utilized dropouts after the network's dense layers. After building the model, classification of the breast histopathology images using the Softmax activation function.

5 Result Analysis

The model configurations are described in this section. There are several accuracy metrics that stand out, including precision, recall and the f1-score of the best model. The classification report, confusion matrix and performance graphs are examined for additional analysis of the best model. The method which down samples the image size to 128x128 so that higher accuracy using 24 sized each batch. However, applied different pretrained transfer learning model in the dataset. The accuracy comparison chart for different transfer learning model and the proposed model as follows:

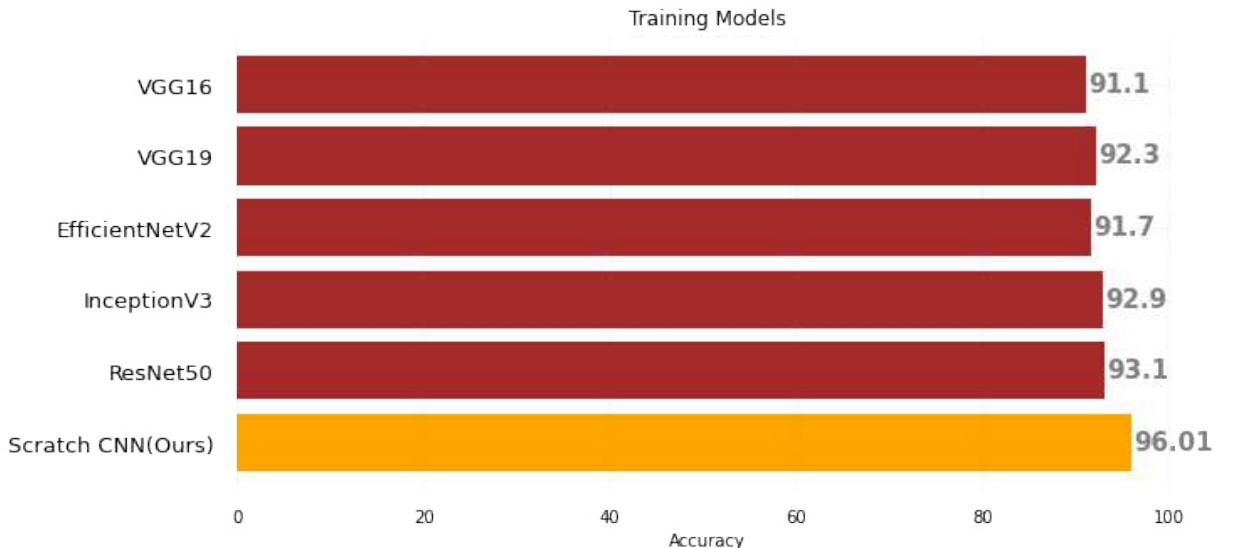


Fig. 4: Accuracy Comparison of Different Models.

An optimizer is a technique that modifies the properties of a neural network, such as a function or algorithm. Accuracy is improved and there is decreased overall loss. Thus, utilizing the following three optimizers in figure 5, Adam, SGD and RMSProp are illustrated. Adam's accuracy in the proposed model was the highest according to the following figure.

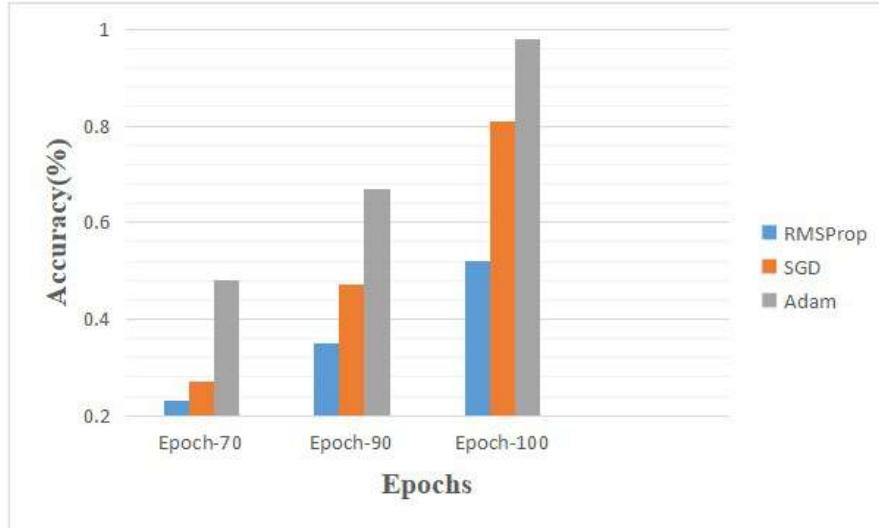


Fig. 5: Optimizer's Accuracy Comparison

The batch size is the quantity of samples processed before to a model modification. The number of epochs is the total number of complete iterations through the training dataset. The minimum and maximum sizes of a batch must be one and the number of samples in the training dataset, respectively. The batch sizes that used for this classification are depicted in figure 6. From the comparison curve, batch size 24 provide highest accuracy among others.



Fig. 6: Accuracy versus Batch Size.

Prior research endeavors were found to be inadequate in their implementation of batch normalization, a technique that accelerates the learning rate of neural networks while simultaneously offering regularization benefits to counteract overfitting. Despite this, subsequent authors opted to utilize hashing in [22], a decision that was made to prevent precision loss, however this approach has not

been implemented. The utilization of a model comprising EfficientNetB2 and RestNet50, as documented in [23], was already implemented in our research however the accuracy achieved was only 93.1 % which was not satisfactory. The implementation of batch normalization in the proposed model was aimed at optimizing the velocity and stability of the training process of artificial neural networks through normalization of the inputs at each layer by rescaling them. Hence, developed a new CNN model from scratch, resulting in a marked improvement in the performance of training data, exhibiting a two times increase in efficiency as compared to the previous efforts documented in [24]. The comparative analysis of the outcome of the classification procedure is presented in the following manner.

For binary classification:

The experiment's accuracy for binary classification was tested by comparing it to the findings of other authors, which are displayed in Table 3. The classification accuracy is 99.10%, 99.15%, 99.22%, and 98.99% respectively for the independence of a very crucial factor called magnification of images by 40, 100, 200 and 400 respectively. The classification result, independent of the magnification factor achieved accuracy of 99.5%.

Table 3: Classification Accuracy Comparison for Binary Class

Author	Model	40X	100X	200X	400X
Pratiher et al., [22]	L-Isomap and SSAEm	96.8	98.1	98.2	97
Bardou et al., [24]	CNN	94.65	98.33	94.07	97.12
	Ensemble CNN model	94.54	97.85	93.77	96.15
Yun Jianget al., [13]	BHCNet-3 + Exp	98.12	98.80	98.88	98.21
	BHCNet-3 + Cos	98.75	98.88	99.17	98.76
Proposed Method	CNN (Scratch)	99.10	99.15	99.22	98.99

For multiclass classification:

The experiment's results for multiclass classification accuracy were measured and compared to those of other authors, as shown in Table 4.

Table 4: Classification Accuracy Comparison for Multi Class

Author	Model	40X	100X	200X	400X
Bardou et al., [24]	CNN	86.34	84.00	79.93	79.74
	Ensemble CNN model	88.23	84.64	83.31	83.98
Yun Jianget al., [13]	BHCNet-3 + Exp	94.43	94.45	92.27	91.15
Abhijeet Patil et al., [25]	A-MIL	82.95	86.45	86.56	84.43
Richa Upadhyay et al., [23]	MPCS-OP (RN-50)	93.00	93.26	92.28	88.74
Proposed Method	CNN (Scratch)	96.36	96.43	96.12	95.91

As seen in the table, the proposed model's accuracy variance for various magnification factors is significantly less than that of other authors and also outperformed them in terms of accuracy. The experiment's results for the independent magnification factor are represented by the classification report in table 5 and confusion matrix and accuracy curve in figure 7, 8.

Table 5: Classification report for multi class

Author	Precision	Recall	F1-score	Support
adenosis	0.94	0.94	0.94	18
ductal carcinoma	0.85	1.00	0.92	22
fibroadenoma	1.00	0.93	0.96	28
lobular carcinoma	1.00	0.90	0.95	20
mucinous carcinoma	0.92	1.00	0.96	24
papillary carcinoma	1.00	0.93	0.97	15
phyllodes tumor	1.00	0.95	0.98	22
tubular adenoma	1.00	1.00	1.00	19

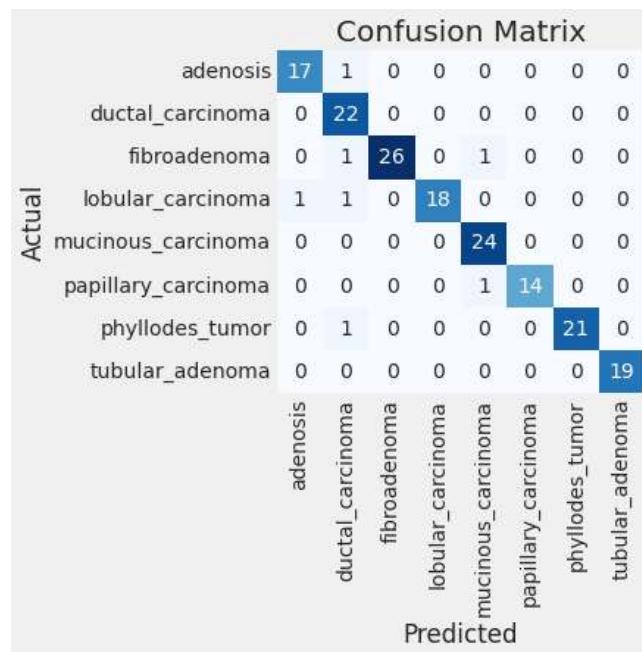


Fig. 7: Confusion Matrix of multi class classification.



Fig. 8: Accuracy Curve of multi class classification.

The approach suggested in this work is reliable to the problems of classification, according to experimental outcomes. The comparison findings with various baseline models identifies that the strategy proposed in the model performs better.

6 Conclusion

This study posits the deployment of a deep learning based network, constructed entirely from scratch, that boasts dense layers in higher level representation, resulting in a marked improvement over conventional classification systems. The proposed scratch method has been subjected to numerous performance evaluations, comparing it with existing technique. It has been found that the model enhance the generalizability and robustness of classification in dealing with imbalanced dataset of breast cancer histopathology images. The deep scratch network exhibits higher accuracy 96.1 % compared to transfer learning models and has the potential to effectively classify both majority and minority classes. Despite the challenges posed by the BreakHis dataset, including low light images and varying magnification factors, there is a need for continuous improvement of histopathology images for better accuracy. The proposed future implementation of attention in the scratch model is aimed at further increasing classification precision and providing a more nuanced classification of histopathology images.

References

1. I. A. for Research on Cancer, "World Fact Sheet," <https://gco.iarc.fr/today/data/factsheets/populations/900-world-fact-sheets.pdf/>, 2020, [Online; accessed 26-June-2022].
2. W. H. O. (WHO), " 20-Breast-fact-sheet," <https://gco.iarc.fr/404>, 2020, [Online; accessed 26-June-2022].
3. R. Karthiga and K. Narasimhan, "Automated diagnosis of breast cancer using wavelet based entropy features," in *2018 Second international conference on electronics, communication and aerospace technology (ICECA)*. IEEE, 2018, pp. 274–279.
4. J. V. Horvat, D. M. Keating, H. Rodrigues-Duarte, E. A. Morris, and V. L. Mango, "Calcifications at digital breast tomosynthesis: imaging features and biopsy techniques," *Radiographics*, vol. 39, no. 2, pp. 307–318, 2019.
5. R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights into imaging*, vol. 9, pp. 611–629, 2018.
6. F. A. Spanhol, L. S. Oliveira, P. R. Cavalin, C. Petitjean, and L. Heutte, "Deep features for breast cancer histopathological image classification," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2017, pp. 1868–1873.
7. S. Mitra and B. U. Shankar, "Medical image analysis for cancer management in natural computing framework," *Information Sciences*, vol. 306, pp. 111–131, 2015.
8. P. Filipczuk, T. Fevens, A. Krzyżak, and R. Monczak, "Computer-aided breast cancer diagnosis based on the analysis of cytological images of fine needle biopsies," *IEEE transactions on medical imaging*, vol. 32, no. 12, pp. 2169–2178, 2013.
9. F. A. Spanhol, L. S. Oliveira, C. Petitjean, and L. Heutte, "A dataset for breast cancer histopathological image classification," *Ieee transactions on biomedical engineering*, vol. 63, no. 7, pp. 1455–1462, 2015.
10. E. D. Carvalho, O. Antonio Filho, R. R. Silva, F. H. Araujo, J. O. Diniz, A. C. Silva, A. C. Paiva, and M. Gattass, "Breast cancer diagnosis from histopathological images using textural features and cbir," *Artificial intelligence in medicine*, vol. 105, p. 101845, 2020.
11. S. Boumaraf, X. Liu, Y. Wan, Z. Zheng, C. Ferkous, X. Ma, Z. Li, and D. Bardou, "Conventional machine learning versus deep learning for magnification dependent histopathological breast cancer image classification: A comparative study with visual explanation," *Diagnostics*, vol. 11, no. 3, p. 528, 2021.
12. S. Saxena, S. Shukla, and M. Gyanchandani, "Pre-trained convolutional neural networks as feature extractors for diagnosis of breast cancer using histopathology," *International Journal of Imaging Systems and Technology*, vol. 30, no. 3, pp. 577–591, 2020.
13. Y. Jiang, L. Chen, H. Zhang, and X. Xiao, "Breast cancer histopathological image classification using convolutional neural networks with small se-resnet module," *PloS one*, vol. 14, no. 3, p. e0214587, 2019.
14. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

15. T. Kausar, M. Wang, M. Idrees, and Y. Lu, "Hwdcnn: Multi-class recognition in breast histopathology with haar wavelet decomposed image based convolution neural network," *Biocybernetics and Biomedical Engineering*, vol. 39, no. 4, pp. 967–982, 2019.
16. S. Saxena, S. Shukla, and M. Gyanchandani, "Breast cancer histopathology image classification using kernelized weighted extreme learning machine," *International Journal of Imaging Systems and Technology*, vol. 31, no. 1, pp. 168–179, 2021.
17. G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.
18. H. Rezaeilouyeh, A. Mollahosseini, and M. H. Mahoor, "Microscopic medical image classification framework via deep learning and shearlet transform," *Journal of Medical Imaging*, vol. 3, no. 4, pp. 044501–044501, 2016.
19. M. Desai and M. Shah, "An anatomization on breast cancer detection and diagnosis employing multi-layer perceptron neural network (mlp) and convolutional neural network (cnn)," *Clinical eHealth*, vol. 4, pp. 1–11, 2021.
20. F. A. Spanhol, L. S. Oliveira, C. Petitjean, and L. Heutte, "A dataset for breast cancer histopathological image classification," *Ieee transactions on biomedical engineering*, vol. 63, no. 7, pp. 1455–1462, 2015.
21. L. Chang, G. Zhou, O. Soufan, and J. Xia, "mirnet 2.0: network-based visual analytics for mirna functional analysis and systems biology," *Nucleic acids research*, vol. 48, no. W1, pp. W244–W251, 2020.
22. S. Pratiher and S. Chatteraj, "Manifold learning & stacked sparse autoencoder for robust breast cancer classification from histopathological images," *arXiv preprint arXiv:1806.06876*, 2018.
23. A. Patil, D. Tamboli, S. Meena, D. Anand, and A. Sethi, "Breast cancer histopathology image classification and localization using multiple instance learning," in *2019 IEEE International WIE conference on electrical and computer engineering (WIECON-ECE)*. IEEE, 2019, pp. 1–4.
24. P. C. Chhipa, R. Upadhyay, G. G. Pihlgren, R. Saini, S. Uchida, and M. Liwicki, "Magnification prior: a self-supervised method for learning representations on breast cancer histopathological images," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2717–2727.
25. D. Bardou, K. Zhang, and S. M. Ahmad, "Classification of breast cancer based on histology images using convolutional neural networks," *Ieee Access*, vol. 6, pp. 24680–24693, 2018.

Rough Target Region Extraction with Background Learning

Ryo Nakamura^{1,2}, Yoshiaki Ueda^{1,3}, Masaru Tanaka⁵, and Jun Fujiki^{1,4}

¹ Fukuoka University, Fukuoka Nanakuma 8191, Japan

² sd210501@cis.fukuoka-u.ac.jp

³ uedayos@fukuoka-u.ac.jp

⁴ fujiki@fukuoka-u.ac.jp

⁵ Prof. Masaru Tanaka deceased in June 2021.

Abstract. Object localization is a fundamental and important task in computer vision, that is used as a pre-processing step for object detection and semantic segmentation. However, fully supervised object localization requires bounding boxes and pixel-level labels, and these annotations are expensive. For this reason, Weakly Supervised Object Localization (WSOL) with image-level (weak) supervision has been the focus of much research in recent years. However, WSOL requires a large dataset to detect the region of an object in images with high performance. When the large dataset is unavailable, it is difficult to localize the image with high performance. This paper proposes a method for extracting target regions using small amounts of target and background images with image-level labels. The proposed method enables the detection of object locations with high performance using relatively less training images by classifying multiple patches cut from the image. This object localization method differs from the typical WSOL method that takes a single image as input and detects the location of an object because it assumes a small patch of area as input. The label of the patch cropped from the image must be labeled with the ground truth. However, the proposed method uses labels attached to images because ground truth labeling is costly. Instead, in the proposed method, the network learns by learning many "background" labeled background patches, and learns to induce the network to classify the mislabeled background patches that resemble ground truth as background. We call this key idea Decision-Boundary Induction(DBI). Moreover, learning many background patches for such a DBI is what we call background learning. In our experiments, we verified that decision boundaries are induced, and accordingly, we could roughly extract the target region. Also, we showed that the Loc. Acc. is higher than that of WSOL.

Keywords: Weakly supervised object localization, Patch-based training, Background learning, Noisy label training

1 Introduction

Object localization is a fundamental and important task in computer vision and is used as preprocessing for object detection [18, 17, 4, 11, 10, 5], semantic segmentation [14, 12, 9, 8], etc. For this, methods of deep learning, such as convolutional

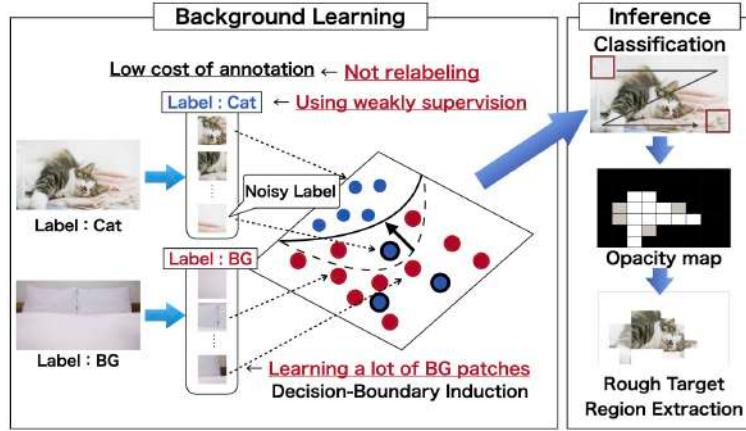


Fig. 1. Overview of proposal method: We consider the task of classifying randomly cropped multiple small patches from an image as whether they are in the target or background (BG). In order to classify patches with high accuracy, it is necessary to attach ground truth to the patches. However, in background learning, we do not provide the ground truth. Instead, the network induces the decision boundary to classify patches of label noise as background by learning many background patches cropped from the background image. Note that label noise is the background patch labeled with the target label. In inference, the background learning network roughly extracts the target region by outputting the backgroundness (sigmoid value) of patches in the target image as a sliding window formula.

neural networks (CNN), are widely used. To achieve high accuracy, these need labeled training datasets: Pixel-level labels and bounding boxes of target objects. When we carry out pixel-level labeling, it often becomes a burden compared to image-level labeling (i.e., attaching labels to images in one-by-one ways).

As a methodology to overcome such problems, Weakly Supervised Object Localization (WSOL) [22, 3, 19, 1, 21, 20, 16, 13] has received recent attention from researchers in the field. This aims for high accuracy object-localization, and we have only to prepare training datasets consisting of image-level labels, and thus we can save time and human resources despite the high accuracy. Typically, WSOL estimates regions in the images, which are recognized to be important for the classification of images. The regions are then used in object localization. For this procedure, we need huge datasets, and thus a large part of the total cost is passed on to the large size of the datasets. This implies that we have to label many images to use WSOL, which would also be a heavy task.

To avoid such difficulty, we propose a novel method to identify the location of the (target) object to be recognized (see Fig. 1). The method consists of two parts, explained below. One is to randomly cut out small parts of images, each of which will be called a patch, and then classify them into two categories with attached labels: ‘foreground’ (target) and ‘background.’ More specifically, all patches from an image will be labeled as the common name of an object (e.g., ‘cat,’ ‘dog,’ ‘horse,’ ‘owl,’ and so on) if the object in the consideration is in the image, and they will be labeled as ‘background’ otherwise. An image showing a

cat might contain a background area. One should notice that all patches from the image are labeled as ‘cat’ even if the patch does not contain any part of the cat. In such a case, the false labels of the patches will be called ‘label noises.’ On the other hand, all patches from an image showing no objects in consideration are assigned with ‘background,’ in coincidence with the true label. Therefore one can easily collect a large dataset consisting of truly-labeled patches, namely the ones with ‘background’ labels, since it is easy to prepare images showing no objects.

The other is the method we call ‘background learning,’ which enables our model to discriminate labels to be attached to new patches by training with the large dataset of patches truly labeled ‘background’ combined with a relatively small dataset including label-noises. Now, assigning new patches to either true target or true background is what we call object localization and will be realized with the network trained using our background learning. It would be natural to design all patches with label noises, and the ones with background labels should be similar to each other. The true labels of such patches are both ‘background.’ The main ingredients in the dataset are cut out from background images, which is why we call the method ‘background learning’.

All patches in the labeled training dataset for background learning are classified into three classes (see Fig. 2.):

- One consists of patches showing (a part of) objects (thus, these are cut out from images showing objects).
- Another is label-noises, the class consisting of patches cut out from images showing objects (and thus labeled as the target), but the patches themselves show no parts of the object.
- The other consists of patches cut out from background images (images showing no objects), which are truly labeled as ‘background.’

In this paper, each state of the classes is said to be positive (i.e., showing objects), false positive (i.e., showing no object but cut out from an image showing objects), and negative (i.e., cut out from background image), respectively. With these terminologies, background learning is a method to determine whether the states of patches showing objects are positive or false positive. Equivalently, it is a method to detect false positives (or extract positives) among states of patches showing objects.

For our experiments, we prepared 240 images showing one of cats, dogs, horses, or owls and 240 background images. We trained networks by WSOL and our background learning separately. As a result, we showed that our background learning successfully extracted target regions (regions lying in positive patches). For the object localization task, we showed that the performance of background learning is superior to WSOL in the sense of Localization Accuracy (Loc. Acc.).

2 Related work

Weakly Supervised Object Localization(WSOL) aims to learn to localize the object using only image-level labels. A popular method for WSOL is Class Acti-

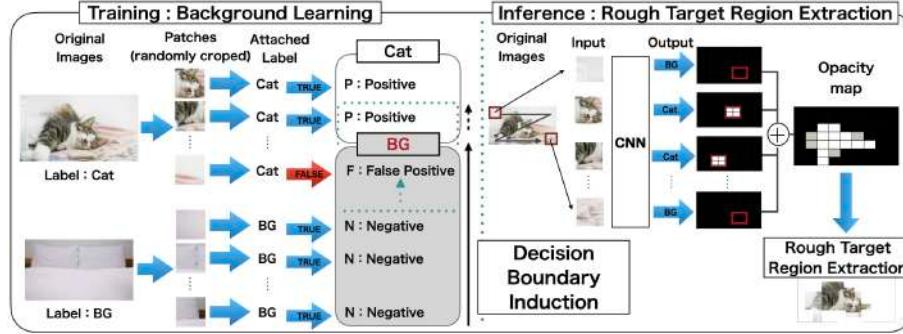


Fig. 2. Explanation of rough target area extraction with background learning. In background learning, a large amount of training data is prepared by randomly cropping multiple small-area patches from the image. Then, annotations to patches are not annotated with the correct labels, instead using image labels, which does not increase the annotation cost. In fully supervised learning, a patch of F is classified as a "cat," but learning a large number of patches of N that resemble F induces it to be classified as a "background" (Decision-Boundary Induction). For rough target extraction, CNN with background training is used to output target patches using the sliding window method to generate a targetness map of the corresponding patch region.

vation Map (CAM) [22], which generates localization maps by aggregating deep feature maps using fully connected layers by class. Hwang and Kim [7] simplify the network structure of CAM by removing the last unnecessary fully connected layer. CAM methods are simple and effective, but they can only classify the small classification region of the object. In order to improve the activation map of CAM, HaS [16] and CutMix [20] adopted the region dropout-based strategy from the input image so that the network focuses on the more related region of the object. ADL [3] focuses on the problem that learning while deleting classifiable regions with high performance requires high computational resources and eliminates feature maps corresponding to discriminable regions to localize objects with a lightweight model that is efficient and has low network parameters.

3 Rough Target Region Extraction with Background Learning

3.1 Background Learning

Overview. Fig. 2 shows the proposed method. In background learning, we consider the problem of classifying whether the small patches of the image are target or background using a background image. Note that background images are relatively easy to obtain because the images are not required to include the object. When creating the training dataset, multiple patches are randomly cropped from the target and background images, and the patches are annotated with image labels. Then, the patch cropped from the target image has two patches: the target and the background. But we do not relabel the background patches with the target label. (i.e. the incorrect label is annotated). Instead, we use the data

imbalance induced by learning many background patches with the F. We then induce the network to classify the mislabeled background patches as background (We call this induction DBI: Decision-Boundary Induction).

Define target/background. In this paper, "target" and "background" mean the target object to be extracted and the non-target object to be extracted. In a specific example, if the target is the cat, the cat's area in the image represents the target, and the other image areas represent the background. Therefore, if an image contains a cat and a dog, the cat area would be the target, while the other dog and background areas would be the background.

About the type of patches to be cropped. The cropped patch from the image is a small area image of the target or background image. The size of the small area is an optional parameter. Also, the number of patches to be cropped is an optional parameter. We use image labels to label the patches. Therefore, cropped patches have three types of patterns, as shown in Fig. 2. The first is a positive (P) patch that includes the target region cropped from the target image. (P) patches are labeled with the target label. The second is a false positive (F) patch that does not include the target region cropped from the target image. (F) patches are labeled with the target label. Not labeling (F) patches with ground truth are to avoid labeling costs. The third is a negative (N) patch cropped from the background image. (F) patches are labeled with the background label.

Background learning purpose (Decision-Boundary Induction). The purpose of background learning is to induce the "background" classification result of the network when inputting the F patch. Therefore, we induce the classification of the network by training a large amount of N patch on it, causing bias in the learning. In fully supervised learning, the network learns that background F patches labeled as targets are to be classified as backgrounds based on their labels. However, to classify it as "background," we need to relabel F with a background label. But relabeling increases the annotation cost. Therefore we want to train the network to correctly classify P, F, and N patches into target and background patches without relabeling. In this paper, the problem of classifying such P, F, and N patches is called the PFN classification problem. Then, we use the imbalance of the number of training data, i.e., the property of [2] (Decision-Boundary Induction), which induces the class with the largest number of training data to be classified.

Training. For learning patches with label noise such as PFN, we use fully supervised learning, which is widely used in deep learning. In this learning, the error function is optimized to classify the patches so that they correctly answer the label of the image to be cropped, without relabeling. In the paper, binary cross-entropy is used for the error function and Stochastic Gradient Descent (SGD) for optimization.

3.2 Rough Target Region Extraction

The CNN with background training roughly calculates the target region by the following procedure. First, the target image is slided window by a specified step width (in this paper, 8 pixels in height and width), and the backgroundness of



Fig. 3. Sample image of the dataset used in the experiments

each patch is output. Next, for each pixel in the target image, add the final output c , which is the output of the sigmoid function. When this score is calculated for all pixels, the maximum value is M , and the minimum value is m . The “Targetness” is defined as follows for each pixel.

$$\text{Targetness} = \frac{c - m}{M - m} \in [0, 1] \quad (1)$$

In this paper, we consider the targetness to be the probability that each pixel is in the region to be classified. The target region can be visualized as an opacity map by using the targetness as the value of α channel, representing the opacity (see Fig. 2).

4 Experiments

4.1 Experimental setting

Data collection. In order to verify the effectiveness of decision-boundary induction, we prepare the dataset, which includes target (cat, dog, owl, horse) images with forest and playground background regions and background images with a different scene from the target backgrounds. Since there are no publicly available open datasets with such limited backgrounds, we collected 240 targets and 480 backgrounds (240 treated as background areas of targets and 240 treated as background images) from Fricker, manually generated masks of targets and background areas, and used the masks to composite them with targets and background images to construct a data set (see Fig. 3).

Create train/test dataset. The training dataset is a two-category dataset of target and background images, each with 240 images (a total of 480 images). The test dataset uses target images from the training dataset and Ground truth masks labeled with targets at the pixel level to measure the performance of extracting target regions from the target images in the training dataset. This mask can compare target region extraction performance and extract patches related to PFN from the mask information. Note that the masks containing

Table 1. Performance evaluation of rough target region extraction with background learning with Loc. Acc.: The result that performs well in comparing background patches is assumed to be bold.

Model	Target	BG : Forest		BG : Playground	
		TG:BG=1:1	TG:BG=1:2	TG:BG=1:1	TG:BG=1:2
VGG16	Cat	0.81	0.98	0.90	0.91
	Dog	0.61	0.93	0.83	0.79
	Owl	0.85	0.81	0.76	0.75
	Horse	0.78	0.53	0.75	0.55
	Avg.	0.76	0.81	0.81	0.75
ResNet50	Cat	0.88	0.90	0.88	0.90
	Dog	0.96	0.90	0.86	0.76
	Owl	0.83	0.76	0.83	0.80
	Horse	0.53	0.81	0.78	0.57
	Avg.	0.80	0.84	0.84	0.76

the target and background merge with the background and evaluate target area extraction performance, but not for training.

Experiment details. We use the VGG16 [15] and ResNet50 [6] models in this experiment. We train the models with the binary cross-entropy loss for 150 epochs using SGD with a learning rate of 0.01. The mini-batch size is 64 for the WSOL methods and 512 for our method. The reason for this is to align the updates of the network parameters. HaS [16] and Cutmix [20] are localized with CAM [22]. In the proposed method, 8 patches are randomly cropped with size $48 \times 48 \times 3\text{ch}$ from a single image ($256 \times 256 \times 3\text{ch}$).

Evaluation. In the evaluation of target area extraction, we evaluate the performance of target area extraction on trained data rather than on the performance of target area extraction on unlearned target images. The number of target images used is 240, the same as the number of training data. We use the localization accuracy metric (Loc. Acc.) to evaluate the roughness of the target region extraction. Loc. Acc. is a metric that calculates the proportion of images with an Intersection over Union (IoU) of 40% or higher. The threshold value of IoU when calculating Loc. Acc. is $[0.05, 0.15, \dots, 0.95]$, and the best value is used as the experimental result.

4.2 Evaluation of the effectiveness of Decision-Boundary Induction

We show through experiments that Decision-Boudary Induction can be used by adjusting the amount of background patches in the dataset, and that the use of DBI can lead to improved performance in target region extraction. For this experiment, we conducted the following on datasets with one and two times the ratio of background patches to target patches.

- Quantitative evaluation of the extraction performance of the target region by Loc. Acc.
- Qualitative evaluation visualizing relative frequencies of backgroundness of patches of PFN.

- Qualitative evaluation to compare target area extraction results

Evaluation of target area extraction performance with background learning. Table 1 shows the results of Loc. Acc. for each condition when the background patches are trained with the ratio of background patches 1x and 2x compared to the target patches and the target regions are extracted. Cat results showed that the Loc. Acc. was higher in all cases when the background was learned 2x. Dog’s results show that for VGG16, the Loc Acc is higher when the background is trained 2x only when the background region is Forest, and 1x is higher for all other cases. In the Owl results, the Loc. Acc. was higher when the background was trained 1x for VGG16, ResNet50, and the background was Forest and Playground. In the Horse results, the Loc. Acc. was higher when the background region was Forest, the Loc. Acc. was higher when the background region was Playground and the Loc. Acc. was higher when the background region was Playground. As shown above, it can be confirmed that adjusting the number of background patches according to the target and the model used contributes to the performance improvement of Loc. Acc.

We consider the bias of the background pattern of the image to be related to the reason that Forest and Playground did better with 2x background patches and 1x background patches, respectively. In the case of Forest, since F contains many similar patterns, such as diverse leaves and trees, it is necessary to learn many N to induce identification. On the other hand, Playground contains many instances of playground equipment, trees, sand, etc., and the patterns are distributed, so the discrimination induction works effectively with a relatively small number of N images.

Comparison by backgroundness relative frequency graph for each patch. Fig.4 is the result of visualizing the sigmoid (backgroundness) of P, F, and N patches as relative frequencies (see Fig.4.) when background learning is performed using Forest as the background on ResNet50, which had high Loc. Acc. in Table 1. In this experiment, for patches, P is defined as if the target is included in 10% or more of the patches cropped from the target image, and F is defined as otherwise. The data for each PFN is 200, for a total of 600. We denote patches extracted from the image containing the region to be identified as X and patches extracted from the background image as Y to clarify the training data structure used for background training in the graph. The CNN trained with X and Y is denoted as $\text{Model}(X, Y)$.

The Fig.4 shows that the relative frequencies of F and N above 0.8 of the sigmoid are higher when the targets are Cat and Dog by learning the background patches twice. In Table 1, we can verify that Loc. Acc. is also high following the results. On the other hand, when the target is Owl, learning as 2x background patches did not increase the relative frequencies of backgroundness of F and N above 0.8. We consider that one of the reasons why the owl case has not worked is that the features of the forest and the Owl are similar. Owls are mimic animals as they hide and hunt in the forest. Therefore, as a result, we consider that learning to classify the P, F, and N patches became difficult, the learning was unstable, and thus the hypothetical results were not observed.

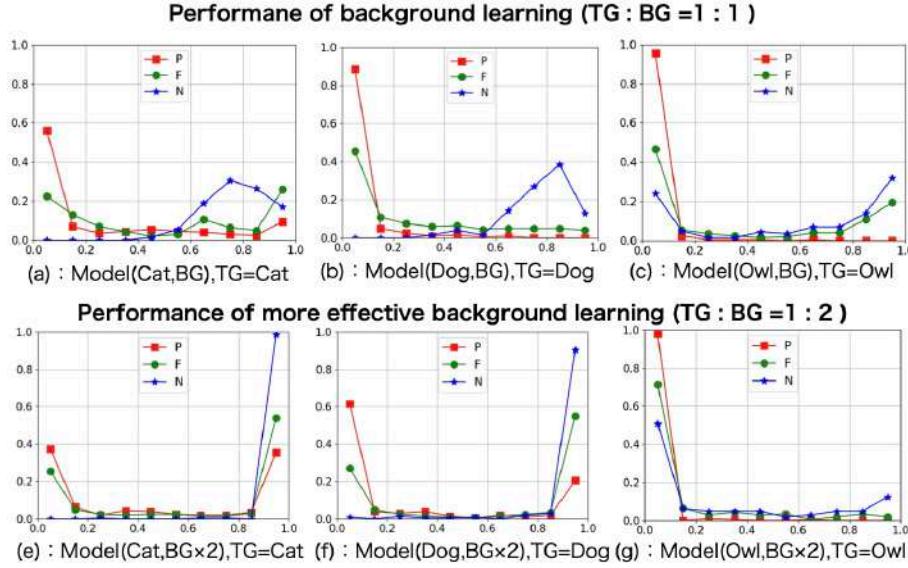


Fig. 4. Relative frequency graph of backgroundness of P, F, and N patches (This experiment uses ResNet50 and Forest background region). The horizontal axis represents the backgroundness (sigmoid value of the background class), and the vertical axis represents the relative frequency. We used forest for the background and ResNet50 for the network.

Comparison by target area extraction map. Fig.5 shows the target region extracted image using the trained network of (a)-(h) in Fig.4. (a)-(d) are the target region extraction images when the number of target patches and background patches are the same. (e)-(h) are the target region extraction images when more background patches are trained. The Cat, Dog, and Horse results show that when the background is trained strongly, the regions are extracted to remove the background region. On the other hand, the Owl results show that when the background is learned too much, the target region is extracted in such a way that the backgroundness body region of the Owl is removed. The effectiveness of can also be verified through qualitative results on the amount of background patches (Fig.4).

4.3 Comparison of target region extraction performance

We compare the proposed method with the WSOL method by Loc. Acc. to show that the proposed method can extract the target region with a smaller amount of images than the WSOL method (The results are shown in Table 2). Avg. in the table is the average of Loc. Acc. for the four targets (Cat, Dog, Owl, and Horse). For each model and target, the highest value of Loc. Acc. is indicated by bold and the second highest by underline.

In the results of Forest in Table 2, our method has the highest Loc. Acc. for all targets in VGG16 and ResNet50, where we adjusted the number of background patches and used DBI effectively. Avg. results show that our method occupies at least the top two positions. It is interesting to note that the results show a higher

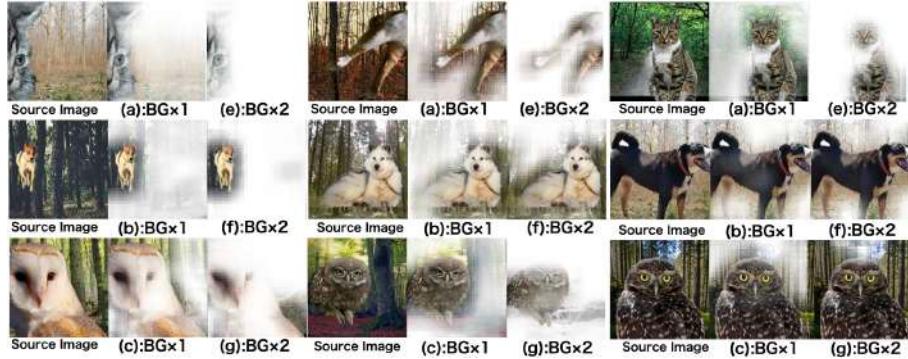


Fig. 5. The results of extracting the target region when background learning: (a)-(h) are the results of the model learned in Fig. 4. It can be shown that background learning induces the classification of the background in the target image. However, if the background learning is too effective, it also induces the classification of target regions that look like the background.

Table 2. Loc. Acc. comparison on a dataset where the background region.

Model	Method	Forest				Playground				Avg.	
		Cat	Dog	Owl	Horse	Avg.	Cat	Dog	Owl		
VGG16	CAM [22]	0.83	0.58	0.67	0.42	0.62	0.72	0.57	0.68	0.51	0.62
	ADL [3]	0.73	0.75	0.67	0.38	0.63	0.74	0.60	0.69	0.46	0.62
	HaS [16]	0.70	0.65	0.78	0.55	0.67	0.86	0.64	0.68	0.46	0.66
	Cutmix [20]	0.89	0.65	0.73	0.44	0.68	0.88	0.76	0.69	0.46	0.70
	Ours(BGx1)	0.81	0.61	0.85	0.78	0.76	0.90	0.83	0.76	0.75	0.81
	Ours(BGx2)	0.98	0.93	0.81	0.53	0.81	0.91	0.79	0.75	0.55	0.75
ResNet50	CAM [22]	0.66	0.66	0.75	0.44	0.63	0.85	0.72	0.74	0.50	0.70
	ADL [3]	0.85	0.73	0.71	0.55	0.71	0.89	0.81	0.78	0.62	0.78
	HaS [16]	0.70	0.59	0.68	0.52	0.62	0.72	0.66	0.69	0.54	0.65
	Cutmix [20]	0.70	0.50	0.66	0.47	0.58	0.86	0.66	0.74	0.53	0.70
	Ours(BGx1)	0.88	0.96	0.83	0.53	0.80	0.88	0.86	0.83	0.78	0.84
	Ours(BGx2)	0.90	0.90	0.76	0.81	0.84	0.90	0.76	0.80	0.57	0.76

Avg. than WSOL, even though the parameters for the amount of background patch are roughly chosen.

The Playground results in Table 2 also show that when DBI is used effectively, our method has the highest Loc. Acc. for all targets in VGG16 and ResNet50. The difference from Forest results was that ResNet50 had the second-best performance with ADL of Avg. The reason for this is the difference in the features of the background since there are more variations of objects, such as playground equipment, the ground, and trees, in the Playground than in the Forest. This can be considered to increase the learning of the target object.

Fig. 6 shows the results of the visualization images where each target is extracted using WSOL and our method. The images extracted by CAM and ADL, among the conventional WSOL methods compared, tended to be extracted for frequently appearing parts, such as cat's whiskers, dog's face, owl's face, and horse's feet. Region dropout-based methods that mask part of the image, such

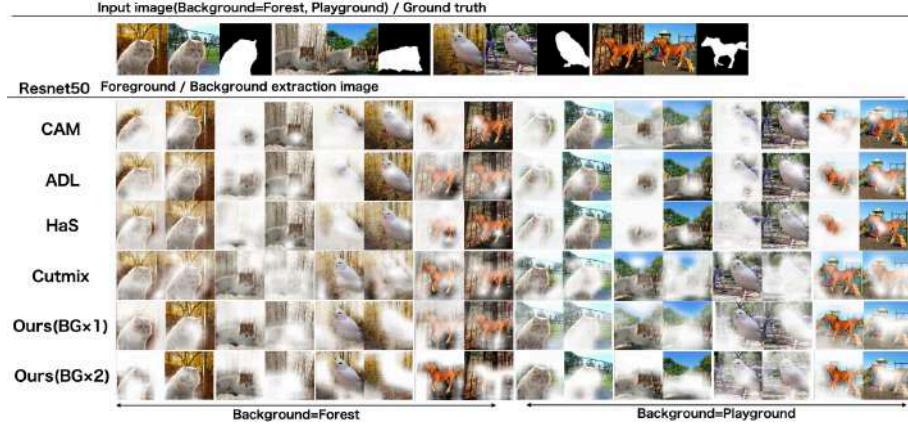


Fig. 6. Visualization results of target region extraction for existing WSOL methods and our method. We use ResNet50 for the network.

Table 3. The relationship between patch size and the performance of target area extraction with background learning. We study the relationship between the patch size {32, 48, 64, 96, 128}pixels.

Patch size	Forest				Playground			
	Cat		Dog		Cat		Dog	
	VGG16	ResNet50	VGG16	ResNet50	VGG16	ResNet50	VGG16	ResNet50
32	0.954	0.979	0.914	0.902	0.786	0.942	0.967	0.7
48	0.958	0.975	0.858	0.793	0.962	0.979	0.931	0.943
64	0.967	0.975	0.846	0.846	0.958	0.975	0.894	0.923
96	0.913	0.86	0.7	0.663	0.942	0.876	0.801	0.7
128	0.786	0.777	0.627	0.570	0.847	0.802	0.570	0.542

as HaS and Cutmix, increased the target area to be extracted, but Cutmix also tends to extract more background areas as foreground, and HaS tended to extract some parts of the body as targets, but still tended to extract parts that appeared frequently.

4.4 Ablation Study

Relationship Study with Patch Size (Table 3). We studied the effect of patch size on the extraction of the target area. Specifically, we changed the size of patches extracted from images in {32, 64, 96, 128} and studied the relationship between Loc. Acc. We used VGG16 and ResNet50 as the networks. We used VGG16 and ResNet50 as the network and cat and dog as the target images. Then, we used the forest and playground as the background. The background patch is cropped twice as many times as the target patch to create the dataset. The results are shown in Table 3. The experimental results show that the patch sizes with the highest Loc. Acc. were 32 for the forest and 48 for the playground.

We consider the reason that the accuracy increases as the patch size decreases is because the proportion of patches that contain the target decreases.

Table 4. Study of Background Patch Ratio and Performance of Target Area Extraction with Background Learning.

	Forest		Playground	
	Cat	Dog	Cat	Dog
Patch size	VGG16 ResNet50	VGG16 ResNet50	VGG16 ResNet50	VGG16 ResNet50
BG×1.0	0.93	0.9	0.923	0.846
BG×2.0	0.95	0.954	0.914	0.902
BG×4.0	0.983	0.975	0.971	0.951
			0.925	0.917
			0.939	0.906

Background learning aims to DBI a background patch F with a target label by learning many N true background patches. Therefore, the smaller the F patches are, the more likely they produce DBI. Therefore, the smaller the patch size, the higher Loc. Acc. However, if the patch size is too small, DBI will also occur in the body of the target image, as shown by the owl in Fig. 5, so a moderately small patch is important for target area extraction.

Also, patch size differs depending on the background because of the background's complexity. The forest has a bias toward trees, leaves, and other patterns that appear in the background. Small patches will cause pattern bias in the clipped patches when there are many background patterns. Since it is difficult to perform DBI on a background with few patterns, we believe that increasing the size of the patch increases the number of patterns in the image and reduces pattern bias, which is effective in improving accuracy.

Study of Relationships with Background Patches Ratio (Table 4). We studied the relationship between the ratio of background patches to target patches and the performance of target region extraction. Changing the ratio of each patch is equivalent to adjusting the strength of the DBI effect. Specifically, we changed the proportion of background patches by {1.0, 2.0, 4.0} and studied the relationship between Loc. Acc. We used VGG16 and ResNet50 as the network and cat and dog as the target images are cat and dog. Then, we used the forest and playground as the background. The patch size is 32×32. The results are shown in Table 4. The results for forest show that increasing the ratio of background patches results in a better Loc. Acc. In the playground, no consistent trend was found comparing the models.

In the forest results, the DBI works correctly and improves accuracy. On the other hand, the playground results showed some variation depending on the model and target. We consider this because the patch size of 32×32 is relatively small, and the image does not contain a variety of background patterns. If there is a large piece of playground equipment in the background image, a large percentage of the image will contain the equipment if the patches are cut out randomly. If the patch size is large, a sandbox and playground equipment appear as patterns in the patch, which can be classified as background because it contains equipment frequently appearing as background. However, if the patch size is small, the number of patches containing only playground equipment patterns increases, making it difficult to distinguish other patterns from the background, in our opinion.

4.5 Limitation

Our proposed method can extract the target region with higher performance than the conventional method in a situation where only a small number of images are available, but it has some challenges.

First, it is not easy to classify the target body and background in the patch input. Fig. 4 shows the relative frequency of the PFN backgroundness when the synthetic data is background trained. Comparing background learning and the effect of background learning enhancement, the backgroundness of most F improved, but the backgroundness of some P also increased. A possible reason for the increased backgroundness of P is that P has patches of the target body, which are not discernable from the background patches, so the DBI is also working on the patches of the body. This problem represents the limitation of learning only patches. To deal with this problem, it is necessary to incorporate a mechanism that can determine the target's body from the target's structural information based on the relationship of the positions of the cropped patches.

Second, there is a need to establish clear indicators for use in selecting background images for background studies. In the background learning, F and P classification results are decided by the background image prepared as the dataset. One approach to effectively DBI F is to use a large background image, but this approach is likely to cause an unbalance in the number of images in the target and background images, and thus DBI for P as well. Currently, we perform background learning by preparing random background images, but to perform DBI effectively, it is important to incorporate a mechanism to select an effective N for DBI. Due to the lack of a clear metric for background images, we consider it difficult to apply the method to large datasets with target images from diverse backgrounds in the current situation.

5 Conclusions

In this paper, we proposed the patch-based rough target region extraction method to extract target object regions with a relatively small number of images and a small annotation cost. The proposed method learns the network to robustly classify whether a small patch in the image is the target or background by using the method we call background learning. Also, the trained network is used to localize objects by determining whether the small areas of the image are targets or backgrounds. There are two important aspects to classifying target and background patches with fewer training images and less annotation cost. First, we do not use a single image to train the network but rather assume small patches randomly cropped from multiple images as input. This assumption is data efficient and enables the neural network to be trained by cropping many patches from several training images. Second, even if background images have noisy labels, background learning can improve classification robustness. Background learning is a method of classifying target and background patches by learning many background patches of ground truth cropped from background images, even when the labeling of patches cropped from the target image is done roughly. Using this learning, we provide a more robust classification of patches without using the

cost of relabeling. For patch-based rough target area extraction, we calculate the backgroundness of the patch by inputting a small region of the image in a sliding window to the network. The result is calculated as the map of backgroundness, and the maps are merged by averaging the maps of backgroundness calculated for all regions. In our experiments, we verified that DBI works and improves the performance of target region extraction on an ideal dataset with similar background images and target object backgrounds. In addition, we verified that our method could extract target regions with higher Loc Acc than the existing WSOL method, although limited.

Acknowledgements I am grateful to Associate Professor Takafumi Amaha of Fukuoka University for advice on writing the paper. I would like to take this opportunity to thank him.

References

1. Bae, W., Noh, J., Kim, G.: Rethinking class activation mapping for weakly supervised object localization. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020. pp. 618–634. Springer International Publishing, Cham (2020)
2. Chawla, N., Japkowicz, N., Kołcz, A.: Editorial: Special issue on learning from imbalanced data sets. SIGKDD Explorations **6**, 1–6 (06 2004). <https://doi.org/10.1145/1007730.1007733>
3. Choe, J., Lee, S., Shim, H.: Attention-based dropout layer for weakly supervised single object localization and semantic segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **43**(12), 4256–4271 (2021). <https://doi.org/10.1109/TPAMI.2020.2999099>
4. Cinbis, R., Verbeek, J., Schmid, C.: Multi-fold mil training for weakly supervised object localization. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2409–2416. IEEE Computer Society, Los Alamitos, CA, USA (jun 2014). <https://doi.org/10.1109/CVPR.2014.309>, <https://doi.ieeecomputersociety.org/10.1109/CVPR.2014.309>
5. Eu Wern Teh, M.R., Wang, Y.: Attention networks for weakly supervised object localization. In: Richard C. Wilson, E.R.H., Smith, W.A.P. (eds.) Proceedings of the British Machine Vision Conference (BMVC). pp. 52.1–52.11. BMVA Press (September 2016). <https://doi.org/10.5244/C.30.52>, <https://dx.doi.org/10.5244/C.30.52>
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition (2016), in CVPR
7. Hwang, S., Kim, H.E.: Self-transfer learning for weakly supervised lesion localization. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016. pp. 239–246. Springer International Publishing, Cham (2016)
8. Khoreva, A., Benenson, R., Omran, M., Hein, M., Schiele, B.: Weakly supervised object boundaries (2016), in CVPR, pages 183–192
9. Kolesnikov, A., Lampert, C.H.: Seed, expand and constrain: Three principles for weakly-supervised image segmentation (2016), in ECCV, pages 695–711
10. Liang, X., Liu, S., Wei, Y., Liu, L., Lin, L., Yan, S.: Towards computational baby learning: A weakly-supervised approach for object detection. In: 2015 IEEE

- International Conference on Computer Vision (ICCV). pp. 999–1007 (2015). <https://doi.org/10.1109/ICCV.2015.120>
11. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Is object localization for free? - weakly-supervised learning with convolutional neural networks. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 685–694 (2015). <https://doi.org/10.1109/CVPR.2015.7298668>
 12. Pathak, D., Krahenbuhl, P., Darrell, T.: Constrained convolutional neural networks for weakly supervised segmentation (2015), in ICCV, pages 1796–1804
 13. Rahimi, A., Shaban, A., Ajanthan, T., Hartley, R., Boots, B.: Pairwise similarity knowledge transfer for weakly supervised object localization. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020. pp. 395–412. Springer International Publishing, Cham (2020)
 14. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. CoRR **abs/1312.6034** (2013)
 15. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2015), iCLR
 16. Singh, K.K., Lee, Y.J.: Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 3544–3553 (2017). <https://doi.org/10.1109/ICCV.2017.381>
 17. Song, H.O., Girshick, R., Jegelka, S., Mairal, J., Harchaoui, Z., Darrell, T.: On learning to localize objects with minimal supervision. In: Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32. p. II–1611–II–1619. ICML’14, JMLR.org (2014)
 18. Wang, C., Ren, W., Huang, K., Tan, T.: Weakly supervised object localization with latent category learning. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014. pp. 431–445. Springer International Publishing, Cham (2014)
 19. Wei, Y., Feng, J., Liang, X., Cheng, M.M., Zhao, Y., Yan, S.: Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6488–6496 (2017). <https://doi.org/10.1109/CVPR.2017.687>
 20. Yun, S., Han, D., Chun, S., Oh, S.J., Yoo, Y., Choe, J.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 6022–6031 (2019). <https://doi.org/10.1109/ICCV.2019.00612>
 21. Zhang, C.L., Cao, Y.H., Wu, J.: Rethinking the route towards weakly supervised object localization. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13457–13466 (2020). <https://doi.org/10.1109/CVPR42600.2020.01347>
 22. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2921–2929 (2016). <https://doi.org/10.1109/CVPR.2016.319>