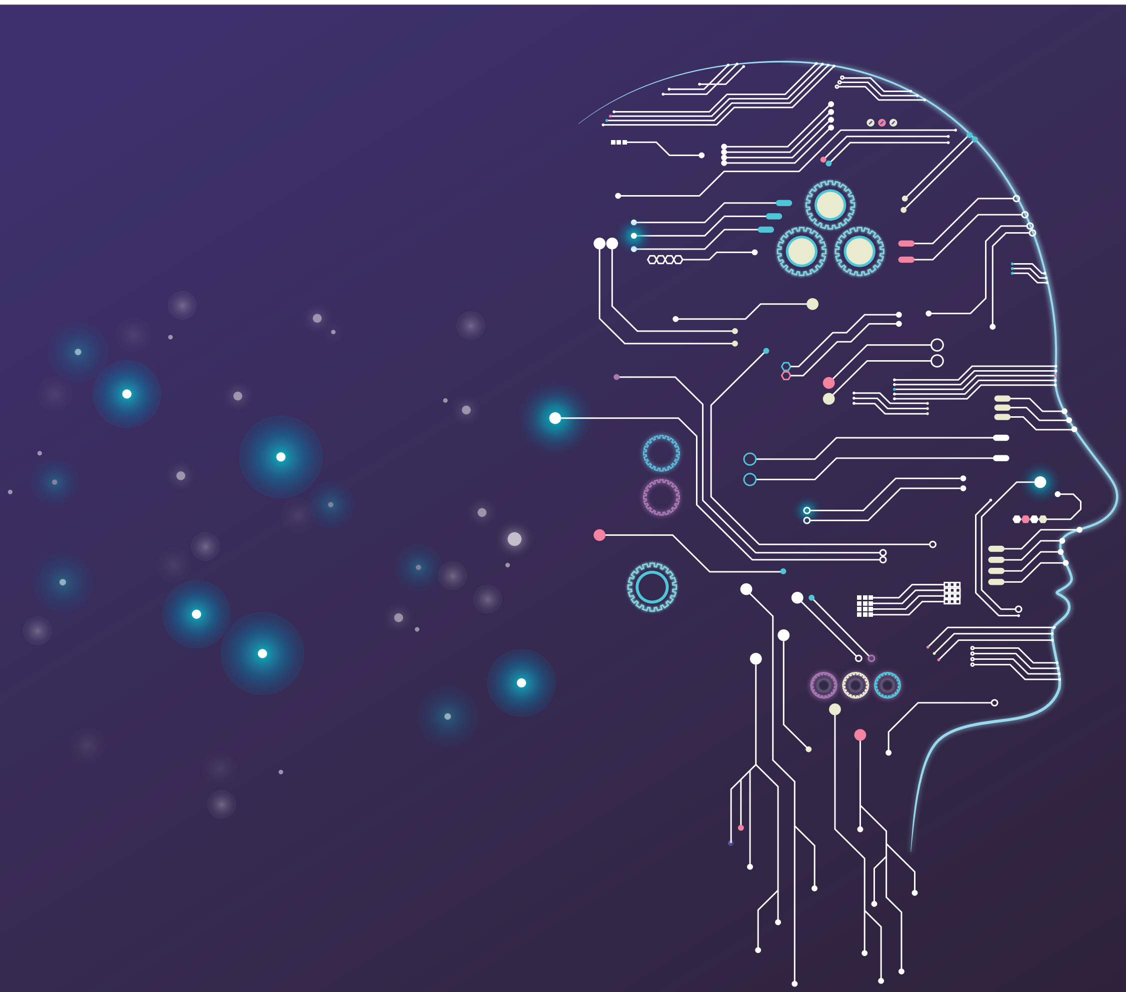


한국인공지능학회 2020 하계 및 추계 학술대회 논문집

하계학술대회: 2020년 8월 27일~29일 (온라인)

추계학술대회: 2020년 11월 19일~21일 (온라인)

한국인공지능학회
Korean Artificial Intelligence Association

한국인공지능학회
2020 하계 및 추계 학술대회 논문집

하계학술대회: 2020년 8월 27일~29일 (온라인)

추계학술대회: 2020년 11월 19일~21일 (온라인)

한국인공지능학회
Korean Artificial Intelligence Association

2020 하계 및 추계학술대회 논문집을 발간하며

2020년은 (사)한국인공지능학회가 창립되고 나서 처음으로 두 차례의 학회를 개최하여 한국인공지능발전에 크게 기여한 의미 있는 한 해였습니다. 인공지능이 4차 산업혁명의 전환을 이룰 핵심기술로 대두되고 있으며 그 기술 발전이 어느 때보다 급속히 이루어지고 있는 이 때에, 한국에서 인공지능 발전의 세계적 추세와 그 발전 성과들을 교류하며 한국인공지능의 발전을 모색한 이번 학술대회들은 국내 인공지능 연구자들에게 크나큰 호평을 받았습니다.

또한 국내 인공지능 연구의 주요 성과들을 교류하고 검토할 수 있는 논문투고 및 승인 과정이 있었습니다. 여기에 참여해서 하계 및 추계학술대회 프로그램 위원으로 활동해 주신 모든 교수님들 그리고 국내 최고의 인공지능전문가들에게 다시금 감사드립니다. 구체적으로 말씀드리면 하계학술대회에 총 72편의 논문이 접수되고 이 중 64편의 논문이 심사를 통과하였습니다. 그리고 64편의 승인논문 목록을 공표하기로 하였습니다. 또한 추계학술대회에 총 31편의 논문이 접수 되고 이 중 28편의 논문이 심사를 통과하였습니다. 그리고 저자들의 의견을 반영하여 24편의 승인논문 목록을 공표하기로 하였습니다. 최종적으로 본 논문집은 최종 승인된 논문들 중 다시 20여개의 논문들을 선별하여 실었습니다.

2020년 학술대회에 투고된 논문들을 보면서 국내 인공지능 연구자들의 연구가 확률모형 및 변분법, 능동학습과 적응 및 제로 샷 학습, 공정한 분류 문제, 강화학습, 연속학습과 메타 학습, 그래프 분석 및 학습, 딥 네트워크를 이용한 의료데이터 분석, 비디오 문답을 비롯한 컴퓨터 비전을 비롯하여 인공지능에서 주요하게 대두되는 수많은 분야들에서 깊이 있게 진행되고 있다는 것을 알 수 있었습니다. 아무쪼록 본 논문집이 나날이 발전하는 한국인공지능발전의 성과를 보여주는 자료가 될 수 있기를 기대합니다.

2021년 2월

(사)한국인공지능학회

회장 유창동

한국 인공지능 연구자들께,

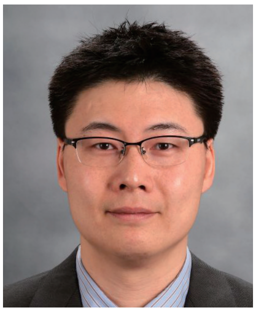
2020년은 모두에게 새로운 시련과 동시에 기회를 준 한 해였습니다. 그 한 가운데에서 열렸던 한국인공지능학회의 하계학술대회는 그 만큼 의미가 깊었습니다. 학회는 8월 27일부터 29일까지 3일간 온라인으로 개최가 되었습니다. 모두가 익숙하지 않은 채로 한국인공지능학회에서는 처음으로 온라인에서 열린 하계학술대회는 여러모로 모두에게 기억될 만한 학술대회였습니다.

하계학술대회는 2건의 기조강연과 4건의 튜토리얼, 11개의 기획 세션으로 구성되었습니다. 11개의 세션은 바이오/의료, 음성/기계학습/신호, 컴퓨터 비젼, 공정한 인공지능, 산업인공지능 등 5개 분야별 세션과 Qualcomm, Neuroscience Inspired AI Lab, ETRI 등 3개의 소속별 세션, 3개의 우수논문 소개 세션으로 구성되었습니다. 어느 한쪽으로 치우침 없이 다양하게 연구해주시고, 본 하계학술대회에 투고해주신 한국을 대표하는 인공지능 연구자들의 결과들을 한자리에서 볼 수 있는 것만으로도 큰 기쁨이었습니다. 또, 이어진 한양대학교, UNIST, 연세대학교의 AI 대학원 소개와 중부발전, IITP AI 사업단 소개, 5개 AI 관련 연구센터 소개는 한국 인공지능 연구의 현재와 미래를 한 번에 볼 수 있는 중요한 자리였습니다. 인공지능에 관심있으신 많은 분들이 총 220여분 본 학술대회에 등록해주셨고, 온라인 상으로 100여명의 참여자가 상시 접속 해주셨습니다. 다소 생소한 온라인 방식의 학회였지만, 오프라인 학회에 못지않은 열기가 랜선을 통해 전달되었던 것은 두말할 나위 없습니다.

반년이 지나 선별된 18편의 논문에 대한 논문집을 발간하게 되어 다시 한 번 그날의 기억을 떠올립니다. 오늘도 또 그 후속의 연구를 진행하고 계실 한국의 인공지능 연구자들의 노고와 열정에 경외심을 느끼며, 그것이 한국, 나아가서는 인류 역사 전반을 풍요롭게 할 것임을 믿어 의심치 않기에 감사의 말씀을 전합니다.

2020 하계학술대회 조직위원장
서병기(UNIST)

한국인공지능학회에서는 2020년 11월 19일부터 11월 20일, 이틀간 추계학술대회를 온라인으로 개최하였습니다. 코로나 19라는 전세계적인 어려운 상황속에서도 세계적인 AI연구자들분들께서 발표를 해주시고, 많은 분들께서 참여해주셨습니다. Keynote 발표를 해주신 Microsoft연구소의 Tie-Yan Liu 박사님과 Ming Zho 박사님을 비롯하여 Invited Talk을 해주신 Wenjun Zeng 박사님, 황승원 교수님, Nan Zhou Duan 박사님, 이준영 박사님, 주한별 박사님에게 감사드립니다.



또한 최신 AI 기술들에 대한 발표를 해주신 25분의 국내 대학/연구소의 교수님 및 연구원분들께도 감사의 말씀을 드립니다. 본 학회에서 처음 시도되는 Industry 세션을 훌륭하게 이끌어주신 Korea Startup Forum 관계자분들께도 감사의 뜻을 전하고 싶습니다. 논문을 투고해 주신 연구자들 그리고 학회에 참석해 주신 많은 분들주신 참석해주신 많은 분들께도 감사의 말씀을 드리며, 본 학술대회가 한국인공지능 발전에 디딤돌이 될 수 있기를 기원합니다.

2020 추계학술대회 조직위원장
이미란(MSR), 김선주(연세대)

2020년 하계/추계학술대회에서 노력해 주신 모든 프로그램 위원분들에게 감사 인사를 드리며 이 글을 시작합니다. 작년에 진행된 하계/학술대회는 한국인공지능이 이룩한 연구성과를 교류하고 그 성과를 엄밀히 평가하는 학술 교류의 장이었습니다. 하계학술대회에서는 논문 모집을 1차와 2차로 구분하여 국내 최고전문가들의 수준 높은 의견 교환 및 평가를 진행하였으며 추계학술대회에서도 훌륭한 평가를 진행하였습니다. 이런 평가에 기초해서 2020 학술대회에서는 하계 10편(제1저자 기준 카이스트, 고려대, 서울대, 연세대), 추계 5편(제1저자 기준 성균관대, 서울대, 카이스트, 연세대)의 우수논문들이 선정되었으며, 추계에서는 특별히 MS 최우수논문을 선정하였습니다.

논문 심사를 통해서 한국인공지능 연구의 수준이 매우 향상되었고, 앞으로도 더 큰 발전 전망을 보이고 있다는 것을 알 수 있었습니다. 이런 성과를 낳게 해 주신 연구자분들과 심사위원분들에게 깊이 감사드리며 앞으로 논문심사에 더 많은 연구자들이 참여할 수 있도록 노력하겠습니다. 특별히 이번에 2020 학술대회의 심사위원으로 크게 노력해 주신 고려대 석흥일 교수님과 카이스트의 황성주, 신진우 교수님, 경북대의 정희철 교수님을 비롯한 여러 심사위원분들에게 깊은 감사의 인사를 드립니다.

또한 위 논문집이 앞으로 학회의 학술지 발간의 큰 디딤돌이 되기를 기대합니다.

2020 하계/추계학술대회 프로그램 위원장

김광수(KAIST)

▣ 하계학술대회 (2020년 8월 27~29일, 온라인) 조직위원회

• **조직위원장**	서병기 교수 (UNIST)		
• **조직위원**	김광수 교수 (KAIST)	김지환 교수 (서강대학교)	문일철 교수 (KAIST)
	석흥일 교수 (고려대학교)	이윤근 소장 (ETRI)	장동의 교수 (KAIST)
	정희철 교수 (경북대학교)	최준원 교수 (한양대학교)	
• **프로그램 위원장**	김광수 교수 (KAIST)		
• **프로그램 위원**	권준석 교수 (중앙대학교)	김수형 교수 (전남대학교)	김지환 교수 (서강대학교)
	문일철 교수 (KAIST)	문태섭 교수 (성균관대학교)	백승렬 교수 (UNIST)
	서병기 교수 (UNIST)	석흥일 교수 (고려대학교)	신진우 교수 (KAIST)
	유창동 교수 (KAIST)	이주호 박사 (AITRICS)	장길진 교수 (경북대학교)
	장동의 교수 (KAIST)	정희철 교수 (경북대학교)	조성호 교수 (KAIST)
	조영임 교수 (가천대학교)	주재걸 교수 (KAIST)	황성주 교수 (KAIST)

▣ 추계학술대회 (2020년 11월19~21일, 온라인) 조직위원회

• **조직위원장**	김선주 교수 (연세대학교)	이미란 전무 (Microsoft Research)	
• **조직위원**	김광수 교수 (KAIST)	김준모 교수 (KAIST)	최재식 교수 (KAIST)
• **프로그램 위원장**	김광수 교수 (KAIST)		
• **프로그램 위원**	권준석 교수 (중앙대)	문일철 교수 (KAIST)	백승렬 교수 (UNIST)
	석흥일 교수 (고려대)	유창동 교수 (KAIST)	이주호 교수 (KAIST)
	장길진 교수 (경북대)	장동의 교수 (KAIST)	정희철 교수 (경북대)
	주재걸 교수 (KAIST)	최재식 교수 (KAIST)	황성주 교수 (KAIST)

▣ 하계학술대회 승인논문 목록

1. Ahmad Wisnu Mulyadi, Eunji Jun and Heung-Il Suk. Uncertainty-Aware Variational-Recurrent Imputation Network for Improving In-Hospital Mortality Prediction on Clinical Time Series (Korea U.)

2. Bae Heesun, Moon Ilchul and Kim Giwoon. COVID-19 Spread Regimes with Temporally Calibrated SIR Model (Korea U., Soon Chun Hyang U. Hospital)

3. Byungju Choi and Jimin Hong. Hierarchical Transformer (Korea U., Humelo)

4. Canh Le and Chan-Hyun Youn. City-Scale Visual Place Recognition with Deep Local Features Based on Multi-Scale Ordered VLAD Pooling (KAIST)

5. Chaehyeon Lee and Heechul Jung. GridMix Data Augmentation (Kyungpook National U.)

6. Changha Lee, Gyusang Cho and Chan-Hyun Youn. Multivariate Load Forecasting for Multi-Client AMI Data Using CNN-LSTM Networks (KAIST)

7. Chang-Hee Han, Hyung-Tak Lee, Hwa-Ah-Nee Lee and Han-Jeong Hwang. Convolutional Neural Network Enhances the Performance of Using Ear-EEGs for Classification of Eyes-Closed and Eyes-Open State (Technical U. of Berlin, Korea U.)

8. Dahyun Kim. Overview of The Paper Related to Audio Visual Scene Aware Dialog (KAIST)

9. Dias Issa and Chang D. Yoo. Node embedding and fairness: A review (KAIST)

10. Dohyun Kim, Jangsup Moon and Taesup Moon. HSCI : Hippocampal Sclerosis Classification with Interpretation (Sungkyunkwan U., Seoul National U. Hospital)

11. Dohyung Kim, Sungho Park, Sunhee Hwang, Minsong Ki, Hyeran Byun and Seokyu Jeon. Fairness-Aware Batch Sampling for Image Classification (Yonsei U.)

12. Dongha Kim, Yongdai Kim, Yongchan Choi and Kunwoong Kim. Purifying Noisy Labels in Classification by Adversarial Search and Semi-supervised Learning (Seoul National U.)

13. Donghyun Lee and Ji-Hwan Kim. Language Model Using Neural Turing Machine Based on Localized Content-based Addressing (Sogang U.)

14. Dongjun Kim, Weonyoung Joo, Seungjae Shin, Kyungwoo Song and Il-Chul Moon. Adversarial Likelihood-Free Inference on Black-Box Generator (KAIST)

15. Eunji Jun, Ahmad Wisnu Mulyadi, Jaehun Choi and Heung-Il Suk. Uncertainty-Gated Stochastic Sequential Model for EHR Data Imputation and In-Hospital Mortality Prediction (Korea U., ETRI)

16. Ga-Young Choi, Won-Seok Kim, Han-Jeong Hwang, Miseon Shim, Hye-Ran Cheon, Dong-June Yeo and Soo-In Choi. Deep Learning Based EEG Neuronavigation to Identify Motor Hotspot (Korea U. Sejong, Seoul National U. Bundang Hospital, Kumho National Institute of Technology)

17. Gyuhyeon Sim, Jinho Choi, Hyesu Lim and Jaegul Choo. A Survey on Interactive Image Segmentation Using Deep Learning (KAIST, Tomocube Inc., Korea U.)

18. Hayeon Lee, Wonjun Yoon, Jinseok Park and Sung Ju Hwang. Learning Spatial Relationships for Cross-Modal Retrieval (KAIST, Lunit)

19. Heechul Jung, Yoonju Oh, Seongho Jeong, Chaehyeon Lee and Taegyun Jeon. 18. Contrastive Self-supervised Learning for Satellite Imagery (Kyungpook U., SIA)

20. Hogun Kee, Chanho Ahn and Songhwai Oh. Model Selection for Imitation Learning (Seoul National U.)

21. Hong-Gyu Jung and Seong-Whan Lee. Few-Shot Learning with Geometric Constraints (Korea U.)

22. Hyemi Kim, Seungjae Shin, Wanmo Kang and Il-Chul Moon. Analyzing Disentanglement on the perspective of a Linear VAE (KAIST)

23. Jae Hoon Shin, Jee Hang Lee and Sang Wan Lee. Deep Interaction between Reinforcement Learning Algorithms and Human Reinforcement Learning (KAIST, Sangmyung U.)

24. Jaesoon Baik and Junwon Choi. Active Learning for Images Using Consistency-based Acquisition Function (Hangyang U.)

25. Jee Seok Yoon and Heung-Il Suk. A Plug-in Approach to Factorization and Disentanglement (Korea U.)

26. Jee-Weon Jung, Hye-jin Shim, Ju-Ho Kim and Ha-Jin Yu. Comparison of Neural Attention Modules for Deep Speaker Embedding using RawNet in text-independent speaker verification (U. of Seoul)

27. Jeongho Park, Jae Goo Choy and Songhwai Oh. Memory Efficient Reinforcement Learning for Multi-tasks with Deep Virtual Q-Networks (Seoul National U.)

28. Jinheon Baek, Minki Kang and Sung Ju Hwang. Graph Representation Learning with Attention-based Set Pooling (KAIST)

29. Jihun Yun, Aurelie C. Lozano and Eunho Yang. Stochastic Gradient Methods with Block Diagonal Matrix Adaptation (U. of Seoul)

30. Jinheon Baek, Minki Kang and Sung Ju Hwang. Graph Representation Learning with Attention-based Set Pooling (KAIST)

31. Jin Hyeok Yoo, Yecheol Kim, Jisong Kim and Jun Won Choi. Generating Joint Camera-LiDAR Features for 3D Object Detection (Hanyang U.)

32. Jiyeon Lee, Wonjun Ko, Eunsong Kang and Heung-Il Suk. Personalized Regions Selection and Graph Relational Modeling for Early MCI Identification (Korea U.)

33. Joohyung Lee, Youngmoon Jung and Hoirin Kim. Speaker Embedding with Speech Posterior in Speech Enhancement (KAIST)

34. Juneha Jeon and Taesup Moon. Improving Medical Diagnosis Classifier Using Prior Knowledge with Interpretation-driven Regularization (Sungkyunkwan U.)

35. Junho Koh, Jaekyum Kim and Jun Won Choi. Video Object Detection Using Motion Context and Feature Aggregation (Hanyang U.)

36. Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee and Jinwoo Shin. Mitigating Dataset Bias with Biased Classifier (KAIST)

37. Junyeong Kim and Chang D. Yoo. Hierarchical Multimodal Attention Network for Open-ended Multi-modal Video Question Answering (KAIST)

38. Kang Haeyong and Yoo Chang D. Relational Information Bottlenecks For Scene Graph Generation (KAIST)

39. Kibeom Kim, Taehee Kim and Jaegul Choo. Graph Knowledge Integration into a Text Encoder (Korea U., KAIST)

40. Kookhoi Kim. Trends in the 3D Instance Segmentation (KAIST)

41. Kyungchae Lee, Changha Lee and Chan-Hyun Youn. Reasoning based Adaptive Image Compression for Efficient Satellite-Land Communication without Performance Degradation (KAIST)

42. Minsu Abel Yang, Jeehang Lee and Sang Wan Lee. Biological Reinforcement Learning via Predictive Spacetime Encoding (KAIST, Sangmyung U.)

43. Moogab Kim, Naveed Ilyas and Kiseon Kim. AttentionSeg: An Attention-based Convolutional Neural Network for Real-Time Object Segmentation from 3D Point Cloud (GIST)

44. Myeongseob Ko, Byeongchang Jeong, Daegyeom Kim and Cheol E. Han. Analyzing attention-based models for pneumonia detection (Korea U.)

45. Nuri Kim and Songhwai Oh. Active Multi-Class Object Detection Using Deep Reinforcement Learning (Seoul National U.)

46. Ran Heo and Eunjin Cho. Deepfake detection model based on fake attributes shown in images/videos (Dongyang Mirae U., RMIT U.)

47. Rusty Mina and Chang Yoo. Should you trust Sensitive Attribute Classification Accuracy as an Unfairness Measure? No (KAIST)

48. Sara Kim and Yongdai Kim. Issue of Constrained Fairness Methods (Seoul National U.)

49. Seanie Lee, Dong Bok Lee and Sung Ju Hwang. Neural Contrastive Learning with Natural Adversaries for Factually Consistent Text Summarization (KAIST)

50. Seulgi Hong, Jayeon Lim and Min-Kook Choi. What Is Wrong with Annotation? Quantifying the Annotation Quality via Noisy Label Learning (hutom)

51. Seungbum Hong and Min-Kook Choi. Deep Mutual Learning with Pretrained CNNs using Inhomogeneous Datasets (hutom)

52. Song, Hee, Shin, Jung and Lee. Midbrain dopamine activity during reinforcement learning reflects bias-variance tradeoff (KAIST, Solarluce, Seoul National U. Hospital)

53. Sunghun Kang and Chang D. Yoo. Counterfactual Augmentation for Fair Face Attribute Classification (KAIST)

54. Sunjae Yoon and Chang D. Yoo. VLANet: Video-Language Alignment Network for Weakly-Supervised Video Moment Retrieval (KAIST)

55. Sungmin Cho, Dohwi Kim and Junseok Kwon. Semi-Supervised Classification via Deep Metric Soft-max Learning (Chung-Ang U., Thermoeye Co.)

56. Sungpil Kho, Wonyoung Lee, Wonhyeok Im, Minsong Ki and Hyeran Byun. Vision-based Multi-person Pose Estimation: A Survey (Yonsei U.)

57. Tae-Eui Kam and Keun-Soo Heo. Deep Learning-based Noise Component Detection from Resting-state Functional MRI (Korea U.)

58. Thang Vu and Chang D. Yoo. 3D Instance Segmentation on Point Clouds: A Survey (KAIST)

59. Thanh Nguyen and Chang D Yoo. A survey on meta learning (KAIST)

60. Trung Pham and Chang D. Yoo. The Impact of Attention Mechanism in Fusion Step for Visual Question Answering (KAIST)

61. Tung M. Luu and Chang D. Yoo. Hindsight Goal Ranking on Replay Buffer for Sparse Reward Environment (KAIST)

62. Wonjun Ko and Heung-Il Suk. A Deep Reinforcement Learning Framework for Task-related Signals Selection for Motor Imagery-based Brain–Computer Interfaces (Korea U.)

63. Wonsik Jung, Eunji Jun and Heung-Il Suk. Modeling AD Progression on Longitudinal AD Biomarkers via Deep Recurrent Model (Korea U.)

64. Yongjin Shin, Gihun Lee, Seungjae Shin, Se-Young Yun and Il-Chul Moon. FedWR: Federated Weight Recovery from Ultra-Sparse Network forCommunication Efficient Federated Learning (KAIST)

▣ 추계학술대회 승인논문 목록

1. Daeun Lee and Joon S. Lim. ADHD Classification Based on Deep Learning Using MFCC Coefficients (Gachon U.)

2. Dahyun Kim, Sunjae Yoon, Jiwoo Hong and Chang D Yoo. Multi Window Attention by Distinguishing Movement of Torso and Movement of Arms and Leg (KAIST)

3. Dias Issa and Chang Yoo. Fair-Node2vec: Fair Graph Embedding through Biased Random Walks (KAIST)

4. Dohyeon Lee, Kyungjae Lee, Seungwon Hwang, Sunghyun Park and Seonhoon Kim. Orthogonal Disentanglement of Semantic Symbolic Representation for Query-Document Matching (Yonsei U., Naver Corporation)

5. Dongha Kim, Yongchan Choi and Yongdai Kim. Understanding and improving deep semi-supervised learning methods (Seoul National U.)

6. Dongha Kim and Yongdai Kim. Identifying Effects of Architecture Design to Invariance and Complexity in Deep Neural Networks (Seoul National U.)

7. Dongjun Kim, Kyungwoo Song, Yoonyeong Kim, Yongjin Shin and Il-Chul Moon. Sequential Likelihood-Free Inference with Implicit Surrogate Proposal (KAIST)

8. Eunjin Jeon, Wonjun Ko, Jee Seok Yoon and Heung-Il Suk. Toward Subject Invariant and Class Relevant Representation in BCI via Mutual Information Estimator (Korea U.)

9. Haeyong Kang and Chang D. Yoo. Directional Predicate Attention Embedding for Scene Graph Generation (KAIST)

10. GAN2GAN: Generative Noise Learning for Blind Denoising with Single Noisy Images (Sungkyunkwan U., Yonsei U.)

11. Jewook Lee, Pilhyeon Lee, Seogkyu Jeon, Sungho Park, Kibeom Hong and Hyeran Byun. Generalized Zero-Shot Action Recognition via Sequence Feature Generating (Yonsei U.)

12. Junyeong Kim, Sunjae Yoon, Dahyun Kim and Chang D. Yoo. Structured Co-reference Graph Attention Network for Audio-Visual Scene-aware Dialog (KAIST)

13. Kihyung Joo and Jungseop Son. Gasoline engine misfire detection with machine learning (Hyundai Motors Company)

14. Kyungwoo Song, Yohan Jung, Dongjun Kim and Il-Chul Moon. Implicit Kernel Attention (KAIST)

15. Muhammad Usama and Dong Eui Chang. Learning-Driven Exploration for Reinforcement Learning (KAIST)

16. Sangwon Lee and Gil-Jin Jang. Weakly Supervised U-Net for Sound Event Detection (Kyungpook U.)

17. Sara Kim, Kyusang Yu and Yongdai Kim. Within-group fairness: A new concept for Fairness AI (Seoul National U.)

18. Seogkyu Jeon, Pilhyeon Lee, Kibeom Hong and Hyeran Byun. Disentangling Identity and Aging Factor for Continuous Face Aging (Yonsei U.)

19. Sungmin Cha, Hsiang Hsu, Taebaek Hwang, Flavio Calmon and Taesup Moon. CPR: Classifier-Projection Regularization for Continual Learning (Sungkyunkwan U., Havard U.)

20. Sunghun Kang and Chang Yoo. Fair Attribute Classification via Intervened Facial Image Conversion (KAIST)

21. Sungmin Cho, Hyeseong Kim, Ji Soo Kim and Junseok Kwon. Noise Injected Markov Chain Monte Carlo Augmented by Adversarial Attack for Visual Tracking (Chung-Ang U.)

22. Trung Pham and Chang Yoo. The Power of Transfer Learning In The Big Data Era (KAIST)

23. Xiaowei Xing and Dong Eui Chang. The Adaptive Dynamic Programming Toolbox (KAIST)

24. Youngjae Park, Jinhee Park and Gil-Jin Jang. Colorectal Cancer Classification using U-Net and LinkNet on Pathological Images (Kyungpook U.)

▣하계학술대회 논문

▣추계학술대회 논문

한국인공지능학회 2020 하계 및 추계 학술대회 논문집

하계학술대회 논문

Multivariate Load Forecasting for Multi-Client AMI Data Using CNN-LSTM Networks

Changha Lee[1], Gyusang Cho[1] and Chan-Hyun Youn[1]

[1] School of Electrical Engineering, KAIST, Daejeon, South Korea, {changha.lee, cks1463, chyoun}@ kaist.ac.kr

Abstract

Efficient and precise prediction on future energy consumption with Advanced Metering Infrastructure (AMI) data can leverage more information and be used for efficient system management with enhanced usability. Due to the nature of the data, short-term and long-term time series relative features are included in the adjacent data as well as the value itself. We apply a novel architecture of deep neural network(DNN) with CNN and LSTM layer to predict the short or long-term features of the AMI data.

Keywords— *Load Forecasting, Multivariate Prediction, Deep Learning, Machine Learning*

I. Introduction

Recent consumption of electrical energy has been on the rise due to the increase in population, economic growth, and the advent of new electric devices. The electricity industry also grows with the increasing demand for electrical energy, developing from the traditional electric management system to a new electric management system following the invention of two-way communication and automatic meter. This is called a smart grid. Smart grid is an intelligent power grid system that enhances efficiency by interacting with suppliers and consumers by incorporating information and communication technology into the production, transportation and consumption processes of electricity. Research has been actively conducted to provide more efficient system management and intelligent services by building intelligent power grid systems.

Load Forecasting is an essential task for the development of modern power systems. Technologies such as renewable energy, electric vehicles, and energy storage systems being used in modern power systems make it necessary to predict the time-varying load demands. As the complexity and uncertainty of the development of modern power systems increases, it is challenging to predict the load requirements of time series data of time-variability. More accurate predictive models have been studied in analyzing time series data, starting with the basic algorithm, such as Linear Regression (LR) [1] and Autoregressive Integrated Moving Average (ARIMA) [2]. Recently, with advances in deep running, neural layers such as Recurrent Neural Network (RNN) or Long Short-Term Memory (LSTM) [4] are being studied, and models for analyzing time series data such as Bi-LSTM [6], and CNN-LSTM [3] are being studied and developed.

CNN-LSTM is a neural network combining convolutional neural network (CNN) with LSTM. Therefore, CNN-LSTM extracts spatial and temporal features and learns both data characteristics. This neural network is widely used to time-series data analysis. For example, according to [7], a combined model of CNN and LSTM was used to accurately detect arrhythmia in the ECG which is a subject of medical field. The hybrid deep neural network consisting of CNN and LSTM is studied for remaining useful life prognostics in [5].

Our proposed CNN-LSTM method uses CNN and LSTM layer to predict multivariate future data from short-term to long-term. First, the data of one dimension for time is reconstructed into two dimensional data considering spatial feature. The convolution layer is used to extract spatial feature from input data. The outputs of CNN are used as the input of LSTM layer. Finally, the outputs of LSTM are connected to Fully-Connected layer to predict multivariate predictive load. We propose a novel hybrid cnn-lstm network to predict multivariate power consumption in multi-client data environment.

II. Method

Our target application is the prediction of multivariate time-series data in multi-client environment. The proposed model learns data characteristics to predict short or long term value from an univariable data preprocessed by 36 timestamp window. univariable data consists of only power consumption according to time. The train dataset forwards into model and the prediction value is obtained. The predictive value is used to calculate the gradient to update deep learning model parameter.

A. 2D Reformator

First of all, the proposed model transforms time-series data of which one dimension for time into separate time interval to make two dimension for time. When setting windowed n-length data as a $X_i = \{x_0, x_2, ..., x_{n-1}\}$, we first modify the data dimension as a $X_i \in \mathbb{R}^{\sqrt{n}\times\sqrt{n}}$. The $X_{i,w,h}$ means the value x_{w+h} in X_i where $0 \leq w, h \leq \sqrt{n}-1$. We set $\sqrt{n}-1$ as a K for the rest of this paper. Windowing length n should be a square number to ensure natural number of dimension. This reformation procedure is simple method to assist spatial feature extractor. In multiple client environment, a spatial feature of two dimensional data is various according to each clients.

B. CNN-LSTM model

The overall CNN-LSTM network architecture is described in Fig. 1. The proposed model consists of multiple layer including CNN and LSTM. The 2-dimensional reshaped data forwards a convolution layer. The output of convolution layer is shown in Equation 1.

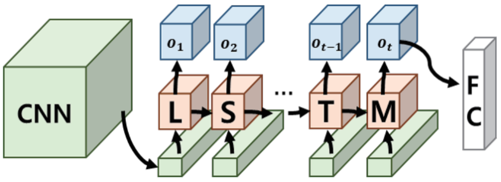


Fig. 1. The description of CNN-LSTM network architecture.

$$y_{i,j} = \sigma(b_j + \sum_{k=0}^{K-1} w_{j,k} * X_{i,k}) \tag{1}$$

where $X_{i,k} = \{X_{i,k,0}, ...X_{i,k,K-1}\}$, operator $*$ is the cross-correlation, b is a bias, w is a trainable weight of the kernel, and σ is an activation function ReLU in this paper. The output in Equation (1) means a spatial feature vector according to convolution of kernel and input.

The output of convolution is fed into LSTM to extract temporal feature. A LSTM unit has main components consisting of an input gate, a forget gate, an output gate, and a memory cell. When setting input of LSTM as the previous output s meaning spatial features, the input gate, forget gate, and output gate are defined as follows:

$$i_t = \sigma_g(W_i^i s_t + b_i^i + W_i^h h_{t-1} + b_i^h) \tag{2}$$

$$f_t = \sigma_g(W_f^i s_t + b_f^i + W_f^h h_{t-1} + b_f^h) \tag{3}$$

$$o_t = \sigma_g(W_o^i s_t + b_o^i + W_o^h h_{t-1} + b_o^h) \tag{4}$$

where W_x^i and W_x^h or b_x^i and b_x^h sequentially mean the connecting weights or bias for input and hidden at any x unit, and σ_g is an activation layer for gate. We use the sigmoid activation layer for gates. The dimensions of weight matrices are defined as follows: $W^i \in \mathbb{R}^{h\times n}$ and $W^h \in \mathbb{R}^{h\times h}$ where h is the number of hidden units. The gate activation variables construct LSTM cell and hidden units as following Equation (5) and Equation (6).

$$c_t = f_t \odot c_{t-1} + i_t \odot \sigma_h(W_c^i s_t + b_c^i + W_c^h s_t + b_c^h) \tag{5}$$

$$h_t = o_t \odot \sigma_h(c_t) \tag{6}$$

where operator $\odot$ is the element-wise multiplication, and σ_h is specified by tanh activation layer according to LSTM. The outputs $h_t \in \mathbb{R}^h$ indicates temporal feature of the data.

The last layer of CNN-LSTM is a fully connected layer to predict multivariate loads. Considering h_t has data $\{h_0, h_1, ..., h_{h-1}\}$, a output of fully connected layer is defined as follows:

$$y_j = \sum_{x=0}^{h-1} w_{x,j} h_x + b_j \tag{7}$$

The output y_j in Equation (7) corresponds one of multivariate prediction value. Therefore, the entire predictive value is expressed as follows:

$$Y_i = \{y_0, y_1, ..., y_j, ..., y_p\} \tag{8}$$

where p is a prediction length. Whether the target application is a short-term or a long-term forecast depends on the size of the p value.

III. Experiments

A. Dataset Description

We evaluated proposed CNN-LSTM and other comparable methods on the AMI meter dataset collected by Korea Electric Power Corporation (KEPCO). The automatic smart meter records active power value at every 15 minutes. Each time-series data was preprocessed with sliding window algorithm. The sliding window algorithm is used to create input data and target data. The input of dataset consists of windowed 36 variables according to time step. According to Equation 8, the range of predicted length is [1, 2, 3, 10, 36]. The number of clients in the training is 48 different clients, and the number of test clients consist of 12 different clients from the training dataset. The dataset includes around 1.3 million windowed data for training and around 0.4 million windowed data for test.

B. Performance comparison with CNN-LSTM

We conducted experiments to validate that the proposed network shows better performance than other CNN-LSTM and deep learning-based models. The deep learning-based models, such as MLP, LSTM, and GRU is used to predict

Method	Description	Params
LSTM	LSTM+Dense(p)	18.2-20.5K
GRU	GRU+Dense(p)	13.9-16.2K
CNN-LSTM [3]	Conv1D+MaxPooling+Conv1D+MaxPooling+LSTM+Dense(32)+Dense(p)	43.5-44.7K
CNN-LSTM (Ours)	Reshape+Conv2D+LSTM+Dense(p)	33.7-36.0K

Table 1. The model architecture description according to prediction length p. According to value p, last layer has different trainable parameters.

p	1		2		3		10		36	
Method	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
LSTM	0.0127	0.0552	0.0179	0.0632	0.0204	0.0669	0.0358	0.0877	0.0686	0.1319
GRU	0.0137	0.0595	0.0175	**0.0619**	0.0196	0.0658	0.0366	0.0888	0.0678	0.1326
CNN-LSTM [3]	0.0200	0.0663	0.0222	0.0677	0.0238	0.0713	0.0353	0.0880	**0.0670**	0.1320
CNN-LSTM (Ours)	**0.0117**	**0.0519**	**0.0172**	0.0665	**0.0195**	**0.0633**	**0.0352**	**0.0858**	0.0671	**0.1299**

Table 2. The experimental results for test dataset.

short or long term values. The other CNN-LSTM model was re-deployed with reference [3] and we reproduced results for the windowed AMI meter dataset. The learning rate starts at 0.01 and decreases to one-tenth of the current rate at one-third and two-thirds of the total learning step. We train the networks for 30 epochs and 128 mini-batch size.

For fair comparison, we unified the configuration of some layers as following Table 3. The other layers, such as fully connected or pooling layers, follow a description as referred in [3].

Common Layer	Filter	Kernel size	Stride
Convolution1D	64	(2,1)	1
Convolution2D	64	(2,2)	(1,1)
LSTM	64	-	-
GRU	64	-	-

Table 3. The configuration of common layer.

The model architecture and the number of model parameter are introduced in Table 1. As can be seen in Table 1, the Params varies according to p.The CNN-LSTM model we proposed reduced computational complexity by 20% less trainingable parameters than the previous model [3].

Experimental results with other deep learning methods are shown in Table 2. We observed that the proposed model performs better in the short- to mid-term. Additionally, for training datasets consisting of only load data, LSTM and GRU were also observed to be performing well enough and rather poor performance by traditional CNN-LSTM [3] techniques. This means that existing CNN-LSTM performs well only for multivariable and long-term data. When data constitute only one feature rather than multiple features, it can be seen that multivariable data can perform sufficiently high even in a network of simple layers, such as LSTM, GRU, and the proposed CNN-LSTM.

IV. Conclusion

We focus on predicting future energy within previous AMI data, with capturing short-term and long-term features. It is to be cost-efficient for acquiring more precise-future data with high-accuracy network within small amount of parameters in the network. Consequently, we are able to draw connections of the past and future data by proposed CNN and LSTM network. Additionally, this paper provides the information needed to select a model for predicting time series data. However, our results should be verified in a commonly used set of benchmark data. Therefore, future work should focus on the general performance and be compared with other deep learning models.

References

[1] Arthur S Goldberger. Best linear unbiased prediction in the generalized linear regression model. *Journal of the American Statistical Association*, 57(298):369–375, 1962.

[2] Xin Jin, Yao Dong, Jie Wu, and Jujie Wang. An improved combined forecasting method for electric power load based on autoregressive integrated moving average model. In *2010 International Conference of Information Science and Management Engineering*, volume 2, pages 476–480. IEEE, 2010.

[3] Tae-Young Kim and Sung-Bae Cho. Predicting residential energy consumption using cnn-lstm neural networks. *Energy*, 182:72–81, 2019.

[4] Weicong Kong, Zhao Yang Dong, Youwei Jia, David J Hill, Yan Xu, and Yuan Zhang. Short-term residential load forecasting based on lstm recurrent neural network. *IEEE Transactions on Smart Grid*, 10(1):841–851, 2017.

[5] Zhengmin Kong, Yande Cui, Zhou Xia, and He Lv. Convolution and long short-term memory hybrid deep neural networks for remaining useful life prognostics. *Applied Sciences*, 9(19):4156, 2019.

[6] Tuong Le, Minh Thanh Vo, Bay Vo, Eenjun Hwang, Seungmin Rho, and Sung Wook Baik. Improving electric energy consumption prediction using cnn and bi-lstm. *Applied Sciences*, 9(20):4237, 2019.

[7] Shu Lih Oh, Eddie YK Ng, Ru San Tan, and U Rajendra Acharya. Automated diagnosis of arrhythmia using combination of cnn and lstm techniques with variable length heart beats. *Computers in biology and medicine*, 102:278–287, 2018.

Overview of The Paper Related to Audio Visual Scene Aware Dialog

Dahyun Kim[1] and Chang D. Yoo[1]

[1] Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea, {dahyun.kim, cd_yoo}@kaist.ac.kr

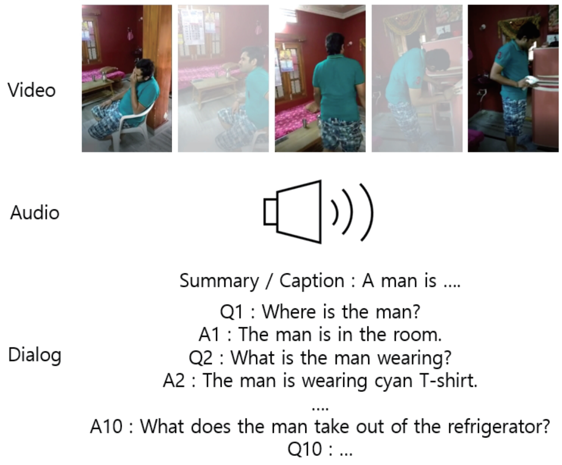
Abstract

To apply AI to practical applications, research using various data is needed. For this research, we introduce Audio Visual Scene Aware Dialog Task using Multimodal data. In order to generate Answer for Question through Visual Video, Audio, and Dialog, the design of a model with a comprehensive understanding of multimodal data is required. Introduce various paper related AVSD task and explain the motivation of each paper. And comparing the results of each study, suggest how to effectively deal with multimodal data. Finally, this paper suggests future research direction through the lack of previous research.

Keywords— *Deep Learning, Machine Learning, Audio Visual Scene Aware Dialog, AVSD, VQA, question answering*

I. Introduction

AI has developed in various fields. AI has been applied to various existing fields such as Visual Task and Audio Task and has made many improvements in performance. And because of this development, there are many efforts to apply AI to the real application. However, to apply AI to actual applications, various tasks must be combined. For example, in order to communicate with a person, the model needs to understand what people saying and the situation related to the conversation at the same time. It is necessary to understand and use various modalities. As such, we introduce the Audio-Visual Scene Aware Dialog (AVSD) [5] Task, one of the datasets using multimodal data. And in this paper, we compare the researches on AVSD. AVSD Dataset consists of Visual Video, Audio, and Dialog. Visual Video is a video of about 30 seconds in which one or more people act. Audio has sound information related to Visual Video. And Dialog consists of Summary, Caption, and History. Summary and Caption are sentences that describe the overall video. History is about two people recording video-related questions and answers. History is a question and answer related to the video. A person who asks the question does not see the video and asks 10 questions to understand the video. The other watches videos and answer questions. AVSD Task is intended to create a model for making answer for the question, through this dataset. The model should have a comprehensive understanding of Visual video, Audio, and History. And the model should generate an answer according to these multimodal data and question, The AVSD Task is conducted in DSTC7[1] and DSTC8, and various researches are conducted. Section2 introduces the motivation of researches. Section 3 compares the performance of the researches introduced in Section 2, and Section4 discusses conclusions and future research directions.



Summary / Caption : A man is ....
Q1 : Where is the man?
A1 : The man is in the room.
Q2 : What is the man wearing?
A2 : The man is wearing cyan T-shirt.
....
A10 : What does the man take out of the refrigerator?
Q10 : ...

Fig. 1. Example of AVSD dataset

II. Motivation of papers

End-to-End Audio-Visual Scene–Aware Dialog using Multimodal Attention – Based Video Features [3] (Model 1) introduces models that enable end to end learning using AVSD Dataset. Later, this model is used as a baseline model for comparison in various other researches

A Simple Baseline for Audio–Visual Scene–Aware Dialog [8] (Model 2) uses only four frames by subsampling, not all video data when using visual video data. This subsampled video frames have enough information to help improve performance and dramatically reduce training speed. In addition, to use multimodal data, multimodal attention

Model	BLEU1	BLEU2	BLEU3	BLEU4	METEOR	ROUGE_L	CIDEr
Without summary / caption							
Model1[3]	0.256	0.131	0.109	0.078	0.113	0.277	0.727
Model2[8]	0.285	0.187	0.131	0.096	0.128	0.311	0.941
Model4[2]	0.305	0.200	0.144	0.109	0.138	0.366	1.132
Model6[6]	0.677	0.556	0.462	0.387	0.249	0.544	1.022
Using caption in training							
Model3[4]	0.727	0.593	0.488	0.405	0.273	0.566	0.118
Using summary / caption							
Model5[7]	0.307	0.210	0.151	0.113	0.145	0.339	1.180
Model6[6]	0.746	0.626	0.528	0.445	0.286	0.598	1.240

Table 1. The Performance of models

is used using factor graph attention.

Joint Student-Teacher Learning for Audio-Visual Scene-Aware Dialog[4] (Model3) uses the Student-Teacher Learning for training models. Teacher network, along with video and history, uses caption for training. And the Student network is trained to mimic the Attentional multimodal fusion vector and the generated answer in the Teacher network using only video and history. By doing so, The model produces results similar to those with captions without captions.

Learning Question-Guided Video Representation for Multi-Turn Video Question Answering [2] (Model4) effectively utilizes the video features of the portion related to the Question. It may be time inefficient to encode the entire video sequence. For summarizing the video frame features efficiently, compute the similarity between video and question, use the score in attention mechanism. Calculate similarity score between video feature and token about the question. Similarity score makes gate that emphasizes frame related to question

CMU Sinbad's Submission for the DSTC7 AVSD Challenge [7] (Model5) yields state-of-the-art accuracy in DSTC7. Multimodal feature integration improves performance. For fusing contribution from a different modality, this paper uses hierarchical attention. Since this paper, hierarchical attention is used to use multimodal data in researches related to AVSD.

Bridging Text and Video: A Universal Multimodal Transformer for Video-Audio Scene-Aware Dialog[6] (Model6) yields state-of-the-art accuracy in DSTC8. Learn joint representations among different modalities as well as generate informative and fluent responses. This paper design a universal multimodal transformer to encode different modalities and generate a response at the same time.

III. Evaluation

In this section, we introduce the performance of models explained earlier. To compare the quality of the generated answer, some measures are used. The objective measures used are BLEU, METEOR, ROUGE_L, and CIDEr. These objective measures calculate the similarity between the sentence generated by the model and the ground true sentence generated by the person.

The overall performance is as shown in Table1. Most models introduced earlier do not use caption and summary. And then, the performance of Bridging Text and Video: A Universal Multimodal Transformer for Video-Audio Scene-Aware Dialog[6] (Model6) is the best. And even with caption and summary, The same model's performance is the best.

Most researches use various data through multimodal attention. And for this multimodal attention, various methods are used, such as factor graph attention and hierarchical attention. Because performance improvement is achieved through various attention methods, it can be understood that various data are used effectively. And the most effective way to perform is transformer. Unlike other methods, different types of data are attached and used together, rather than encoding different types of data separately. This would have enabled a comprehensive understanding of various data.

IV. Conclusion and Discussion

Multimodal data must be used to apply AI to actual applications. In order to proceed with these studies, AVSD dataset using multimodal data is introduced. We introduced the papers related AVSD and explain the motivation and its performance of each research. Most researches focus on a comprehensive understanding of multimodal data. For this, various methods such as attention and transformer are used, and many performance improvements are made compared to the Baseline model.

However, some further research is still needed. When looking at the generated answer, there is something in common about the wrong answer. For example, questions about a person's action, or questions about the object's characteristics, do not answer correctly. To solve this problem, further research will be required on how to use features of data or to perform analysis from the typical aspects of data.

REFERENCES

[1] Huda Alamri, Chiori Hori, Tim K Marks, Dhruv Batra, and Devi Parikh. Audio visual scene-aware dialog (avsd) track for natural language generation in dstc7. In *DSTC7 at AAAI2019 Workshop*, volume 2, 2018.

[2] Guan-Lin Chao, Abhinav Rastogi, Semih Yavuz, Dilek Hakkani-Tür, Jindong Chen, and Ian Lane. Learning question-guided video representation for multi-turn video question answering. *arXiv preprint arXiv:1907.13280*, 2019.

[3] Chiori Hori, Huda Alamri, Jue Wang, Gordon Wichern, Takaaki Hori, Anoop Cherian, Tim K Marks, Vincent Cartillier, Raphael Gontijo Lopes, Abhishek Das, et al. End-to-end audio visual scene-aware dialog using multimodal attention-based video features. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2352–2356. IEEE, 2019.

[4] Chiori Hori, Anoop Cherian, Tim K Marks, and Takaaki Hori. Joint student-teacher learning for audio-visual scene-aware dialog. In *INTERSPEECH*, pages 1886–1890, 2019.

[5] B. Klein, G. Lev, G. Sadeh, and L. Wolf. Associating neural word embeddings with deep image representations using fisher vectors. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4437–4446, 2015.

[6] Zekang Li, Zongjia Li, Jinchao Zhang, Yang Feng, Cheng Niu, and Jie Zhou. Bridging text and video: A universal multimodal transformer for video-audio scene-aware dialog. *arXiv preprint arXiv:2002.00163*, 2020.

[7] Ramon Sanabria, Shruti Palaskar, and Florian Metze. Cmu sinbads submission for the dstc7 avsd challenge. In *DSTC7 at AAAI2019 workshop*, volume 6, 2019.

[8] Idan Schwartz, Alexander G Schwing, and Tamir Hazan. A simple baseline for audio-visual scene-aware dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12548–12558, 2019.

Node embedding and fairness: A review

Dias Issa[1] and Chang D. Yoo[1]

[1] Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea,
{dias.issa , cd$_{yoo}$}$@kaist.ac.kr$

Abstract

A great variety of research has been dedicated to developing node embedding techniques in recent years. Nevertheless, extremely few works addressed the issue of mitigating potential bias in produced embeddings. In this paper, we review some of the prominent node embedding algorithms along with pioneering works in the direction of fair node embedding.

Keywords— *Graph Deep Learning, Node embedding, Fairness*

I. Introduction

Node (graph) embedding techniques are utilized in various fields including computational biology, linguistics, and social networks [3]. The original aim of graph embedding algorithms is in mapping the nodes of a graph to a vector space with a specific number of dimensions as a hyperparameter. The embedding vectors could be utilized for different purposes such as link prediction, node classification, and recommendation.

A great variety of techniques were proposed in previous research. Part of them such as Structural Deep Network Embedding (SDNE) [10] or Large-scale Information Network Embedding (LINE) [9] technique employ deep autoencoders, first and second-order network proximities for generating graph embeddings. While others, such as DeepWalk [7] or Node2vec [4] utilize the approach of random walks. Nevertheless, most of the current graph embedding models do not consider fairness constraints and produce biased embedding vectors, which in turn leads to discriminatory predictions of machine learning models utilizing these embedding vectors.

In next section several prominent background works in node embedding field are reviewed. After that, the pioneering works in the direction of introduction of fairness constraint in node embedding are examined. Finally, the analysis and conclusions are provided.

II. Graph Embedding algorithms

A. *Deepwalk: Online learning of social representations*

Deepwalk [7] is one of the most popular graph embedding algorithms that was first of it's kind widely used as a benchmark for comparison of graph embedding techniques. This approach is one of the representatives of the family of graph embedding algorithms that utilize walks. In graph theory, the walk is the concept that allows traversing graph by moving from one vertex to another using their common edge. Perrozi et al. [7] employed a random walk technique, which kind of walk in a graph where the next node in the path is chosen randomly among all neighbors of the current node. The key idea of a Deepwalk [7] algorithm is in the perception of random walks in a graph as sentences in a document. In natural language processing (NLP) field such algorithms as Word2vec [6] predicts the probability of a word to appear in a sentence, given the surrounding words. Perrozi et al. [7] utilized this idea in graph setting as the estimation of the probability of appearance of a node in a random walk given the previous nodes. Similarly to Word2vec, Deepwalk learns feature vectors for nodes in the graph that are utilized for estimation of this probability. In other words, Deepwalk learns node embeddings that could be further utilized for downstream tasks as node classification, link prediction, recommendation, etc.

B. *node2vec: Scalable feature learning for networks*

Node2vec is another extremely popular algorithm that utilizes the idea of Deepwalk of representing random walks in a graph as sentences in a document for further feature learning using NLP techniques. However, the significant difference lies in the random walk sampling strategy. While Deepwalk samples the next node in a random walk by randomly choosing among the neighbors of a current node, Node2vec employs search bias variable a. This variable has two parameters: p and q. The first parameter p represents the priority of a breadth-first-search (BFS) procedure, while the second parameter q represents the priority of a depth-first-search (DFS) procedure. Therefore, probabilities $1/p$ and $1/q$ impact the next node appointment decision. Figure 1 depicts BFS and DFS algorithms. It could be clearly seen that BFS is appropriate for learning a local neighborhood, while DFS is appropriate for learning a global neighborhood. This subtle difference allowed Node2vec significantly outperform Deepwalk in such tasks as link prediction and node classification.

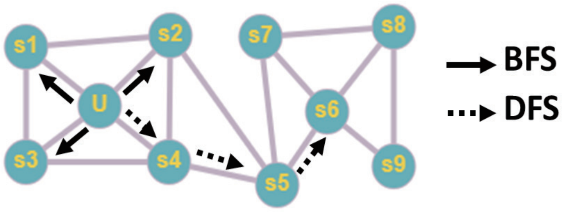


Fig. 1. BFS and DFS procedures [4].

C. *Graphgan: Graph representation learning with generative adversarial nets*

GraphGAN [11] in contrast to previous graph embedding techniques works in a slightly different way. Wang et al. [11] try to learn feature vectors for nodes of graph utilizing the widely known Generative Adversarial Network (GAN) approach. The generator in GraphGAN tries to predict the neighbors of a given root node utilizing its feature vector (embedding) and feature vectors of other nodes in a graph. The discriminator tries to identify whether the input neighbors are real neighbors or neighbors generated by the generator. Figure 2 illustrates the idea of GraphGAN architecture. This architecture allowed Wang et al. [11] to produce graph embeddings out of real-world datasets that demonstrate significant gain over previous algorithms in link prediction, node classification and recommendation tasks.

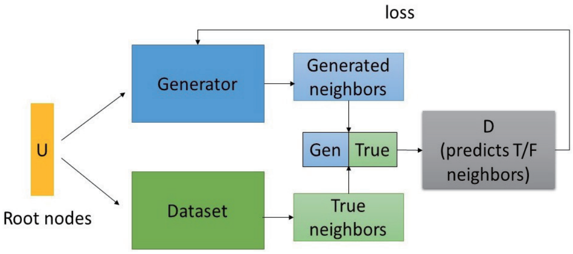


Fig. 2. The principles of GraphGAN performance.

D. *HARP: Hierarchical Representation Learning for Networks*

Chen at al. [2] introduced the improvement for previously implemented graph embedding algorithms. The problem of previous models could be in a risk of getting stuck in a local minima due to non-convex optimization using stochastic gradient descent. Therefore, the authors introduced the technique of graph coarsening, where similar nodes are merged into supernodes. Then the embedding procedure is done using graph of the supernodes. These supernode embedding vectors are then utilized as during initial weight initialization of nodes that form these supernodes.

Due to such strategy, HARP could be utilized as a preprocessing stage for previously implemented graph embedding techniques. This approach allowed Chen et al. [2] to increase the performance up to 14% of Macro F1 score compared to original graph embedding techniques.

III. Towards Fair Node Embedding

Most of the current graph embedding generation algorithms that are applied for various tasks such as link prediction, node classification, and recommendation do not consider fairness constraints into account. Graph generating models utilize real-world graphs as ground truth for learning, however, according to Rahman et al. [8] real-world graphs could be significantly unfair, and the algorithms utilizing these graphs further propagate these biases resulting in discriminatory predictions [8, 1].

A. *Compositional fairness constraints for graph embeddings*

The first research work about the fairness of graph embeddings was introduced by Bose and Hamilton [1]. The authors did not consider the issue of generating fair graph embeddings, they focused on the post-processing of feature vectors generated for nodes of a graph. The authors aimed to train filter-models to filter out the sensitive information out of generated embeddings. For example, for the task of movie recommendation, the graph of the dataset consists of nodes, representing users and movies, and edges between these nodes, representing the recommendation procedure [1]. The authors filtered-out the information about the gender, age, and occupation of the users from users' embedding vectors. The main idea of the authors' paper was in a combination of such filters in order to produce embeddings, which are invariant with respect to several fairness constraints.

Figure 3 illustrates this approach. Firstly, embeddings are fed to filter networks. Secondly, each filter network filter-out its targeted sensitive attribute and produce the output embedding invariant to a single sensitive attribute. Thirdly, the mean of outputs of filter networks is fed to discriminators, where each of the discriminators tries to predict its targeted sensitive attribute from the given input embedding vector. Finally, the loss from the discriminators is utilized for boosting the performance of the filter networks.

Figure 4 illustrates the prediction bias of the 3 different models trained on movie recommendation dataset MovieLens1M, which contains 1 million movie ratings by 6000 users on 4000 movies [5]. Prediction bias is calculated for each movie as an absolute difference in average predicted scores for users with different sensitive attributes. Then,

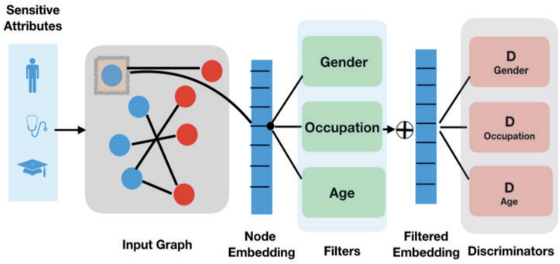


Fig. 3. The approach of Bose and Hamilton [1].

prediction bias is averaged across all movies. For example, for gender as a sensitive attribute, prediction bias is formulated as average absolute difference in predicted ratings for male vs. female users, across all movies.

The first baseline model produces graph embeddings without consideration of any fairness constraints, the second model outputs the feature vectors invariant for a single sensitive attribute, the third model produces graph embeddings invariant to a combination of several sensitive attributes. Three sensitive attributes were chosen: gender, age, and occupation. The interesting observation is that the compositionally trained adversary outperformed the single adversary, thus, removing more sensitive information. This leads to the fact that gender, age, and occupation are correlated in this dataset.

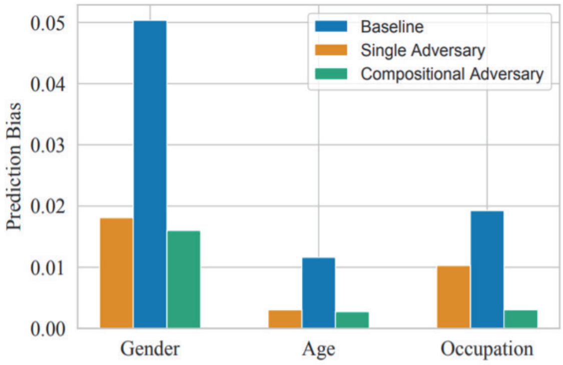


Fig. 4. Prediction bias for different sensitive attributes in movie recommendation dataset [1].

B. *Fairwalk: towards fair graph embedding*

In contrast to previous work, Rahman et al. [8] propose the method of generation of fair graph embeddings using a fair version of the node2vec algorithm, referred to as Fairwalk. Their idea is based on the modification of the random walk policy by introducing sensitive attribute bias. During the random walk procedure, before the procedure of randomly choosing the next node among the given set of neighborhood nodes, Rahman et al. [8] divide neighbors into sets based on their sensitive attribute value. After that, the authors randomly choose one of the sets and allow random walk procedure to continue by randomly selecting the next node among the chosen set of neighbors. For example, for gender, as a sensitive attribute, on each step of selection of the next node in random walk procedure for node *u*, its neighbors are divided into *female* and *male* neighbors. Depending on which set is selected, the next node of a random walk is chosen among either *female* or *male* neighbors. Utilizing this approach, the authors were able to reduce bias in friendship recommendation task in Instagram graph dataset in terms of demographic parity, which is a group fairness measure that requires the acceptance rates of different groups to be equal.

IV. Analysis and Conclusions

Most of the prominent node embedding algorithms do not take fairness into account. The works of Rahman et al. [8] and Bose and Hamilton [1] are pioneers in the field of fair node embedding, however, there exist several disadvantages of these methods. The drawback of Fairwalk [8] algorithm resides in that it could use only a single fairness constraint for the embedding generation process. This leads to the issue that Fairwalk could not produce embedding vectors invariant for several sensitive attributes. While the approach of Bose and Hamilton [1] applies filtering algorithms on already generated embedding vectors. Therefore, there is no fair graph embedding generation approaches, which enforce invariance to several sensitive attributes during the embedding generation procedure. Addressing this gap could be a potential direction for future research.

To sum up, there are very few works in the field of fair node embedding, and a wide range of issues that should be solved. We believe that the research on multi-constraint fair graph embedding generation techniques is very promising direction for future studies.

References

[1] Avishek Bose and William Hamilton. Compositional fairness constraints for graph embeddings. In *Proceedings of the International Conference on Machine Learning*, pages 715–724, 2019.

[2] Haochen Chen, Bryan Perozzi, Yifan Hu, and Steven Skiena. Harp: Hierarchical representation learning for networks. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[3] Palash Goyal and Emilio Ferrara. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151:78–94, 2018.

[4] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[5] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.

[6] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[7] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

[8] Tahleen Rahman, Bartlomiej Surma, Michael Backes, and Yang Zhang. Fairwalk: towards fair graph embedding. In *Proceedings of the 2019 International Joint Conferences on Artifical Intelligence (IJCAI). IJCAI*, pages 3289–3295, 2019.

[9] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pages 1067–1077, 2015.

[10] Daixin Wang, Peng Cui, and Wenwu Zhu. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1225–1234, 2016.

[11] Hongwei Wang, Jia Wang, Jialin Wang, Miao Zhao, Weinan Zhang, Fuzheng Zhang, Xing Xie, and Minyi Guo. Graphgan: Graph representation learning with generative adversarial nets. In *Proceedings of the Thirty-second AAAI conference on artificial intelligence*, 2018.

Purifying Noisy Labels in Classification by Adversarial Search and Semi-supervised Learning

Dongha Kim[1], Yongchan Choi[1], Kunwoong Kim[1] and Yongdai Kim[1]

[1] Department of Statistics, Seoul National University, Seoul, South Korea, {dongha0718, pminer90, kwkimm.online, ydkim903}@snu.ac.kr

Abstract

The label noise problem in classification has been studied for several decades and recently many methods have been proposed to resolve this problem for deep neural networks (DNNs). In this study, we propose a new and novel learning framework so called adversarial search and semi-supervised learning (ASSL). ASSL exploits the memorization capability of DNNs, which is known to be a major drawback of DNNs for noisy label problems, to purify noisy labels by developing a new technique closely related to the adversarial training method. And ASSL uses the semi-supervised learning by regarding the confusing samples as unlabeled samples rather than excluding them out of the train data. To demonstrate the superiority of ASSL we conduct various experiments on the well-known benchmark data sets whose results show that ASSL outperforms previous state-of-the-art baselines by large margins. Especially, we observe that ASSL is very robust to data with highly noised labels, e.g. for heavy noise scenarios, ASSL improves the state-of-the-art test accuracies by more than 10% on image data sets and more than 25% on sentence data sets.

Keywords*— Deep Learning, Noisy label problem, Adversarial training, Semi-supervised learning*

I. Introduction

Learning with deep neural networks (DNNs) has achieved impressive results on many classification problems [21, 13], but its success highly relies on massive train samples with high-quality labels such as ImageNet [7] and MS-COCO [25]. Since annotating is usually done manually by human experts, it is expensive, time-consuming and even impossible to obtain huge clean labeled data. On the other hand, it is possible to access numerous data whose labels are easy to collect but relatively inaccurate. We are able to get these noisy-annotated data through internet search engines [9, 19, 34, 42] or hashtags, which is much cheaper and time-efficient than human labeling. Hence, it is crucial to exploit noisy labeled data to obtain a good classifier.

In this study, we provide a new and novel learning framework with noisy labeled data which is called ASSL (Adversarial Search and Semi-supervised Learning). ASSL consists of two steps. First, we conduct a supervised learning with noisy labeled data to get a prediction model which is smooth but over-fits the train data. With this model, we refine the train data using our new technique, called RAS (Refinement with Adversarial Search). Then, we extract good labeled samples from noisy data and treat the remained samples as unlabeled samples by disregarding their labels. Secondly, with the reformed data obtained from the first step which consist of labeled and unlabeled data, we train a prediction model by use of a semi-supervised learning method.

The most appealing feature of ASSL compared to other recent studies is that ASSL takes advantage of the memorization capability of DNNs that DNNs can easily over-fit train data. It is one of the most prominent issues for DNNs with noisy labeled samples since memorizing noisy samples would be expected to result in sub-optimal performance. There have been many trials to prevent the undesirable memorization ability of DNNs with noisy labeled samples through various regularization strategies such as imposing penalty term [44] or using large learning rate [35].

Instead of avoiding it, ASSL utilizes the memorization capability to filter out noisy labeled samples. When there are no noisy labels, it is known that over-fitted DNNs generalize well for test data [2]. This seemingly contradictory phenomenon can be understood that over-fitted DNNs can preserve certain smoothness as well [3]. That is, DNNs provide over-fitted but smooth predictive models. ASSL utilizes the property of DNNs to develop RAS for detecting noisy labeled samples.

RAS is motivated by the fact that a smooth over-fitted model changes its values much around noisy samples while it changes less around cleaned samples under the so-called *cluster assumption* [5] meaning that similar samples are likely to be of the same ground-truth label. Based on the idea of the adversarial training method [12, 27] which seeks a direction toward which a given predictive model changes most, RAS measures an instability score for each sample in the train data, which is proportional to how much a given smooth over-fitted predictive model changes toward the adversarial direction. Then, RAS regards samples with large instability scores as noisy labeled samples. Note that RAS does not require clean-annotated validation data set when it calculates the instability scores.

There are many approaches which require some additional information of noisy data. For example, there are some studies to assume that the true transition probability matrix between true and noisy labels is known [30], or to require small but clean-annotated train samples [42]. In contrast, ASSL does not require any such assumption and thus is easily applicable in practice.

By conducting various experiments on the well-known benchmark data sets including image data and sentence data, we demonstrate empirically that ASSL achieves the state-of-the-art performance with large margins for most considering cases. Particularly, ASSL is very powerful when the noise level is high. For heavy noise scenarios, ASSL improves the state-of-the-art test accuracies by more than 10% on image data sets and more than 25% on sentence data sets.

This paper is organized as follows. In Section ii., we briefly review related works, including methods for combating samples with noisy annotations and semi-supervised learning algorithms. Our proposed method, ASSL including RAS, is explained in Section iii., results of various experiments are given in Section iv. and conclusions follow in Section v..

II. RELATED WORKS

Noisy label problem Learning a classifier with noisy annotated data has been studied for several decades [1, 28, 48]. However, such methods are not easily applicable to DNNs because the excessive complexity of DNNs leads to memorizing all the train data including noisy labels, which hampers prediction power.

Recently, many noisy-annotation-robust deep learning approaches have been proposed to overcome this issue. Several methods focused on the noise transition matrix and correcting the objective function accordingly [10, 30]. [41] detected noisy samples by using feature maps of a given DNN model and assigned small weights in the loss function to reduce the undesirable effects caused by noise labels. [37] proposed a special loss function which is able to abstain on confusing samples to improve the classification performance of DNNs. [35] and [44] developed methods to estimate the ground-truth labels and network parameters simultaneously. There was also an attempt to employ the meta-learning algorithm to get a noise-tolerant classifier [24].

Semi-supervised learning Deep learning has suffered from collecting large amount of labeled data which requires both cost and time. Thus, it becomes important to develop semi-supervised methodologies that learn a model by using not only small labeled data but large available unlabeled data.

Various semi-supervised learning (SSL) methods have been proposed for deep learning. [31] used a specially designed auto-encoder to extract essential features for classification. Generative models including variational auto-encoder [18] or generative adversarial networks [11] were applied to SSL to enhance a classifier [17, 26, 33, 6, 22]. In addition, there were SSL methods to encourage a model to be robust to perturbation of data or parameter [32, 23, 27, 43], and to strengthen a model using ensemble of the past models [23, 36]. [39] and [4] adopted and generalized the idea of the MixUp algorithm [46] which trains a model with convex combinations of randomly selected sample pairs to shrink the model to be linear.

III. THE PROPOSED ASSL METHOD

Our goal is to learn a K-class classification model for data with noisy labels. Let $\mathscr{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ be a noisy train data set, where $\mathbf{x}_i$ is the i-th input sample and $y_i \in \{1, 2, \ldots, K\}$ is the corresponding (noisy) label. Also we denote the train data set with label k as $\mathscr{D}_k$, i.e. $\mathscr{D}_k = \{(\mathbf{x}, y) \in \mathscr{D} : y = k\}$. A discriminative network parametrized by θ is denoted by $f(\mathbf{x}; \theta)$, which maps an input $\mathbf{x}$ to a K-dimensional simplex with the softmax layer.

After explaining the motivation of ASSL, the three steps of ASSL will be described in detail in the consecutive subsections. Additionally, we propose an iteration method to enhance the original ASSL.

A. Motivation

We explain the main motivation about how ASSL utilizes a smooth over-fitted model. Consider as a toy example, a binary classification problem with 1-dimensional input data. Suppose that an input x is generated from a mixture of two uniform distributions, $x \sim 1/2Unif(-4, -1) + 1/2Unif(1, 4)$ and its ground-truth label is determined by the sign value, that is, $y = I(x > 0)$, where $I(\cdot)$ is the indicator function. We then corrupt some train samples by reversing their labels.

Suppose that there is a classifier $h(x; \eta) = p(y = 1|x; \eta)$ with parameter η which classifies all the train samples perfectly and is smooth enough (Figure 1(a)). As can be seen in Figure 1(a), $h(x; \eta)$ increases or decreases rapidly around noisy samples while it is stable around clean samples, which gives a clue how to distinguish pure and impure samples by looking at the derivative of $h(x; \eta)$ at each datum. The derivatives of noisy samples are large or small while those of clean-labeled samples are close to 0. On the other hand, suppose that $h(x; \eta)$ is either too much smooth or too much

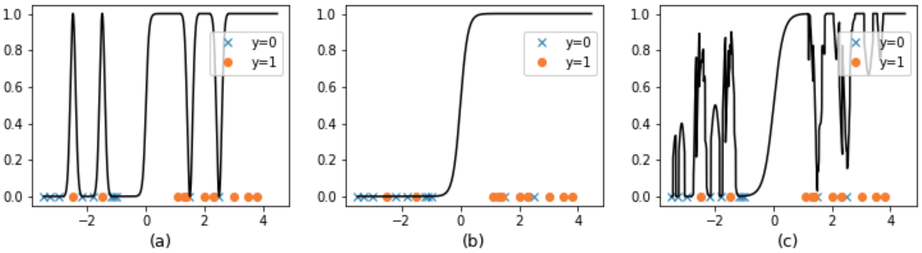


Fig. 1. Example of three types of classifiers for noisy labeled data.

complex, which are illustrated in Figure 1 (b) and (c) respectively. Then the derivative of $h(x;\eta)$ does not give any information about whether a given datum is noisy or clean. This is why we need a not only over-fitted but also smooth classifier for detecting noisy samples.

B. Step 1: Refinement with adversarial search

Over-fitting a model with smoothness Among many recent methods, we adopt MixUp [46] to obtain a smooth over-fitted model for its easiness to implement and computational efficiency.

For given two samples $(\mathbf{x}_1,\tilde{y}_1)$ and $(\mathbf{x}_2,\tilde{y}_2)$ from $\mathscr{D}$, where $\tilde{y}$ is the one-hot coded vector of y, MixUp calculates the following:

$$\begin{aligned}&L_\lambda^{\mathrm{MU}}(\theta,\mathbf{x}_1,\mathbf{x}_2,y_1,y_2)\\&=-\mathrm{CE}\left\{\mathrm{Mix}_\lambda(\tilde{y}_1,\tilde{y}_2),f(\mathrm{Mix}_\lambda(\mathbf{x}_1,\mathbf{x}_2);\theta)\right\},\end{aligned}\tag{1}$$

where $\mathrm{Mix}_\lambda(a,b)=\lambda a+(1-\lambda)b$ for $\lambda\in[0,1]$ and CE is the cross-entropy function. Then we minimize the following objective function with respective to θ:

$$\underset{(\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2)\sim\mathscr{D},\lambda\sim\mathrm{B}(\alpha,\alpha)}{\mathrm{E}}\left[L_\lambda^{\mathrm{MU}}(\theta,\mathbf{x}_1,\mathbf{x}_2,y_1,y_2)\right],$$

where $\mathrm{B}(\alpha,\alpha)$ is the beta distribution with tuning parameter α. [46] mentioned that MixUp with large α increases smoothness of the model but looses its memorization capability. Thus in this study we use a small α (0.25 in practice) to get a smooth but over-fitted model.

RAS method The first step of RAS is to define the instability score which intuitively measures how much $f(\mathbf{x};\widehat{\theta})$ changes its values around a given datum $\mathbf{x}$, where $\widehat{\theta}$ is the estimate obtained in the MixUp procedure. For this purpose, we utilize the idea of the adversarial direction of [12, 27] for a given datum $\mathbf{z}=(\mathbf{x},y)$, given as

$$\mathbf{r}^a(\mathbf{z},\varepsilon)=\underset{\mathbf{r};\|\mathbf{r}\|\le\varepsilon}{\mathrm{argmax}}D_{\mathrm{KL}}\left(\tilde{f}_y(\mathbf{x};\widehat{\theta})||\tilde{f}_y(\mathbf{x}+\mathbf{r};\widehat{\theta})\right),\tag{2}$$

where $\varepsilon>0$ is a maximum perturbation radius, D_{KL} is the KL divergence and $\tilde{f}_y(\mathbf{x};\theta)=(f_y(\mathbf{x};\theta),1-f_y(\mathbf{x};\theta))\in\mathbb{R}^2$. And we define the instability score as $M(\mathbf{z})=|f_y(\mathbf{x}+\mathbf{r}^a(\mathbf{z},\varepsilon);\widehat{\theta})-f_y(\mathbf{x};\widehat{\theta})|$.

The second step of RAS is to pick out samples based on the instability score $M(\mathbf{z})$ for $\mathbf{z}\in\mathscr{D}$ and to regard them as clean labeled samples. Let P_n be the empirical distribution of $\{M(\mathbf{z}),\mathbf{z}\in\mathscr{D}\}$ and let $\gamma\in(0,1)$ be a given constant. A naive approach is to pick samples whose quantiles of the instability scores with respect to P_n are less than γ and to regard them as cleaned samples. A better approach is to apply this procedure to each data set $\mathscr{D}_k$, which is used for experiments unless otherwise stated. That is, we pick cleaned samples from the data set of each class separately.

C. Step 2: Semi-supervised learning with the refined data

To obtain the final predictive model, we apply a SSL method to the refined data obtained in Step 1 in which all unpicked samples are treated as unlabeled data. In this study, we use MixMatch [4] which is known as one of the state-of-the-art methods in deep semi-supervised learning problems. We tried other recently proposed methods such as VAT [27], ICT [39] and UDA [43], and concluded that MixMatch is powerful, efficient and easily applicable to various data sets. For example UDA is also powerful but requires a task-specific data augmentation technique which needs huge computational resources, being hard to extend to other tasks.

Note that ASSL does not remove any samples completely from the train data while many other approaches for noisy label problems remove the samples, e.g. [24, 37]. Since only labels are contaminated, it is beneficial to use them as unlabeled data instead of removing them.

D. Iterative training

We propose an iterative training scheme of ASSL for enhancement of the prediction power. In first iteration, we apply our proposed ASSL method with the train data set $\mathscr{D}$ to have the estimated parameter of the prediction model $\widehat{\theta}_1^{\mathrm{ASSL}}$.

In second iteration, we calculate the predicted labels of all the train data with the estimated model in the first iteration, given as

$$\widehat{y}_i = \underset{1,\ldots,K}{\operatorname{argmax}} f_k(\mathbf{x}_i; \widehat{\theta}_1^{\text{ASSL}}), \quad i = 1, \ldots, n. \tag{3}$$

We then define the new train data set $\tilde{D}$ by replacing all the labels of the train data with the predicted labels obtained in (3), that is, $\tilde{D} = \{(\mathbf{x}_i, \widehat{y}_i)\}_{i=1}^{n}$. And we repeat all the two steps in ASSL with the new train data set $\tilde{D}$ leading to obtain the second estimated parameter denoted by $\widehat{\theta}_2^{\text{ASSL}}$. We repeat this iteration until the performance is saturated.

In experimental analysis we conduct three iterations of ASSL to estimate the prediction model. We observe that iterating more than three times does not give significant improvements.

IV. Experimental analysis

We carry out extensive experiments with ASSL including performance tests and ablation studies by analyzing both image and sequential data sets. We consider CIFAR-10 and CIFAR-100 [20] for image data analysis, and Laptop, Restaurant and Movie [29] for sequential data analysis. Laptop and Restaurant data sets are collected from SemEval-2016[1]. All the three sequential data sets consist of review sentences and their corresponding polarity (*positive* or *negative*). All experiments are implemented by using the PyTorch framework. As mentioned in Section D., we iterate ASSL three times.

For CIFAR-10 and CIFAR-100, we use WideResNet28-2 ([45], 1.47M params) and WideResNet28-10 ([45], 36.54M params) respectively as the backbone network. Note that other studies analyze CIFAR-10 with ResNet-34 [13], PreActResNet-32 [14] and WideResNet28-10 [45] , which are much bigger than WideResNet28-2. As for sequential data sets, we use a pre-trained BERT [8] as a encoder. After encoding them to 300-dimensional vectors, we use 4-hidden layered DNN (300-300-300-150-150-2) equipped with the ReLU activation function and the batch normalization [15]. We do not learn the BERT during the training step, i.e. freeze the parameters of the BERT.

As other studies did, we use 10% randomly selected clean-annotated data from the train data as validation data in order to find the optimal hyperparameters in RAS and MixMatch.

A. *Data description*

In this section we describe the label contamination strategies for each data set.

CIFAR-10 & 100 According to the common settings of other recent papers [35, 47, 24, 37, 44], we consider two settings for giving label noises: symmetric and asymmetric. In the symmetric noise setting, for each sample in the train data set its label is contaminated with probability r to a random label generated from the uniform distribution on 1 to K ($K = 10$ for CIFAR-10 an $K = 100$ for CIFAR-100). In the asymmetric noise setting for CIFAR-10, with probability r a noisy label is generated by one of the following mappings: *truck→automobile, bird→airplane, deer→horse* and *cat↔dog*. As for CIFAR-100, a noisy label is generated by flipping a given label to the next label with probability r according to the transition chain: *class1→class2→* $\cdots$ *→class100→class1*.

Laptop, Restaurant and Movie For all the three data sets, we only consider the symmetric noisy annotation strategy as [40] did. The strategy applied is slightly different from the above method: for a given datum, with probability r we do not generate a random label from the uniform distribution but reverse its label, e.g. *positive→negative* or *negative→positive*.

B. *Prediction performance analysis*

CIFAR-10 & 100 The results are summarized in Table 1 and 2. ASSL achieves the highest test accuracies in most cases compared to other existing methods. Note again that ASSL utilizes a much smaller network than the architectures used in other studies for CIFAR-10. In particular, ASSL improves the existed state-of-the-art test accuracies with large margins when the noise level is high. The margins between ASSL and other methods are at least 3% for both the symmetric setting with $r \geq 0.7$ and the asymmetric setting with $r = 0.5$. Moreover, for the extreme noise case ($r = 0.9$ for CIFAR-10 and $r = 0.8$ for CIFAR-100), ASSL improves the existed state-of-the-art accuracies by more than 10%.

We find that the iterative training in Section D. strengthens the performance except that r is small. For a small r, applying the iterative training worsens the results but the extent of degradation is not significant. On the other hand, the iterative training leads to impressive improvements for the heavy noise settings. It even records more than 13% margin for the symmetric noise setting with $r = 0.9$. for CIFAR-10 and $r = 0.8$ for CIFAR-100. This is because that the quality of the relabeled train data is improved much when the noise rate is high.

Laptop, Restaurant and Movie Table 3-5 present the accuracies of ASSL and other methods over the three sequential data sets. The study of [40] is the first trial to learn a model with noisy labels for sequential-level data to the best of our knowledge, thus we only present the performance results conducted in [40] as baselines. Since the architectures used by [40] and our study are different, we also list the baseline results achieved only with cross-entropy loss for fair comparison.

While other methods including NetAB get worse drastically as the noise level increases, ASSL is not much affected

[1] http://alt.qcri.org/semeval2016/task5/

Noise type	Symmetric					Asymmetric				
Method	r=0.1	r=0.3	r=0.5	r=0.7	r=0.9	r=0.1	r=0.2	r=0.3	r=0.4	r=0.5
Cross Entropy [35]	91.0	88.4	85.0	78.4	41.1	91.8	90.8	90.0	87.1	77.3
CNN-CRF [38]	-	-	-	-	-	92.0	91.5	90.7	89.5	84.0
Joint Optim. [35]	92.7	91.4	89.6	85.9	58.0	93.2	92.7	92.4	91.5	84.6
PENCIL [44]	93.26	92.09	90.29	87.10	61.21	93.00	92.43	91.84	91.01	80.51
MLNT [24]	93.24	92.50	90.65	87.11	59.09	93.61	93.25	92.82	92.30	82.09
DAC [37]	-	-	-	-	-	94.23	93.20	92.07	89.88	-
ASSL (1st iter.)	**94.96**	92.65	91.38	85.31	57.60	**94.52**	93.61	91.33	88.04	81.48
ASSL (2nd iter.)	94.00	93.33	92.70	90.43	70.64	93.95	93.72	92.43	91.91	85.54
ASSL (3rd iter.)	93.67	**93.34**	**92.77**	**90.90**	**71.49**	94.34	**94.12**	**93.01**	**92.51**	**85.66**

Table 1. Test accuracies (%) on CIFAR-10 for various methods trained with *symmetric* and *asymmetric* label noises. We ran three implementations and reported the mean accuracy.

Noise type	*Symmetric*				*Asymmetric*			
Method	r=0.2	r=0.4	r=0.6	r=0.8	r=0.1	r=0.2	r=0.3	r=0.4
Cross Entropy [47]	58.72	48.20	37.41	18.10	66.54	59.20	51.40	42.74
$\mathscr{L}_q$ [47]	66.81	61.77	53.16	29.16	68.36	66.59	61.45	47.22
Trunc $\mathscr{L}_q$ [47]	67.61	62.64	54.04	29.60	68.86	66.59	61.87	47.66
PENCIL [44]	73.86	69.12	57.79	fail	75.93	**74.70**	72.52	63.61
DAC [37]	**75.75**	68.20	59.44	34.06	75.59	73.22	71.38	65.34
ASSL (1st iter.)	74.33	66.32	55.70	31.39	**77.28**	72.64	67.93	56.50
ASSL (2nd iter.)	74.98	69.11	65.57	43.28	77.23	73.42	73.34	64.83
ASSL (3rd iter.)	74.35	**69.54**	**65.60**	**44.88**	76.41	73.48	**73.59**	**65.94**

Table 2. Test accuracies (%) on CIFAR-100 for various methods trained with *symmetric* and *asymmetric* label noises. We ran three implementations and reported the mean accuracy.

Method	r=0.1	r=0.2	r=0.3	r=0.4	r=0.5
CNN [16]	75.0	74.0	66.0	64.0	60.0
NetAB [40]	76.5	75.0	68.5	65.5	60.5
Baseline	78.45	74.69	69.80	63.22	53.99
ASSL (1st iter.)	84.36	84.57	83.90	82.70	84.58
ASSL (2nd iter.)	**85.20**	**86.47**	84.99	83.72	84.75
ASSL (3rd iter.)	85.20	86.30	**85.20**	**84.80**	**86.05**

Table 3. Test accuracies (%) on Laptop dataset for various methods trained with *symmetric* label noises. We ran five implementations and reported the mean accuracy.

Method	r=0.1	r=0.2	r=0.3	r=0.4	r=0.5
CNN [16]	77.0	72.0	70.0	69.0	52.0
NetAB [40]	79.0	76.0	72.0	69.5	56.0
Baseline	83.59	81.17	75.52	69.12	57.39
ASSL (1st iter.)	**90.33**	86.81	87.47	**88.35**	86.59
ASSL (2nd iter.)	88.80	**88.39**	**87.88**	86.92	**88.38**
ASSL (3rd iter.)	88.10	87.90	87.50	87.90	88.38

Table 4. Test accuracies (%) on Restaurant dataset for various methods trained with *symmetric* label noises. We ran five implementations and reported the mean accuracy.

by the noise level. For heavy noise cases, ASSL achieves substantial improvements with large margins more than 25% compared to NetAB and baseline. It is also notable that the accuracy of ASSL with high noise level ($r = 0.5$) is competitive to those of ASSL with low noise level, which reassures that ASSL is powerful to analyzing highly noisy labeled data set.

Method	r=0.1	r=0.2	r=0.3	r=0.4	r=0.5
CNN [16]	73.0	70.0	66.0	59.0	49.0
NetAB [40]	75.5	75.0	70.5	69.0	50.0
Baseline	76.11	77.21	71.79	65.64	52.33
ASSL (1st iter.)	**80.60**	79.93	77.95	80.45	78.88
ASSL (2nd iter.)	80.40	79.12	**79.90**	80.20	79.12
ASSL (3rd iter.)	78.03	**80.50**	79.40	**81.20**	**80.03**

Table 5. Test accuracies (%) on Movie dataset for various methods trained with *symmetric* label noises. We ran five implementations and reported the mean accuracy.

C. *RAS analysis*

In this subsection, we investigate various aspects of RAS. Recall that RAS can apply the instability scores to either $\mathscr{D}$ or each $\mathscr{D}_k$ separately. We compare these two strategies by visualizing the distribution of the number of labeled samples at each class in the refined data. We consider the first iteration of ASSL over CIFAR-100 with the symmetric noise setting. Figure 2 summarizes the results when $\gamma = 0.2$ and $\gamma = 0.5$. Results for other γ values are in the supplementary material. We can observe that the distributions of the number of samples decided to be cleaned labeled samples by applying RAS to individual $\mathscr{D}_k$ have relatively smaller variances than those with $\mathscr{D}$. The results imply that the distribution of the instability score is quite different for each class and balancing the numbers of cleaned labeled samples for each class is beneficial.

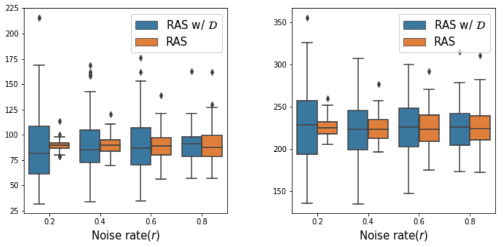


Fig. 2. Boxplots of the numbers of labeled samples at each class of the refined data by RAS applied to $\mathscr{D}$ (RAS w/ $\mathscr{D}$) and RAS applied to individual $\mathscr{D}_k$ with (**Left**) $\gamma = 0.2$ and (**Right**) $\gamma = 0.5$.

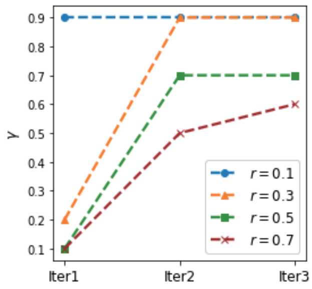


Fig. 3. The optimal γ values for various noise levels.

D. *Ablation study*

About the quantile point γ in RAS We investigate the relation between the quantile point γ and the noise level r. Note that the larger the γ value is the more the samples are regarded as clean labeled samples. We consider CIFAR-10 with the symmetric noise setting. Figure 3 draws the values of the optimal γ for various values of r at each iteration of the iterative training. It is observed that when there are not many noisy labels in the train data, it is beneficial to use a large value of γ and vice versa. Also, it can be seen that more labeled samples in the refined data are helpful as the iterative training proceeds, which suggests that the iterative training procedure keeps purifying the train data.

When validation data are not available, we can resort to prior knowledge about the noise level to choose γ. So, it is interesting to see that how sensitive the accuracies of ASSL according to the choice of γ. Figure 4 draws the accuracies of the ASSL after the first iteration for CIFAR-10 with the symmetric noise settings. Unless the noise level is high, the results are not too sensitive to the choice of γ, which recommends that a relatively small value of γ is desirable when no validation data are given.

Importance of SSL framework We carry out one more experiment to stress the importance of using seemingly noisy labeled samples as unlabeled samples by comparing SSL method and SL method. In Step 3 of ASSL, we train prediction models by two methods: 1) MixMatch (ASSL) and 2) MixMatch only with labeled samples in the refined data (ASSL-SL). Table 6 lists the results over CIFAR-10 for the symmetric noise setting with $r = 0.5$ and CIFAR-100 with the symmetric noise setting with $r = 0.4$. The results confirm the importance of using a SSL method in ASSL.

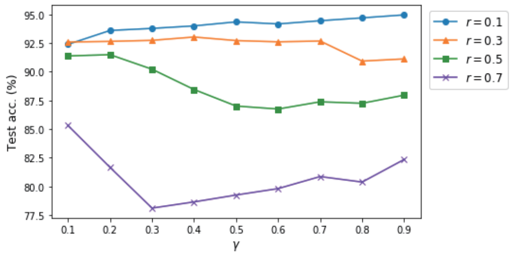


Fig. 4. Plot of test accuracies for various γ values.

	CIFAR-10			
Method	γ=0.1	γ=0.3	γ=0.5	γ=0.7
ASSL	**91.38**	**90.22**	**87.65**	**87.38**
ASSL-SL	84.40	85.50	86.48	87.01
	CIFAR-100			
Method	γ=0.1	γ=0.3	γ=0.5	γ=0.7
ASSL	**59.62**	**66.18**	**66.32**	**63.6**
ASSL-SL	44.81	59.45	60.17	61.01

Table 6. Test accuracies (%) of the original ASSL (ASSL) and ASSL with only labeled samples in the refined data (ASSL-SL).

V. Conclusion

In this paper, we proposed a new and novel learning framework, called ASSL, to learn a prediction model when the train data have some noisy annotations. We found that ASSL exhibited significantly improved performance compared to other recent approaches in most experiments and was especially superior on highly noisy-labeled cases with large margins.

An interesting issue is to apply ASSL without validation data. Note that validation data are used to select the γ in RAS and to learn a prediction model by SSL. We have seen that the choice of γ is not sensitive to the final results unless it is too large. For SSL, we may use a part of refined data as validation data. We leave this problem as a future work.

References

[1] Dana Angluin and Philip Laird. Learning from noisy examples. *Machine Learningyy*, 2(4):343–370, 1988.

[2] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

[3] Mikhail Belkin, Alexander Rakhlin, and Alexandre B Tsybakov. Does data interpolation contradict statistical optimality? *arXiv preprint arXiv:1806.09471*, 2018.

[4] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. *arXiv preprint arXiv:1905.02249*, 2019.

[5] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

[6] Zihang Dai, Zhilin Yang, Fan Yang, William W Cohen, and Ruslan R Salakhutdinov. Good semi-supervised learning that requires a bad gan. In *Advances in Neural Information Processing Systems*, pages 6513–6523, 2017.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[9] Rob Fergus, Li Fei-Fei, Pietro Perona, and Andrew Zisserman. Learning object categories from internet image searches. *Proceedings of the IEEE*, 98(8):1453–1466, 2010.

[10] Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. 2016.

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[12] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[16] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics.

[17] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.

[18] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[19] Jonathan Krause, Benjamin Sapp, Andrew Howard, Howard Zhou, Alexander Toshev, Tom Duerig, James Philbin, and Li Fei-Fei. The unreasonable effectiveness of noisy data for fine-grained recognition. In *European Conference on Computer Vision*, pages 301–320. Springer, 2016.

[20] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[22] Abhishek Kumar, Prasanna Sattigeri, and Tom Fletcher. Semi-supervised learning with gans: manifold invariance with improved inference. In *Advances in Neural Information Processing Systems*, pages 5534–5544, 2017.

[23] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

[24] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S Kankanhalli. Learning to learn from noisy labeled data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5051–5059, 2019.

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[26] Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. Auxiliary deep generative models. *arXiv preprint arXiv:1602.05473*, 2016.

[27] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

[28] David F Nettleton, Albert Orriols-Puig, and Albert Fornells. A study of the effect of different types of noise on the precision of supervised learning techniques. *Artificial intelligence review*, 33(4):275–306, 2010.

[29] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *arXiv preprint cs/0506075*, 2005.

[30] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1944–1952, 2017.

[31] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. In *Advances in Neural Information Processing Systems*, pages 3546–3554, 2015.

[32] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 1163–1171, 2016.

[33] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

[34] Florian Schroff, Antonio Criminisi, and Andrew Zisserman. Harvesting image databases from the web. *IEEE transactions on pattern analysis and machine intelligence*, 33(4):754–766, 2010.

[35] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5552–5560, 2018.

[36] 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*, pages 1195–1204, 2017.

[37] Sunil Thulasidasan, Tanmoy Bhattacharya, Jeff Bilmes, Gopinath Chennupati, and Jamal Mohd-Yusof. Combating label noise in deep learning using abstention. *arXiv preprint arXiv:1905.10964*, 2019.

[38] Arash Vahdat. Toward robustness against label noise in training deep discriminative neural networks. In *Advances in Neural Information Processing Systems*, pages 5596–5605, 2017.

[39] Vikas Verma, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, IJCAI'19, pages 3635–3641. AAAI Press, 2019.

[40] Haihui Wang, Bing Liu, Chaozhuo Li, Yan Yang, and Tianrui Li. Learning with noisy labels for sentence-level sentiment classification. In *EMNLP/IJCNLP*, 2019.

[41] Yisen Wang, Weiyang Liu, Xingjun Ma, James Bailey, Hongyuan Zha, Le Song, and Shu-Tao Xia. Iterative learning with open-set noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8688–8696, 2018.

[42] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2691–2699, 2015.

[43] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation. *arXiv preprint arXiv:1904.12848*, 2019.

[44] Kun Yi and Jianxin Wu. Probabilistic end-to-end noise correction for learning with noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7017–7025, 2019.

[45] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[46] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[47] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in neural information processing systems*, pages 8778–8788, 2018.

[48] Xingquan Zhu and Xindong Wu. Class noise vs. attribute noise: A quantitative study. *Artificial intelligence review*, 22(3):177–210, 2004.

City-Scale Visual Place Recognition with Deep Local Features Based on Multi-Scale Ordered VLAD Pooling

Duc Canh Le[1] and Chan-Hyun Youn[1]

[1] Department of Electrical Engineering, KAIST, Daejeon, Korea {canhld,chyoun}@ kaist.ac.kr

Abstract

Visual place recognition is the task of recognizing a place depicted in an image based on its pure visual appearance without metadata. In visual place recognition, the challenges lie upon not only the changes in lighting conditions, camera viewpoint, and scale, but also the characteristic of scene level images and the distinct features of the area. To resolve these challenges, one must consider both the local discriminativeness and the global semantic context of images. On the other hand, the diversity of the datasets is also particularly important to develop more general models and advance the progress of the field. In this paper, we present a fully-automated system for place recognition at a city-scale based on content-based image retrieval. Our main contributions to the community lie in three aspects. Firstly, we take a comprehensive analysis of visual place recognition and sketch out the unique challenges of the task compared to general image retrieval tasks. Next, we propose yet a simple pooling approach on top of convolutional neural network activations to embed the spatial information into the image representation vector. Finally, we introduce new datasets for place recognition, which are particularly essential for application-based research. Furthermore, throughout extensive experiments, various issues in both image retrieval and place recognition are analyzed and discussed to give some insights for improving the performance of retrieval models in reality.

Keywords— *visual place recognition, information retrieval, features pooling, image representation, datasets*

I. Introduction

Visual place recognition has received a significant amount of attention in the past few years both in computer vision and robotics communities , motivated by applications in autonomous driving, augmented reality and simultaneous localization and mapping (SLAM) [1, 21]. Generally, there are two major approaches for visual place recognition: image retrieval-based and large-scale 3D model-based.

In the large-scale 3D model approach, the scenes are represented via 3D models with image descriptors attached to 3D points and the localization task is cast as 2D-to-3D image registration where the full 6 DOF camera poses are recovered from the models [23]. The method has the advantage of high accuracy, nevertheless, building and maintaining a large-scale 3D model is extremely expensive. On the other hand, the image retrieval based methods approximate the location of a query image with locations of similar images in a large-scale geo-tagged database. Compared to 3D-based methods, maintaining a database of geo-tagged images is much easier, therefore, the approach has the advantage of scalability. Moreover, it has been shown in [23] that retrieval-based method can also recover both position and full camera poses of the query image if sufficient database images are correctly retrieved.

In general image retrieval, the goal is to find the image with the most similar visual appearance to the instance depicted in the query. Modern retrieval systems index an image to a compact vector, which captures both the local and the semantic visual appearance of the image. By measuring the distance between these vectors, we can compute the similarity between the corresponding images. The fundamental issue in image retrieval is how to create a canonical form that efficiently unify similar images and distinguish those are different.

Visual place recognition exposes unique challenges besides to those existing in general image retrieval. Nowadays, challenges of visual place recognition come from not only the visual differences between query and database images but also from the overlapping between irrelevant images, and the different appearance of the same place among images [11]. For example, in city scenes, common dynamic objects such as cars, humans, bikes, trains appear in most of the images; or a picture of same place may shift depending on the camera model, the viewpoint, the time in the day or the season it was taken.

To address these issues, we propose a pooling scheme on top of convoluional neural network (CNN) activations that produce an effective vector-form of scene images. Our method takes advantage of local features in CNN, pool them with a statistical model to achieve the canonical form, and finally exploit the multi-scale transform to retain the spatial details. While the final vector may include residual information, the experiment show that it effectively eliminates the unique issues in visual place recognition.

In addition, one of the challenges in the research of visual place recognition in the early days was how to collect enough geo-tagged images to build the database. In recent years, map providers such as Google or Naver introduce the street-view feature in their map platform, which allows users to view the street-level spherical panorama image in most urban areas in the world. Subsequently, several city-scale databases have been constructed [22, 1, 18]. Nevertheless, our significant concern when starting this research is whether or not the characteristics of cities affects the recognition methods. Encouraged by this question in mind, we generate the dataset of our city, Daejeon, with database images from Google Streetview and query images were taken by phones. We later show that our concern does make sense, as the state-of-the-art methods on existing datasets perform worse in our dataset.

The paper is organized as follow: In Section ii. we describe in detail the challenges of visual place recognition and discuss previous works. In Section iii., we present our pooling scheme. In Section iv., we first describe our newly collected dataset and then show demonstrate the benefits of proposed method over prior works. Finally, the conclusion and discussion on the future of visual place recognition are given in Section v..

II. Background and Related Works

Content-based image retrieval (CBIR) has been a core problem in the multimedia field over two decades. Modern CBIR systems usually consist of two-stage: offline stage and online stage. In the offline stage, a visual database is constructed by crawling images from various sources to create the plaint-image pool, and indexing images to database for efficient searching. In the online stage, the query image is given and similar images in the database are retrieved by scoring the images in the database and optionally re-rank the top images with the highest scores by geometric verification [5].

Nowadays, the challenge of CBIR lies in developing an efficient representation that is robust to different appearances of an image due to occlusions or changes in illumination, view-point, and scales. To achieve the invariant, local features is extracted from images. A local features describe a very small patch of the image (e.g. 16×16 pixels) and is robust to change in illumination and scale. Notable local features are SIFT [12], SURF [3], BRIEF [4]; and recently off-the-shell CNN activations shows outstanding performance when used as local features [17].

While local features is good for matching images, it is not suitable for image retrieval due to extra cost of point-to-point matching. To solve this problem, methods are developed to aggregate local features into a compact global feature. Most of the successful feature aggregation model are statistical-based, namely bag of visual words (BoVW) [19], Fisher vector (FV) [14] and vector of locally aggregated descriptors (VLAD) [9]. These models capture the distribution of local features over a codebook and use it as the global representation of images.

In recent years, end-to-end deep learning model introduces simple aggregation strategy on CNN feature and achieve decent global features by jointly learn both extractions and aggregation. [2] trained the CNN with classification loss and used the pooled vector at the last fully-connected layer as the global representation of the images. The authors of [16, 20, 7] replaced the fully-connected layer with pooling layer and trained the network with the triplet-ranking loss to produce a global image representation that is suitable for image retrieval. Remarkably, [1] proposed the NetVLAD layer, which pools the features in VLAD manner and ready to plug into any CNN architecture. For visual place recognition, the authors showed that the whole network can be weakly supervised trained with triplet ranking loss and noisy data with only GPS geotagged.

When applied to visual place recognition, CBIR reveals unique challenges due to the characteristic of urban environment. The issue come from not only the visual differences between query and database images but also from the overlapping between irrelevant images, and the different appearance of the same place among images [10]. For example, in city scenes, common dynamic objects such as cars, humans, bikes, trains appear in most of the images; or a picture of same place may shift depending on the camera model, the viewpoint, the time in the day or the season it was taken. Many efforts have been made in the recent decade to solve these challenges with attentive features model, on which the features extractor is trained to detect key-points at the salient areas of the images, i.e. static objects such as walls or buildings. However, in [11, 22, 10] and in this paper, we show that the attentive model itself is not strong enough for better place recognition. The underlying reasons lie in two issues: the repetitive structures , and the reflection phenomenon.

A. *Repetitive Structures*

Repetitive structures may refer to the common patterns that appear frequently in many different geographical places and are easily matched to other instances of the same type. The examples of repetitive structures in the city are generic windows, building fences or trees.There are two key reasons why the repetitive pattern is problematic in the retrieval algorithm. First, the features from these pat-

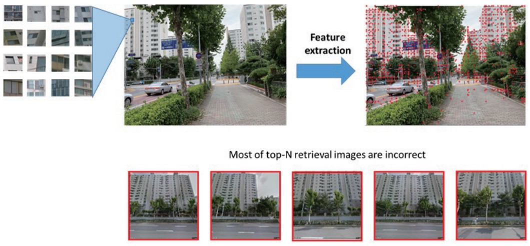


Figure 1. Typical building facades in our database that create difficulty to distinguish buildings; many features extracted from the image belong to repetitive patterns, thus, the retrieval system return many false-positive images

terns dominate in the image and therefore, degrade the contributions of other important features. For example, in the BoVW model, the final representation vector is actually the histogram of features over the pre-trained codebook. Then if two images are completely different but they are dominant by the same repetitive structure, the BoVW model is likely to give them a relatively high similarity score. The same issue can be found in VLAD model, as they are both pure statistical-based. The second reasons are the representation of images in the database usually ignores the spatial relations between features. This can be improved by post-processing methods such as spatial verification with RANSAC [5] with the penalty on computational complexity. However, post-processing is not always effective because sometimes the correct images cannot even reach to spatial verification step.

There are efforts to reduce the negative impact of repetitive structures on retrieval algorithm [10, 11, 22, 19]. Especially, the author of BoVW [19] proposed to use term frequency – inverted documents frequency (TF-IDF) weighting scheme on the visual words of the codebook, in which the more frequently visual words are down-weighted. Formally, suppose there is a codebook of V visual words, then each image is represented by a vector:

$$v_d = (t_1, t_2, \ldots, t_V)^T \tag{1}$$

of weighted visual word frequencies with components:

$$t_i = \frac{n_{id}}{n_d} \times \log \frac{N}{N_i} \tag{2}$$

Where n_{id} is the number of occurrences of visual words i in image d, n_d is the total number of visual words in the image d, N_i is the number of images containing term i, and N is the number of the whole database. TF-IDF degrades the impact of repetitive patterns with the assumption that they are common and dominated in the whole database. However, this assumption does not always hold owing to two factors: (1) repetitive patterns may appear in many images but not major of the database, and (2) the features from these patterns are not necessarily identical. Therefore, [49] and [2] suggested explicitly detecting and down-weighting the features from repetitive structures (that called burstiness features). Recently, [10, 11] introduces a method to learn the good features for visual place recognition and the model implicitly learns to down weight the features from the repetitive structures.

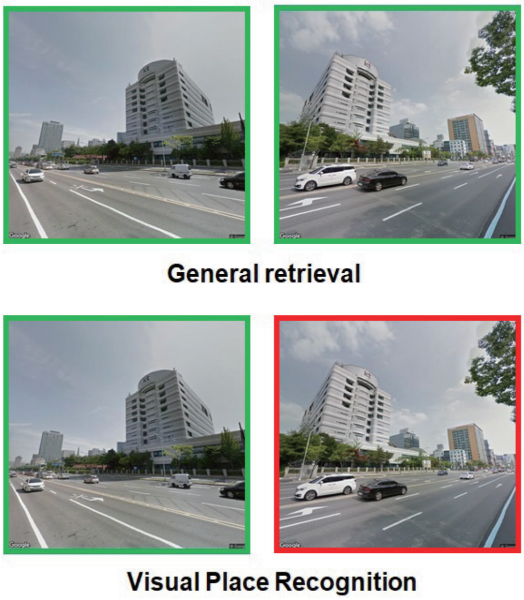


Figure 2. Two image of the same building from different perspectives; in general retrieval, they are considered similar, but in visual place recognition, they are false positives

All of the prior works are statistical-inspired and aim to break the domination of burstiness features. On the other hand, our approach attempts to embed the spatial information among features, which includes the burstiness features, to the final representation. We later show that our method and prior methods are complementary and one can combine both to further improve the efficiency of place recognition algorithms.

B. *Reflection Phenomenon*

Reflection phenomenon usually happens when images contain a large building. In some general image retrieval tasks, e.g. product retrieval, landmark recognition, and visual search, etc., the global context information is not an essential concern. For example, if one queries the visual search engine with an image of Eiffel tower, then all retrieved images should merely contain the tower no matter the distance or the perspective of these images to the tower. Similarly, if one wants to search the information of a product from an image, he only cares about the product itself but not the semantic context of the image. However, in visual place recognition, the ultimate goal is to detect the location of the query image, therefore, the semantic context information of images does make sense and should be thoughtfully concerned.

The reflection phenomenon is very unique to visual place recognition compared to other issues. Similar to the repetitive phenomenon, it can also be improved with spatial verification with the penalty on computational cost and processing time. Nevertheless, in spatial verification, we verify the relation among all features, which is quite not necessary in this case. Indeed, all we need is a semantic representation of images to degrade the negative impact of both repetitive structures and reflection phenomena. In the next Section, we present yet a simple technique to generate a semantic representation of an image with multi-scales ordered VLAD pooling of the image deep local features.

III. Methodology

A. Pooling Features on the Top of CNN Activations

In recent years, CNN has shown breakthrough performance on many computer vision tasks compared to traditional models. Many research describes CNN as a "brain-inspired" computing model, however, in computer vision, CNN yet still follows the conventional pipeline of computer vision and pattern recognition: extracting image features and learning the patterns on top of these features. The underlying idea of CNN compared to the traditional machine learning approach is: in traditional machine learning, the features are extracted manually and the classification model is learnable, but in CNN, both the features and the classification model are immensely learnable. Therefore, the last fully connected layer can be replaced with other machine learning models to perform tasks other than image classification or improve the accuracy of the model [17].

Our paper focuses on Streetview images, where the human-made constructions and their locations are significant to recognize the scene. In scene level images, the image usually consists of many objects on a background. The background occupies a large portion of the image but usually has a uniform appearance and does not tell us much about the uniqueness of the image. On the other hand, foreground objects, even engage relatively few pixels in the image, contains far more useful information. In scene-level images, there is no priority among objects, and the spatial relations among objects are essential for the semantically understanding the image. For the visual place recognition task, we expect the retrieved images should not only contain the same buildings but also have a similar viewpoint with the query image. To adding the spatial information to the image, we exploit the ordered-pooling approach, where the images are orderly compared at multi-scales. We refer our method as Spatial Pyramid VLAD Pooling (SPVP).

B. Spatial Pyramid VLAD Pooling

Our work also relies on reusing pre-trained CNN features as the off-the-shell local features, but instead of pooling them in a trivial manner as previous works, we aim to develop a more sophisticated aggregating method. At first glance, one can use BoVW [19] or VLAD [9], which has been already thoroughly studied and shown sufficient performance with hand-crafted local features. However, these statistical pooling methods suffer from missing spatial information of the images. Inspired by the Spatial Pyramid Pooling Network (SPP-net) [7] and R-MAC [20] which extracts the image features at a single scale but pool them over regions of increasing scales, we propose to pool the CNN activations in VLAD manner at multi-scales. The detail of our framework is shown in Fig. 3.

Our framework is built on top of Deep Local Features (DELF) [13] with the backbone network is Resnet50 [8] following by an attentive layer. In DELF, the $2048-D$ activations are taken at the output of the $conv4\_x$ block of Resnet50 [8] and fed to the attentive layer to select semantically meaningful features for place recognition. The attentive model can be trained implicitly with classification loss. The selected features are L2-normalized, and then their dimension is reduced to 40 with principal component analysis, and finally L2-normalized again. We use both the pre-trained FCN model on the Google Landmark v2 dataset and the PCA model from the authors1 to extract the features from the images and from there develop our representation technique. In our framework, the default image size is 640×640 and the DELF outputs a sparse $640 \times 640 \times 40$ features map. Our representation has total of three scales, corresponding to activations of original 640×640 image and 320×320 and 160×160 patches. Note that we only pass the image through the extractor network once and then build the pyramid on the final activations.

Next, we need to pool the activations of the image and its patches to summarize the representation by a single feature vector of reasonable dimension. For this, we adopt VLAD [9] instead of mean pooling [16] or max-pooling [20]. We randomly select 10 million features to build a codebook of 256 visual words with k-mean, and at each level, we pool the features of each patch in VLAD manner with respect to the codebook. Each DELF feature is $40-D$, results in a $10240-D$ VLAD vector for each patch, which is too high. Therefore, we train the PCA on top of all VLAD vectors to reduce the dimensionality of these vectors to either 256, 512, 1024 or 2048.

Intuitively, one can describe our methods as multi-patches pooling in the sense that we divide the image into multiple patches and simultaneously compare image-to-image and patch-to-patch. This strategy indeed works in visual place recognition by following justifications: (1) the patch-to-patch comparison can preserve the spatial relations among objects in the images and ensure that if two

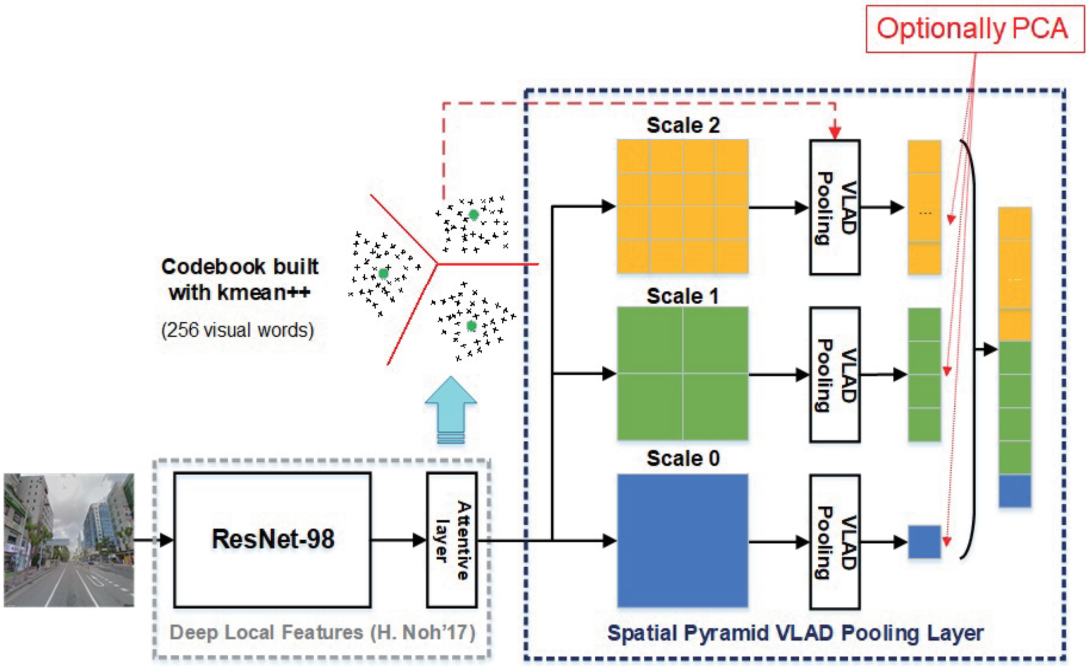


Figure 3. The overall framework of our Spatial Pyramid VLAD Pooling approach for visual place recognition; in sort, we pool the CNN features with VLAD in an ordered manner

images are matched by patch-to-patch comparison, they are semantically similar; and (2) even if in the database, there are no image with perfectly view-point that can be matched with patch-to-patch, the global image-to-image comparison yet still works and provide us the most similar images in the database. In other words, our pooling method always guarantees the best match images are retrieved for a given query.

Relation to previous studies: Our approach is inspired by SPP-net [7] and R-MAC [20], nevertheless, these approaches are not specialized for place recognition and their pooling operator are trivial (max-pooling). On the other hand, our pooling method is designed particularly for place recognition and we pool the CNN features with VLAD. The most related method to ours is the one of Gong et. al. [6], which proposes to pool the CNN features with VLAD at multi-scale in an orderless manner. We show the difference between ordered and orderless pooling in Figure 3-3. In short, the orderless pooling summarizes the patch-level VLAD vector while the ordered pooling concatenates them. Nevertheless, we argue that their work is not applied well to visual place recognition, because orderless pooling is good to handle object at different scales but do not preserve the spatial information of the image. The second key difference is in their work, the representation vector at the first scale is corresponding to $4096 - D$ CNN activation for the entire 256×256 image, and the VLAD is only applied at higher scales of the pyramid. Meanwhile, we use VLAD on all scales.

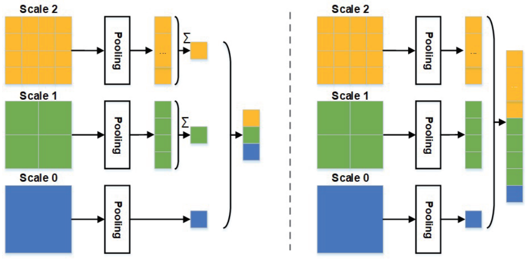


Figure 4. Our ordered pooling (Left) vs Gong. et. al. [6] orderless pooling (Right); even achieving more compact vector size, ordredless pooling suffers from missing of spatial information

IV. Experiments

A. Dataset and Evaluation Metric

After the crawling process, there is a total of 86.885 locations are captured, results in a total of 695.080 street-level images with GPS tags. We use these images to create the experiment datasets with 263.064 images, covers $40km^2$ of the city's urban area. We split those images into two sets: the database with 259.064 images and the query set with 4.000 images, equally distributed over 500 locations in the area. The visualized locations the dataset are

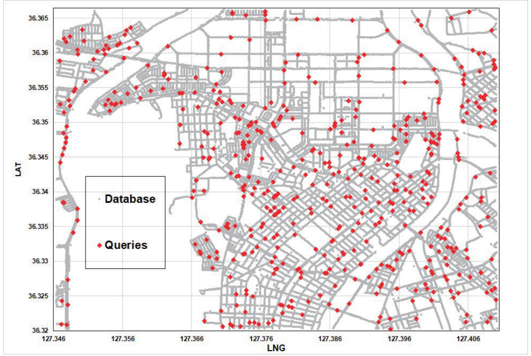


Figure 5. Locations of the database (blue) and query (red) images in the dataset; the dataset includes 259.064 database images and the query set with 4.000 images, distributed over 500 locations in the area

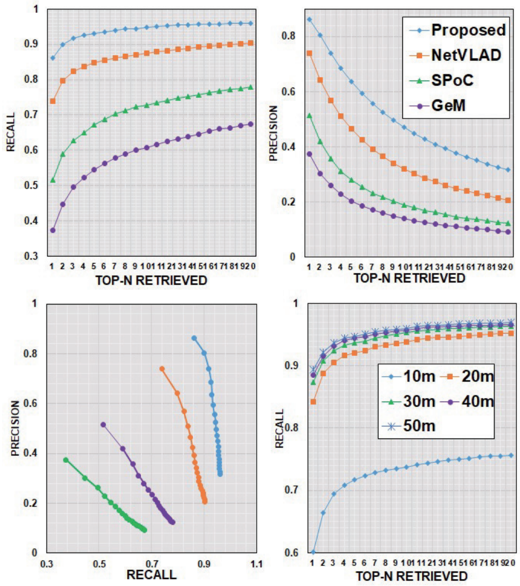


Figure 6. SPVP vs. STOA – recall and precision vs number of retrieved images; SPVP outperforms all current state-of-the-art pooling methods by a large margin; On the last figure, SPVP also perform reasonably with different error threshold

shown in Figure 4-1 , respectively. In the experiment, we use GPS as the growth truth location for queries. For each image in the query set, correct locations of the image are all street-view points with the distance to the image's GPS location less than D meters with D is from 10 to 50.

For each query, we retrieve top-N most similar images from the database and say that this query is correctly recognized at N if at least one in top-N retrieved images is at the correct point. Then we calculate the percentage of correctly recognized queries ($Recall@N$) as well as the percentage of correct images in the retrieved result ($Precision@N$) with respect to the values of N and methods. Formally, let $Q_i : i = (1-M)$ is the set of queries, $R_k^{(Q_j)} : k = (1-K)$ is the retrieved result vector of the query Q_j with $R_k^{(Q_j)}$ is either 1 for correct images or 0 for wrong images. Then with $N \leq K$, the Recall@N is calculated by:

$$Recall@N = \frac{\sum_{i=1}^{M}(OR_{k=1}^{N} R_k^{Q_i})}{M} \tag{3}$$

and the Precision@N is calculated as:

$$Recall@N = \frac{\sum_{i=1}^{M}(\sum_{k=1}^{N} R_k^{Q_i})}{M} \tag{4}$$

B. Quantitative Experiment

We demonstrate our proposed SPVP scheme on our collected dataset. We also employ popular pooling methods on top of CNN activations and state-of-the-art end-to-end learning method to compare with our pooling scheme, namely max-pooling (MAC) [20], mean (SPoC) and generalized mean (GeM) [15] pooling, and NetVLAD [1]. We report the recall and the precision curve of all approaches in Fig. 6.

On our dataset, SPVP outperforms all previous pooling methods, including the state-of-the-art NetVLAD, by a large margin. SPVP achieves top-1 recall of 86% and top-20 of 96%, which is leading over NetVLAD (74% and 90%), SPoC (49% and 79%), and GeM (37% and 68%). Remarkably, MAC performs extremely poor on our dataset with top-1 and top-20 recall are both smaller than 1% (that we do not include it in the charts!). Similarly, SPVP exclusively reaches higher performance than other approaches in both precision and precision-recall experiment. For example, at the 40% precision, GeM, SPoC, and NetVLAD attain recalls of 40%, 60%, and 85%, while SPVP reaches 95% recall at the same precision. Nevertheless, the improvement of our model comes with the cost of memory: the length of representation vectors in our approach is 51.200, much higher than SPoC (40), GeM (40), and NetVLAD (4.096). We believe this drawback can be resolved when combining our approach with dimension reduction techniques like PCA.

In the previous experiments, we set the distance error threshold D=25m. However, in practice, this value can vary depending on the application. Therefore, we examine our method with different distance thresholds from 10m to 50m and step size of 10m, and report the recall value at each distance threshold in Fig. 6. The result indicates that the recall is significantly reduced with $D = 10m$ with the value at top-1 and top-20 is 60% and 75%. Nevertheless, we argue that this is normal because of two reasons: (1) we collect the database with a $10m \times 10m$ grid, therefore $D = 10m$ is smaller than the resolution of the database; and (2) even the GPS (which we use as the growth truth in our experiment) itself has error of 10m - 15m. When $D > 10m$, our

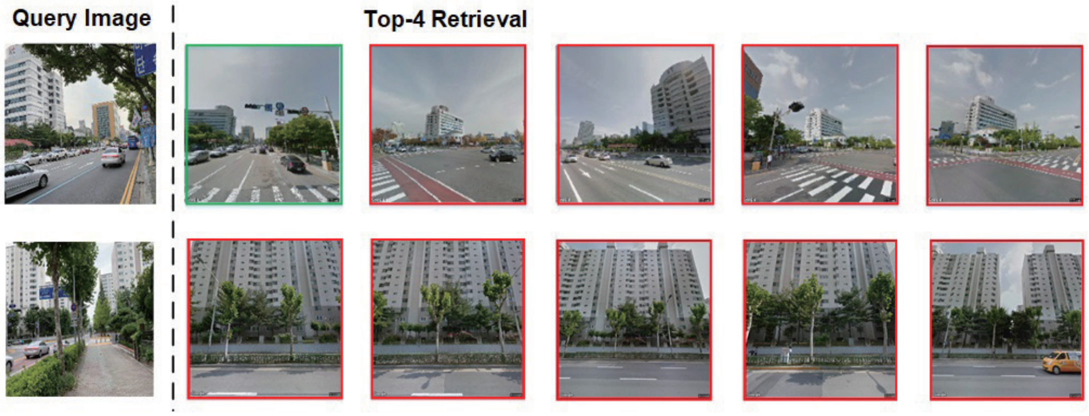


Figure 7. Visual failure cases of NetVLAD [1] due to missing of semantical information of the baseline model; the green bound denotes the correct image and red bound denotes the wrong image; in the above query, the model suffers from the “reflection effects”, and in the below query, the model suffers from the repetitive structures

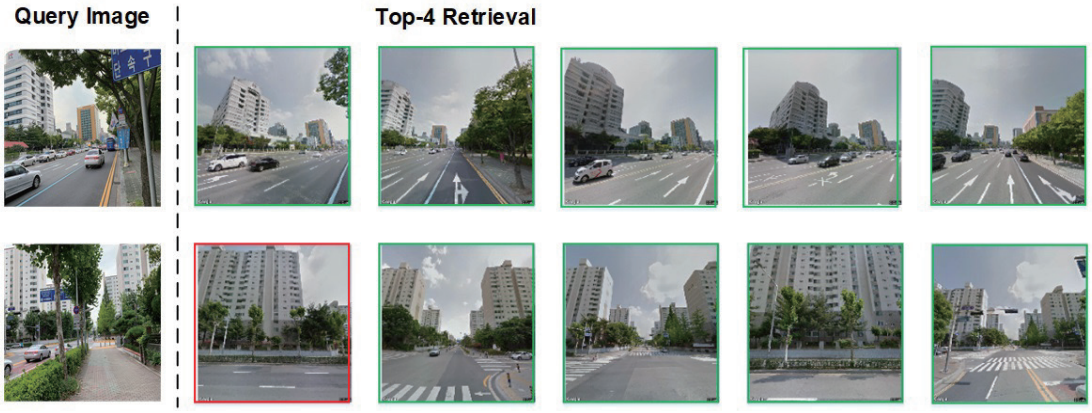


Figure 8. SPVP can overcome the repetitive structures as well as provide more semantic representation of images with most of retrieved images are at the correct locations; moreover, even SPVP is designed in mind to deal with the repetitive patterns and spatial information loss, it still performs well on ordinary queries

approach consistently produces good performance with recall values from 85% to 95%.

C. *Qualitative Experiment*

When we did experiment on some specific queries, we demonstrate that SPVP can overcome the repetitive structures as well as provide more semantic representation of images. Figure Fig. 7 shows the queries and its top-4 retrieval result with NetVLAD from the database with the green bound denotes the correct image and red bound denotes the wrong image. On the same queries, as shown in Fig. 8, SPVP does not suffer from repetitive structures as well as the view-point changes and most of the retrieved images are at correct locations

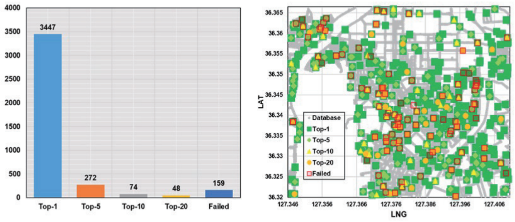


Figure 9. Statistic of recognition result and heatmap of the queries 86% of queries are successfully recognized at top-1 and 4% are totally failed under top-20; both good queries and failed queries are equally likely over the whole map without any bias

D. *Statiscal Analysis of the Dataset*

Finally, we report the statistical of the experiment on the dataset and show the result in Fig. 9. Among 4000 queries,

7

3447 queries (86%) are successful recognized at top-1, 272 queries (6.8%) at top-5, 47 queries (1.1%) top-10, and 48 queries (1.1%) at top-20. The remaining 159 queries (4%) are totally failed under top-20. We show the heat-map of the queries in Fig. 9 with the color of green indicates good recognition performance queries (successful recognized at top-1) and red for failed queries. We can observe that the distribution of both good queries and failed queries are equally over the whole map. Therefore, we conclude that our large scale dataset is fair enough to be utilized in other experiments out of this thesis.

V. Conclusion and Discussion

A. *Concluding Remark*

In this thesis, we present a fully-automated system for place recognition at the city-scale based on content-based image retrieval. Our contributions for the community lie in three aspects: (1) we take a comprehensive analysis of visual place recognition and sketch out the unique challenges of the task; (2) we propose yet a simple pooling approach on top of CNN features to embed the spatial information into the image representation vector; and (3) we introduce new datasets for place recognition, which are particularly essential for application-based research.

B. *Discussion*

1) Visual place recognition in reality: Visual place recognition has many potential applications in a wide range of services. In robotics, visual simultaneous localization and mapping (SLAM) have become an emerging field of research in recent years. SLAM is the task of constructing and updating the map of an environment while simultaneously maintaining the location of the agent within it. Visual SLAM does the task by using visual information only, therefore, techniques of visual place recognition can be easily adopted in visual SLAM. On the other hand, visual place recognition has been utilized in commercial photo collection services. Google Photos is the cloud-based service that allows users storing and organizing their photos on the cloud. In Google Photos, the location of the image can be either the GPS or predicted based on its visual appearance when GPS is not available. In the future, we believe that visual place recognition can be useful in many other emerging platforms such as smart city and smart building (indoor localization).

2) Deep learning and image retrieval: Deep learning and image retrieval: In Section ii., we have shown that deep learning is radically changing the general framework of image retrieval, nevertheless, how much should deep learning be adopted in the framework, is somewhat an interesting question. In [1], the authors state that "the core behind the idea that makes the success of deep learning is end-to-end learning", and suggest that the general framework can be replaced with single-pass deep learning models. However, in terms of representation learning, the representation from single-pass models is not general and coupled to one category of image retrieval. On the other hand, in the traditional framework, we learn the general image representation first and deploy the task-specific models on top of this representation, therefore, we can exploit more sophisticated techniques tailored with the task to improve the retrieval performance, e.g. spatial information embedding or exploiting the side features. Another major disadvantage of the single-pass models is they require complex training datasets with strong supervision.

3) Legacy of the datasets: In this thesis, we build our dataset with Google Streetview Imagery, following pioneers at [22, 1, 18]. However, the newest Google Maps Platform Terms of Service states that

> "...the customer will not extract, export, or otherwise scrape Google Maps Content for use outside the Services. For example, Customer will not: (i) pre-fetch, index, store, reshare, or rehost Google Maps Content outside the services; (ii) bulk download Google Maps tiles, Street View images, geocodes, directions, distance matrix results, roads information, place information, elevation values, and time zone details; (iii) copy and save business names, addresses, or user reviews; or (iv) use Google Maps Content with text-to-speech services"

Therefore, even we do not use the Google Map API, our datasets may still violate the Googe Terms of Uses. Nevertheless, there are open map platforms that support building the Streetview database and therefore, can be utilized in the future for constructing datasets for place recognition.

VI. Acknowledgement

References

[1] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1437–1451, jun 2018.

[2] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. Neural Codes for Image Retrieval. *Proceedings of European Conference on Computer Vision*, apr 2014.

[3] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. In *Proceedings of European Conference on Computer Vision*, volume 3951 LNCS, pages 404–417. Springer, Berlin, Heidelberg, 2006.

[4] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. BRIEF: Binary robust independent elementary features. In *Proceedings of European Conference on Computer Vision*, volume 6314 LNCS, pages 778–792, 2010.

[5] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395, jun 1981.

[6] Yunchao Gong, Liwei Wang, Ruiqi Guo, and Svetlana Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 8695 LNCS, pages 392–407. Springer Verlag, 2014.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1904–1916, sep 2015.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2016-Decem, pages 770–778. IEEE, jun 2016.

[9] Herve Jegou, Matthijs Douze, Cordelia Schmid, and Patrick Perez. Aggregating local descriptors into a compact image representation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 3304–3311. IEEE, jun 2010.

[10] Hyo Jin Kim, Enrique Dunn, and Jan-Michael Frahm. Predicting Good Features for Image Geo-Localization Using Per-Bundle VLAD. In *2015 IEEE International Conference on Computer Vision (ICCV)*, volume 2015 Inter, pages 1170–1178. IEEE, dec 2015.

[11] Hyo Jin Kim, Enrique Dunn, and Jan-Michael Frahm. Learned Contextual Feature Reweighting for Image Geo-Localization. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3251–3260. IEEE, jul 2017.

[12] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, nov 2004.

[13] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-Scale Image Retrieval with Attentive Deep Local Features. In *Proceedings of IEEE International Conference on Computer Vision*, pages 3476–3485. IEEE, oct 2017.

[14] Florent Perronnin, Yan Liu, Jorge Sanchez, and Herve Poirier. Large-scale image retrieval with compressed Fisher vectors. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 3384–3391. IEEE, jun 2010.

[15] Filip Radenovic, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Revisiting Oxford and Paris: Large-Scale Image Retrieval Benchmarking. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5706–5715. IEEE, jun 2018.

[16] Filip Radenovic, Giorgos Tolias, and Ondrej Chum. Fine-Tuning CNN Image Retrieval with No Human Annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7):1655–1668, jul 2019.

[17] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN features off-the-shelf: An astounding baseline for recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 512–519. IEEE Computer Society, sep 2014.

[18] Grant Schindler, Matthew Brown, and Richard Szeliski. City-Scale Location Recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–7. IEEE, jun 2007.

[19] Sivic and Zisserman. Video Google: a text retrieval approach to object matching in videos. In *Proceedings of IEEE International Conference on Computer Vision*, pages 1470–1477 vol.2. IEEE, 2003.

[20] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of CNN activations. In *International Conference on Learning Representations*, nov 2015.

[21] Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 Place Recognition by View Synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(2):257–271, feb 2018.

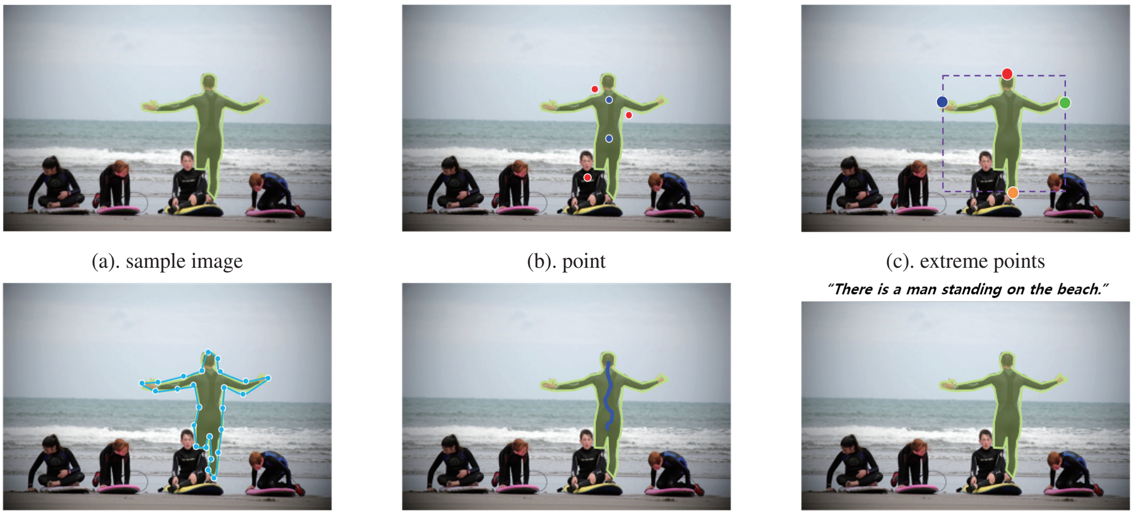
[22] Akihiko Torii, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. Visual Place Recognition with Repetitive Structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(11):2346–2359, nov 2015.

[23] Akihiko Torii, Hajime Taira, Josef Sivic, Marc Pollefeys, Masatoshi Okutomi, Tomas Pajdla, and Torsten Sattler. Are Large-Scale 3D Models Really Necessary for Accurate Visual Localization? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017-Janua:1–1, 2019.

A Survey on Interactive Image Segmentation Using Deep Learning

Gyuhyeon Sim[1], Jinho Choi*[2], Hyesu Lim*[3] and Jaegul Choo[1]

[1] Graduate School of AI, KAIST, Daejeon, Republic of Korea, {ghsim, jchoo}@ kaist.ac.kr
[2] Tomocube Inc, Daejeon, Republic of Korea, jhchoi@tomocube.com
[3] Department of Computer Science and Engineering, Korea University, Seoul, Republic of Korea, limhyesu98@korea.ac.kr



(a). sample image (b). point (c). extreme points

"There is a man standing on the beach."

(d). polygon (e). scribble (f). natural language

Fig. 1. Different types of user interactions that specify the same object of interest: (a) sample image where the target object is highlighted in the yellow mask; (b) user annotations denoting foreground and background are painted in red and blue point, respectively; (c) extreme points (the left-, top-, right- and bottom-most pixels) of the target object are marked with blue, red, green and orange color, respectively; (d) the target object is wrapped with polygon composed with sky blue dots; (e) the object of interest is specified by scribble; and (f) the target object is described using natural language.

Abstract

Interactive image segmentation is the task of extracting the object of interest accurately. The target object can be specified by various user interactions. As a result of the recent success of deep learning in computer vision, deep learning based models have been introduced to the field of interactive image segmentation. Such models outperform the traditional method that mainly uses hand-crafted features. In this survey, we briefly introduce the recent literature of the interactive image segmentation using deep learning techniques. We group them based on different types of user interactions as well: point based, extreme points based, polygon based, scribble based and natural language based interaction.

Keywords— *Interactive Image Segmentation, Computer Vision, Deep Learning, Image Segmentation*

I. Introduction

Convolutional neural networks(CNNs) pan out the image classification task with learning 1.2 million images in the ImageNet which is a dataset of 15 million labeled high-resolution images [16]. Since then, the CNNs have been applied to several computer vision tasks (e.g., object detection, semantic segmentation, instance segmentation), and have achieved remarkable accuracy in these tasks [30, 14, 9]. Among these, the semantic segmentation, which is a sub-task of the image segmentation, is the process of partitioning all image pixels into sets of pixels with the same meaning. This semantic segmentation plays an important role in the fields where the visual understanding should be required, such as 1) medical image analy-

*These authors contributed equally.

sis for diagnosis [11], 2) urban-scene understanding in autonomous driving [10].

Image segmentation is formulated as a pixel-wise classification. Not only does this make the image segmentation more challenging than the image classification, but supervised settings in this task burden the human by letting them annotate all pixels in images. According to Microsoft COCO [6], a dataset for object recognition, they annotate all images in three steps which are category labeling, instance spotting, and instance segmentation. The 3-steps annotation pipeline takes about 20k, 10k, and 55k worker hours respectively for annotating 2.5 million instances. Although such large datasets(MS COCO [6], PASCAL VOC [12], Cityscape dataset [10], BSDS [26], etc.) for the image segmentation are released, it is difficult to use them directly as a training dataset because of limited types of object and context. Thus, the need for novel methods is even larger for facilitating more rapid annotation.

Interactive image segmentation has been spotlighted as a resolver of this issue. The interactive image segmentation instead focuses on extracting the object of interest accurately with minimal user effort [31]. Thus, the model for this helps the annotator to get satisfying image labels within the few clicks by the following steps: 1) the model gets user interaction in the form of bounding boxes or positive and negative scribbles; 2) then the initial prediction is returned to the user based on the input image and the user interactions; 3) the users refine the given prediction iteratively by adding the guidance until they get the satisfying prediction.

Even before the deep neural networks(DNNs) based approach appears, massive traditional methods tackled this problem by using the hand-crafted features such as a color distribution [31], edge information [27] and texture [7]. Specifically, the interactive image segmentation problem can be formulated by the graph optimization called Interactive Graph Cuts [5], and be solved by min-cut/max-flow algorithm [4]. This leads to Graph optimization based approaches being more popular such as graph cut [4] and random walk [13] utilized strokes [5, 37], bounding boxes [31, 36, 17], or boundaries [27]. However, these methods are limited in that they take mainly low-level features, so require substantial number of user interactions to earn good representation of foreground/background [41].

Xu et al. [41] first suggested DNNs based interactive instance-level segmentation model that performs superior to graph based models. They trained Fully Convolutional Networks [23] (FCNs) to segment an object relying on positive and negative clicks provided by user interactions, which are encoded by Euclidean distance transformation.

Our survey includes the necessary settings for training the behavior to extract objects in response to the user's interaction in the DNNs based approach. We also cover the most recently published literatures until 2020, and group these literatures into the following several categories (e.g., point, extreme points, polygon, scribble, and natural language) by the types of user interactions, illustrated in Fig. 1. The characteristics of classified literature are reported together briefly.

II. Simulating and Transforming User Interactions

In this section, we provide necessary settings which are needed to train user interactive model. Dissimilar to the general segmentation network, the interactive segmentation model focuses on extracting the object of interest based on the user's interaction. Thus, in addition to the segmentation network, additional components are needed to make the network aware of user interactions. For this purpose, Xu et al. [41] proposes a method of sampling clicks and representing them as input features which are forwarded to the network with the input image. Although each of them has slight difference according to its specific interaction way, these two components are widely used in the recent DNNs based approach [19, 18, 21, 15, 34].

A. Simulating user interactions

It is very expensive to collect human interactions for training from real users. Thus, the clicks have to be simulated in the way of mimicking real users. The general setting of user interactions simulation is addressed in [41]. Positive clicks indicate foreground objects and negative clicks indicate background. According to [41], the number of positive and negative clicks are randomly selected in $[1, N_{pos}]$ and $[0, N_{neg}]$ respectively where N_{pos} and N_{neg} are the maximum integer number of positive and negative clicks and they are given as hyperparameters. The candidates of positive and negative clicks are determined by the ground-truth mask. Let n_{pos} and n_{neg} denote selected number of positive and negative clicks and P_1, P_2, N_1, and N_2 denote the number of pixels that are given as hyperparamters for conditioning the candidates. The n_{pos} positive clicks are sampled from its candidates which are defined by two conditions; 1) pixels at P_1 pixels away from the ground-truth foreground object boundaries and 2) pixels that are P_2 pixels away from other pixels which are previously clicked as positive. The n_{neg} negative clicks are randomly sampled from its candidates which is defined within the ground-truth background. 1) Pixels away from the foreground object by N_1 pixels and 2) pixels that are N_2 pixels away from previously clicked as negative pixels are considered as ground-truth background. This general setting is widely used in the interactive segmentation models [42, 21, 34, 24]. Each of them has slight differences according to its specific model design.

B. Click representation

Similar to the click simulation, there are a couple of common ways of click representation called interaction map. Xu et al. [41] suggested transforming user interactions into Euclidean distance maps. Each pixel value in the Euclidean distance map is the distance between that pixel location and the location of the closest click. Two separate Euclidean distance maps are generated from positive and negative clicks respectively. they are concatenated with the input image and form a network input. Another approach of click representation is addressed in DEXTR [25]. From each positive and negative clicks, two separate heatmaps are generated by centering 2D Gaussian around each click. Similar to the previous approach [41], the heatmaps are concatenated with the input image.

III. Interactive Image Segmentation based on Deep Learning

As illustrated in Fig. 1, interactive image segmentation can be performed with various interaction ways from the user, and the segmentation model varies depending on the interaction way. We group recent literature into the following several categories by the type of user interaction: A. point based interaction, B. extreme points based interaction, C. polygon based interaction, D. scribble based interaction and E. natural language based interaction. We introduce each interaction method and briefly report the models corresponding to that method.

A. Point based interaction

Point based interactive segmentation models predict the mask of interested objects by taking clicks or from the user. that models can also receive scratch as a input, in that the scratch is defined by a set of points. In the case of Xu et al. [41], a user provides initial positive or negative clicks or scribble with the input image. Then the user gets an initial prediction and add new clicks iteratively for refinement until a satisfactory result is returned.

Xu et al. [41] first proposed an interactive segmentation algorithm based on DNNs, where clicks on foreground and background are received as an input of the model. The clicks are converted to a minimum euclidean distance map where each pixel means the euclidean distance from the closest user click. Then, the input image and transformed user interactions are concatenated and passed to the network as input. Liew et al. [19] proposed a RIS-Net which expands the field-of-view of the given user interaction by combining local regional information near the clicks with the multiscale global context. Similar to predicting the global context first in [19], Wang et al. [39] use one CNN to obtain the initial segmentation. Taking the user interactions and initial segmentation, another CNN is used to obtain more refined results. Lin et al. [21] expanded the existing network by attaching a first click attention network while emphasizing the importance of the first click. Jang and Kim [15] proposed a novel scheme named Backpropagating Refinement Scheme(BRS). The BRS edit the interaction map with backpropagation to avoid the user-annotated location from being labeled incorrectly. Most recently, Sofiiuk et al. [34] reduce computational costs required by BRS, introducing an auxiliary variables added in the middle of the networks.

B. Extreme points based interaction

In these methods, a model predicts the instance mask of the object from four extreme points given by users. Extreme points [29] are the left-most, right-most, top, and bottom pixels of the target object. Several interactive segmentation methods require users to draw the bounding box. However, drawing bounding boxes is cognitively demanding tasks that take 25.5 seconds [35]. Otherwise, this extreme clicking protocol only takes 7s seconds per instance.

Maninis and Caelles first propose an extreme points-based annotation tool called DEXTR [25]. They first crop the input by the bounding box inferred from extreme points, generate a heatmap by centering 2D Gaussian around each of four points, and concatenate it to the cropped input image. Instead of predicting instance masks of objects in an input image one by one, Agustsson et al. [2] propose a method that predicts the segmentation of masks of all objects at a time using Mask-RCNN [14]. They remove Region Proposal Network (RPN) [30] and directly obtain Region-of-Interest (ROI) from extreme points. Wang et al. [40] combine the extreme points-based method and level set optimization. They also allow users to refine the predicted masks using a motion vector, which is produced when the annotator drags incorrect points.

C. Polygon based interaction

Some other works [1, 22, 8] extend the traditional polygon-based annotation tools [33], which have been utilized to generate the ground-truth label of several public benchmark dataset [28, 20]. Polygon-RNN [8] is a semi-automatic annotation tool that predicts the polygon of the target object. Given the bounding box provided by the user, the model sequentially produces the vertices of the polygon using a Recurrent Neural Network. This tool allows users to correct initial predictions by moving the wrong vertices. While the traditional polygon-based annotation tools require 30 to 40 clicks to annotate an object, Polygon-RNN reduced the number of interactions by the factor of 4.7. Polygon-RNN++ [1] improves the proposed model by using reinforcement learning and increasing the output resolution of the predicted polygon. Ling et al. [22] propose Curve-GCN that utilizes graph convolutional networks and show superior performances than other polygon-based methods.

D. Scribble based interaction

User interactions can also be given by scribbles. In [38], the model takes two types of guidance from users. First, a user provides a bounding box then the initial segmentation is obtained. After that, the user may provide additional guidance with scribbles for the foreground and background. With (supervised) or without (unsupervised) user-given scribbles, the model is fine-tuned to the given image. They use a weighted loss function with higher weights on user-provided scribbles. For efficient annotation, Andriluka et al. [3] proposed an annotation interface called Mask Paint which combines the strength of traditional polygon drawing and free-form drawing tools. In Magic Paint [3], which is a Mask Paint with an automatic propagation assistant, users are allowed to mark the interior of the foreground object or the background instead of focusing on boundaries. The label of user-given scribbles is automatically propagated through the full image by computing global pixel similarity between labeled and unlabeled pixels.

E. Natural Language based interaction

Natural language can also be an approach to user guidance for refinement. Rupprecht et al. [32] allow users to give feedback in the form of natural language to fixed CNN at test time. User-given feedback is called as *guide* and the CNN is guided through a *guiding block*. The guiding block is trained to control the strength of activations per channel so that it can pay attention to the objects mentioned in the guide. The text guide is simulated by comparing the initial prediction and the ground truth mask. From the difference between them, the text guide which consists of class and location information is generated. Extracting important features from the given sentence through RNN, the inference network makes more accurate predictions.

IV. CONCLUSION

We have surveyed recent deep learning based interactive segmentation models. First, we gave a brief outline of the interactive segmentation. In section 2, stating the difference between the general segmentation and the interactive segmentation, we discussed common ways of simulating and transforming user interactions. Finally, we categorized deep learning based interactive segmentation models by the types of user interactions; point based, extreme points based, polygon based, scribble based, and natural language based interaction.

REFERENCES

[1] David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. Efficient interactive annotation of segmentation datasets with polygon-rnn++. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 859–868, 2018.

[2] Eirikur Agustsson, Jasper RR Uijlings, and Vittorio Ferrari. Interactive full image segmentation by considering all regions jointly. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11622–11631, 2019.

[3] Mykhaylo Andriluka, Stefano Pellegrini, Stefan Popov, and Vittorio Ferrari. Efficient full image interactive segmentation by leveraging within-image appearance similarity. *arXiv preprint arXiv:2007.08173*, 2020.

[4] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004.

[5] Yuri Y Boykov and M-P Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *Proceedings eighth IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 105–112, 2001.

[6] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1209–1218, 2018.

[7] Vicent Caselles, Ron Kimmel, and Guillermo Sapiro. Geodesic active contours. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 694–699, 1995.

[8] Lluis Castrejon, Kaustav Kundu, Raquel Urtasun, and Sanja Fidler. Annotating object instances with a polygon-rnn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 2, 2017.

[9] 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, 2018.

[10] 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 (CVPR)*, pages 3213–3223, 2016.

[11] M de Bruijne. Machine learning approaches in medical image analysis: From detection to diagnosis. *Medical Image Analysis*, 33:94–97, 2016.

[12] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[13] Leo Grady. Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1768–1783, 2006.

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, pages 2961–2969, 2017.

[15] Won-Dong Jang and Chang-Su Kim. Interactive image segmentation via backpropagating refinement scheme. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5297–5306, 2019.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012.

[17] Victor Lempitsky, Pushmeet Kohli, Carsten Rother, and Toby Sharp. Image segmentation with a bounding box prior. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 277–284, 2009.

[18] Zhuwen Li, Qifeng Chen, and Vladlen Koltun. Interactive image segmentation with latent diversity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 577–585, 2018.

[19] JunHao Liew, Yunchao Wei, Wei Xiong, Sim-Heng Ong, and Jiashi Feng. Regional interactive image segmentation networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2746–2754, 2017.

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014.

[21] Zheng Lin, Zhao Zhang, Lin-Zhuo Chen, Ming-Ming Cheng, and Shao-Ping Lu. Interactive image segmentation with first click attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13339–13348, 2020.

[22] Huan Ling, Jun Gao, Amlan Kar, Wenzheng Chen, and Sanja Fidler. Fast interactive object annotation with curve-gcn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5257–5266, 2019.

[23] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.

[24] Soumajit Majumder and Angela Yao. Content-aware multi-level guidance for interactive instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11602–11611, 2019.

[25] Kevis-Kokitsi Maninis, Sergi Caelles, Jordi Pont-Tuset, and Luc Van Gool. Deep extreme cut: From extreme points to object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 616–625, 2018.

[26] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[27] Eric N Mortensen and William A Barrett. Intelligent scissors for image composition. In *Proceedings of the 22nd annual Conference on Computer Graphics and Interactive Techniques*, pages 191–198, 1995.

[28] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 891–898, 2014.

[29] Dim P Papadopoulos, Jasper RR Uijlings, Frank Keller, and Vittorio Ferrari. Extreme clicking for efficient object annotation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4940–4949, 2017.

[30] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 91–99, 2015.

[31] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. " grabcut" interactive foreground extraction using iterated graph cuts. *ACM transactions on graphics (TOG)*, 23(3):309–314, 2004.

[32] Christian Rupprecht, Iro Laina, Nassir Navab, Gregory D Hager, and Federico Tombari. Guide me: Interacting with deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8551–8561, 2018.

[33] Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1-3):157–173, 2008.

[34] Konstantin Sofiiuk, Ilia Petrov, Olga Barinova, and Anton Konushin. f-brs: Rethinking backpropagating refinement for interactive segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8623–8632, 2020.

[35] Hao Su, Jia Deng, and Li Fei-Fei. Crowdsourcing annotations for visual object detection. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[36] Meng Tang, Lena Gorelick, Olga Veksler, and Yuri Boykov. Grabcut in one cut. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1769–1776, 2013.

[37] Vladimir Vezhnevets and Vadim Konouchine. Growcut: Interactive multi-label nd image segmentation by cellular automata. *Graphicon*, 1(4):150–156, 2005.

[38] Guotai Wang, Wenqi Li, Maria A Zuluaga, Rosalind Pratt, Premal A Patel, Michael Aertsen, Tom Doel, Anna L David, Jan Deprest, Sébastien Ourselin, et al. Interactive medical image segmentation using deep learning with image-specific fine tuning. *IEEE Transactions on Medical Imaging*, 37(7):1562–1573, 2018.

[39] Guotai Wang, Maria A Zuluaga, Wenqi Li, Rosalind Pratt, Premal A Patel, Michael Aertsen, Tom Doel, Anna L David, Jan Deprest, Sébastien Ourselin, et al. Deepigeos: a deep interactive geodesic framework for medical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7):1559–1572, 2018.

[40] Zian Wang, David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. Object instance annotation with deep extreme level set evolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7500–7508, 2019.

[41] Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas S Huang. Deep interactive object selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 373–381, 2016.

[42] Shiyin Zhang, Jun Hao Liew, Yunchao Wei, Shikui Wei, and Yao Zhao. Interactive object segmentation with inside-outside guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12234–12244, 2020.

Learning Spatial Relationships for Cross-Modal Retrieval

Hayeon Lee*[1], Wonjun Yoon*[2], Jinseok Park[3] and Sung Ju Hwang[4]

[1] School of Computing, KAIST, Daejeon, Korea, hayeon926@ kaist.ac.kr
[2] Lunit, Seoul, Korea, wonjun@lunit.io
[3] School of Computing, KAIST, Daejeon, Korea, jspark@ mmc.kaist.ac.kr
[4] Graduate School of AI, KAIST, Daejeon, Korea, sjhwang82@ kaist.ac.kr

Abstract

Understanding the relationships between objects in an image is an important problem in visual recognition. While such spatial relationships have been exploited for various vision tasks such as visual question answering, they have not been leveraged much for cross-modal retrieval. In this paper, we tackle the problem of cross-modal retrieval that considers the spatial relationships between visual objects. To this end, we propose a CNN architecture which we refer to as Object Phase Module (OPM), which encodes relative locations of objects in a given image. To validate the efficacy of our method, we compiled a novel dataset, R-CLEVR, whose captions describe spatial relationships between objects, on which our model significantly outperforms existing cross-modal retrieval methods. Further experimental study on MS-COCO shows that such focus on spatial relationships could be useful on datasets with relatively small number of spatial relationships as well.

Keywords— *Cross-Modal Retrieval, Visual-Semantic Embedding*

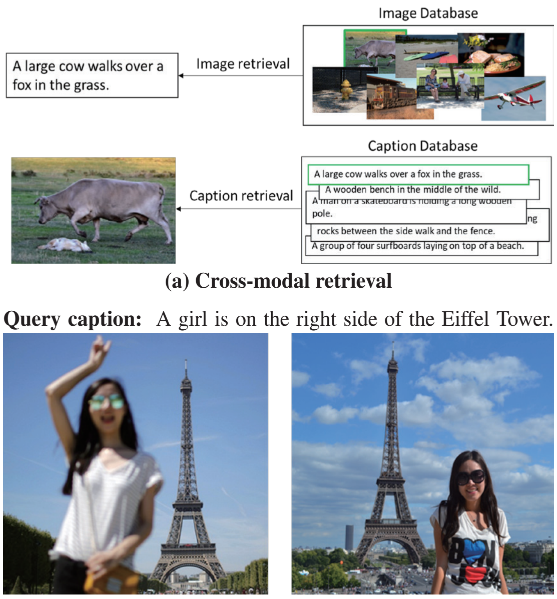


Image Database
A large cow walks over a fox in the grass.
Image retrieval
Caption Database
A large cow walks over a fox in the grass.
A wooden bench in the middle of the wild.
A man on a skateboard is holding a long wooden pole.
ng
rocks between the side walk and the fence.
A group of four surfboards laying on top of a beach.
Caption retrieval

(a) Cross-modal retrieval

Query caption: A girl is on the right side of the Eiffel Tower.

(b) Image retrieval with spatial relationships

Fig. 1. (a) Cross-modal retrieval task requires to find the best matching instance from the other domain, given an instance from either the image or text domain. (b) For certain Cross-modal retrieval tasks, we need to understand the spatial relationships between objects for accurate retrieval.

I. Introduction

Various computer vision tasks such as image retrieval, text-to-image synthesis, and visual question answering (VQA) [1, 5, 25] requires reasoning over multi-modal information from both images and texts. While recent advances in deep convolutional networks and text encoding methods have made it possible to learn to accurately represent the visual and textual inputs separately, the main bottleneck in multi-modal learning with images and texts, is on learning a semantic space [21] that jointly embeds them to accurately describe the relationships between visual and textual data. Obtaining a good semantic space is essential in solving practical cross-modal tasks such as visual localization of phrases [4, 36] where we need to localize objects without object-level annotations.

Since each image is not a simple set of objects but rather can be considered as a graph of objects interconnected by relationships [24], it is essential to ensure that the learned semantic embedding space capture such semantic relationships between visual entities. Recent work [13, 14, 35, 36, 38] has proposed to learn such semantic-visual embeddings by learning a two-path network where each path encodes visual and textual data respectively. However, to the best of our knowledge, no work has considered the relations among the objects when encoding an image in such a two-path network for learning a semantic-visual embedding space. For instance, if the image contains two objects

*These authors contributed equally to this work.

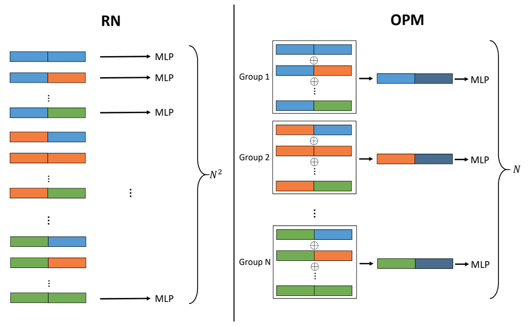


Fig. 2. Comparison of RN and OPM (Ours) architecture.

of the same type in two different locations, the network will struggle to recognize how they are related and simply understand that there exist two objects of the same type somewhere in an image as shown in Figure 1. This is an important problem since many real-world scenarios such as grasping and navigation where the environment contains multiple objects of the same class, it may not be always possible to differentiate the objects or scenes solely by their properties, and we may need to consider their relations as well [3, 18, 28].

Our goal is to entail spatial relationships among objects in a joint embedding space. Relation Network(RN) [27] tackles this problem by learning representations that consider pairwise relationships between spatial features, and achieve impressive performance on the VQA datasets that requires to do demanding relational reasoning. The RN basically learns a network that focus on the common properties for relational reasoning given two pairs of features corresponding to different spatial locations which may correspond to objects, such as their relative spatial location (See Figure 2). However, this pairwise relational encoding has limited scalability since it requires to generate $O(N^2)$ feature maps, where N is the number of spatial bins, and thus cannot be used with even moderately large visual encoder network (e.g. VGG-19). To overcome this limitation, instead of performing pairwise encoding, we propose to encode each spatial feature using all other features combined into a single set vector. This reduces the complexity of the feature map to $O(N)$, enabling it to be coupled with large encoders (e.g. ResNet). We refer to this set-wise encoding as *Object Phase Module (OPM)*. We validate OPM to cross-modal retrieval and phase grounding tasks. For accurate cross-modal retrieval, relational reasoning of spatial relationships between object is important but has been relatively neglected in previous work. One reason for this is because for existing datasets, relational spatial reasoning is not crucial for accurate retrieval and genral semantics plays a major role in defining the image-text relationships, or only describe absolute location of objects (e.g. "zebra in the left"). Thus, to better validate the spatial reasoning ability for cross-modal retrieval, we compile a new dataset, Retrieval-CLEVR (R-CLEVR), that requires to understand the spatial relationships between objects for accurate cross-modal retrieval. R-CLEVR consists of 93,000 images, each of which has three sentences describing relative locations of objects (See Figure 4), and come in three subsets categorized by the difficulty in spatial reasoning. On this R-CLEVR dataset, Our end-to-end network architecture for cross-modal retrieval with OPM achieved significantly higher accuracy over state-of-the-art baseline methods. Further validation of our model shows that our model obtains good performance on a general domain as well, in which the relationships are less concerned. Specifically, our model achieves state-of-the-art and second-best performance on the image-to-text and text-to-image retrieval respectively. Finally, we validate our model on the phrase grounding task [13, 26], whose results demonstrate our model's effectiveness over state-of-the-art methods.

In summary, our contribution is threefold:

- We present Object Phase Module (OPM), that efficiently encodes the spatial relationships among objects, with set-wise encoding of spatial relationships for learning visual-semantic embeddings.
- We present a new dataset, R-CLEVR, that contains visual objects and textual descriptions that describe spatial relationships among them, that can be utilized for cross-modal retrieval and phase grounding task.
- We validate the effectiveness of our model by comparing its performance with state-of-the-art methods for cross-modal retrieval on R-CLEVR and MS-COCO, and phrase grounding tasks on R-CLEVR, on which it largely outperforms existing approaches.

II. Related Work

Visual-semantic embedding Learning joint-embedding space for visual and textual modalities is an extensively studied topic, due to its applicability to various tasks that requires multi-modal reasoning such as cross-modal retrieval and visual question answering. Recent deep learning based approaches [12, 13, 14, 17, 20, 33, 34, 35] have shown that learning of visual-semantic embedding can be done via a two-path network architecture [34], with seprate paths for image and text encoder that are conjoined at the upper layer, where upper layers learn common metric space that maps the visual instances with their corresponding semantic label embeddings. The image encoder is commonly implemented as convolutional neural networks such as ResNet [15] or VGG [29]. The text encoder is often implemented as a recurrent neural network (RNN), such as LSTM [16], GRU [7] and SRU [32], which receives the word embeddings generated from Word2Vec [23] as input. Recent work [13, 14, 35] use hard negative mining along

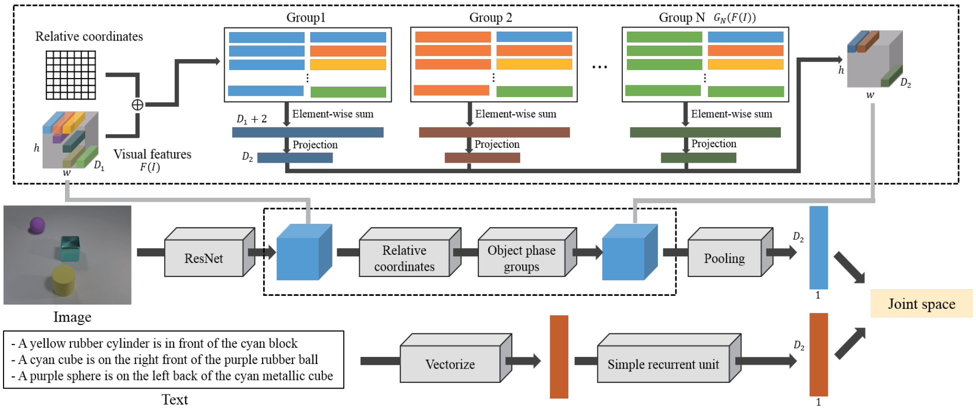


Fig. 3. Proposed architecture for learning semantic-visual embedding with Object Phase Module (OPM). Top box with dashed lines describes the object phase module and bottom part shows the overall structure for our semantic-visual embedding.

with contrastive loss, which successfully improved the performance of cross-modal retrieval tasks. Engilberge . [13] adopt selective spatial pooling [11, 10] originally proposed for weakly-supervised object localization to enhance visual feature extraction. DAN [17] propose to use an attention module to focus on the relevant part of the features from the different modalities. A few work have tried to improve on the textual encoding as well, such as CHAIN-VSE [35] which propose a convolution network-based text encoder using a variation of the inception module [31].

Learning spatial relation Several studies [8, 28, 38] have proposed to learn spatial relations among objects. Learning of spatial relationships is important for many tasks, such as phase grounding whose objective is to localize the corresponding regions or objects given texts that describe the relationships between the objects. However, the previous studies assume a supervised setting where the image is given with ground-truth bounding boxes with corresponding phrases which may be unavailable for real-world datasets. Moreover, none of the aforementioned work has considered how to entail spatial relationships of objects in an image in the image encoder. Recently, Adam et al. [27] proposed to capture relations among objects, using a specific module that encodes pair-wise relationships between all pairs of objects in a given image. Specifically, the model pairs visual feature for each object with the feature for another object, and use it to an input to an MLP that learns the correct relationship between the two. Using this relational module, RN tackles visual question answering (VQA) [1] that requires to reason the relationships between objects, by encoding the visual-relational features conditioned on the text embedding of the question, and achieves superhuman performances on their own dataset. However, RN is limited in that it uses pairwise encoding, which quadratically increases with the increase in feature dimensions, and is not applicable to applications that require high-dimensional representations. Contrarily, our object phase module considers the relation of all other features to the target feature, and thus has much lower complexity and better captures the target-specific relationships.

Localization of phrases with embedding Recently, building on the line of visual-semantic embedding research, several work [4, 11, 13, 36] adopt weekly-supervised approaches to localize entities corresponding to the descriptions of phrases in the form of attention masks without strong supervision (e.g. bounding boxes). To extract features that capture spatial information, these previous work uses selective spatial pooling [11, 10]. However, the spatial pooling methods are suboptimal since they do not directly preserve spatial visual correspondence between objects. In contrast to such previous approaches, we have designed a module that directly learns relative phases of the objects by explicitly learning the semantic correspondence between text descriptions and visual features. We relocate the features relatively to global features, and group them such that we can concretely infer the relationships between objects, rather than predicting them from visual attributes as done in previous work, which does not truly learn the relations among objects.

III. Approach

In this section, we describe our model for cross-modal retrieval. Our model has four main components: 1) base feature extractor, 2) Object Phase Module (OPM), 3) text encoder and 4) triplet ranking loss. The proposed network follows the architecture of the two-branch network [34] that consists of a visual encoder and a text encoder in two separate branches merged into one. The distinctive component of our cross-modal retrieval model that differentiate it from the existing model, is OPM, which is a variant of the Relation Network (RN) that captures spatial relationships between objects. In particular, OPM captures the relative coordinates of objects and use the object phase group which denotes a conditional vector given global feature space to induce a relation of each feature with global ones. Figure 3 illustrates the overview of our model.

A. *Visual feature extractor*

To extract visual features that preserve the spatial information of objects in the given image, we use a all-convolutional version of ResNet-152 [15] pretrained with ImageNet [9] as a base image encoder F, from which we removed all full-connected layers and global average pooling. Let an image $I \in (0,255)^{W\times H\times 3}$ as an input, where W,H are the width and height of the image. Then, the base image encoder produces a 3D visual feature map $F(I) \in \mathbb{R}^{h\times w\times D_1}$, where $w = \frac{W}{32}$, $h = \frac{H}{32}$, and D_1 is the dimension with the size 2048.

B. *Object phase module*

In this subsection, we introduce our main contribution, Object Phase Module (OPM), which reconstructs a 3D feature map $F(I)$ that explicitly captures the spatial relationships among visual features. The model has two distinctive features: relative coordinates and intertwined group features.

Relative coordinates If the network is given the locations of objects via bounding boxes or coordinates, it will be straightforward to capture spatial relationships among objects. However, for cross-modal retrieval, often the only available information about the objects' locations are image-level text descriptions. To tackle this challenge, we use normalized coordinates of each feature $f_{i=1,\dots,N=w\times h}$ in the 3D feature map $F(I)$ to memorize the positions of objects and infer their spatial phases. Let the coordinate (x_i,y_i) represents the spatial position of the each feature f_i. Then, we normalize (x_i,y_i) to the value ranging from -1 to 1 and concatenate this normalized coordinates at the end of the feature f_i, to yield a D_1+2-dimensional features. Through our ablation study in Table 3, we found that this simple technique significantly enhances the cross-modal retrieval performance.

Object phase groups In order to encode relative coordinates of other objects with respect to the target object, we create a group G_i that corresponds to each feature f_i. In the group G_i, we generate all possible pairs of the central feature f_i and other features (including itself), f_j for $j = 1,\dots,N$, by concatenating them with their coordinates. Thus, each group G_i consist of N pairs,

$$G_i(F(I)) = \{p_{i,1},\ p_{i,2},\ \dots,\ p_{i,N}\} \tag{1}$$

where $p_{i,j} = (f_i \oplus x_i \oplus\ y_i \oplus f_j \oplus x_j \oplus y_j)$, $\oplus$ is a concatenation of the two features and their coordinates.

To reflect the relations of visual features evenly to each central feature, we apply an element-wise sum on the each group, which yields a $2\times(D_1+2)$ dimensional vector. Then, each vector is passed to the projection layer P_θ with learnable parameters θ that shares the weights as shown in Figure 3. This will generate a feature map whose dimensionality is $w\times h\times D_2$, where $D_2 = 2400$ is the number of outputs from the projection layer. We feed the output to the max pooling by generating the final joint embedding vector. We relocate the functional form of each f_i as G_i denoting a conditional vector given the global feature space, and it is essential for P_θ to induce a relation of each feature with global ones. Then, P_θ followed by max pooling is forced to be trained to maximize its response over the relevant image regions. Throughout the cross-modal retrieval experiments in Section B., we demonstrate that the network can learn spatial relationships between the objects.

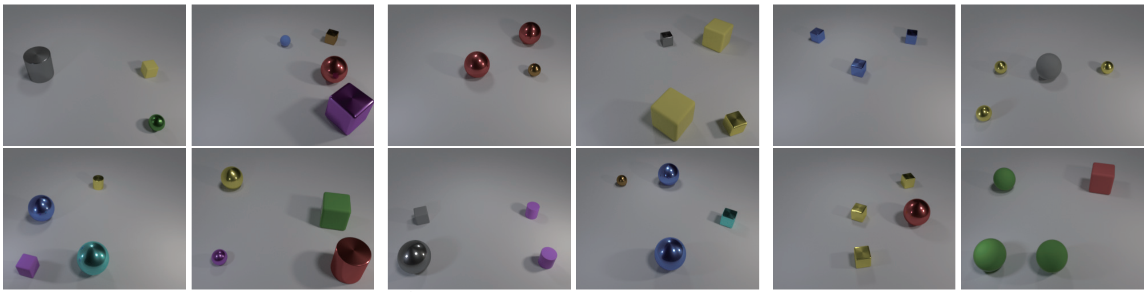
Module	# of input features	# of FLOPs $\times 10^8$
RN [27]	$O(N^2)$	234.4
Ours	$O(N)$	4.9

Table 1. Comparison of the number of features inserted into the projection layer and FLOPs.

RN and OPM Relation Network (RN) [27] is a module that constrains functional form of the network by intertwining all object features with their relative coordinates. This architecture is shown to be effective on the VQA task which requires relational reasoning, but it is difficult to apply RN directly to the semantic-visual embedding tasks. This is because RN concatenates a text embedding as a condition to the image encoder, to obtain a single vector that encodes features with respect to the given question. This yields significant bias to the text when learning the joint embedding for texts and images.

In addition, the original feature extractor used in RN is a four layer vanilla network, which outputs less than 100-dimension vectors. This design allows the network to consider the potential relations between all objects, which results in N^2 pairs. In general, however, the networks used for semantic-visual embeddings are often large (VGG, ResNet), which is required to extract richer visual features, resulting in high-dimensional features (orders of thousands). When this high-dimensional feature is fed to a projection layer of $O(N^2)$ complexity, this will require excessive amount of floating point operations (FLOPs) as shown in Table 1.

Even if obtaining N^2 relational pairs could provide one-to-one relationships between objects, we have found that grouping the features in relation to the target ones in the global feature space is significantly more efficient (Table 2), as it reduces the number of input features to the projection layer by $O(N)$. Moreover, this set-wise encoding technique enables scalable visual-semantic embedding task, which yields each state-of-the-art and second-best performance for MS-COCO in Table 4. Note that MS-COCO is less concerned about the relationships of objects, as it is designed as general benchmark for high-level semantic-visual recognition tasks.



Easy [no duplicate]
A yellow cube in front of the green shiny ball.
A green ball in front of the small yellow rubber cube.
A cylinder at the left back of the small green shiny sphere.

Middle [two duplicate]
A red shiny ball is left to the brown shiny sphere.
A big red sphere is in front of the small brown ball.
A small brown sphere right to the large red shiny sphere.

Hard [three duplicate]
A blue cube is left behind of the small shiny rectangle.
A blue rectangle is left behind of the small blue cube.
A blue shiny cube is left front of the small blue shiny cube.

Fig. 4. Three types of R-CLEVR Dataset. An image involves three to four objects including duplicate objects. The text listed under each difficulty is the description of the first image of each difficulty.

C. *Textual encoding*

Given any natural language text S that describes the spatial relation of various objects in the image, our text encoder T converts it into a vector representation. For example, S can be expressed in forms of a phrase or sentence such as ‘a gray rubber cube on the left front of the green metal cylinder’ and ‘a yellow rubber cylinder is in front of the cyan block’. We tokenize such texts into tokens $\{s_1, s_2, ..., s_{N_T}\}$ with the number of tokens N_T. Each token s_{n_T} is mapped to the index i_{n_T} according to the dictionary and a recurrent unit take the index set $\{i_1, i_2, ...i_{N_T}\}$ as an input. We adopt the recurrent unit structure [13] to embed the text in a continuous, joint space. The structure consists of simple recurrent units (SRU) [32] and L2 normalization. The text encoder produces the text embedding vector $T(S) \in \mathbb{R}^{D_2}$ with dimension D_2=2400 in the joint metric space.

D. *Triplet ranking loss with hard negatives*

Following the previous work [20, 21, 30], we use contrastive triplet ranking loss. Since VSE++ [14] shows that incorporating hard negatives can significantly enhance the retrieval performance, we follow the same strategy. Let B=$\{(I_n, S_n)\}_{n=1}^{N_B}$ be a batch training set with N_B (image, text) pairs and v_n be the embedding of image I_n from the image encoder and w_n be the embedding of text S_n from the text encoder.

We then use the contrastive triplet ranking loss with hard negatives defined as:

$$\mathscr{L} = \frac{1}{N_B}\sum_{n=1}^{N_B}\Big(\max_{m\in N_B, m\neq n} L_{cont}(v_n, w_n, w_m) + \max_{m\in N_B, m\neq n} L_{cont}(w_n, v_n, v_m)\Big) \tag{2}$$

where

$$\mathscr{L}_{cont}(q,p,n) = max(0, \alpha - cos(q,p) + cos(q,n)). \tag{3}$$

where α is the margin, q is the query, p is a positive instance, n is a negative instance and $cos(a,b)$ is the cosine similarity. The contrastive triplet ranking loss encourages to learn that positive (image, text) embedding pairs to be closer at least margin m apart than negative (image, text) embedding pairs in a joint metric space.

IV. Experiments

We validate our cross-modal retrieval model on two datasets, R-CLEVR and MS-COCO and weekly-supervised visual grounding of phrases on R-CELVR. R-CLEVR is a novel dataset we have compiled to evaluate spatial relational reasoning for cross-modal retrieval, since existing datasets such as MS-COCO contain few instances that require spatial reasoning, as they mostly focus on high-level semantics.

A. *Datasets*

R-CLEVR In order to analyze and demonstrate our model, we design a dataset for cross-modal retrieval that requires to reason spatial relationships between objects, which we name as Relational-CLEVR (R-CLEVR). We modify image and sentence generation codes provided from CLEVR [19]. R-CLEVR is composed of 93,000 images where 90,000 images are used for training and 3,000 images are reserved for validation. Each image is annotated by three captions which describes two different objects with their spatial relationships and the image in the validation set has additional three phrases for phrase grounding tasks. We divide the dataset into three subsets, to vary the difficulty of spatial reasoning by including different number of the same objects that share one of the four properties: shape, color, material, and size as shown in Figure 4.

MS-COCO MS-COCO dataset [22] consists of a total of 123,287 images each of which comes with five annotations

Easy	Caption Retrieval					Image Retrieval				
Model	R@1	R@5	R@10	Med	Mean	R@1	R@5	R@10	Med	Mean
CHAIN-VSE [35]	85.8	97.7	99.1	1.0	1.4	82.5	98.1	99.5	1.0	1.4
VSE++ [14]	92.8	99.2	99.8	1.0	1.2	91.0	99.6	100.0	1.0	1.1
Engilberge [13]	98.3	100.0	100.0	1.0	1.0	97.1	100.0	100.0	1.0	1.0
ResNet + OPM (Ours)	**99.8**	**100.0**	**100.0**	1.0	1.0	**99.6**	**100.0**	**100.0**	1.0	1.0
Middle	Caption Retrieval					Image Retrieval				
Model	R@1	R@5	R@10	Med	Mean	R@1	R@5	R@10	Med	Mean
CHAIN-VSE [35]	80.9	96.9	98.9	1.0	1.7	78.6	98.6	99.6	1.0	1.5
VSE++ [14]	90.5	99.1	99.7	1.0	1.2	88.5	99.5	99.8	1.0	1.3
Engilberge [13]	94.6	100.0	100.0	1.0	1.0	92.9	100.0	100.0	1.0	1.0
ResNet + OPM (Ours)	**98.2**	**100.0**	**100.0**	1.0	1.1	**97.5**	**100.0**	**100.0**	1.0	1.1
Hard	Caption Retrieval					Image Retrieval				
Model	R@1	R@5	R@10	Med	Mean	R@1	R@5	R@10	Med	Mean
CHAIN-VSE [35]	41.2	73.9	88.8	2.0	4.3	45.5	86.1	98.2	2.0	2.8
VSE++ [14]	52.8	85.5	95.7	1.0	2.9	53.1	91.7	99.1	1.0	2.3
Engilberge [13]	54.2	84.2	95.6	1.0	2.9	53.5	89.9	99.7	1.0	2.4
CNN + RN [27]	49.6	80.8	93.7	2.0	3.4	46.7	86.0	98.7	2.0	2.8
CNN + OPM (Ours)	56.0	83.3	94.8	1.0	3.0	52.5	89.3	99.5	1.0	2.5
ResNet + OPM (Ours)	**66.7**	**92.6**	**98.7**	1.0	2.1	**65.7**	**96.3**	**100.0**	1.0	1.8

Table 2. Results of cross-modal retrieval experiments on the three difficulty levels of R-CLEVR.

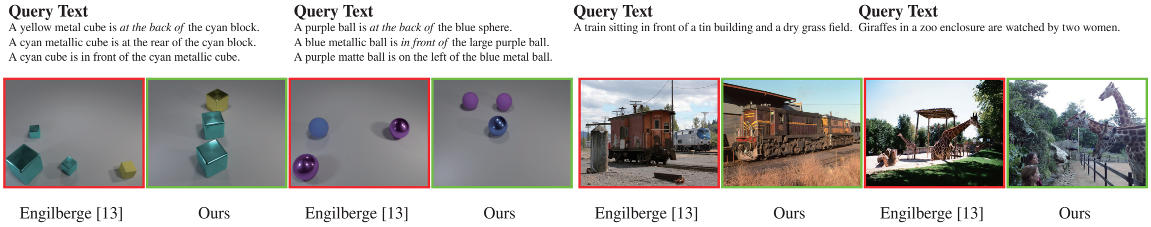
collected from different annotators. For the cross-modal retrieval task, we used the training/validation/test split on the MS-COCO proposed Karpathy and Li [20].

B. Cross-modal retrieval on R-CLEVR

Evaluation metric For evaluating the performance of the proposed model, we conducted experiment on cross-modal retrieval tasks with R-CLEVR dataset. The goal of caption retrieval is to find the corresponding caption when a query image is given and vice versa. We used the same evaluation metric of MS-COCO retrieval task in Vendrov [33]. Recall at k (R@k) of the caption retrieval computes the percentage of containing the correct caption among retrieved first k captions and vice versa. The comparison results of other visual-semantic retrieval methods are denoted in Table 2.

Experimental analysis As shown in Table 2, we observed that our method outperformed other methods for all three different classes of R-CLEVR. For Hard type dataset, compared to other methods, our model resulted at least 12.2%, 8.4%, 3.1% higher values for R@1, R@5, R@10 for the caption retrieval and 12.2%, 6.4%, 0.1% for the image retrieval task. For RN, due to the limitation of scalability, we adapted a backbone network as four layer vanilla convolutional layers which trained for VQA tasks on CLEVR and fine-tuned it for cross-modal retrieval tasks, denoted as CNN + RN in Table 2). Even when using a small convolution network as a backbone, our module (CNN + OPM), outperformed CNN + RN by a large margin and had comparable performance with other methods which employ high dimensional features of ResNet or VGG Net. Considering it is difficult to distinguish duplicated objects only with vision attributes such as color or shape of an object, the results show that OPM definitely supports the model to learn spatial relationships and achieve the state-of-the-art accuracy for the retrieval tasks on the R-CLEVR. The qualitative results of R@1 for the image retrieval tasks are shown in Figure 5. For example, the first sentence in ground truth text set says 'a yellow cube is at the back of the cyan block'. Engilberge finds image which yellow cube is positioned next to cyan blocks in it, while our model properly reflects the text. Figure 6 is qualitative result of R@1 for caption retrieval task of our model and Engilberge Similar to Figure 5, our model retrieves the sentence describing objects' relative positions in image space.

Self-comparison according to components of OPM We conducted an ablation study to demonstrate the role of each component of our proposed model. The results are reported on Table 3. sPool of the first column removed OPM from suggested structure, adapting spatial selective pooling [10] to visual feature map generated from ResNet convolutional part and projected a visual feature to joint space with projection layer. OPG and RC each means object phase group and relative coordinate component and sPool was added one by one to perform ablation studies. We can observed that even with the application of OPG or RC only, the R@1 accuracy is increased by more than 10% for both caption and image retrieval. Furthermore, we measured the accuracy according to the type of pooling. When using global max pooling, the accuracy for R@1 was 1.6% higher than the accuracy when using sPool. sPool localizes interesting objects selecting values of relevant features extracted from corresponding spatial positions[10, 11]. However, the encoded features from OPM are no more simple spatial information at the exact location but the features in relation to all other features. Thus, we believe that sPool cannot bring additional benefits to the features once the relationships with others are considered, but maxPool which is channel-wise hard attention shows better performance.



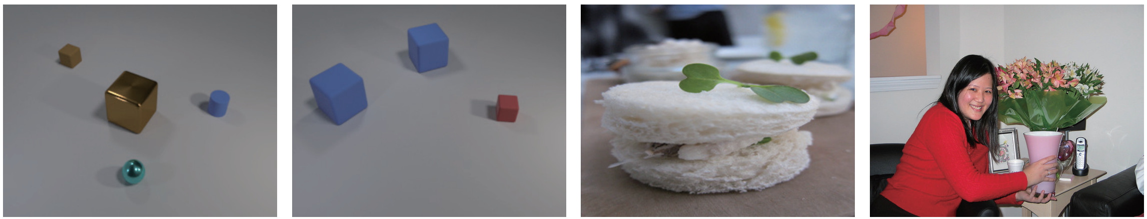
Query Text
A yellow metal cube is *at the back of* the cyan block.
A cyan metallic cube is at the rear of the cyan block.
A cyan cube is in front of the cyan metallic cube.

Query Text
A purple ball is *at the back of* the blue sphere.
A blue metallic ball is *in front of* the large purple ball.
A purple matte ball is on the left of the blue metal ball.

Query Text
A train sitting in front of a tin building and a dry grass field.

Query Text
Giraffes in a zoo enclosure are watched by two women.

Fig. 5. Image Retrieval (R@1) on R-CLEVR and MS-COCO. The images are retrieved from given query texts by models. The correct cases are bordered with green boxes and the fail cases are bordered with red boxes. The left four images are cases where compared model found inappropriate image, but our model found the proper ones.



Engilberge [13] A blue cylinder is *on the left back of* the tiny cyan ball. A brown block is *on the right front of* the large blue matte cylinder. A small cyan shiny ball is in front of the brown cube.

Ours A tiny brown rubber block is *on the left rear of* the cyan shiny ball. A large brown metallic cube is *on the left of* the blue cylinder. A small blue cylinder is *on the right rear of* the cyan metallic ball.

Engilberge [13] A red rubber block is *on the left back of* the blue cube. A blue cube is *on the right front of* the red rubber cube. A red rubber cube is on the right of the *red* cube.

Ours A tiny red matte block is *on the right of* the blue cube. A blue cube is *on the left back of* the tiny red block. A large blue rubber cube is *on the left front of* the blue cube.

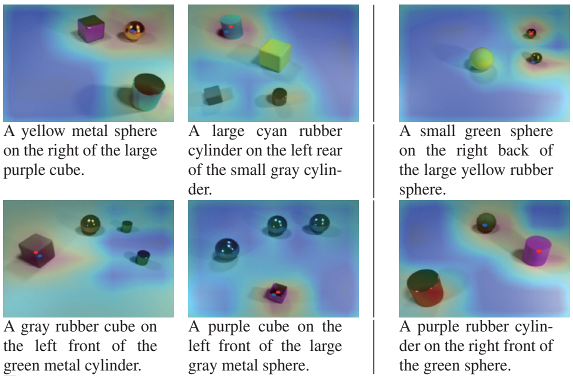
Engilberge [13] A sandwich is laying on its wrapper ready to be eaten.

Ours Two pieces of bread with a leafy green on top of it.

Engilberge [13]A woman contemplating something with a coffee up.

Ours A woman in a red shirt holding a pink vase.

Fig. 6. Caption Retreival (R@1) on R-CLEVR. Phrase is colored as red when the description of retrieved text is inaccurate, while it is denoted as green for the right description.



A yellow metal sphere on the right of the large purple cube.

A large cyan rubber cylinder on the left rear of the small gray cylinder.

A small green sphere on the right back of the large yellow rubber sphere.

A gray rubber cube on the left front of the green metal cylinder.

A purple cube on the left front of the large gray metal sphere.

A purple rubber cylinder on the right front of the green sphere.

Fig. 7. PCK examples. The red dot marks the center point of ground-truth object of the query phrase and the blue dot is the point of the highest prediction value by our system for the same phrase. The left four present correct cases and the right two shows fail cases.

C. *Cross-modal retrieval on MS-COCO*

In this subsection, we verified our approach on the general benchmark dataset for cross-modal retrieval, MS-COCO. Comparing with recent other methods in Table 4, our model showed comparable results with the state of the art, Engilberge . In other words, it was shown to achieve the best and the second best on each caption and image retrieval test. Since the sentences in MS-COCO are less concerned with the relationships between objects, our model showed moderate improvement with this dataset. When we analyze results qualitatively as shown in 5 and 6, we can observe that our method retrieves proper ones than the comparison method for the case that requires the model to consider the spatial relationships between objects. We evaluated our model with a larger image size (330 × 330) to achieve our best performance as shown in Engilberge .

D. *Weekly-supervised visual grounding of phrases*

In this task, we aimed to localize the image content described by a given textual phrase which includes spatial relationship information. While making bounding box annotation requires time-consuming and expensive work, our approach can ground an object in the form of an attention mask without those kind of object-level annotation.

Following Engilberge , we reused parameters of our model trained for the semantic-visual embedding to the phrase grounding. First, we obtained response values by computing correlation between each visual-relational feature from OPM and a text embedding. Then we yielded a heatmap with the response values to overlay with the input image as shown in Figure 7. The red point is the center coordinate of the object depicted by the given phrase. The blue one is the prediction point which has the highest value in the heatmap, considered as the most related region with the phrase. Since we predicted outputs pixel-by-pixel instead of bounding boxes, we used a Percentage of Correct Keypoints (PCK) [6, 37] as evaluation metric. PCK is percentage of correct cases that the prediction point(the blue one) is closer than T pixels to the ground-truth (in short, P@T). For three type data classes, the results of P@10 and P@20 was reported on Table 5.

Model	Caption Retrieval					Image Retrieval				
	R@1	R@5	R@10	Med	Mean	R@1	R@5	R@10	Med	Mean
sPool	55.1	85.0	95.0	1.0	2.9	53.1	89.8	99.6	1.0	2.4
sPool + OPG	65.1	92.6	98.3	1.0	2.1	64.4	96.3	99.8	1.0	1.8
sPool + RC	65.5	90.8	98.5	1.0	2.2	64.6	96.3	99.7	1.0	1.8
sPool + RC + OPG	65.1	91.9	98.7	1.0	2.2	64.7	95.9	99.7	1.0	1.9
maxPool + RC + OPG	**66.7**	**92.6**	**98.7**	1.0	2.1	**65.7**	**96.3**	**100.0**	1.0	1.8

Table 3. Ablation study on R-CLEVR dataset. Object Phase Group (OPG) and relative coordinate (RC) component are helpful for interference.

Model	Caption Retrieval				Image Retrieval			
	R@1	R@5	R@10	Med	R@1	R@5	R@10	Med
Embedding network [34]	50.4	79.3	89.4	-	39.8	75.3	86.8	-
2-Way Net [12]	55.8	75.2	-	-	39.7	63.3	-	-
LayerNorm [2]	48.5	80.6	89.8	5.1	38.9	74.3	86.3	7.6
CHAIN-VSE [35]	61.2	89.5	95.8	1.0	46.6	81.9	90.92	2.0
VSE++ [14]	64.6	90.0	95.7	1.0	52.0	84.3	92.0	1.0
Engilberge . [13] (400×400)	69.8	91.9	96.6	1.0	**55.8**	**86.9**	**94.0**	1.0
Ours	68.5	91.4	96.2	1.0	52.2	83.8	92.3	1.0
Ours (330×330)	**71.6**	**92.0**	**96.8**	1.0	53.4	85.9	93.6	1.0

Table 4. Cross-modal retrieval results on MS-COCO dataset. We marked results of state of the art as BOLD. We found that resized image (330x330) shows better performance than our original evaluation.

Model	Easy		Middle		Hard	
	P@10	P@20	P@10	P@20	P@10	P@20
Engilberge [13]	10.3	18.1	9.9	17.6	9.7	16.0
Ours	**16.7**	**26.3**	**17.6**	**26.6**	**16.0**	**25.8**

Table 5. PCK results. Our architecture outperforms Engilberge for phrasing grounding task on every task and evaluation criterion.

Our model with OPM showed better performance at three types of task by at least 6.4% point than Engilberge without relation-inferring module, for both p@10 and p@20. As denoted in Figure 6, 1st, 2nd row images are successful cases. In the first image of them, there is one purple cube, yellow metal sphere and a cyan cylinder in front of them. The predicted point(blue dot) is pointing the same object as ground truth point (red dot) and attention mask is covering the object indicated by phrase (*yellow sphere*). The 3rd row images are fail cases. It can be inferred that the network confused the modifier and the modified phrase in the first two cases and for third image, the pointed object and correct object share common properties (color and shape) except position. This experiment shows potential in solving weekly supervised groundings of phrases containing relative information without object-level annotation.

V. Conclusion

In this paper, we propose a novel semantic-visual embedding network, Object Phase Module (OPM) which entails spatial relationships among objects in an image, and a novel dataset R-CLEVR that contains multiple objects with spatial relationships among the objects described in the captions, that requires to do spatial reasoning. Throughout extensive experiments on cross-modal retrieval and phrase grounding, we have demonstrated that our OPM outperforms state-of-the-art visual-semantic embedding methods for both tasks on R-CLEVR. Further, it shows comparable performance with state-of-the-art methods on MS-COCO, in which a large portion of the examples do not require spatial reasoning. Further qualitative analysis of the allocated attentions suggest that the previous methods mainly focus on objects' properties, but OPM considers both the object properties and spatial relationships. As future work, we plan to extend the problem and the model to consider spatio-temporal relationships between objects in videos.

References

[1] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*, 2015.

[2] Lei Jimmy Ba, Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016.

[3] Abdeslam Boularias, Felix Duvallet, Jean Oh, and Anthony Stentz. Learning qualitative spatial relations for robotic navigation. In *Proceedings of the International Joint Conference on Artificial Intelligence (ICJAI)*, 2016.

[4] Kan Chen, Jiyang Gao, and Ram Nevatia. Knowledge aided consistency for weakly supervised phrase grounding. In *CVPR*, 2018.

[5] Kevin Chen, Christopher B Choy, Manolis Savva, Angel X Chang, Thomas Funkhouser, and Silvio Savarese. Text2shape: Generating shapes from natural language by learning joint embeddings. *arXiv preprint arXiv:1803.08495*, 2018.

[6] Christopher B. Choy, JunYoung Gwak, Silvio Savarese, and Manmohan Chandraker. Universal correspondence network. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2414–2422, 2016.

[7] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Gated feedback recurrent neural networks. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 2067–2075. JMLR.org, 2015.

[8] Volkan Cirik, Taylor Berg-Kirkpatrick, and Louis-Philippe Morency. Using syntax to ground referring expressions in natural images. *AAAI*, 2018.

[9] 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.

[10] T. Durand, N. Thome, and M. Cord. Weldon: Weakly supervised learning of deep convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4743–4752, June 2016.

[11] Thibaut Durand, Taylor Mordan, Nicolas Thome, and Matthieu Cord. WILDCAT: Weakly Supervised Learning of Deep ConvNets for Image Classification, Pointwise Localization and Segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[12] Aviv Eisenschtat and Lior Wolf. Linking image and text with 2-way nets. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[13] Martin Engilberge, Louis Chevallier, Patrick Pérez, and Matthieu Cord. Finding beans in burgers: Deep semantic-visual embedding with localization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[14] Fartash Faghri, David J. Fleet, Jamie Kiros, and Sanja Fidler. VSE++: improving visual-semantic embeddings with hard negatives. In *British Machine Vision Conference 2018, BMVC*, 2018.

[15] 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.

[16] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[17] Jeonghee Kim Hyeonseob Nam, Jung-Woo Ha. Dual attention networks for multimodal reasoning and matching. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[18] Michael Janner, Karthik Narasimhan, and Regina Barzilay. Representation learning for grounded spatial reasoning. *Transactions of the Association for Computational Linguistics*, 6:49–61, 2018.

[19] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross B. Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[20] Andrej Karpathy and Fei-Fei Li. Deep visual-semantic alignments for generating image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, abs/1412.2306, 2017.

[21] Ryan Kiros, Ruslan Salakhutdinov, and Richard S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *CoRR*, abs/1411.2539, 2014.

[22] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[23] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems (NIPS)*, pages 3111–3119. 2013.

[24] Michael Bernstein Li Fei-Fei Ranjay Krishna, Ines Chami. Referring relationships. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[25] Scott Reed, Zeynep Akata, Bernt Schiele, and Honglak Lee. Learning deep representations of fine-grained visual descriptions. In *IEEE Computer Vision and Pattern Recognition*, 2016.

[26] Anna Rohrbach, Marcus Rohrbach, Ronghang Hu, Trevor Darrell, and Bernt Schiele. Grounding of textual phrases in images by reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[27] Adam Santoro, David Raposo, David G. T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Tim Lillicrap. A simple neural network module for relational reasoning. pages 4974–4983, 2017.

[28] Mohit Shridhar and David Hsu. Grounding spatio-semantic referring expressions for human-robot interaction. *CoRR*, abs/1707.05720, 2017.

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[30] Richard Socher, Andrej Karpathy, Quoc V. Le, Christopher D. Manning, and Andrew Y. Ng. Grounded compositional semantics for finding and describing images with sentences. *TACL*, 2:207–218, 2014.

[31] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[32] Sida I. Wang Hui Dai Tao Lei, Yu Zhang and Yoav Artzi. Simple recurrent units for highly parallelizable recurrence. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

[33] Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. Order-embeddings of images and language. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

[34] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning two-branch neural networks for image-text matching tasks. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 2018.

[35] Jônatas Wehrmann and Rodrigo C. Barros. Bidirectional retrieval made simple. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[36] Fanyi Xiao, Leonid Sigal, and Yong Jae Lee. Weakly-supervised visual grounding of phrases with linguistic structures. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[37] Yi Yang and Deva Ramanan. Articulated human detection with flexible mixtures of parts. *The IEEE Trans. Pattern Anal. Mach. Intell.*, 2013.

[38] Hanwang Zhang, Yulei Niu, and Shih-Fu Chang. Grounding referring expressions in images by variational context. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

COVID-19 Spread Regimes with Temporally Calibrated SIR Model

Hee-Sun Bae[1], Il-Chul Moon[1] and Gi-Woon Kim[2]

[1] Department of Industrial and Systems Engineering, KAIST, South Korea {cat2507,icmoon}@ kaist.ac.kr
[2] Depeartment of Emergency Medicine, Soon Chun Hyang University Hospital, South Korea

Abstract

After COVID-19 outbreaks, media covers the daily report of COVID-19 spread, and authorities estimate expected spread in short term. This information relies primarily on Susceptible-Infectious-Recovery (SIR) Model, and modelers often fit the parameters of the model over entire spread process or over a moving window period of fixed duration. Although COVID-19 has its own spread factors, these coefficients are mitigated by various response efforts depending on countries and periods. Therefore, parameters of SIR model must be optimized by considering such societal background and temporal clusters, or *regime*. This report captures the spread regime by applying regime detection to JHU COVID-19 dataset. We detect the spread *regime* by hidden Markov model, and we fit the dataset for each regime by l-BFGS optimizer. We found that some countries encountered the second wave while most countries still face the first one. We also provide the R_0 coefficient, which has different values depending on the country's societal and temporal contexts.

Keywords— *COVID-19, Hidden Markov Model, Simulation Calibration, SIR Model*

I. Introduction

After COVID-19 outbreaks, media covers daily reports of the virus spread[9], and authorities estimate spread trend in the near future. The estimates are often extrapolated from Susceptible-Infectious-Recovery (SIR) model[6], which is the simplest yet quite effective model in such estimation. The inference on the model parameters of the SIR model becomes a non-convex optimization problem[4], so there is no closed-form solution in the parameter inference. Hence, modelers utilize l-BFGS[7] or MCMC[2] through iterative optimum searches to infer it.

This practice can lead to misguided parameter searches if the SIR model is used to estimate the entire period of the spread. COVID-19 has its own fatality rate and the route of transmission. The diseases with the airborne transmission tend to have high basic reproduction numbers(R_0), i.e. Measles with $5 \leq R_0 \leq 18$ [1]If the fatality rate is high, R_0 tends to be low, i.e. MERS-CoV with $R_0 \leq 1$[8]. This inherent spread factor is limited because public health authorities enforce various policies, such as social distancing and tracking-and-testing. Therefore, in spite of being infected by the same virus, the spread shows different patterns as these policies are implemented at different levels depending on countries and periods.

This report analyzes the COVID-19 spread by inferring the parameters of the SIR model by countries and by temporal clusters, or *regime*. This report captures the spread regime by fitting the JHU COVID-19 dataset (JHU) with detected regimes from a hidden Markov model[3], and we fit the SIR model for a specific pair of regime and country by the l-BFGS optimizer. Also, we provide the R_0 coefficients, which are different across countries' societal circumstances and periods.

II. Method

A. SIR Model Variation

The original SIR Model does not have a compartment of *Death*[6], so we added another transition from *Infectious* to *Death* because of the demand on the analyses of fatality. Each variable is time-dependent, so we have four compartment variables of S_t, I_t, R_t, and D_t ($t =$ day), each variable representing population of the susceptible, the infectious, the recovered and the dead. We hypothesized $S_t + I_t + R_t + D_t = N$, the total population, meaning that there are no other causes of death.

Our model requires three parameters to infer. β is the number of people infected from a single patient. γ is the transition rate of infected patients to either *Recovery* or *Death*. α is the fatality rate from the infected patients. In addition, our report mainly focuses on $R_0 = \beta/\gamma$ [5]. We specify SIRD model as below.

$$\begin{aligned} \frac{dS}{dt} &= -\frac{\beta IS}{N}, & \frac{dI}{dt} &= \frac{\beta IS}{N} - \gamma I, \\ \frac{dR}{dt} &= \gamma(1-\alpha)I, & \frac{dD}{dt} &= \gamma\alpha I \end{aligned} \tag{1}$$

B. *Calibration Procedure*

We specified the calibration procedure as below. It assumes that we have accurate reports on compartmental population over-time with accurate total population.

1. Start with the regime number of $K = 1$
2. Optimize parameters of α, β, γ of the model.
3. Infer the parameters of hidden Markov model by setting observation to be the deviation between model estimates and reported population. The inference algorithm follows the Gibbs sampling, which is a variant of Monte-Carlo Markov-Chain (MCMC). We assumed emission probability distribution to be Gaussian distribution and state transition probability distribution to be multinomial distribution.
    (a) For each regime from the hidden Markov Model, optimize α, β, and γ with respect to minimizing the RMSE of the data. We used l-BFGS optimizer, a gradient-based optimizer.
4. Calculate RMSE of the infected, the recovery, and the dead part for the entire period
5. Repeat if RMSE is improved over 5% with $K = K + 1$.

Since we used hidden Markov Model to detect regime, initial assignment affected a lot to finally detected regime, and wrongly assigned case returned wrongly detected regimes. Therefore, 30 times replications are experimented to exclude results from wrongly detected regime.

C. *Dataset*

We used the COVID-19 dataset from Johns Hopkins Coronavirus Resource Center. Additionally, we set the initial susceptible population data with World Bank Open Data.

Though we analyzed more countries, we mainly focus on 6 countries, China, South Korea, US, Italy, Iran and Brazil for this report.

III. Result

A. *Spread Regime Detection*

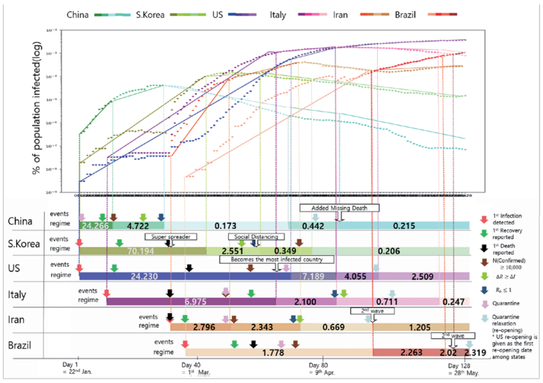


Fig. 1. Six countries were selected by considering importance and location. Color-coding is applied to distinguish the spread regimes and the countries. The color intensity is adjusted to reflect each regime's R_0 value. (Upper) Percentage of infected population in log-scale. Dots are reported numbers in the JHU dataset. Lines are estimates from the calibrated SIRD model by spread regimes. (Lower) Regimes and events of analyzed countries. Regimes are identified from hidden Markov models to minimize the estimation error. R_0 is the calibration result of the SIRD model. Events are gathered from the news outlets by corroborating several sources. The relaxation of Quarantine is given as the first day of the policy implementation.

Figure 1 shows the estimated and the reported log-percentage of infected patients in total population. Countries are facing different spread regimes. For instance, South Korea has passed four regimes and is an interesting case of reducing R_0 because it had an extremely high R_0, 70.194, because of a super spreader during the early spread regime. This high R_0 can be interpreted from multiple perspectives. Whereas R_0 can be high if there is a super spreader, R_0 can also be high if S. Korea was able to track and confirm the infected as many as possible. The reported dataset cannot indicate the actual number of the infected individuals, so we cannot calculate 'the true R_0' unless the complete tracking and confirmation was achieved. However, the highest R_0 indicates such aggressiveness in the track-and-test policy implementation.

Iran shows a clear indication of the second wave from the perspective of the spread regime. Iran's R_0 decreased from 2.796 to 0.669 through three regimes, but now Iran faces the fourth regime, increasing R_0 by 1.205. This means that the spread was contained in the third regime with R_0 below one, but the infected number is currently growing. .

B. *Calibration of SIRD Model*

Table 1. R^2 value regarding concept of regime

R^2	Without Regime			With Regime		
Countries	I	R	D	I	R	D
China	.0	.602	.623	.933	.981	.679
S.Korea	.0	.435	.317	.932	.991	.993
US	.338	.694	.443	.997	.965	.980
Italy	.015	.696	.303	.964	.998	.994
Iran	.024	.384	.159	.963	.999	.976
Brazil	.767	.767	.720	.990	.989	.996

Table 1 shows the R^2 value of our model, which explains the fitness of our model to the reported dataset: the number of infectious, recovered and dead population by countries. Table 1 compare R^2 with regimes and without regimes to emphasize the necessity of dynamic calibrations. We denote that nearly every compartment of populations in every country has $R^2 \geq 0.90$. The result is not replicated 1) because we show there is only stochastic effect from the inference of the hidden Markov model and 2) because we show the best regime finding method in the below.

Since the parameter inference of the hidden Markov model is sensitive to the initial setup, we replicated 30 different cases to select the optimal regime detection with increasing K. We follow the Occam's Razor by setting the lowest K when the RMSE is converged without improvement above 5%.

IV. Conclusion

Our result suggests that in some countries, re-opening policies is implemented without the reduction of R_0 less than one. Our regime detection also indicates the problem of early policy implementation in Iran, as Iran shows the obvious second wave of the growing infected population. We note that other countries also have faced such increment, for instance, South Korea after re-opening. However, our model indicates that the second wave of South Korea is not as significant as that of Iran because our model did not separate another regime for the second wave in South Korea. From the perspective of a policy recommendation, our results suggest that the daily estimate of R_0 may be too sensitive to be a reliable indicator. The daily peak of R_0 may be high, but such daily peaks may not persist sufficiently to suggest a meaningful period of spread that would warrant a change in regime as was the case of South Korea. Additionally, the trend of daily estimates is a sequence of R_0 values, and recommendations are often made by its trend. The spread regime identification may specify the required meaningful duration of the trend by considering the fluctuation of R_0 and its trend.

V. Limitations and Further studies

Our report relies on stochastic models and inference algorithms, so the analyzed data is derived from a set of sub-optimal parameters. Also, while we report R^2 value above 90% except one case, we utilize the l-BFGS algorithm that may not result in the optimal parameters for the SIRD model. Also, the SIRD model can be further complicated to reflect the actual situation. One fundamental limitation comes from the fact that we have to rely on the *reported infected population*, which is not the true counting of patients. Since the number of infected population data is the number of *reported* infected population, different testing strategies from countries can cause differences in the reported infectivity rate. Therefore, our cross-national analyses is less accurate than our domestic analyses of a single country. We also added the qualitative analyses on policy events corroborating more than two media sources, but these events can be interpreted differently depending on the social, cultural, and political contexts.

References

[1] Roy M Anderson and Robert M May. *Infectious diseases of humans: dynamics and control*. Oxford university press, 1992.

[2] Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan. An introduction to mcmc for machine learning. *Machine learning*, 50(1-2):5–43, 2003.

[3] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[4] Giuseppe C Calafiore, Carlo Novara, and Corrado Possieri. A modified sir model for the covid-19 contagion in italy. *arXiv preprint arXiv:2003.14391*, 2020.

[5] Odo Diekmann, Johan Andre Peter Heesterbeek, and Johan AJ Metz. On the definition and the computation of the basic reproduction ratio r 0 in models for infectious diseases in heterogeneous populations. *Journal of mathematical biology*, 28(4):365–382, 1990.

[6] W. O. Kermack and A. G. Mckendrick. A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character (1905-1934)*, 115(772):700–721, 1927.

[7] Jorge Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of computation*, 35(151):773–782, 1980.

[8] WHO. Who mers-cov global summary and risk assessment. *WHO/MERS/RA/16.1*, December 5th, 2016.

[9] Geneva World Health Organization. Coronavirus disease 2019 (covid-19). situation report 1-138. URL: https://www.who.int/emergencies/diseases/novel-coronavirus-2019/situation-reports.

Graph Knowledge Integration into a Text Encoder

Kibeom Kim[1], Taehee Kim[2] and Jaegul Choo[2]

[1] Korea University, kkbyg@korea.ac.kr
[2] Korea Advanced Institute of Science and Technology, {taeheekim, jchoo}@kaist.ac.kr

Abstract

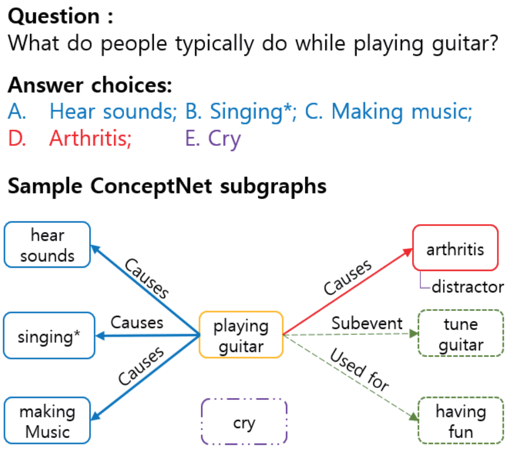
Machine reading comprehension(MRC) is a task in which relevant document is given and answers to questions related to the document. Unlike MRC, commonsense question answering is considered a more difficult task because it has to answer the question without a given context. In this paper, we propose a novel framework that effectively integrates information from the external knowledge graph (e.g. ConceptNet) into a neural model. We first extract relevant paths from external knowledge graphs, in order to enhance the model performance. After that, the extracted paths are encoded by adopting knowledge graph embedding (e.g. RotatE), and integrated into the text encoder (e.g. BERT) to deal with multi-modality. Experimenting on the CommonsenseQA dataset, our proposed model significantly improved the baselines.

Keywords— *Deep Learning, Question Answering, Commonsense Reasoning*

I. Introduction

With the recent development of deep learning, many areas of natural language processing(NLP) have achieved rapid growth especially in MRC tasks such as SQuAD [12] and RACE [5] which do not require external knowledge. Recently, NLP researchers are involved in more difficult task such as commonsense reasoning which requires external knowledge to answer the question. one of the dataset which demands commonsense reasoning is CommonsenseQA dataset [17] that require an understanding of common sense without relevant document or context(i.e., a passage in MRC datasets).

CommonsenseQA [17] was created from a knowledge graph called ConceptNet [15]. ConceptNet [15] is a knowledge graph consisting of nodes (i.e. 'concept') and edges (i.e. 'relation') related to real-world knowledge. CommonsenseQA [17] consists of five multiple-choice questions, four of which are composed of target concepts that have the same relation from one source concept extracted from ConceptNet [15]. In particular, three of the answer choices are meaningfully similar, making the problem more difficult.



Question :
What do people typically do while playing guitar?

Answer choices:
A. Hear sounds; B. Singing*; C. Making music;
D. Arthritis; E. Cry

Sample ConceptNet subgraphs

Fig. 1. Sample ConceptNet subgraphs for CommonsenseQA question.

For example, the source concept 'playing guitar' in Figure 1 is connected to three target concepts('singing', 'making music' and 'hear sound' in Figure 1) by 'causes' relation in ConceptNet [15]. The fourth answer choice('arthritis' in Figure 1) is the target concept extracted from the ConceptNet [15], which also has 'causes' relation from source concept('playing guitar') as same as the three target concepts('singing', 'making music' and 'hear sound' in Figure 1), but has a different contextual meaning. Finally, the fifth answer choice may not have been extracted from ConceptNet [15], but annotators deciding and adding one themselves. Due to this method of data generation, the CommonsenseQA [17] is considered a more difficult task for the neural network model than the conventional MRC tasks.

On the other hand, BERT [2] and T5 [11] are pre-trained with extremely large corpus through self-supervised learning. Self-supervised learning dramatically reduced the cost of data labeling because it makes labels on its own from training data. In addition, the model performance was dramatically improved compared to the unidirectional language model by randomly masking the input data during the pre-training process and make them predict through the bidirectional representation. It has been recently reported that, after pre-training, knowledge becomes implicitly embedded in the parameters of the model [13]. Through the enormous amount of data and large models(i.e. large number of model parameters) with the T5 [11] model, it achieved significant performance improvement in OpenQA(e.g. TriviaQA [3]) task as well without having access to external knowledge. However, this approach [13] requires a huge number of parameters (e.g. 11 billion parameters), which demand enormous computing resources and data. Furthermore, a way of simply relying on pre-training, it is hard to learn knowledge beyond what is in the learned pre-trained corpus. To solve this problem, injecting information using external knowledge graphs has recently emerged as a popular research topic. In this paper, we propose a framework that can effectively integrate knowledge graph in BERT-based [2, 8, 6] text encoder.

II. Integrating Knowledge Graph to Text Encoder

In this section, we first outline the problem setting of CommonsenseQA [17], and introduce the workflow of our framework.

There can be many ways to put external graph knowledge in a typical text encoder. We first do not use the ConceptNet [15] graph as it is, but prune it into the appropriate size. To integrate textual embedding and graphical embedding into our neural network model, we need to create another embedding for graph knowledge. We applied the RotatE [16] knowledge graph embedding method, which embeds the graph of ConceptNet. Next, we extract concepts and relation(i.e. triple) by our knowledge extraction method that can help solving CommonsenseQA [17] questions. The extracted graph knowledge (i.e. triple) is expressed in a pre-trained RotatE [16] graph embedding and incorporated and injected into the model. Our method has exceeded existing baseline models [7, 9]. and achieved considerable performance.

A. *Graph Pruning*

The connection between each node in ConceptNet [15] is represented by a unit called triple, which means that it is connected in the form of node-edge-node (i.e. concept - relation - concept). For example, ’(playing guitars, causes, singing)’ and ’(playing guitars, used for, having fun)’ are triples in Figure 1. ConceptNet [15] was created by many annotators, and is a huge graph containing 32 million triples and 36 relations (e.g. IsA, UsedFor, Causes, RelatedTo, etc.). In addition, we found that about half of the nodes were one degree or less. It means that about half the nodes were connected to only one node. Because training graph embedding from these sparse graphs can lead to poor performance, we pruned the graphs to leave the necessary nodes.

We prune ConceptNet [15] using the following method to use only the necessary graph and to increase the performance of the graph embedding.

1. The word contained in the concept can be more than one, and we extract only the concept consisting of not more than four subwords.
2. CommonsenseQA [17] uses 22 relations from ConceptNet, while we use similar meanings. Consolidate relation to form a new graph using only a total of 15 relation.

B. *Knowledge Extraction*

So far, the graph itself was scaled down through pruning the graph, and now the important thing is how to extract subgraph and put it as an input. We assumed that in the question of answering questions, there would be one triple that would help a model to find answers. In CommonsenseQA [17], a question concept is given separately from the question, answer questions. We assume that if there is a triple from this question concept to answer choice, this would be the key information. So, for each answer choice, we explored whether it is connected to the question concept, and added the corresponding triple as the input of the model.

C. *Model Architecture*

Among the various variants of GNN, GCN [4] expresses nodes as vectors, combines information from adjacent nodes, and updates their own node vector. KagNet [7] and Graph-Based Reasoning [9] tried to solve the CommonsenseQA [17] dataset by applying the plain GCN [4]. The reason why plain GCN [4] was used in both papers is that, as seen in existing work [10, 20], relational GCNs [14] are mainly over-parameterized and do not represent knowledge graph effectively. The plain GCN [4] does not fully reflect the graph structure, because it explicitly means that it does not take into account the edge feature(i.e. relation) and direction of the graph.

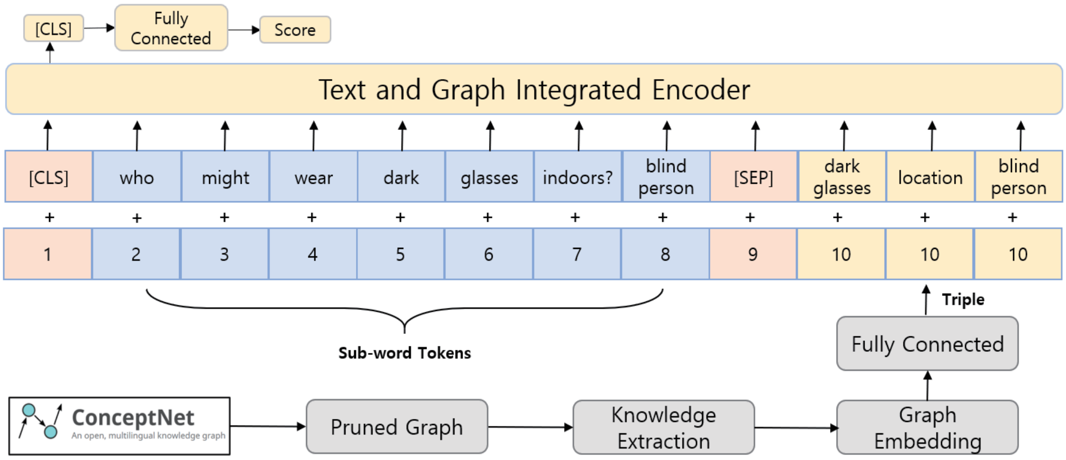


Fig. 2. Illustration of the Integrated Text and graph architecture. The example of the CommonsenseQA is 'Who might wear dark glasses?', and the answer choices are {A. blind person*; B. glove box; C. movie studio; D. ray charles; E. glove compartment}. Among them, 'blind person' marked with asterisk (*) is the correct answer choice. The format of the input is [CLS] question + answer choice1 [SEP] related triple. The blue boxed token is the text encoder part, and the yellow boxed token represents the graph encoder part.

Related to these problems, there have been recent attempts to incorporate graph structures with transformers without using GNNs.(e.g. COMET [1], Graph-BERT [19]) As shown in the above papers, the Transformers [18] can reflect graph knowledge well enough. Just as GCN [4] expresses the adjacent relationship between nodes, the Transformer [18] can represent the relation between each adjacent node with a self-attention and attention mask. Inspired by these work, we propose a new model framework that injects graph knowledge into the transformer based text encoder [2, 8, 6].

The transformer [18] based models (e.g. ALBERT [6], RoBERTa [8]) have a much higher number of model parameters compared to the previous models. The reason why these models can train well even if there are many parameters is because they are pre-trained with extremely large corpus. However, the models may not be sufficiently trained because the data extracted from the knowledge graph is much smaller than the pre-train corpus used in the transformer based model. Therefore, we have adopted ALBERT [6] as the backbone model, a model that has a relatively small number of parameters and allows us to train parameters efficiently among transformer based models. ALBERT [6] shares parameters on all layers, through cross layer parameter sharing. Thanks to these features, we assume that even a small number of knowledge graph data could be well trained with ALBERT [6].

To integrate graph knowledge into existing ALBERT, we used 80 token (i.e. sequence length) of textual input, and put the next token together as an graphical input. We used ALBERT's pre-trained embedding, but token on the graph side did not use ALBERT's pre-trained embedding. This is because the large text corpus used in pre-train phase is too different in the form and content of the data extracted from the knowledge graph(e.g. triple in knowledge graph). Instead of using ALBERT's pre-trained parameter, we use graph embedding, previously trained through RotatE [16], as input token. Because pre-trained graph embedding is already well-learned on graph structure, it can be said that it has similar meaning to pre-train step of ALBERT [6], and performance has improved when actually using pre-trained graph embedding as input.

III. Experiments

A. Results and Analysis

The results on CommonsenseQA development dataset is shown in Table 1. The performance of the model organized in Table 1 is all single models, including ours.

Group	Model	Dev Acc
Group 1	RoBERTa	78.5
Group 1	ALBERT	80.5
Group 2	KagNet [7]	61.0
Group 2	Graph-based Reasoning [9]	79.3
Group 2	Our Model	81.9

Table 1. Accuracy on CommonsenseQA development set. **Group 1**: models without external knowledge, **Group 2**: models with external knowledge graph(e.g. ConceptNet)

Our model has achieved considerable performance, far exceeding the existing baseline. To compare methods, groups were divided into two according to the applied methods. **Group 1**: models without external knowledge, **Group 2**: models with external knowledge graph(e.g. ConceptNet). ALBERT is the model that is the baseline of our model, and we have improved by 1.4%. This improvement proves the effectiveness of our methods.

B. *Ablation study*

In this section, we perform an ablation study to check the effect of pre-trained knowledge graph embedding. We conducted an experiment not to use graph embedding in the proposed model.

Models	Dev Acc
Our Model	81.98
- Pre-trained graph embedding	81.32

Table 2. Ablation study for pre-trained graph embedding.

As a result of Table 1, you can see better performance when you pre-learn through Rotate. Given that our proposed model has risen by a total of 1.4 points in accuracy over the original ALBERT model, 0.6 points can be considered as a significant difference in performance. We trained our pre-trained graph embedding(RotatE) using negative sample size 128, hidden dim 500, gama 9, batch size 1024, leading rate 0.0001, max step 200,000. We trained graph embedding by referring to the RotatE official code.

IV. Conclusion

We propose a new framework that effectively integrates knowledge graph into the text encoder model. The model integrates triple into one token after the last text input token of ALBERT [6]. Through our integration method, the model predict the correct answer well by using the knowledge extracted from the graph. To extract helpful to answer the question, proceed with the following steps. First, we prune the graph, and integrate redundant relation to reduce scarcity. Second, to extract knowledge from the graph, we find one hop link from the question concept to the answer choice. Third, in order to train graph knowledge well, we pre-train graph embedding through RotatE [16] knowledge embedding. We have achieved results far above the existing baseline in the CommonsenseQA [17] dataset.

References

[1] Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. Comet: Commonsense transformers for automatic knowledge graph construction. *arXiv preprint arXiv:1906.05317*, 2019.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[3] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

[4] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[5] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*, 2017.

[6] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[7] Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. Kagnet: Knowledge-aware graph networks for commonsense reasoning. *arXiv preprint arXiv:1909.02151*, 2019.

[8] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[9] Shangwen Lv, Daya Guo, Jingjing Xu, Duyu Tang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, and Songlin Hu. Graph-based reasoning over heterogeneous external knowledge for commonsense question answering. In *AAAI*, pages 8449–8456, 2020.

[10] Diego Marcheggiani and Ivan Titov. Encoding sentences with graph convolutional networks for semantic role labeling. *arXiv preprint arXiv:1703.04826*, 2017.

[11] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

[12] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[13] Adam Roberts, Colin Raffel, and Noam Shazeer. How much knowledge can you pack into the parameters of a language model? *arXiv preprint arXiv:2002.08910*, 2020.

[14] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer, 2018.

[15] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. *arXiv preprint arXiv:1612.03975*, 2016.

[16] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*, 2019.

[17] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*, 2018.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[19] Jiawei Zhang, Haopeng Zhang, Li Sun, and Congying Xia. Graph-bert: Only attention is needed for learning graph representations. *arXiv preprint arXiv:2001.05140*, 2020.

[20] Yuhao Zhang, Peng Qi, and Christopher D Manning. Graph convolution over pruned dependency trees improves relation extraction. *arXiv preprint arXiv:1809.10185*, 2018.

Graph-Knowledge Integration into a Natural Language Encoder

Kibeom Kim[1], Taehee Kim[2] and Jaegul Choo[2]

[1] Korea University, kkbyg@korea.ac.kr
[2] KAIST, {taeheekim, jchoo}@kaist.ac.kr

Abstract

Given a question, machine reading comprehension (MRC) is the task of identifying its answer from a given document or context. On the other hand, common-sense question answering is considered a more difficult task because it has to answer the question without a given context. This paper proposes a novel framework that effectively integrates information from the external knowledge graph (e.g., ConceptNet) into a neural model. We first extract relevant paths from external knowledge graphs to provide the model with relevant information. Next, the extracted paths are encoded by knowledge graph embedding, e.g., RotatE. The path embeddings are then integrated into the text encoder (e.g., BERT) and trained as multimodal models. Our experiments on the CommonsenseQA dataset demonstrate that our proposed model significantly outperforms the baseline methods.

Keywords— *Deep Learning, Question Answering, Common-Sense Reasoning*

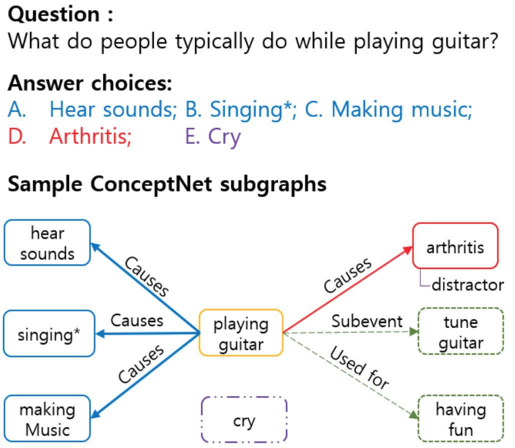
I. Introduction

With the recent development of deep learning, numerous areas of natural language processing (NLP) have achieved rapid growth, especially in MRC tasks such as SQuAD [12] and RACE [5] which do not require external knowledge. Recently, NLP researchers are involved in more difficult tasks such as common-sense reasoning, which requires external knowledge to answer the question. The CommonsenseQA [17] dataset is one such data set that requires an understanding of common sense without a relevant document or context (i.e., a passage in MRC datasets).

CommonsenseQA [17] is a dataset involving a knowledge graph called ConceptNet [15].

ConceptNet [15] is a knowledge graph consisting of nodes (i.e., 'concept') and edges (i.e., 'relation') relevant to real-world knowledge.

CommonsenseQA [17] consists of five multiple-choice questions, Four of them were made by sampled from ConceptNet [15]. Among them, in particular, three answer choices have similar meaning, making the problem more difficult.



Question :
What do people typically do while playing guitar?

Answer choices:
A. Hear sounds; B. Singing*; C. Making music;
D. Arthritis; E. Cry

Sample ConceptNet subgraphs

Fig. 1. Sample ConceptNet subgraphs for CommonsenseQA question.

For example, the source concept 'playing guitar' in Fig. 1 is connected to three target concepts ('singing,' 'making music,' and 'hear sound' in Fig. 1) by 'causes' relation in ConceptNet [15]. These three target concepts become three answer choices each. The fourth answer choice ('arthritis' in Fig. 1) is the target concept extracted from the ConceptNet [15], which also connects 'causes' relation from the source concept('playing guitar'), but it has an entirely different meaning from the other three triples. Finally, the fifth answer choice may not be extracted from ConceptNet [15], but annotators deciding and adding one themselves. Due to this data generation method, the Connmonsense QA [17] is considered a more difficult task for the neural network model than the conventional MRC tasks.

On the other hand, BERT [2] and T5 [11] are pre-trained with a large corpus through self-supervised learning. Self-supervised learning dramatically reduced data labeling cost because it makes labels on its own from training data. As a result, the performance of the self-supervised models has been dramatically improved. It has been recently reported that, after pretraining, knowledge is learned in the form of model parameters [13]. Through a large amount of training data and model parameters, e.g., T5 [11], it achieved significant performance improvement in open-domain QA task such as TriviaQA [3], without accessing external knowledge. However, this approach [13] requires a large number of parameters, e.g., 11 billion parameter, along with enormous computing resources and data. Furthermore, this approach has limitation that it cannot learn beyond what is in the pre-trained corpus since it relies only on pre-training. To solve the problem, utilizing external knowledge graphs by combining them with a language model has recently emerged as a popular research topic. In this paper, we propose a novel framework that can effectively integrate a knowledge graph in BERT-based [2, 8, 6] text encoder.

II. Integrating Knowledge Graph to Text Encoder

This section first introduces the problem setting of CommonsenseQA [17] and describe the workflow of our framework.

Different approaches may exist to put an external graph knowledge in a standard text encoder.

To integrate relevant knowledge graph into our natural-language encoder model, our model first needs to properly represent the knowledge graph as an embedding vector, which will be used as additional input to the encoder. To this end, we applied the RotatE [16] knowledge graph embedding method, which computes the embedding vectors of entities and relations existing in a given knowledge graph, e.g., ConceptNet [15]. Afterwards, given a particular question in CommonsenseQA [17], our method selects triples, composed of entities and their relations, from ConceptNet that are relevant to the given question. The selected triples are using a pre-trained RotatE [16] graph embedding and incorporated into the model. Our method has outperformed existing baseline models [7, 9], achieving the comparable accuracy to the state-of-the-art models.

A. *Graph Pruning*

The connection between each node in ConceptNet [15] is represented by a unit called a triple, which means that it is connected in the form of node-edge-node (i.e., concept - relation - concept). Fig. 1 shows the example triples such as '(playing guitars, causes, singing),' and '(playing guitars, used for, having fun).' ConceptNet [15] was created by numerous annotators and is a large-scale graph containing 32 million triples and 36 relations (e.g., IsA, UsedFor, Causes, RelatedTo, etc.). Besides, we found that about a half of the nodes have one degree or less, meaning that they were connected to only one or no node. Because training graph embedding from these sparse graphs can lead to poor performance, we pruned the graphs to utilize only the necessary nodes. We prune ConceptNet [15] using the following method to use only the necessary graph and increase the performance of the graph embedding.

1. The word contained in the concept can be more than one, and we extract only the concept consisting of at most three subwords.
2. CommonsenseQA [17] uses 22 relations from ConceptNet [15]. While we reduce relations to form a new graph using only a total of 15 relations.

B. *Knowledge Extraction*

So far, the graph itself was scaled down through pruning the graph, and now the important thing is how to extract the subgraph and put it as an input. We assumed that there would be one triple that would help a model to find answers. In CommonsenseQA [17], a question concept is given separately from the question, answer questions. We assume that if there is a triple from this question concept to answer choice, this would be the key information. So, we explored whether it is connected to the question concept for each answer choice and added the corresponding triple as the input of the model.

C. *Model Architecture*

Among the various variants of GNN, GCN [4] expresses nodes as vectors, combines information from adjacent nodes, and updates their node vector. KagNet [7] and Graph-Based Reasoning [9] tried to solve the CommonsenseQA [17] dataset by applying plain GCN [4]. The reason why plain GCN [4] was used in both papers is that, as seen in existing work [10, 20], relational GCNs [14] are mainly over-parameterized and do not represent knowledge graph effectively. Plain GCN [4] does not fully reflect the graph structure because it explicitly means that it does not consider the edge feature (i.e., relation) and direction of the graph.

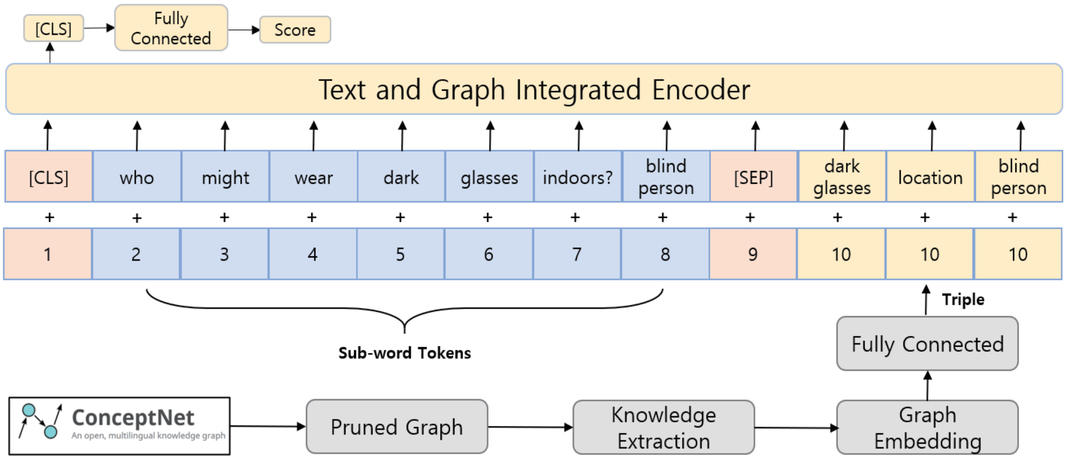


Fig. 2. Illustration of the integrated text and graph multi-model architecture. The example of the CommonsenseQA is 'Who might wear dark glasses?,' and the answer choices are {A. blind person*; B. glove box; C. movie studio; D. ray charles; E. glove compartment}. Among them, 'blind person' marked with an asterisk (*) is the correct answer choice. The format of the input is [CLS] question + answer choice1 [SEP] related triple. The blue boxed token is the text encoder part, and the yellow boxed token represents the graph encoder part.

There have been recent attempts to incorporate graph structures with transformers without using GNNs (e.g., COMET [1], Graph-BERT [19]). As shown in the above papers, Transformers [18] can reflect graph knowledge well enough. Just as GCN [4] expresses the adjacent relationship between nodes, the Transformer [18] can represent the relation between each adjacent node with a self-attention and attention mask. Inspired by this work, we propose a new model framework that injects graph knowledge into the transformer-based text encoder [2, 8, 6]..

Transformer [18] based models (e.g., ALBERT [6], RoBERTa [8]) have a much higher number of model parameters compared to the previous models. These models can train well even if there are many parameters because they are pre-trained with an extremely large corpus. However, the models may not be sufficiently trained because the data extracted from the knowledge graph is much smaller than the pre-train corpus used in the transformer-based model. Therefore, we have adopted ALBERT [6] as the backbone model, a model with a relatively small number of parameters, and allows us to train parameters efficiently among transformer-based models. ALBERT [6] shares parameters on all layers through cross-layer parameter sharing. Thanks to these features, we assume that even a small number of knowledge graph data could be well trained with ALBERT [6].

To integrate graph knowledge into existing ALBERT [6], we used 80 tokens (i.e., sequence length) of textual input and put the next token together as graph embedding. We used pre-trained embedding of ALBERT [6], but the token on the graph side did not use pre-trained embedding of ALBERT [6]. The large text corpus used in the pre-train phase is too different in the form and content of the data extracted from the knowledge graph (e.g., triple in knowledge graph). Instead of using the pre-trained parameter of ALBERT [6], we use graph embedding, previously trained through RotatE [16], as graph embedding input tokens. Because pre-trained graph embedding is already well-learned on the graph structure, it can be said that it has a similar meaning to the pre-train step of ALBERT [6], and performance has improved when using pre-trained graph embedding as input.

III. Experiments

A. Results and Analysis

The results on CommonsenseQA [17] development dataset is shown in Table 1. The performance of the model organized in Table 1 is all single models, including ours.

Group	Model	Dev Acc
Group 1	RoBERTa [8]	78.5
Group 1	ALBERT [6]	80.5
Group 2	KagNet [7]	61.0
Group 2	Graph-based Reasoning [9]	79.3
Group 2	Our Model	81.9

Table 1. Accuracy on CommonsenseQA development set. **Group 1**: models without external knowledge, **Group 2**: models with external knowledge graph (e.g., ConceptNet)

Our model has achieved comparable performance, far exceeding the existing baseline. We divide groups into two according to the applied methods. **Group 1**: models without external knowledge, **Group 2**: models with external knowledge graph (e.g., ConceptNet [15]). ALBERT [6] is the baseline of our model, and we have improved by 1.4%. This improvement proves the effectiveness of our methods.

B. Ablation study

In this section, we perform an ablation study to check the effect of pre-trained knowledge graph embedding. We experimented not to use graph embedding in the proposed model.

Models	Dev Acc
Our Model	81.98
- Pre-trained graph embedding	81.32

Table 2. Ablation study for pre-trained graph embedding.

As a result of Table 1, we can see better performance when pre-learn the model through RotatE [16]. Given that our proposed model has risen by a total of 1.4 points in accuracy over the original ALBERT [6] model, 0.6 points can be considered as a significant difference in performance. We trained our pre-trained graph embedding (i.e., RotatE [16]) using negative sample size 128, hidden dim 500, gamma 9, batch size 1024, leading rate 0.0001, max step 200,000. We trained graph embedding by referring to the RotatE [16] official code.

IV. CONCLUSIONS

We propose a new framework that effectively integrates the knowledge graph into the text encoder model. The model integrates triple into one token after the last text input token of ALBERT [6]. Through our integration method, the model predicts the correct answer well using the knowledge extracted from the graph. To extract helpful to answer the question, proceed with the following steps. First, we prune the graph, and integrate redundant relation to reduce scarcity. Second, to extract knowledge from the graph, we find one hop link from the question concept to the answer choice. Third, in order to train graph knowledge well, we pre-train graph embedding through RotatE [16] knowledge embedding. We have achieved results far above the existing baseline in the CommonsenseQA [17] dataset.

REFERENCES

[1] Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. Comet: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, 2019.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[3] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, 2017.

[4] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[5] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, 2017.

[6] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2019.

[7] Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. Kagnet: Knowledge-aware graph networks for commonsense reasoning. 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)*, pages 2822–2832, 2019.

[8] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[9] Shangwen Lv, Daya Guo, Jingjing Xu, Duyu Tang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, and Songlin Hu. Graph-based reasoning over heterogeneous external knowledge for commonsense question answering. In *AAAI*, pages 8449–8456, 2020.

[10] Diego Marcheggiani and Ivan Titov. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515, 2017.

[11] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

[12] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, 2016.

[13] Adam Roberts, Colin Raffel, and Noam Shazeer. How much knowledge can you pack into the parameters of a language model? *arXiv preprint arXiv:2002.08910*, 2020.

[14] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer, 2018.

[15] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: an open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 4444–4451, 2017.

[16] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*, 2018.

[17] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, 2019.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[19] Jiawei Zhang, Haopeng Zhang, Li Sun, and Congying Xia. Graph-bert: Only attention is needed for learning graph representations. *arXiv preprint arXiv:2001.05140*, 2020.

[20] Yuhao Zhang, Peng Qi, and Christopher D Manning. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215, 2018.

Reasoning based Adaptive Image Compression for Efficient Satellite-Land Communication without Performance Degradation

KyungChae Lee[1], Changha Lee[1] and Chan-Hyun Youn[1]

[1] School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon, Korea,
{kyungchae.lee, changha.lee, chyoun}@kaist.ac.kr

Abstract

In the era of multinational cooperation, gathering and analyzing the satellite images are getting easier and more important. Typical procedure of the satellite image analysis include transmission of the bulky image data from satellite to the ground producing significant overhead. To reduce the amount of the transmission overhead while making no harm to the analysis result, we propose a novel image compression scheme RDIC in this paper. RDIC is a reasoning based image compression scheme that compresses an image according to the pixel importance score acquired from the analysis model itself. From the experimental results we showed that our RDIC scheme successfully captures the important regions in an image showing high compression rate and low accuracy loss.

Keywords*— *Image Compression, Explainable AI, Satellite Images

I. Introduction

Satellite image analysis is a crucial task for gathering information world-wide in the era of multinational cooperation. Since most, if not all, satellites have little computational resources satellite image analysis is typically done on the ground where powerful computing stations exist. Thus, analyzing a satellite image starts with image transmission from satellite to the ground station.

The transmission latency is quite large when it comes to the satellite environment since they are very far away from the ground and therefore have very limited communication bandwidth. Therefore, there is a need for an image compression technique in order to mitigate the transmission overhead. However, compressing an image often causes information loss resulting in analysis performance degradation which is not wanted behavior.

One of the most popular image compression technique is JPEG[8]. JPEG is an image compression standard that has been used in various domains enabling lightweight image compression with acceptable visual image quality. However, since the jpeg compression basically works with filtering out the high frequency color components in the image[8], color distortion is inevitable after the pipeline of encoding and decoding. This color distortion causes performance degradation on image analysis tools such as deep neural networks.

In this paper we propose an adaptive image compression technique that can reduce the transmission latency while preserving the accuracy of the analysis tools. This can be done by adaptively choosing the region of interest(RoI) where the analysis tool values the most and compress those regions with high quality and others with low quality. The details of choosing the RoI will be delivered in Section iii. We showed that with our proposed algorithm we can significantly reduce the image file size while successfully preserving the analysis accuracy on satellite images.

II. Related Works

A. JPEG Image compression for Satellite Imageries

As mentioned in the Section i., jpeg image compression is one of the most popular standard in the domain and there are numerous works that applied jpeg compression on satellite images[11]. Work done by Tada et. al.[11] evaluated the effect of the jpeg compression with power spectrum comparison showing the degree of image distortion after the compression. What is not dealt with tada's work is that the image compression may cause the deep neural network malfunction even though the compressed image looks fine with human eyes. Researches like [3, 7] report that jpeg compression does affect the performance of the deep neural network. We also conducted a preliminary experiment that shows the impact of jpeg compression on Faster R-CNN object detection network for satellite images in Section iv.. Thus, it is clear that compressing a satellite image should be dealt carefully in order to get the maximum analyzing performance on the ground.

B. Image Compression with Accuracy Consideration

There are some works that deals with the image compression considering the outcome of the analysis tools on the receiver side[9, 4, 5, 10, 6]. Both works are targetting the different domain but the basic idea is similar. Work [6] uses a image compression for efficient image transmission between mobile client and the edge server when offloading the compute intensive image object detection task from client to the server. Here, they propose a Dynamic Region of Interest for adaptive image compression where the object detection result for the previous frame is used for determining the important area of the current frame. The selected RoI is then compressed with higher quality(less compression), and the rest of the image area are compressed with lower quality(more compression). By doing so, [6] achieves real-time image offloading in edge assisted augmented reality(AR) service.

Work [9] and others [4, 5, 10] are more related to our work that those directly aim the same domain we are dealing with: Satellite image compression without performance degradation. Paper [9] proposes a fuzzy c-means image segmentation and adaptive image compression according to the segmentation result which in turn compresses the background more and important objects less.

However, works like [9, 4, 5, 10] simply focuses on first incestigatin the image and do not directly consider the structure or the characteristics of the analysis tools on the ground. This inconsistency of image interpretation between land and satellite would result in performance degradation. On the other hand, work done by [6] uses the previous result from the analysis tool, so it directly considers the analysis tool when compressing an image. Nonetheless, algorithm of [6] is very hard to be applied to satellite imageries since satellite images typically captures different places and even if the time series data can be produced, huge transmission overhead makes data from last time step less valuable.

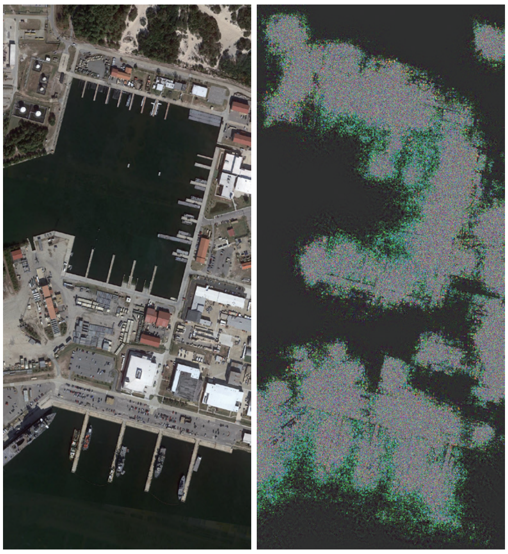
III. Methodology

In this paper, we propose a novel image compression scheme, reasoning based dynamic image compression(RDIC), which makes use of the layer-wise relevance propagation[1] which is one of the explainable AI techniques. Layer-wise relevance propagation works by back-propagating the neural network result and it's relevance score as in the equation 1 and point the salient part of the input image where the model got the most valuable information getting the result.

$$R_i^{(l)} = \sum_j \frac{z_{ij}}{\sum_{i'} z_{i'j} + \varepsilon sign(\sum_{i'} z_{i'j})} R_j^{(l+1)} \qquad (1)$$

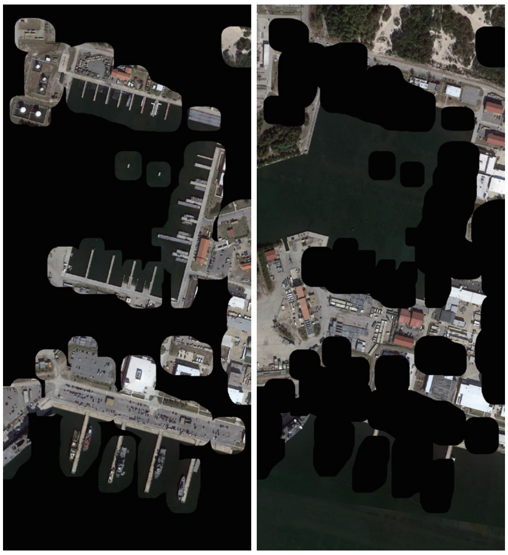
An example of the result of applying epsilon lrp, layer-wise relevance propagation, to the object detection model SCRDet[13] is shown in the Figure. 1. As can be seen in the figure salient part containing target objects(ship, harbor, cars, etc...) are highlighted. Instead of highlighting the whole foreground objects by using epsilon lrp we could highlight the image region that is needed by the model for image analysis.

From the result of the epsilon lrp relevance propagation, we now calculate the region of interest(RoI) which will act as a mask determining the compression quality. The outcome of the epsilon lrp is a bitmap which indicates the pixel-wise relevance score within the range of negative infinity and positive infinity. This unboundedness makes the calculation of the importance mask difficult with raw outcome of the elrp. Thus, we first took the absolute value of the outcome and then noramlized with the mean value of the outcome. From the normalized outcome values, we then create a mask M which indicates the pixel area where the normalized relevance value is bigger than 0 as shown in the equation 2. However, as we can see from the Fig. 1-(b), the outcome of the elrp is basically very noisy and therefore the resulting mask is also very noisy. The noisiness of the mask can significantly degrade the performance of the target object detection model so we implemented a dilation operation, one of the conventional CV techniques, for acquiring smooth RoI masks. The example of the final RoI mask can be seen in the Fig. 2. As can be seen in the figure, salient areas containing the target objects are successfully highlighted. Furthermore, in the background region we can see that salient objects are also included but the category



(a) Input image (b) Epsilon lrp result

Fig. 1. Example of applying epsilon lrp model explanation to the satellite image input and object dection model



(a) Salient region (b) Background

Fig. 2. Example of calculated final RoI mask

of the salient objects does not belong to the target category.

$$M(i,j) = \begin{cases} 1 & abs(elrp(I)(i,j)) >= mean(abs(elrp(I))) \\ 0 & \text{else, I: Input image} \end{cases} \quad (2)$$

After the calculation of the RoI for determining the compression criteria, we conducted the dynamic image compression where we compress the RoI region with high quality and background region with low quality. Here, the quality of the compression follows the quality definition of the computer vision OpenCV[2].

IV. Experimental Results

For the evaluation of our proposed image compression scheme we have conducted an experiment comparing the mean average precision performance of the object detection model on datasets compressed with different methods.

For the dataset we used a DOTA dataset[12] which is an open dataset consisting of numerous images taken by a plane that is similar to the satellite imageries.

For the target object detection model we chose faster r-cnn model following the paper [13].

We compared the evaluation result of the faster r-cnn model on the dataset first, and then we compressed the original dataset with two different methods: Original JPEG compression, and proposed reasoning based dynamic image compression(RDIC). Here, original jpeg compression is done with the quality of 100, and RDIC consist of two different quality 100, 50 each for RoI and BG regions. We then compared both the mAP score and the total file size of the dataset which can be seen in the Fig. 3 and the Table. 1.

	Original	JPEG	RDIC
File Size(MB)	3324	1671	942
mAP(%)	57.75	57.75	56.54

Table 1. Example of caption for table.

The figure Fig. 3 shows the average precision of each classes. As we can see from the figure, the performance of the object detection model is mostly preserved after the compression. Interesting part is that in case of classes like soccer-ball-field and large-vehicle, jpeg and RDIC compressed version of ours resulted in a better precision score. Except for this unexpected outcome, we can see that the average precision gets lower when applied compression to the dataset. However, if we look into the file size analysis in the Table. 1, we can see that compared to the original dataset, JPEG compression provides identical performance while the size of the dataset is reduced to 50.27 percent of the original dataset. Our proposed RDIC loses about 1.21 percent point of the accuracy while reducing the filesize into 942 Mega Bytes which is 56.4 percent of the JPEG compressed dataset, and 27.9 percent of the original dataset. This significant reduction of the filesize allows the satellite image transmission to be about four times faster than usual with only a 1.2 percent point loss of the detection model accuracy.

V. Conclusion

Satellite imageries are big in their size which causes a huge transmission latency hindering the fast and easy analysis of the image. In this paper we propose a novel image compression scheme based on the model reasoning that allows us to compress the satellite image with minimum accuracy loss and high compression rate. Our scheme starts from analyzing the target model by relevance propagation for RoI searching. According to the RoI we then conducted a dynamic image compression which will compress the important part of an image with high quality and others with high compression rate. The evaluation results show that our scheme successfully capture the important region in the image according to the model we use. Since the epsilon lrp method and other techniques we used is not bound to a single object detection model, our scheme is also easy to apply on various other applications and neural network models.

References

[1] Alexander Binder, Sebastian Bach, Gregoire Montavon, Klaus-Robert Müller, and Wojciech Samek. Layer-wise relevance propagation for deep neural network architectures. In *Information Science and Applications (ICISA) 2016*, pages 913–922. Springer, 2016.

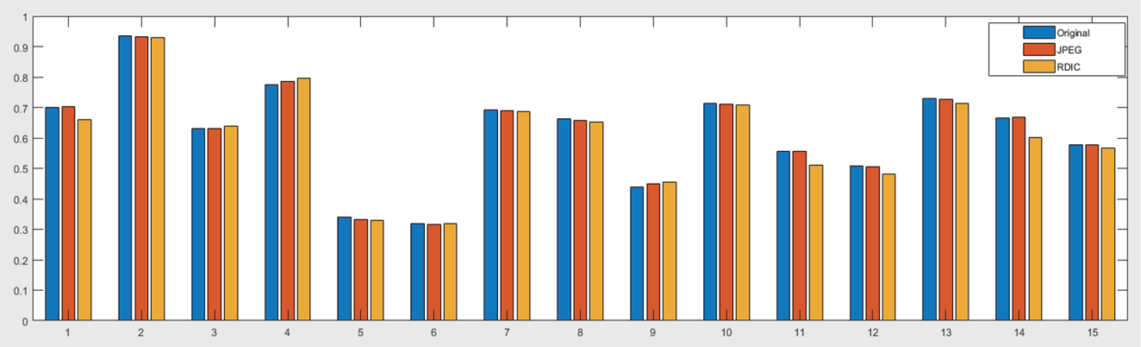


Fig. 3. Experimental results for comparing the mAP score for each test cases: Original Dataset(Blue), JPEG compressed dataset(Orange), RDIC compressed dataset(Yellow). Class number(1-16) stands for classes [’roundabout’, ’tennis-court’, ’storage-tank’, ’soccer-ball-field’, ’small-vehicle’, ’ship’, ’plane’, ’large-vehicle’, ’helicopter’, ’harbor’, ’ground-track-field’, ’bridge’, ’basketball-court’, ’baseball-diamond’, ’mAP’]

[2] G. Bradski. The OpenCV Library. *Dr. Dobb’s Journal of Software Tools*, 2000.

[3] Mathieu Dejean-Servières, Karol Desnos, Kamel Abdelouahab, Wassim Hamidouche, Luce Morin, and Maxime Pelcat. Study of the impact of standard image compression techniques on performance of image classification with a convolutional neural network. 2017.

[4] Xun Du, Adriana Dapena, and Stanley C Ahalt. Content-based image compression for atr applications. In *Algorithms for Synthetic Aperture Radar Imagery VII*, volume 4053, pages 696–704. International Society for Optics and Photonics, 2000.

[5] Xun Du, Honglin Li, and Stanley C Ahalt. Content-based image compression. In *Algorithms for Synthetic Aperture Radar Imagery VIII*, volume 4382, pages 92–102. International Society for Optics and Photonics, 2001.

[6] Luyang Liu, Hongyu Li, and Marco Gruteser. Edge assisted real-time object detection for mobile augmented reality. In *The 25th Annual International Conference on Mobile Computing and Networking*, pages 1–16, 2019.

[7] Zihao Liu, Tao Liu, Wujie Wen, Lei Jiang, Jie Xu, Yanzhi Wang, and Gang Quan. Deepn-jpeg: A deep neural network favorable jpeg-based image compression framework. In *Proceedings of the 55th Annual Design Automation Conference*, pages 1–6, 2018.

[8] William B Pennebaker and Joan L Mitchell. *JPEG: Still image data compression standard*. Springer Science & Business Media, 1992.

[9] Katari Clement Emmanuel Sanjay Raj, Sarma Venkataraman, and Geeta Varadan. A fuzzy approach to region of interest coding in jpeg 2000 for automatic target recognition applications from high-resolution satellite images. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 193–200. IEEE, 2008.

[10] S Richard F Sims, Jonathan A Mills, and Pankaj N Topiwala. Evaluation of video compression technologies for atr. In *Automatic Target Recognition XIV*, volume 5426, pages 370–378. International Society for Optics and Photonics, 2004.

[11] T Tada, K Cho, H Shimoda, T Sakata, and S Sobue. An evaluation of jpeg compression for on-line satellite images transmission. In *Proceedings of IGARSS’93-IEEE International Geoscience and Remote Sensing Symposium*, pages 1515–1518. IEEE, 1993.

[12] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3974–3983, 2018.

[13] Xue Yang, Jirui Yang, Junchi Yan, Yue Zhang, Tengfei Zhan, Zhi Guo, Sun Xian, and Kun Fu. Scrdet: Towards more robust detection for small, cluttered and rotated objects. In *Proc. ICCV*, 2019.

AttentionSeg: An Attention-based Convolutional Neural Network for Real-Time Object Segmentation from 3D Point Cloud

Moogab Kim[1], Naveed Ilyas[1] and Kiseon Kim[1]

[1] School of Electrical Engineering and Computer Science, Gwangju Institute Science and Technology, Gwangju, Republic of Korea,
{anrkq1024, naveedilyas, kskim}@gist.ac.kr

Abstract

The imbalance of object segmentation accuracy among car and smaller objects such as pedestrian and cyclist is an inherent problem due to difference of appearance and size. To address this challenge, we propose an attention-based convolutional neural network (CNN) for object segmentation (AttentionSeg). The proposed network comprises of spatial attention module (SAM) and channel attention module (CAM) to aggregate spatial and semantic information from low-level and high-level layers, respectively. Our network enables to extract more distinct feature on the smaller objects to enhance segmentation performance. We evaluate our method on KITTI dataset. Experiment shows that our network is effective to improve segmentation performance on pedestrian and cyclist while achieving real-time speed of 9.1ms.

Keywords— *Object Segmentation, 3D Point Cloud, Attention*

I. Introduction

Object segmentation from 3D point cloud aims to perform point-wise classification to provide abundant representation on objects of interest (i.e., car, pedestrian, and cyclist). Especially, it is an important task for autonomous vehicles to determine path and control planning by recognizing surrounding objects of the vehicles. Nonetheless, segmenting the objects is challenging to achieve competitive segmentation performance in real-time due to sparsity and hugeness of the point cloud.

To perform 3D recognition such as segmentation and detection, previous approaches directly processed the point cloud [1] or voxelized the point cloud as pixels in an image [2]. However, these methods suffer from expensive computation as they process all sparse and huge points, thus not able to be applicable for real-time application. Recently, authors in [3] and [4] conducted object segmentation from 3D point cloud by spherically transforming the point cloud to 2D range-images to apply CNN-based approach. They employed efficient convolutional and deconvolutional operations to extract distinct features on road-objects inspired from [5]. They achieved significant performance on a large size object (car), however, the performance degraded on small size objects such as pedestrian and cyclist since their approaches lost spatial information on the small objects due to multiple down-samplings.

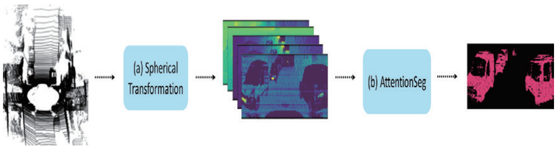


Fig. 1. An overview of proposed method. (a) The spherical transformation is performed on 3D point cloud to obtain 2D range-images. (b) AttentionSeg predicts objects in point-wise for object segmentation.

Based on the above observations, we therefore propose an attention-based approach to improve segmentation performance, especially on smaller objects (pedestrian and cyclist). The proposed approach is composed of two modules: (i) SAM and (ii) CAM, by leveraging CNN architecture of [3]. SAM is employed to provide high-level features with spatial details on the smaller objects from low-level features via skip-connections [6]. Whereas, CAM is used to enhance semantic features by modeling channel interdependency between pedestrian and cyclist in high-level features [7]. Further, we reduce the number of down-sampling to maintain spatial information of the small size objects by removing the first max pooling layer [4]. In summary, the contribution of our research are as follows:

- We design an attention-based CNN for object segmentation to obtain spatial to semantic features for enhanced segmentation accuracy.
- The proposed model with SAM (skip-connection) and CAM increases the ability of the network to segment small size objects such as pedestrian and cyclist by obtaining the spatial information and enhancing the final semantic features, respectively.

II. Proposed Method

The overall architecture of proposed method is illustrated in Fig. 1. Firstly, we spherically transform 3D point cloud to 2D range-images to efficiently process the point cloud. Secondly, the proposed method employs an architecture [3] as a backbone network while using SAM and CAM to model spatial information and channel interdependency for discriminant feature representation, respectively. In addition, we feed the range-images into proposed AttentionSeg to obtain point-wise classes (car, pedestrian, and cyclist). We dedicate subsections for spherical transformation and an attention-based CNN architecture in detail.

A. *Spherical Transformation from 3D Point Cloud to 2D Range-Images*

To efficiently process 3D point cloud instead of directly applying CNN on the point cloud [3][4], spherical transformation is applied to obtain 2D range-images. Each point of the point cloud can be represent as a set of Cartesian coordinate,(x,y,z). Therefore, the formula for spherical transformation can be defined as:

$$\alpha = \arcsin\left(\frac{z}{\sqrt{x^2+y^2+z^2}}\right), \hat{\alpha} = \lfloor\frac{\alpha}{\Delta\alpha}\rfloor \tag{1}$$

$$\beta = \arcsin\left(\frac{y}{\sqrt{x^2+y^2}}\right), \hat{\beta} = \lfloor\frac{\beta}{\Delta\beta}\rfloor \tag{2}$$

where α and β are azimuth and zenith angles. $\hat{\alpha}$ and $\hat{\beta}$ represent the position of a point on 2D range-images. $\Delta\alpha$ and $\Delta\beta$ are resolutions for discretizing the point cloud. We obtain spherically transformed range-images with dimension of $H \times W \times C$, where H, W, and C encode the height, width, and channel, respectively. Since the point cloud is generated by a Velodyne HDL-64E LiDAR with 64 vertical channels, $H = 64$. Road-objects are annotated based on 3D bounding boxes in a front view area of 90°. We discretize the area into 512 grids, which determine $W = 512$. Each point in the point cloud is represented by Cartesian coordinates (x,y,z), distance $d = \sqrt{x^2+y^2+z^2}$, and intensity, which are used as channels $(C = 5)$. We therefore utilize $64 \times 512 \times 5$ range-images as input data.

B. *AttentionSeg: An Attention-based CNN Architecture for Real-Time Object Segmentation*

An attention-based CNN architecture for object segmentation (AttentionSeg), has an encoder-decoder architecture as illustrated in Fig. 2. Our model is able to focus on smaller objects such as pedestrian and cyclist by extracting spatial-aware to semantic features. The proposed model consists of a backbone network and two modules: SAM and CAM. The backbone network extracts general features motivated from [3]. The SAM is used to introduce spatial details from low-level features to high-level features via skip-connection while The CAM is employed to enhance semantic representation of the final feature.

1) Backbone Network: Most popular backbone network for a segmentation task [8] has an encoder-decoder architecture. The encoder extracts general features by down-sampling input images, whereas the decoder upsamples the features to generate specified features for pixel-wise classification with the same resolution as the input images. However, the backbone network employs general convolution and deconvolution layers for encoder and decoder, not applicable for real-time application due to expensive computation. To satisfy the real-time condition, authors in [3][4] exploit efficient convolution and deconvolution modules, namely FireModule [5] and FireDeconv [3], for the encoder and decoder with reduced computation cost. However, the structure [3] loses spatial information due to multiple max pooling layers, which disseminate spatial information to aggregate contextual information, thus not suitable for accurate segmentation [4]. Hence, based on these considerations, we reduce 4 times down-sampling to 3 times by removing the first max pooling layer in our backbone network.

For the encoder, the structure of FireModule [5] is depicted in Fig. 3 (a). The FireModule applies 1×1 convolution (squeeze layer) to input features with C channels to reduce the number of channels to $\frac{C}{4}$. And then 1×1 and 3×3 convolutional layers (expand layer) are applied in parallel to obtain two feature maps with channel dimension equal to $\frac{C}{2}$. Finally, the two features are concatenated for generating features with C channels.

For the decoder, we utilize the FireDeconv [3], which is as the same as the FireModule except for a transposed convolution layer between the squeeze layer and the expand layer as illustrated in Fig 3. (b). Moreover, the number of up-sampling is reduced due to reduction of the number of down-sampling, which alleviates computation cost. After recovering the original resolution, we perform 1×1 convolution to generate 4 channels where each channel represents each class (background, car, pedestrian, and cyclist).

2) Spatial Attention Module: One difficulty in segmenting objects such as pedestrian and cyclist arises due to smallness of appearance. To address this problem, we employ spatial attention module (SAM) [6]. Low-level features contain rich spatial information, however, the information disseminates as passed to high-level layers. To incorporate spatial-aware features, we employ SAM as illustrated in Fig. 4 (a). First, we apply an average pooling to low-level features along channel axis and then apply 1×1 convolutional layer to produce a spatial attention map. The spatial attention map with rich spatial information is transferred via skip-connection to high-level layers for the distinct representation of smaller objects by element-wise multiplication.

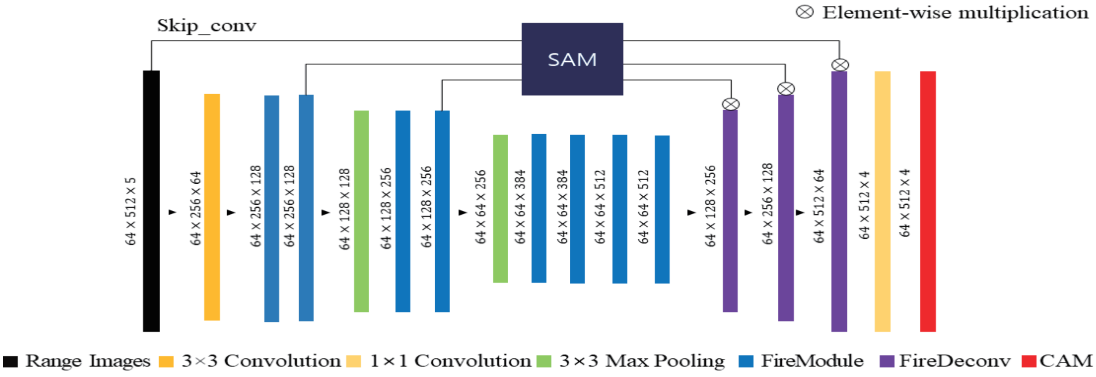


Fig. 2. An architecture of the proposed network. Each convolution layer is followed by ReLU activation. (Best viewed in color)

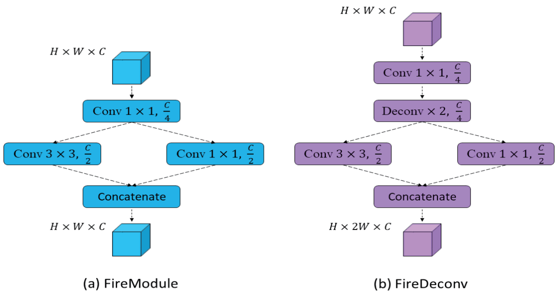

$H \times W \times C$
Conv 1 × 1, $\frac{C}{4}$
Conv 3 × 3, $\frac{C}{2}$
Conv 1 × 1, $\frac{C}{2}$
Concatenate
$H \times W \times C$
(a) FireModule
$H \times W \times C$
Conv 1 × 1, $\frac{C}{4}$
Deconv × 2, $\frac{C}{4}$
Conv 3 × 3, $\frac{C}{2}$
Conv 1 × 1, $\frac{C}{2}$
Concatenate
$H \times 2W \times C$
(b) FireDeconv

Fig. 3. (a) An efficient convolution module (FireModule). (b) An efficient deconvolution module (FireDeconv).

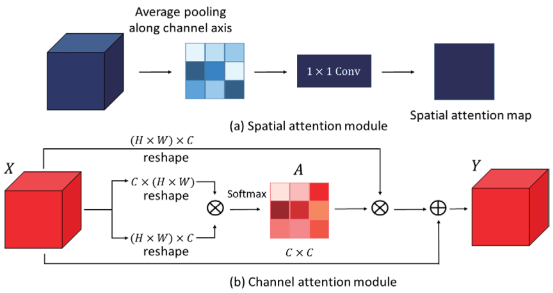


Fig. 4. (a) A structure of spatial attention module. (b) A structure of channel attention module.

3) Channel Attention Module: Channel attention module (CAM) is used to enhance semantic feature representation as shown in Fig. 4. (b). In high-level layers, each channel map can be considered as a class-specific response and the semantic response are related with each other [7]. Therefore, the feature representation on specific classes can be enhanced by modeling interdependency among the channels. Firstly, we reshape input features $X \in R^{H\times W\times C}$ to $R^{N\times C}$. Next, we conduct a matrix multiplication between the reshaped X and the transpose of X. We apply a softmax to get an attention map $A^{C\times C}$.

$$a_{ji} = \frac{exp\,(X_iX_j)}{\sum_{i=1}^{C} exp\,(X_iX_j)} \tag{3}$$

where a_{ji} measures the i^{th} channel's impact on the j^{th} channel. Moreover, we multiply the $X^{N\times C}$ by $A^{C\times C}$ and reshape the result to $R^{H\times W\times C}$. Lastly, the result is multiplied by a learnable value (α) and perform an element-wise summation with the original X to obtain the output $Y \in R^{H\times W\times C}$.

$$Y_j = \alpha \sum_{i=1}^{C} (a_{ji}X_i) + X_j \tag{4}$$

where Y_j represents the final feature map. The equation means that the final feature of each channel is a weighted sum of features of all channels and an original feature, which models semantic interdependency between the channels. Therefore, CAM is applied to the final features generated by 1×1 convolutional layer. Finally, we apply softmax activation to the feature maps so as to obtain point-wise prediction.

III. Experiments

In this section, we present details of experiments. We show the experimental results on KITTI dataset quantitatively and qualitatively.

A. Training Details

To train proposed model, we use 10,848 range-images transformed from KITTI dataset [9] and split the images into 8,057 training set and 2,791 validation set. We implement our model on Pytorch platform and train the model for around 12 hours by using NVIDIA TITAN Xp GPU. Further, we use focal loss [10] to measure the loss of predicted value. The focal loss is given as follows:

$$FL\,(p_c) = -\,(1-p_c)^{\gamma}\,log\,(p_c) \tag{5}$$

where p_c is a predicted probability according to the c class and γ is focusing parameter equal to 2.

B. *Experimental Results*

We evaluate the AttentionSeg on KITTI dataset. Intersection over union (IoU) is used as evaluation metrics, given as follows:

$$IoU_c = \frac{G_c \cap P_c}{G_c \cup P_c} \tag{6}$$

where G_c and P_c are ground-truth and predicted point of the class-c. Further, average runtime (AR) of images in validation set is measured. We compare segmentation performance of AttentionSeg with SqueezeSeg and SqueezeSeg w/o max pooling (MP) as shown in Table 1. Whereas the AR comparison is shown in Table 2. The SqueezeSeg w/o MP improves segmentation accuracy compared to original SqueezeSeg on car and pedestrian classes, by 0.7% and 5.3%. Further, AttentionSeg outperforms SqueezeSeg w/o MP for smaller objects such as pedestrian and cyclist by 4.8% and 4.2% despite slight performance degradation on car. Whereas, we achieve the comparable AR at the cost of segmenting the smaller objects more accurately. Further, the qualitative results are shown in Fig. 5. The qualitative results on pedestrian and cyclist justify the better performance of AttentionSeg as depicted by yellow boxes.

	Car	Pedestrian	Cyclist
SqueezeSeg [3]	58.1	1.8	17.8
SqueezeSeg w/o MP	58.8	7.1	17.3
AttentionSeg (Proposed)	57.2	11.9	21.5

Table 1. Comparison of segmentation performance $(IoU\%)$.

	Average Runtime (ms)
SqueezeSeg [3]	8.9
SqueezeSeg w/o MP	8.4
AttentionSeg (Proposed)	9.1

Table 2. Comparison of average runtime.

IV. Conclusion and Future Work

In this work, we proposed an attention-based convolutional neural network for real-time object segmentation from 3D point cloud. To extract discriminant features on pedestrian and cyclist, we employed spatial and channel attention module. Furthermore, we reduced the number of down-sampling to maintain spatial information on the human-related classes. The segmentation results showed that proposed method improved segmentation performance on pedestrian and cyclist in real-time. In future, we focus on extracting object-level information to improve segmentation performance of overall objects.

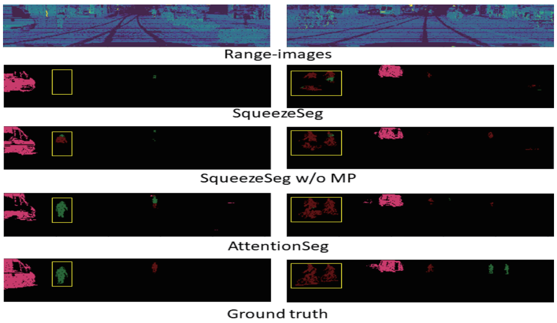


Fig. 5. Visual comparison of segmentation. The black, pink, green, and brown indicate background, car, pedestrian, and cyclist, respectively.

Acknowledgement

This research was a part of the project titled 'Development of Automatic Identification Monitoring System for Fishing Gears', funded by the Ministry of Oceans and Fisheries, Korea.

References

[1] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *In Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017.

[2] Y. Zhou and O. Tuzel. Voxelnet: End-to-end learning for po-int cloud based 3d object detection. *In Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018.

[3] B. Wu, A. Wan, X. Yue, and K. Keurtzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. *In International Conference on Robotics and automation*, 2018.

[4] Y. Wang, T. Shi, P.Yun, L. Tai, and M. Liu. Pointseg: Real-time semantic segmentation based on 3d lidar point cloud. *arXiv:1807.06288*, 2018.

[5] F. N Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. Squeezenet: Alexnet-level accurate with 50x fewer parameters and <0.5mb model size. *arXiv:1602.07360*, 2016.

[6] M. Liu and H. Yin. Feature pyramid encoding network for real time semantic segmentation. *arXiv:1909.08599*, 2019.

[7] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu. Dual attention network for scene segmentation. *In Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019.

[8] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *In Medical Image Computing and Computer-Assisted Intervention*, vol. 9351:234–241, 2015.

[9] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. *In Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

[10] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

Deepfake detection model based on fake attributes shown in images/videos

Ran Heo[1], Eunjin Cho[2]
[1]Dept. of Chemical & Biomolecular Engineering, Dongyang Mirae University,
hran9462@gmail.com
[2]Dept. of Mechanical Engineering, RMIT University, jintazcho@gmail.com

Abstract

Deepfake refers to the fake images/videos created with deep learning. Deepfake video caused social problems as it has spread through the internet, including Social Network Service(SNS). [4,5,6] Therefore, the importance of deepfake detection models has emerged, so many detection models have been suggested. However, these models are not useful in the real world because they can only detect a few parts of deepfake creation algorithms. (Figure 2) In this paper, we suggest the model that uses CNN(Convolutional Neural Network) for fake attribute detection and Transformer for fake/real judgment to cope with various deepfake creation algorithms real-world. Our purpose is to prove that the suggested model is useful in the real-world. Our experiment focused on the improved model's detection performance to detect an increasing number of various deepfake creation models. Researchers' participation is required to cope with multiple deepfake creation algorithms as well. This paper trained deepfake detection models with the presented architecture to prove that the detection performance is similar to compare with the models with high-end GPU[37, 38]. The evaluation result was that Recall increased from 2% to 75% after the detection model improvement, the FPR rate decreased 1.6% to 0.1 and AUC increased from 0.02 to 0.77, which is similar to the model with high-end GPU [37, 38]. [1]

Keywords - Deepfake Detection, Face synthesis Detection, Convolutional Neural Network, Transformer

1. Introduction

Fake face generation algorithms are becoming more complicated and diversified since late 2017 when Reddit user is called 'Deepfakes' proposed face generation method with deep learning. The most representative fake face generation methods are Face synthesis, which generates a whole human head, including face and Face-swap, which swaps faces and facial expressions of two different people. There are deepfake videos of politicians like Obama, Putin, Hillary, and celebrities' face-synthesized pornos created with the Face-swap method. [1, 2, 3] Such videos may cause social problems such as fake news creation, defamation[4, 5, 6], so deepfake related videos legislation[7, 8, 9] and fake face detection have become critical issues to solve these problems.

Deepfake creation algorithm	Method
Began[11]	Face synthesis
CausalGAN[12]	Face synthesis
faceswap/deepfake[13]	Face swap
StarGAN[14]	Face synthesis
Enrique Sanchez and Michel Valstar[15]	Face synthesis
MWGAN[16]	Face synthesis
ALAE[17]	Face synthesis
StyleGAN[18]	Face synthesis
MSG-GAN[19]	Face synthesis
FQGAN[20]	Face synthesis
ProGAN[21]	Face synthesis
StyleGAN v2[22]	Face synthesis
COCO-GAN[23]	Face synthesis
VAEGAN[24]	Face synthesis
HoloGAN[25]	Face synthesis
SPA-GAN[26]	Face synthesis
FTGAN[27]	Face synthesis
SEGAN[28]	Face synthesis
StarGAN V2[29]	Face synthesis
LSGAN[30]	Face synthesis
DCGAN[31]	Face synthesis
WGAN[32]	Face synthesis
GAN2play[33]	Face synthesis
Glow[34]	Face synthesis
GANnotation[35]	Face synthesis
deferred neural rendering[36]	Face synthesis
neural texture[36]	Face synthesis

Table 1. Various deepfake creation algorithms

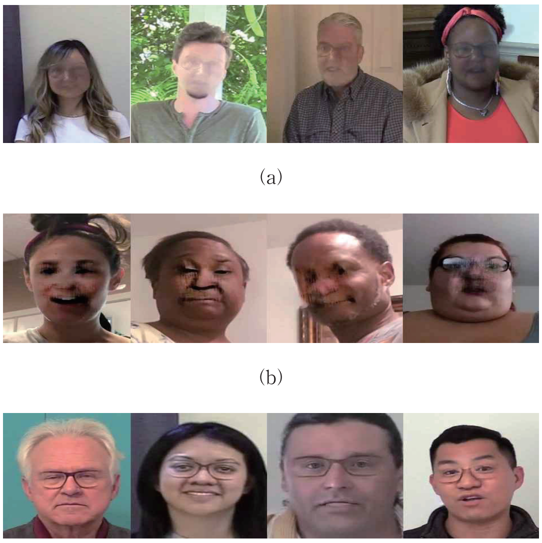
Current detection models stated in table 2, shows

[1] github(Entire model code, instruction, Process Video are available): https://github.com/teamnova-ailab/Deepfake-detection-model-based-on-fake-attributes-shown-in-image-video/

a high detection rate of specific fake images created by the algorithm, which created a training image. However, such detection models specialized in just one algorithm, so they can not guarantee the detection performance of fake faces created by different algorithms.[10] As Fake face creation algorithms are various continuously, existing detection models cannot detect fake faces in the real world. Therefore detection model which guarantee detection accuracy of new fake face creation algorithms is required.

We defined images from Figure 1 and awkward images due to the flaws of the fake face creation process as fake attributes presented images. In this paper, the detection model comprises of CNN models that detect each fake attribute and an Transformer model which judge Real/Fake in the basis of those attributes. This model is useful when a new fake face creation method appears because it is possible to add a new CNN model that trains a new fake feature. Model structure and code are available in GitHub so researchers can contribute this model to improve detection accuracy.

The proposed detection model is enough to train with Nvidia RTX 2070 SUPER when other detection models used Nvidia Titan X GPU[37], Nvidia Tesla P40 GPU[38]. It is because of training with



(a)

(b)

(c)

Figure 1. Example of Image with fake features (a) Face blur Images not showing eyes, nose and mouth (b) Face noise showing Images (c) Images Glasses without glass legs

Detection model	Data set used
McCloskey and Albright (2018) [39]	NIST MFC2018
Yu et al. (2019) [40]	Own (ProGAN, SNGAN, CramerGAN, MMDGAN)
Wang et al. (2019) [38]	FF++, DFDC, Own (PGGAN, StyleGAN2, StarGAN, STGAN, StyleGAN, STGAN)
Stehouwer et al. (2019) [41]	DFFD (ProGAN, StyleGAN)
Nataraj et al. (2019) [42]	100K-Faces (StyleGAN)
Neves et al. (2019) [43]	100K-Faces (StyleGAN) FSRemovalDB (StyleGAN)
Marra et al. (2019) [44]	Own (CycleGAN, ProGAN, Glow, StarGAN, StyleGAN)
Zhou et al. (2018) [45]	Own
Afchar et al. (2018) [46]	Own
Güera and Delp (2018) [47]	Own
Yang et al. (2019) [48]	UADFV
Li et al. (2019) [49]	UADFV DeepfakeTIMIT
Rössler et al. (2019) [50]	FF++
Matern et al. (2019) [51]	Own
Nguyen et al. (2019) [52]	FF++
Agarwal and Farid (2019) [53]	Own (FaceSwap, HQ)
Sabir et al. (2019) [54]	FF++
Bharati et al. (2016) [55]	Own (Celebrity Retouching, ND-IIITD Retouching)
Tariq et al. (2018) [56]	Own (ProGAN, Adobe Photoshop)
Wang et al. (2019) [57]	Own (InterFaceGAN/StyleGAN)
Jain et al. (2019) [58]	Own (ND-IIITD Retouching, StarGAN)
Marra et al. (2019) [59]	Own (Glow/StarGAN)
Zhang et al. (2019) [60]	Own (StarGAN/CycleGAN)
Amerini et al. (2019) [61]	FF++

Table 2. Detection model and Dataset used in Detection model

filter applied images that are suitable for each fake attribute, so unnecessary attribute extraction is reduced, and necessary attribute extraction becomes easy. Therefore it is possible to train with 100, 200 images, which are relatively small amounts of images, and possibly build models despite the hardware not being high-end. This process disentangles the restriction which occurs during the deepfake detection study and paves the way for more researchers to develop the model.

2. Related Works

2.1. Fake face generation method

2.1.1. Face synthesis

Face synthesis is a GAN(Generative Adversarial Networks) based method to generate a whole face, including hair. GAN consists of Generator, which generates images and Discriminator, which judges Real/Fake. A generator consists of Encoder and Decoder, where Encoder learns image attributes and decoder reconstructs image based on those attributes. Specifically, there are two face datasets called A and B. Encoder trains facial features in each dataset, then A decoder creates A face, and B decoder creates B face. (Figure 2) Studies that focus on maintaining hairstyle, hair color, eye color, mustache, and facial expressions[63, 64, 65] are actively processed these days.

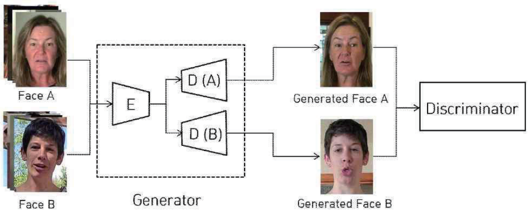


Figure 2. face synthesis algorithm E : Encoder , D (A) : A Decoder, D (B) : B Decoder

2.1.2. face swap

Face swap is a face generation method that swaps a Source face to a Target face in original images/videos. GAN generates faces to convert to Target Face during this process. In the first step, it detects and aligns the Source face in original images/videos, which is an input value of Encoder. The decoder generates Target's Face based on the attributes extracted from Encoder. The generated face is adjusted and inserted into the original image, then smooth the boundary of it. (Figure 3) We can implement this method to both images/Videos and generate faces using various GAN algorithms in Figure 3 (c). However, it may display face color mismatch or resolution mismatch and unnatural synthesized boundary part between the generated face and original image. (Figure 4)

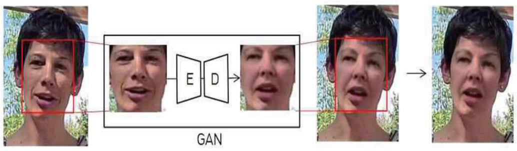
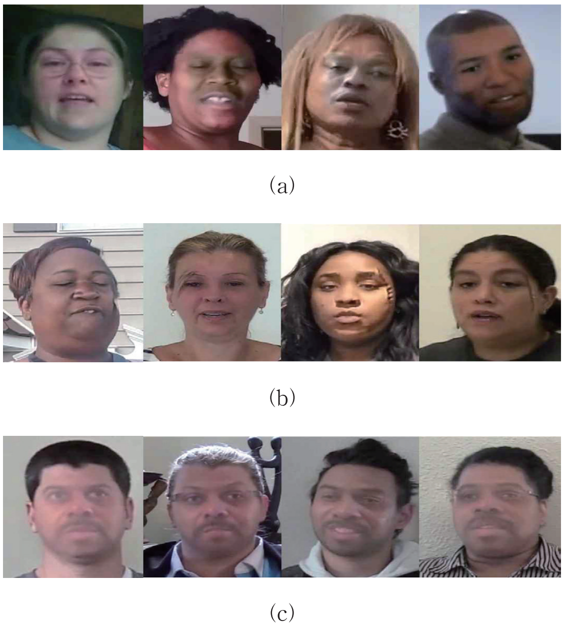


Figure 3. face swap algorithm (a) Face detection (b) Face Crop and Face Align (c) GAN (d) Wrapping face (e) Boundary smoothing


(a)

(b)

(c)

Figure 4. Fake features shown in face swap (a) Face color mismatch between original image and generated face (b) Unnatural synthesized boundary (c) resolution mismatch and unnatural synthesized boundary part between generated face and original image

2.2. Detection Model

2.2.1. Fake face Detection model with similar architecture to this paper

Figure 2 states various fake face detection models. Li et al.[66], Güera and Delp[47], Sabir et al. [48] use CNN and Transformer to attempt fake face detection in fake videos. Li et al.[66] attempts fak e video detection with eye blink interval based on the fact that fake video has less number of eye blink that real video. To do this, CNN extracts features from eye images then Transformer checks eye blink interval.

Güera and Delp[47] attempt to detect fake videos based on attributes shown in videos. CNN extracts features per frame, then Transformer detects fake videos in time flow to train video features.

Sabir et al.[48] has a similar architecture to Güera and Delp[47], but the difference is that it adds Face crop and Face align preprocess to make CNN easy to extract features. Those three detection models incorporate the CNN models and the Transformer models to utilize characteristics in videos to detect unnatural part in temporal flow.

2.2.2. Face detection model with better performance GPU than proposed method

Dang et al.[37] intends to build a model which detects fake face well even if data imbalance problem occurs because there are more real images/videos than fake images/videos. It proposes HF-MANFA, which incorporates MANFA and XGBoost, then attempts to prove the performance through the experiment.

Wang et al.[38] offers a model that can detect well in four types of deformation(noise, blur, compression, resize) in fake images and real images. It proposes a fake face detection model that recognizes neuron activity during learning, and it performs experiments on how well detect those four types of deformations. It also confirms how much the proposed model can identify new data that does not use in learning.

2.2.1. Techniques used in Detection model

2.2.1.1. CNN (Convolutional Neural Network)

We use CNN detector to capture fake attributes, a combination of SSD[67] and Faster-RCNN[68]. Faster-RCNN is 2 Stage-Detector, and it detects fake charactristics in 2 stages. In the first stage, it uses RPN(Region Proposal Network) to draw bounding boxes in an area with fake features while image comes as an input value. The second stage checks if there is any fake feature in bounding boxes through ROI pooling. Faster-RCNN[68] is accurate because it goes the Convolution process twice but has a disadvantage of a slow process.

SSD[67] is a 1 Stage-Detector that detects fake features in only one stage. It confirms if there is any fake feature in pre-sized Default boxes during each Convolution process while the input image comes in. It has an advantage of speed because there is only one convolution process, but the image with less Convolution process has low-dimensional features(straight-line, curve, do), so detection rate possibly decreases in images that require high-dimensional features(eyes, nose, mouth, face).

Considering such attributes, we use the SSD model[67] for features discovered in eyes, nose, mouth then uses the Faster-RCNN model[68] for only a few features found in the face that results in detection model speed increase. It is because the features discovered in eyes, nose, mouth must detect fake features again after eyes, nose, mouth detection. Where there are features found in the face, we used SSD on low-dimensional features and Faster-RCNN on high-dimensional features.

2.2.1.2. Transformer

We use BERT(Bidirectional Encoder Representations from Transformers)[69] for the Real/Fake judgment Transformer model and BERT's fake and real image judgment-making process. It judges Real/Fake based on the relationship in fake features through Self-Attention of Transformer[70], and Feed-Forward Network.

Explicitly, the fake features detected by CNN Detector are computed 'i' times Scaled Dot-Product Attention and determined which features are related. It merges the result of 'i' times computation and multiplies the weight matrix then consolidate (Multi-Head Attention) results. We use the Multi-Head Attention result value of each feature as Position-Wise Feed-Forward Network input value to Figure out how fake features are related.

3. Methods

This paper suggests the detection model consists of

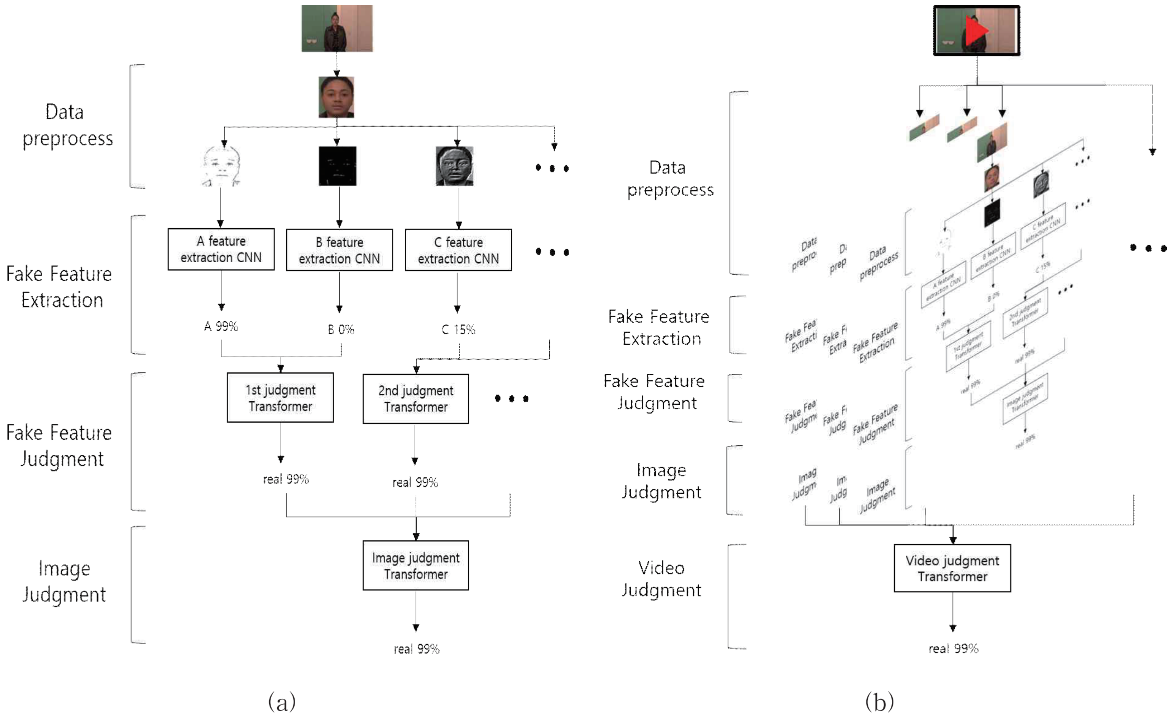


Figure 5. Whole detection model architecture (a) Image detection process (b) Video detection process

five stages: preprocess, detection of fake features, fake features determination, image, and video judgment. The overall model architecture presented in Figure 5, and the model architecture briefly explained step by step.

3.1. Data preprocess

We use face detection to crop facial part when images/videos come in. The most optimized filters for each feature applied in the face images, and then such images are moved to the fake feature detection step. Researchers observe and classify fake feature data to find the most optimized filter combination then find the most suitable filter combination for each feature. The filter combination consists of one filter to multiple filters. When all the real images and fake images are in a sortable state, then nominate the filter as suitable.

3.2. Fake Feature Extraction

We extract fake features discovered in images/videos with the CNN model in this stage. Then we classify the data in each fake feature and construct CNN models that detect such features. Images that applied filter for each feature are used as the input value for the CNN model to classify through whether the feature exists or not. Considering process speed, each model that detects fake features processed in parallel, and the results returns. Various generate methods, and fake features are various in each generation method, so real-time detection is difficult.

3.3. Fake Feature Judgment

The Transformer model judges Real/Fake based on the fake feature information which is previously detected. Fake feature information is bounded to related one and used as Transformer model input value. This process is essential because there is a single fake feature that judges Real/Fake and a combination of multiple fake features that judges Real/Fake.

3.4. Image Judgment

Image judgment uses 3.3(fake feature judgment) result as the input value. Fake feature judgments divide into multiple cases, so the entire decision can be made when image judgment with Transformer. When images come in as input value, image judgment finishes. When the input value is a video, iterates as many times as the number of video frames times the process of 3.1(data preprocess) to 3.4(image judge) and video judgment starts.

3.5. Video Judgment

Video judgment uses the result value of the selected number of frame images. Video is a set of multiple images, so each image's information is gathered to judge the Real/Fake of video.

4. Experiments

4.1. Experimental Environment

4.1.1. Dataset

The Models in Table 3 generates data to construct the detection model training data. Since there are fake images created with various real-world models, more fake feature detection makes a more useful detection model available in reality. Therefore we set detection model training data to include various data in more generate algorithms. We excluded less than 100 generated data or less than 100 numbers of fake features and created CNN Detector using 17 of the total 28 data generated by the algorithms presented in Table 3.

For a straightforward approach and prevent biased results, we use public data called generated photo[71] and celeb-A[72] as fake face data set and real face data.

4.1.2. Evaluation

We used FPR(False positive rate), Recall, AUC(Area Under Curve) for evaluation. FPR, the index of the detection probability where it detects real data as fake data. The Recall is an index of the likelihood of detecting fake data as fake. AUC is an index of the ROC curve(Receiver Operating Characteristic curve) area that its x-axis indicates FPR(1), and the y-axis indicates Recall (2). The model's FPR, Recall represented in the coordinate plane by Threshold, is called the ROC curve. AUC is the calculation of the base area of the graph to compare this graph with other models quantitatively.

$$FPR = \frac{FP}{FP+TN} \qquad (1)$$

$$Recall = \frac{TP}{TP+FN} \qquad (2)$$

The critical part of deepfake detection is to detect fake and real faces, respectively, accurately. Therefore, a high Recall (2) model indicates the probability where it detects the fake data to be fake and a low FPR (1), indicating the probability that real data predicted to be fake is an excellent detection model. Besides, if AUC is closer to 1, it is an excellent detection model because AUC is the ROC curve area, and the ROC curve comprehensively represents by the Threshold.

4.1.3. Training and Test Environment

Training and test environment is stated below.

OS : Ubuntu 18.04
CPU : AMD Ryzen 5 2400G
GPU : Nvidia RTX 2070 SUPER
RAM : DDR4 32GB

4.2. Test for detection rate improvement when fake features are added.

In reality, there are images created with various generate algorithms, and new generate algorithms continuously spread. On the other hand, detection models have lower detection rates when features that have not been learning appear. Therefore, the detection rate should be guaranteed when the detection model improved to prepare for images of new features. To this end, we tested whether the detection rate increased when CNN Detector added to the detection model. The test methods are as follows.

a. Conduct detection of test sets with existing detection models learned from data provided by Facebook [73].
b. Proceed CNN Detector training with the data presented in Table 1 and supplement the model to conduct the test again.
c. Compare the detection rate before and after adding fake features and prove that the proposed model's detection rate increases.

Deepfake creation algorithms	Total number of data	Number of data used to train
Began[11]	512 Images	512
CausalGAN[12]	300 Images	140
faceswap/deepfake[13]	1,576 Images	1,576
StarGAN[14]	2,240 Images	1,747
Enrique Sanchez and Michel Valstar[15]	9,216 images	8,694
MWGAN[16]	1,281 Images	500
ALAE[17]	184 Images	184
StyleGAN[18]	1,000 Images	1,000
MSG-GAN[19]	1,000 Images	1,000
FQGAN[20]	1,000 Images	581
ProGAN[21]	1,000 Images	501
StyleGAN v2[22]	1,000 Images	579
COCO-GAN[23]	1,024 Images	941
VAEGAN[24]	700 Images	637
HoloGAN[25]	640 Images	-
SPA-GAN[26]	401 Images	-
FTGAN[27]	310 Images	-
SEGAN[28]	300 Images	-
StarGAN V2[29]	362 Images	-
LSGAN[30]	100 Images	-
DCGAN[31]	100 Images	-
WGAN[32]	100 Images	-
GAN2play[33]	64 Images	
Glow[34]	3 Images	-
GANnotation[35]	1 Video	-
deferred neural rendering[36]	Webcam	-
neural texture[36]	Webcam	-

Table 3. Total number of data created by deepfake algorithms and data with fake features

Figure 6 states the test result of the trained model with the Facebook dataset and Facebook dataset and dataset presented in Table 3. Recall, which indicates whether fake face data was detected well, was low at 2%, and FPR, which suggests whether real face data was detected incorrectly, was low at 1.6%. AUC, which calculates the base area of the ROC curve that shows Recall and FPR of the detection model according to the Threshold, is low at 0.002. However, Recall increased to 75%, FPR decreased to 0.1%, AUC increased to 0.77 after the model supplement by adding a CNN detector with data stated in Table 3.

The detection rate increases when there are more fake features that can be detected based on test results. The Facebook model can identify 14 fake features, but the model that added CNN Detector as the data in Table 3 can discover 35 counterfeit features. Therefore it proves that detection rate is guaranteed when the CNN detector increases, identifying the fake features.

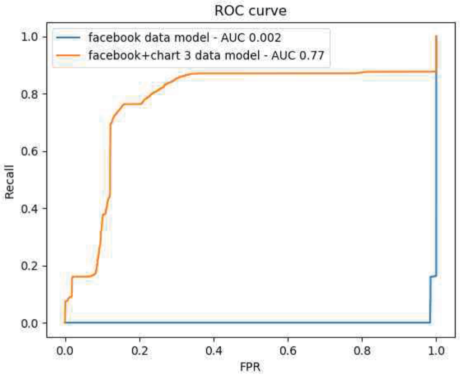


Figure 6. Detection model test result

4.3. Comparison to the model used high-end GPU

We aim to prove that the detection model, which does not use high-end hardware, can perform a high detection rate by comparing them with high-end GPUs. Various researchers need to participate In detecting various fake images in reality, as there are limitations in research if detection models can be learned only with high-end GPUs. Therefore, we aim to ease the constraints of deepfake detection studies and allow more researchers to participate in fake image detection studies.

In the experiment, we compare models using GPUs that performed better than the RTX 2070 GPUs used for model learning [37, 38]. Each model for comparison, as outlined in Table 4, does not experiment using a unified dataset. Moreover, the code is not public, so the comparison is conducted based on AUC, the performance evaluation index listed in the paper. (Table 4)

In the comparison result of the AUC evaluation index of our proposed detection to the detection models in [37] and [38], our proposed model performs approximately 0.13 lower. (Table 5) However, as 4.2 states that the detection model performs better as it adds more fake features than [37], [38].

Therefore, the additional CNN Detector can lead to higher detection rates than detection models usi ng high-end GPUs and requires several researchers to detect various images that appear in real life.

Detection models	GPU	Dataset
Dang et al. [11]	Titan X 12GB GPU	- MANFA dataset - SwapMe and FaceSwap dataset
Wang et al. [12]	Tesla P40 GPU	REAL - CelebA - Flicker-FacesHQ(FFHQ) - FaceForensics++ - DFDC - Celeb-DF FAKE - PGGAN - StyleGAN2 - StarGAN - STGAN - StyleGAN - FF++ - DFDC

Table 4. GPUs and datasets used by detection models to compare

Detection models	AUC
Dang et al. [11]	0.93
Wang et al. [12]	0.906
Proposed model in this paper	0.77

Table 5. Comparison of AUC values between models using high-end GPUs and the model persented in the paper

5. Conclusion

Efforts have made to prevent the abuse of deepfake continuously From 2007 to the present. In this paper, the detection model can be used in practice by a continuous supplement in the detection model consistently, CNN models learn fake features and the detection model structure that final judgment made with Transformer models proposed. Test results for the detection model showed that Recall achieved 85%, FPR 0.1%, and AUC 0.77.

The proposed model mainly focuses on fake features, so there is a limitation when there is a new deepfake creation algorithm. To create a detection model with a high detection rate in new generation algorithms requires the continuous addition of CNN Detector with the participation of researchers.

It is also essential to store and share data, while Transformer integrated training requires previously trained data. If data is not shared, there is a limit to the integrated model creation; otherwise, it creates a non-shared model. To overcome this limitation, we aim to enable data sharing around Github, which has released the detection model code to share data among researchers. We attach the Google drive address to GitHub to allow researchers to download data and upload new data through requests easily. 2)

References

[1] https://www.youtube.com/watch?v=cQ54GDm1eL0 , Accessed : 2020-07-07

[2] https://www.youtube.com/watch?v=RWZmLKw7PG8 , Accessed : 2020-07-07

[3] https://www.youtube.com/watch?v=hKxFqxCaQcM , Accessed : 2020-07-07

[4] Deepfakes porn has serious consequences, https://www.bbc.com/news/technology-42938529 , Accessed : 2020-07-07

[5] How deepfakes undermine truth and threaten democracy, https://www.youtube.com/watch?v=pg5WtBjox-Y , Accessed : 2020-07-07

[6] Fake videos could be the next big problem in the 2020 elections, https://www.cnbc.com/2019/10/15/deepfakes-could-be-problem-for-the-2020-election.html , Accessed : 2020-07-07

[7] https://www.congress.gov/bill/376th-congress/senate-bill/2065/text , Accessed : 2020-07-07

[8] https://www.congress.gov/bill/376th-congress/house-bill/3230/text , Accessed : 2020-07-07

[9] http://www.moj.go.kr/bbs/moj/182/521437/artclView.do , Accessed : 2020-07-07

[10] Ruben Tolosana, Ruben Vera-Rodriguez, Julian Fierrez, Aythami Morales, Javier Ortega-Garcia, "Deep Fakes and Beyond: A Survey of Face Manipulation and Fake Detection", arXiv:2001.00179, 2020, p. 4

[11] David Berthelot, Thomas Schumm, Luke Metz, "Began: Boundary equilibrium generative adversarial networks", arXiv preprint arXiv:1703.10717, 2017

2) github(Entire model code, instruction, Process Video are available): https://github.com/teamnova-ailab/Deepfake-detection-model-based-on-fake-attributes-shown-in-image-video/

[12] Murat Kocaoglu, Christopher Snyder, Alexandros G. Dimakis, Sriram Vishwanath, "CausalGAN: Learning Causal Implicit Generative Models with Adversarial Training", arXiv preprint arXiv:1709.02023, 2017

[13] deepfake/faceswap, https://github.com/deepfakes/faceswap , Accessed : 2020-07-07

[14] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, Jaegul Choo, "StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation", arXiv preprintarXiv:1711.
09020v3, 2018

[15] Enrique Sanchez, Michel Valstar, "Triple consistency loss for pairing distributions in GAN-based face synthesis", arXiv preprint arXiv:1811.03492v1, 2018

[16] Jiezhang Cao, Langyuan Mo, Yifan Zhang, Kui Jia, Chunhua Shen, Mingkui Tan, "Multi-marginal Wasserstein GAN", arXiv preprint arXiv:1911.00888v1 , 2019

[17] Stanislav Pidhorskyi, Donald Adjeroh, Gianfranco Doretto, Adversarial Latent Autoencoders, arXiv preprint arXiv:2004.04467, 2020

[18] T. Karras, S. Laine, and T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks", in Proc. Conference on Computer Vision and Pattern Recognition, 2019.

[19] Animesh Karnewar, Oliver Wang, "MSG-GAN: Multi-Scale Gradient GAN for Stable Image Synthesis ", arXiv preprint arXiv:1903.06048v3, 2019

[20] Yang Zhao, Chunyuan Li, Ping Yu, Jianfeng Gao, Changyou Chen, "Feature Quantization Improves GAN Training", arXiv preprint arXiv:2004.02088v1, 2020

[21] Tero Karras, Timo Aila, Samuli Laine, Jaakko Lehtinen, "Progressive Growing of GANs for Improved Quality, Stability, and Variation", arXiv preprint arXiv:1710.101
96v3, 2018

[22] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, Timo Aila, "Analyzing and Improving the Image Quality of StyleGAN", arXiv preprint arXiv:1912.04958v2, 2020

[23] Chieh Hubert Lin, Chia-Che Chang, Yu-Sheng Chen, Da-Cheng Juan, Wei Wei, Hwann-Tzong Chen, "COCO-GAN: Generation by Parts via Conditional Coordinating", arXiv preprint arXiv:1904.00284v4 , 2020

[24] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, Ole Winther, "Autoencoding beyond pixels using a learned similarity metric", arXiv preprint arXiv:1512.09300, 2016

[25] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, Yong-Liang Yang, "HoloGAN: Unsupervised learning of 3D representations from natural images ", arXiv preprint arXiv:1904.01326v2, 2019

[26] Hajar Emami, Majid Moradi Aliabadi, Ming Dong, Ratna Babu Chinnam, "SPA-GAN: Spatial Attention GAN for Image-to-Image Translation", arXiv preprint arXiv:1908.06616 , 2020

[27] Xiang Chen, Lingbo Qing, Xiaohai He, Xiaodong Luo, Yining Xu, "FTGAN: A Fully-trained Generative Adversarial Networks for Text to Face Generation", arXiv preprint arXiv:1904.05729, 2020

[28] Santiago Pascual, Antonio Bonafonte, Joan Serrà, "SEGAN: Speech Enhancement Generative Adversarial Network", arXiv preprint arXiv:1703.0945
2, 2017

[29] Yunjey Choi, Youngjung Uh, Jaejun Yoo, Jung-Woo Ha, "StarGAN v2: Diverse Image Synthesis for Multiple Domains", arXiv preprint arXiv:1912.018
65v2, 2020

[30] Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, Stephen Paul Smolley, "Least Squares Generative Adversarial Networks", arXiv preprint arXiv:1611.04076, 2017

[31] Alec Radford, Luke Metz, Soumith Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks", arXiv preprint arXiv:1511.06434, 2016

[32] Martin Arjovsky, Soumith Chintala, Léon Bottou, "Wasserstein GAN", arXiv preprint arXiv:1701.07875, 2017

[33] GAN2Play, https://github.com/JimmyYing/GAN2Play , Accessed : 2020-07-07

[34] Diederik P. Kingma, Prafulla Dhariwal, "Glow: Generative Flow with Invertible 1x1 Convolutions", arXiv preprint arXiv:1807.03039v2, 2018

[35] GANnotation, https://github.com/ESanchezLozano/GANnotation , Accessed : 2020-07-07

[36] Justus Thies, Michael Zollhöfer, Matthias Nießner,

"Deferred Neural Rendering: Image Synthesis using Neural Textures", arXiv preprint arXiv:1904.12356v1, 2019

[37] L. Minh Dang, Syed Ibrahim Hassan, Suhyeon Im, Hyeonjoon Moon, "Face image manipulation detection based on a convolutional neural network.", Expert Systems with Applications 389, 156–168, 2019., pp. 159,160,166

[38] Run Wang, Felix Juefei-Xu, Lei Ma, Xiaofei Xie, Yihao Huang, Jian Wang, Yang Liu, "FakeSpotter: A Simple yet Robust Baseline for Spotting AI-Synthesized Fake Faces", arXiv preprint arXiv:1909.06382v2, 2020, pp. 3,4,5

[39] S. McCloskey and M. Albright, "Detecting GAN-Generated Imagery Using Color Cues", arXiv preprint arXiv:1812.08247, 2018.

[40] N. Yu, L. Davis, and M. Fritz, "Attributing Fake Images to GANs: Analyzing Fingerprints in Generated Images", in Proc. International Conference on Computer Vision, 2019.

[41] J. Stehouwer, H. Dang, F. Liu, X. Liu, and A. Jain, "On the Detection of Digital Face Manipulation", arXiv preprint arXiv:1910.01717, 2019.

[42] L. Nataraj, T. Mohammed, B. Manjunath, S. Chandrasekaran, A. Flenner,J. Bappy, and A. Roy-Chowdhury, "Detecting GAN Generated Fake Images Using Co-Occurrence Matrices," arXiv preprint arXiv:1903.06836, 2019.

[43] J. Neves, R. Tolosana, R. Vera-Rodriguez, V. Lopes, and H. Proenc¸a, "Real or Fake? Spoofing State-Of-The-Art Face Synthesis Detection Systems", arXiv preprint arXiv:1911.05351, 2019

[44] F. Marra, C. Saltori, G. Boato, and L. Verdoliva, "Incremental Learning for the Detection and Classification of GAN-Generated Images", in Proc. International Workshop on Information Forensics and Security, 2019

[45] P. Zhou, X. Han, V. Morariu, and L. Davis, "Two-Stream Neural Networks for Tampered Face Detection", in Proc. Conference on Computer Vision and Pattern Recognition Workshops, 2017.

[46] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "MesoNet: a Compact Facial Video Forgery Detection Network," in Proc. International Workshop on Information Forensics and Security, 2018.

[47] D. Güera and E. Delp, "Deepfake Video Detection Using Recurrent Neural Networks", in Proc. International Conference on Advanced Video and Signal Based Surveillance, 2018.

[48] X. Yang, Y. Li, and S. Lyu, "Exposing Deep Fakes Using Inconsistent Head Poses," in Proc. International Conference on Acoustics, Speech and Signal Processing, 2019.

[49] Y. Li and S. Lyu, "Exposing DeepFake Videos By Detecting Face Warping Artifacts," in Proc. Conference on Computer Vision and Pattern Recognition Workshops, 2019

[50] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "FaceForensics++: Learning to Detect Manipulated Facial Images", in Proc. International Conference on Computer Vision, 2019.

[51] F. Matern, C. Riess, and M. Stamminger, "Exploiting Visual Artifacts to Expose DeepFakes and Face Manipulations", in Proc. IEEE Winter Applications of Computer Vision Workshops, 2019.

[52] H. Nguyen, F. Fang, J. Yamagishi, and I. Echizen, "Multi-task Learning For Detecting and Segmenting Manipulated Facial Images and Videos", arXiv preprint arXiv:1906.06876, 2019.

[53] S. Agarwal and H. Farid, "Protecting World Leaders Against Deep Fakes", in Proc. Conference on Computer Vision and Pattern Recognition Workshops, 2019.

[54] Ekraam Sabir, Jiaxin Cheng, Ayush Jaiswal, Wael AbdAlmageed, Iacopo Masi, Prem Natarajan, "Recurrent Convolutional Strategies for Face Manipulation Detection in Videos", arXiv:1905.00582v3 , 2019

[55] A. Bharati, R. Singh, M. Vatsa, and K. Bowyer, "Detecting Facial Retouching Using Supervised Deep Learning", IEEE Transactions on Information Forensics and Security, vol. 11, no. 9, pp. 1903–1913, 2016.

[56] S. Tariq, S. Lee, H. Kim, Y. Shin, and S. Woo, "Detecting Both Machine and Human Created Fake Face Images in the Wild," in Proc. International Workshop on Multimedia Privacy and Security, 2018, pp. 81–87

[57] S. Wang, O. Wang, A. Owens, R. Zhang, and A. Efros, "Detecting Photoshopped Faces by Scripting Photoshop," arXiv preprint arXiv:1906.05856, 2019.

[58] A. Jain, R. Singh, and M. Vatsa, "On Detecting

GANs and Retouching based Synthetic Alterations”, in Proc. International Conference on Biometrics Theory, Applications and Systems, 2018.

[59] F. Marra, C. Saltori, G. Boato, and L. Verdoliva, “Incremental Learning for the Detection and Classification of GAN-Generated Images,” in Proc. International Workshop on Information Forensics and Security, 2019.

[60] X. Zhang, S. Karaman, and S. Chang, “Detecting and Simulating Artifacts in GAN Fake Images”, arXiv preprint arXiv:1907.06515, 2019.

[61] I. Amerini, L. Galteri, R. Caldelli, and A. Bimbo, “Deepfake Video Detection through Optical Flow based CNN”, in Proc. International Conference on Computer Vision, 2019.

[62] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, "Generative Adversarial Networks", arXiv preprint arXiv:1406.2661, 2014

[63] T. Karras, S. Laine, and T. Aila, “A Style-Based Generator Architecture for Generative Adversarial Networks”, in Proc. Conference on Computer Vision and Pattern Recognition, 2019.

[64] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, Jaegul Choo, "StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation", arXiv:1711.09020v3, 2018

[65] Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks",arXiv preprint arXiv:1703.10593v6, 2018

[66] Y. Li, M. Chang, and S. Lyu, “In Ictu Oculi: Exposing AI Generated Fake Face Videos by Detecting Eye Blinking”, in Proc. International Workshop on Information Forensics and Security, 2018.

[62] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, "Generative Adversarial Networks", arXiv preprint arXiv:1406.2661, 2014

[63] T. Karras, S. Laine, and T. Aila, “A Style-Based Generator Architecture for Generative Adversarial Networks”, in Proc. Conference on Computer Vision and Pattern Recognition, 2019.

[64] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, Jaegul Choo, "StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation", arXiv:1711.09020v3, 2018

[65] Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks",arXiv preprint arXiv:1703.10593v6, 2018

[66] Y. Li, M. Chang, and S. Lyu, “In Ictu Oculi: Exposing AI Generated Fake Face Videos by Detecting Eye Blinking”, in Proc. International Workshop on Information Forensics and Security, 2018.

[67] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg, "SSD: Single Shot MultiBox Detector", arXiv preprint arXiv:1512.02325, 2016

[68] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", arXiv preprint arXiv:1506.01497, 2016

[69] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", arXiv preprint arXiv:1810.04805v2, 2019

[70] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, "Attention Is All You Need", arXiv preprint 1706.03762v5, 2017

[71] https://generated.photos/ , Accessed : 2020-07-07

[72] http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html , Accessed : 2020-07-07

Data augmentation for Reinforcement Learning. Brief Survey.

Sanzhar Rakhimkul[2] and Chang Dong Yoo[2]

[2] Korea Advanced Institute of Science and Technology, Daejeon, South Korea. Correspondence to: Sanzhar Rakhimkul
srakhimkul@kaist.ac.kr
[2] Korea Advanced Institute of Science and Technology, Daejeon, South Korea. email: cd_yoo@kaist.ac.kr

Abstract

Deep reinforcement learning agents often fail to perform control tasks in unseen environment scenery directly from learning on high-dimensional states as images. Although convolutional neural network advances in computer vision have shown promising results for successful image classification, the current situation of reinforcement learning agents still lacks data-efficient learning from pixel-based state and generalization to new environments.

Recently, data augmentation techniques demonstrated improved data-efficiency that reached with state-based reinforcement learning agents results and showed state-of-the-art methods across various benchmarks.[2] However, it was empirically proven that different environments and tasks benefit from different kinds of data augmentation techniques. [4] In this paper, we introduce a method that makes a robust training of RL agent with regularizer based on Fourier domain adaptation method. Moreover, we offer a design of efficient learning of a network that augments synthetic images based on influence function and randomized convolution network.

Keywords— *Data augmentation, reinforcement learning, model-free RL*

I. Introduction

In deep reinforcement learning (RL), the recent development of techniques leads to the successful training of agents in simulations. For example, tasks such as playing video games, continuous control of the simulated car from a dashboard camera, and robot control that can grasp, push and pick complex objects in a cluttered environment were a notable success that discovered in recent years. However, when agent trains to perform complex tasks such as continuous multi-tasks of a robotic arm using high dimensional observations such as pixels, it frequently fails to generalize to unseen situations.

Usually, reinforcement learning agent training with the limited environmental interaction in a real-world application as a robotics control is a challenging task. Consequently, frequently the attempts for learning more efficiently on data result in severe over-fitting and limited task performance. Besides, it has empirically shown that reinforcement learning algorithms that use high dimensional observations such as pixels have a lower learning rate compared to the state observation mainly because of sample-inefficiency of the methods.

Data augmentation methods have proven high effectiveness in computer vision and speech fields. [5] The approach for reinforcement learning field has shown improved results in data-efficiency and also in the overall performance of agents.[2] Simple implementation and no additional auxiliary loss provide robust training over various environments for all standard model-free actor-critic algorithms. The nowel papers present that preserving Q-function for augmented and original images leads to faster convergence of agents.[1] More researchers focus on approaches on how to combine or choose augmentation techniques for specific environments and tasks. [4] In the paper, we focus on the current trends in data augmentation techniques for reinforcement learning agents and what are the potential strategies to improve the performance of the system.

II. Literature Review

A. *CURL: Contrastive Unsupervised Representations for Reinforcement Learning*

This work aims to answer the following question - can pixel-based RL be as efficient as RL from coordinate state? Traditionally, it has been widely assumed that pixel-based RL is data inefficient, often taking 100M+ interaction steps to solve benchmark tasks like Atari games. On the contrary, they show for the first time that the answer is yes. [6]

CURL learns contrastive representations jointly with the RL objective. The representation learning is done as an auxiliary task that can be coupled to any model-free RL algorithm. In the paper, they combine contrastive representation learning with two state of the art algorithms (i) Soft Actor Critic (SAC) for continuous control and (ii) Rainbow DQN for discrete control. Contrastive representations are learned by specifying an anchor observation, and then maximizing / minimizing agreement between positive / negative pairs through Noise Contrastive Estimation. A high-level diagram of CURL is shown above. The CURL is the first image-based algorithm to nearly match the sample-efficiency of methods that use stat-based features. The techniques can be used in area like robotics where data-efficiency is paramount. In the research, contrastive learning will have a part as encoder-decoder networks because we believe that it produces the most accurate hidden space representation of high dimensional states.

B. Reinforcement Learning with Augmented Data

In the paper, authors show how data augmentation improves performance and generalization abilities of standard RL algorithms, both on and off-policy. They combine data augmentation with (i) Soft Actor-Critic (SAC) for solving tasks on DeepMind control and (ii) PPO for ProcGen environments. The method does not change the underlying RL pipeline - it only augments the underlying data.[2]

Random crop, stand-alone, has the highest impact on the final performance relative to all other augmentation on DeepMaind control. The paper is a SOTA algorithm that can be applied on top of all reinforcement learning algorithms. Based on the results of the paper, we will use the augmentation techniques with an auxiliary neural network as an influence function to choose the augmentation technique for each environment. We believe that the augmentation as random cropping has a high impact on the way neural networks train useful features for the majority of simulators, even though that augmentation is not the best technique for all. Thus, we believe in an extension of the idea of data augmentation with the neural network to choose the best method for each environment.

C. Image Augmentation Is All You Need: Regularizing Deep Reinforcement Learning from Pixels

In recent research papers related to reinforcement learning techniques based on pixel-based model-free algorithms, they utilize auxiliary losses or pre-train that forces each algorithm to be modified specifically based on its structure. The paper states that high capacity level encoder could lead to severe over-fitting while using lower size encoder could have a low performance for the unseen environment. Author say that, there are a number of approaches currently present in the research world to deal with the learning from pixels problems such as pre-training using self-supervised learning, training with auxiliary losses, and data augmentation. The authors approach using a simple data augmentation technique that improves data efficiency and makes learning faster. While using the augmentation technique, it can reach SAC's state performance on pixel-based SAC using the augmentation method. They called it as DrQ: Data-regularized Q, which can be combined with any model-free reinforcement learning algorithm. In more detail, the authors' key idea of using standard image transformation as well as regularize the Q-function learned by the critic so that different transformations of the same input image have similar Q-function values. While in the paper mentioned various amount of image transformations to the input image the most important one was random shifts, which shows a good balance between simplicity and performance.[1]

Furthermore, to fully exploit the MDP structure, they introduced the optimality invariant state transformation. For every state using transformation, it was possible to generate several surrogate states. Thus the Q-target function should be regularized with the augmented data. They took an averaging the Q-target over K image transformations and Q-function itself over M image transformation. The paper introduced a simple regularization technique that can be easily applied to any model-free reinforcement learning algorithms and get fast generalization and robust training. To conclude the results of the paper, we are feeling that introducing another regularizer to preserve the optimality invariant state the data-efficiency will lead to robust data-efficient training.

D. Automatic Data Augmentation for Generalization in Deep Reinforcement Learning

In the paper, the authors propose UCB-DrAc, a method for automatically determining an effective data augmentation for RL tasks. The main idea is that the upper confidence bound (UCB) algorithm considers the number of augmentation techniques as a multi-armed bandit problem and chooses based on Q-function value and number of times transformation has been selected. Augmentation transformation functions synthesize additional observations for every environment that increase the data-efficiency of images. The method improves training performance in Procgen benchmark on both train and test environments and outperforms recent papers as RAD and CURL. The UCB-DrAc is robust to numerous variations in a simulated environment and SOTA algorithm for continuous and discrete action space among all popular benchmarks.

III. Conclusion

RL agent in model-free learning that has a high-dimensional pixel-based observation matches the state-based observation level when the data augmentation techniques applied to generate transformed images. The simple idea without any additional auxiliary loss and regularization techniques shows decent results. Although in recent research, it proved that preserving Q-function values for the different transformed images as well original image could lead to further data-efficiency and robust learning for complex environments. Further, there are papers which state that applying different augmentation function for various environments leads to higher results. That paper ”Automatic Data Augmentation for Generalization in Deep Reinforcement Learning” is a SOTA algorithm that has a potential for extension by introducing a differentiable transformation matrix. The transformation generalizes all kinds of augmentation techniques presented in various papers into a single function with various parameters.[3] I believe that the transformation function for augmentation with the UCB-DrAc algorithm will generalize over the different environments.

REFERENCES

[1] Ilya Kostrikov, Denis Yarats, and Rob Fergus. Image Augmentation Is All You Need: Regularizing Deep Reinforcement Learning from Pixels. apr 2020.

[2] Michael Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement Learning with Augmented Data. apr 2020.

[3] Donghoon Lee, Hyunsin Park, Trung Pham, and Chang D. Yoo. Learning augmentation network via influence functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[4] Roberta Raileanu, Max Goldstein, Denis Yarats, Ilya Kostrikov, and Rob Fergus. Automatic data augmentation for generalization in deep reinforcement learning, 2020.

[5] Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, Jul 2019.

[6] Aravind Srinivas, Michael Laskin, and Pieter Abbeel. CURL: Contrastive Unsupervised Representations for Reinforcement Learning. apr 2020.

Semi-Supervised Classification via Deep Metric Soft-max Learning

Sungmin cho[1], Dohwi Kim[2] and Junseok Kwon[1]

[1] School of Computer Science and Engineering, Chung-Ang University, Seoul, Korea, {csm8167, jskwon}@cau.ac.kr
[2] Thermoeye Co., Ltd., Seoul, Korea, oranss@thermoeye.co.kr

Abstract

We present a novel method that can solve semi-supervised learning problems via deep metric learning. By minimizing the embedding distance between unlabeled data and sample labeled data, the proposed deep neural network leverages the unlabeled data information effectively. In experiments, we demonstrate that our method is comparable to state-of-the-art methods on the MNIST dataset. Using t-SNE, we analyze the data locations in the embedding space and verify that unlabeled data are accurately classified.

Keywords— *Semi-supervised learning, Deep metric learning*

I. Introduction

Semi-supervised learning tasks aim to accurately train deep neural networks using only a small amount of labeled data and unlabeled data [2], where the key to semi-supervised learning is how to appropriately leverage unlabeled data features. Semi-supervised learning has been applied to diverse areas, in which unlabeled data are available, such as face recognition [4], image retrieval [1], and neural language parsing [11].

In this paper, we propose a novel semi-supervised learning method based on deep metric learning [6], where unlabeled data that are close to labeled sample data in the embedding space considerably improve the classification accuracy. For this objective, we modify the soft-max function proposed by [2] into a metric soft-max function, which uses the largest probability value of the unlabeled data. Because semi-supervised learning has a small amount of labeled data and can cause data imbalance, we adopt the focal loss [12] to address this imbalance problem.

our contribution are two-folds as follows:

- We propose a novel metric classification loss based on a metric soft-max function in a semi-supervised learning framework, in which we measure the similarity between *probability distributions* of unlabeled data and labeled sample data in embedding space.
- We enhance the classification loss using the focal loss.

II. Related work

A. semi-supervised learning

Semi-supervised learning is in between supervised and unsupervised learning, in which datasets are typically divided into labeled and unlabeled data [2]. The key issue in semi-supervised learning is to use labeled data information efficiently to handle unlabeled data. Thus, many methodologies attempted to solve this issue as follows. Using generative models, a new labeled data was generated from unlabeled data by variational autoencoder(VAE) [9], which follows original labeled data distributions. A simple regularization method [16] has been proposed, which generalizes a small training set by inducing noise within the training set, which helps improve the semi-supervised learning performance. Label propagation approaches [7, 8] were proposed for semi-supervised learning. The former proposed a transductive label propagation method satisfies the manifold assumption and can generate pseudo-labels for unlabeled data. The latter used the dynamic graph of labeled and unlabeled data for label propagation.

In contrast to these methods, we solve semi-supervised learning problems in a metric learning framework and present a novel deep neural network.

B. deep metric learning

Metric learning aims to find the optimal distance function to measure the similarity among samples accurately. Deep metric learning uses deep neural networks for metric learning. Chopra *et al*. [3] proposed a Siamese CNN architecture, which extracts features of two images on the embedding space and measures the l2-norm distance between two features. Schroff*et al.* [15] presented a triplet loss-based deep metric learning method to align matching and non-matching face, in which anchor, positive, and negative sample were employed. Using the triplet loss, same class data are close to each other, whereas different class data are far from each other in the embedding space

In contrast to these methods, we used a metric soft-max function To leverage unlabeled data information in the course of deep metric learning process.

III. Proposed Method

The goal of the proposed method is to perform classification and metric learning tasks at the same time. To leverage unlabeled data information, we propose a metric softmax-based objective function, which combines cross entropy with focal loss.

A. metric softmax

We denote the labeled data $X_L = \left\{(x_l^i, y_l^i)\right\}_{i=1}^{n}$ and unlabeled data $X_U = \{(x_u^i)\}_{i=n+1}^{k}$, where $x_l, x_u \in \mathbb{R}^D$, $y_l \in \{1,2,...,C\}$, C is a number of classes, D is a input dimension, and k is a total size of dataset. For training, we use sample data $\{z_c\}_{c=1}^{C} \in x_l$, which contains all classes.

For the labeled data, we adopt a softmax function S to convert features into discrete probability distributions.

$$\mathrm{S}\left(f\left(x_l\right)\right) = \frac{e^{f(x_l)}}{\sum_{\forall \text{ class}} e^{f(x_l)}}, \tag{1}$$

where $f\left(x_l\right)$ denote the deep neural network that extracts features of x_l. For unlabeled data, we use a subset of labeled data, z_s, which are the closer to unlabeled data x_u than the other labeled data, z_r, in the embedding space, as follows:

$$d(x_u, z_s) < d(x_u, z_r), \tag{2}$$

where d denotes a l2-norm distance, as follows:

$$d(x_u, z_c) = ||x_u - z_c||_2. \tag{3}$$

We propose a metric softmax function S_{metric}, which converts the distance between unlabeled and labeled data into discrete probability distributions in the embedding space.

$$\mathrm{S}_{metric}\left(f(x_u), f(z_c)\right) = \frac{e^{-d(f(x_u), f(z_c))}}{\sum_{\forall \text{ class}} e^{-d(f(x_l), f(z_c))}}, \tag{4}$$

where a small distance induces a large probability value in the proposed metric softmax function S_{metric}.

B. objective function

Given $x = \begin{cases} x_l(labeled) \\ x_u(unlabeled) \end{cases}$, we use (x_l, y_l) for training, if x is a labeled data, Otherwise, we use (x_u, z_c). We design the label loss $\mathscr{L}_l$ as a traditional cross entropy loss.

$$\mathscr{L}_l = -\sum_{i=1}^{n} y_l^i \log \mathrm{S}\left(f(x_l^i)\right). \tag{5}$$

We calculate the unlabel loss $\mathscr{L}_u$, as follows:

$$\mathscr{L}_u = -\sum_{i=1}^{n} p_t^i \log(p_t), \tag{6}$$

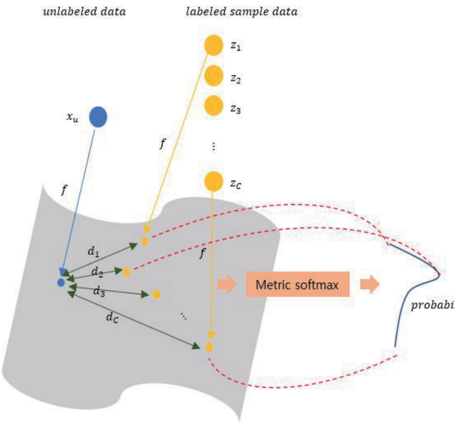


Fig. 1. **Basic idea of the proposed metric softmax**. we embed data using our network f and measure the distance between *probability distributions* of unlabeled data and labeled sample data in embedding space.

where $p_t = \mathrm{S}_{metric}(f(x_u), f(z_c))$ in (4). We enhance $\mathscr{L}_u$ in (6) into $\mathscr{L}_u^f$ using a focal loss to make labeled data more influential in the loss calculation, because labeled data is much less than unlabeled data.

$$\mathscr{L}_u^f = -\sum_{i}^{n} (1 - p_t)^{\alpha} p_t^i \log(p_t), \tag{7}$$

where α is a hyper-parameter. Then, the proposed classification loss is as follows:

$$\mathscr{L} = \mathscr{L}_l + \mathscr{L}_u^f. \tag{8}$$

C. Network Architecture

Module	28 × 28 grayscale
Conv-ReLU-BN (16, 3×3, 1×1)	16 × 28 × 28
Max-Pooling (2,×2, 2×2)	16 × 14 × 14
Conv-ReLU-BN (32, 3×3, 1×1)	32 × 14 × 14
Conv-ReLU-BN (32, 3×3, 1×1)	32 × 14 × 14
Conv-ReLU-BN (64, 3×3, 1×1)	64 × 14 × 14
Conv-ReLU-BN (64, 3×3, 1×1)	64 × 14 × 14
Max-Pooling (2,×2, 2×2)	64 × 7 × 7
Conv-ReLU-BN (128, 3×3, 1×1)	128 × 7 × 7
Conv-ReLU-BN (128, 3×3, 1×1)	128 × 7 × 7
FC-ReLU (128)	128
FC-ReLU (10)	10

Table 1. **The proposed convolutional network architecture**. $(16, 3 \times 3, 1 \times 1)$ denotes the number of means channels, filter size, and stride, respectively.

We implement our network f using a few convolutional neural network layers and fully connected layers, which is similar to that of [6]. Table 1 shows the detailed architecture of the proposed network.

IV. Experiments

For training, we used the MNIST dataset and used 1000 labeled data and 59000 unlabeled data, where 100 for each class. For testing, we used 10000 test data. We adopted the SGD optimization and set the initial learning rate to $1e-4$ and the batch size to 16. Experiments were performed on a Windows 10 64-bit platform with Intel CPU i7 3.60 GHz with NVIDA GeForce GTX 1080 Ti. The proposed method was implemented using Python 3.7 and Pytorch 1.5.0.

A. *Quantitative Comparison*

Module	Test error %
DGN (Kingma *et al.*) [9]	2.40
Pseudo-label (Lee *et al.*) [10]	3.46
SSL with GAN(Odena *et al.*) [14]	3.36
Virtual Adversarial (Miyato *et al.*) [13]	1.36
Association (Haeusser *et al.*) [5]	**0.74**
Ours	1.48

Table 2. **Quantitative results** of semi-supervised classification for the MNIST dataset. We used 1000 labeled data. The best results were written in boldface.

Table 2 shows that our method is the second-best algorithm and quantitatively comparable to other state-of-the art methods.

B. *Metric learning visualization*

Figure 2 shows t-sne results of the proposed metric learning, where the colored points represent labeled data and gray color points are unlabeled data. As shown in the figure, unlabeled data were well clustered based on labeled data.

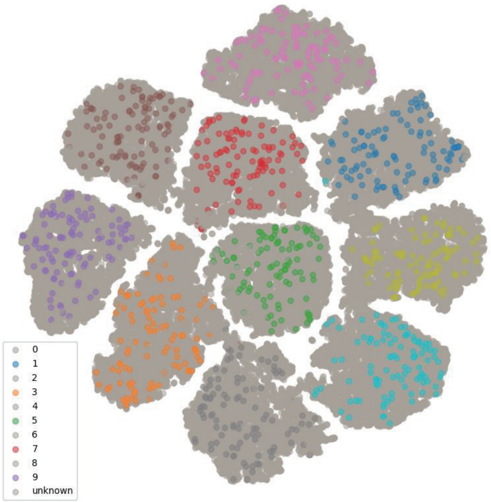


Fig. 2. **t-SNE** illustrates the locations of each unlabeled data and labeled data after learning in the embedding space.

V. Conclusion

We presented a novel method that solve semi-supervised learning problems using deep metric learning. By reducing the distance of the embedding space between unlabeled data and sample labeled data, The networks are able to leverage information of unlabeled data effectively.

Acknowledgement

This research was supported by the the Seoul R&BD Program (IC190027).

References

[1] Cheng Chang, Himanshu Rai, Satya Krishna Gorti, Junwei Ma, Chundi Liu, Guangwei Yu, and Maksims Volkovs. Semi-supervised exploration in image retrieval. *arXiv preprint arXiv:1906.04944*, 2019.

[2] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

[3] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *IEEE conference on computer vision and pattern recognition*, volume 1, pages 539–546, 2005.

[4] Yuan Gao, Jiayi Ma, and Alan L Yuille. Semi-supervised sparse representation based classification for face recognition with insufficient labeled samples. *IEEE Transactions on Image Processing*, 26(5):2545–2560, 2017.

[5] Philip Haeusser, Alexander Mordvintsev, and Daniel Cremers. Learning by association–a versatile semi-supervised training method for neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 89–98, 2017.

[6] Elad Hoffer and Nir Ailon. Semi-supervised deep learning by metric embedding. *arXiv preprint arXiv:1611.01449*, 2016.

[7] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *IEEE conference on computer vision and pattern recognition*, pages 5070–5079, 2019.

[8] Konstantinos Kamnitsas, Daniel C Castro, Loic Le Folgoc, Ian Walker, Ryutaro Tanno, Daniel Rueckert, Ben Glocker, Antonio Criminisi, and Aditya Nori. Semi-supervised learning via compact latent space clustering. *arXiv preprint arXiv:1806.02679*, 2018.

[9] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pages 3581–3589, 2014.

[10] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks.

[11] Percy Liang. *Semi-supervised learning for natural language*. PhD thesis, Massachusetts Institute of Technology, 2005.

[12] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE international conference on computer vision*, pages 2980–2988, 2017.

[13] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, Ken Nakae, and Shin Ishii. Distributional smoothing with virtual adversarial training. *arXiv preprint arXiv:1507.00677*, 2015.

[14] Augustus Odena. Semi-supervised learning with generative adversarial networks. *arXiv preprint arXiv:1606.01583*, 2016.

[15] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[16] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

A survey on meta learning

Thanh X. Nguyen, Chang D. Yoo
Korea Advances Institute of Science and Technology (KAIST)
Daejeon, Korea
thanhnguyen@kaist.ac.kr

Abstract

Humans have the ability to learning new concepts and skills very efficiently from a few examples. In machine learning, meta learning is the research field that aims to achieve this ability, also known as learning to learn. This paper provides a brief survey on meta learning with various approaches and its application on classification and reinforcement learning

Keywords— Meta learning, Machine learning, classification,

I. Introduction

Meta learning is a subfield of machine learning where the algorithm is trained to infer a learning algorithm that works well on an unseen task with very few examples based on meta data. In meta learning, there are two phases: the training phase and the adapting phase. In the training phase, a distribution of tasks is given and the meta learning algorithm is trained to learn how to learn a task, also known as learning to learn. In the adapting phase, an unseen task is given and the meta learning algorithm needs to provide a learning algorithm that can solve the unseen task efficiently. Meta learning can be applied for reinforcement learning or supervised learning, such as classification.

There are three common approaches of meta learning: metric-based, model-based and optimization-based. The metric-based meta learning is similar to nearest neighbor algorithms. It tried to learning a good common kernel to embed the samples of tasks into the latent space in which we can easily distinguish samples using a distance function. The candidates for this approaches are Convolutional Siamese Network [1], Matching Network [2], Relational Network[7], Prototypical Network[8]. Model based meta learning approach is the methods that try to extract and maintain the commonality between tasks and store it inside the model for future use. Therefore, these methods usually have memory in their architecture (external memory or internal memory) or have a special type of weight (fast weight or slow weight). Memory-Augmented Neural Networks [3], Meta Network [4] are the candidates for this approach. Optimization Based meta learning focus to learn the optimization process. It can be learning a good initialization or learning a good optimizer for a group of tasks. LSTM Meta-Learner [5], Model Agnostic Meta Learning [6] and Reptile[9] represent for this approach.

This paper will go through some of the methods, representing for each approach mentioned above. We not only provide a vast knowledge of meta learning to provide an overview of the meta learning but also provide a closed look to each approach to help the reader distinguish between methods and approaches. This survey will be very helpful for new researchers to begin first step in meta learning field and for senior researches to look back on their knowledge. Engineers can also refer to this paper to choose a good algorithm for their product.

II. Meta learning Problem

In this session, we will define the meta learning problem in detail and provide some common terminology. As mentioned before, the meta learning model usually is trained over a distribution of tasks. These tasks need to share some commonality. Each task is associated with a data set D which contains both input features and ground truth labels. The optimal model will be acquired by:

$$\theta^* = argmax_\theta E_{D\sim p(D)}[L_\theta(D)\,]$$

In supervised learning, few-shot classification is an instance of meta-learning. The dataset D is often the combination of support set S and prediction set B. There is a terminology which is called K-Shots N-ways classification task: the support contain K labeled examples for each of N classes

In meta reinforcement learning, the dataset D contains the trajectories given a specific task which is generated by the current policy.

Training process for meta learning is a little bit more complicated than traditional supervised methods. Let's say, we have D = {{xi,yi}}, our classifier is f_θ which output the label distribution given x, $P(y|x)$. The objective function will be:

$$\theta^* = argmax_\theta E_{(x,y)\sim D}[P(y|x)]$$

The step by step procedure is

1. Sample a subset of tasks in D

2. Get the corresponding support set and corresponding prediction set
3. The support set and input feature of prediction set are the model input
4. Compute the loss that minimizes the difference between model input and ground true label. Update model parameters through backpropagation
5. Repeat 1-> 4 until model is converged

III. METRIC BASED

The idea of metric based meta learning is to learn a good kernel k_θ. Given the predicted input x, the predicted y depends on the distance or the similarity of input to the images in the support set after embedding them into the latent space by kernel k_θ.

A. Convolutional Siamese Neural Network

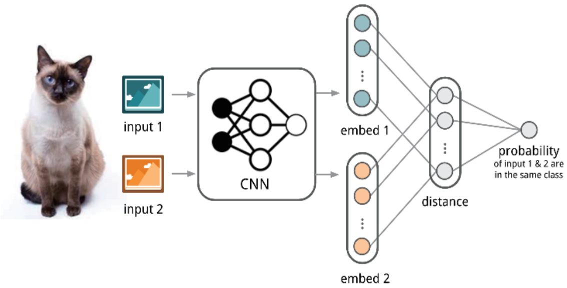


Figure 1 The illustration of Convolutional Siamese Neural Network

Siamese Neural Network is composed of two identical network branches which share weight with each other and receive two input feature vectors. Convolution Siamese Neural Nework uses two branches of convolutional network to embed two input images to feature space and then compare distance between them to see they are in same class are not. The illustration of Convolution Siamese Neural Network is shown in the figure 1. In the testing phase, the tested image is compared in sequence with every image in the support set. The final class of the input is the class of support image which gives the lowest distance.

B. Matching Networks

Matching networks defines a probability distribution over output labels y given a test example x conditioned on the support set. The classifier output is the sum of labels of support samples weighted by attention kernel $k(x, x_i)$ which is proportional to the similarity between x and x_i. The illustration of matching network is shown in the Figure 2.

$$P(y|x,S) = \sum_{(x_i,y_i)\in S} k(x,x_i)y_i \quad ,where\ S = \{(x_i,y_i)\}_{i=1}^{k}$$

The attention kernel is a function of support set feature encoding function, g, and input feature encoding function. One example of attention kernel is the cosine similarity

$$k(x,x_i) = \frac{\exp(cosine(f(x),g(x_i)))}{\sum \exp(cosine(f(x),g(x_j)))}$$

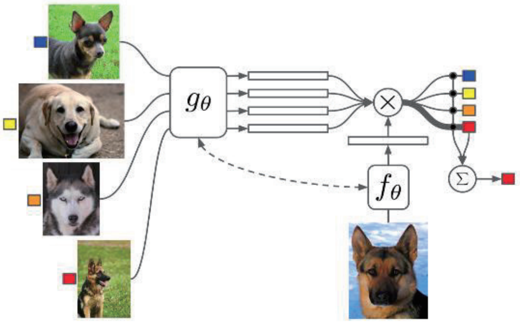


Figure 2 The illustration of Matching Network

IV. MODEL BASED

Model based meta learning depends on a model designed for fast learning. The fast learning can be achieved by internal architecture or embedded memory

A. Memory-Augmented Neural Networks

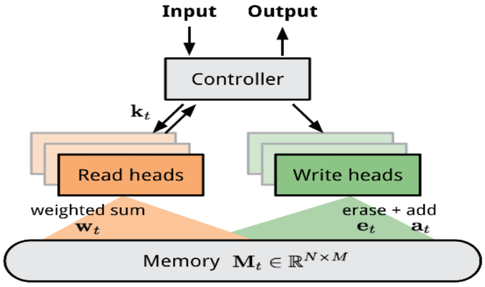


Figure 3 General Architecture of External Memory based meta learning

Memory Augmented Neural Networks use an external memory storage to enhance learning process. With an external memory, the network easier to remember the commonality between tasks, and quickly incorporate new information for new task in testing phase. The general architecture of these networks are presented in Figure 3

B. Meta Networks

Meta networks designed a meta-learning architecture that can rapid adaptive to new task. The network is the combination of slow weights and fast weights. The slow weight are updated by normal stochastic gradient descent. The slow weight aim to store the commonality between tasks. The fast weights are generated by a neural network which utilize the meta information between task. The fast weights aim for adapt to different task.

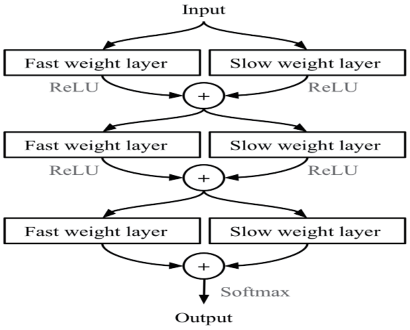


Figure 4 Fast weight and slow weight architecture

V. Optimization Based

Optimization based method change the optimization process by either learn a good optimizer which suitable for meta learning or generate a good weight initialization

A. LSTM Meta Learner

The meta learner is modeled as a LSTM which sequentially received gradient of loss, loss and the weight to produce the updated weight.

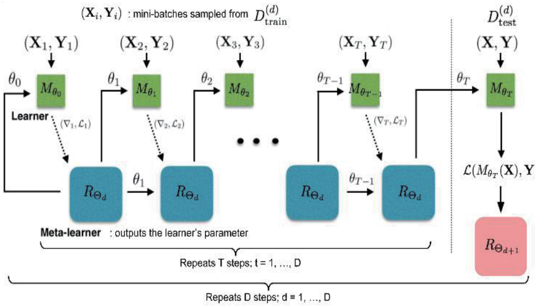

$(\mathbf{X}_i, \mathbf{Y}_i)$: mini-batches sampled from $D^{(d)}_{\text{train}}$
$D^{(d)}_{\text{test}}$
$(\mathbf{X}_1, \mathbf{Y}_1)$
$(\mathbf{X}_2, \mathbf{Y}_2)$
$(\mathbf{X}_3, \mathbf{Y}_3)$
$(\mathbf{X}_T, \mathbf{Y}_T)$
$(\mathbf{X}, \mathbf{Y})$
θ_0
M_{θ_0}
θ_1
M_{θ_1}
θ_2
M_{θ_2}
θ_{T-1}
$M_{\theta_{T-1}}$
θ_T
M_{θ_T}
Learner
$(\nabla_1, \mathcal{L}_1)$
$(\nabla_2, \mathcal{L}_2)$
$(\nabla_T, \mathcal{L}_T)$
R_{Θ_d}
θ_1
R_{Θ_d}
R_{Θ_d}
θ_{T-1}
R_{Θ_d}
$\mathcal{L}(M_{\theta_T}(\mathbf{X}), \mathbf{Y})$
$R_{\Theta_{d+1}}$
Meta-learner : outputs the learner's parameter
Repeats T steps; t = 1, ..., D
Repeats D steps; d = 1, ..., D

Figure 5 LSTM meta learning architecture

B. MAML

Model-Agnostic Meta-learning (MAML) is the paper that focus to learn an initialization for model weight which is suitable for a group of task. Using this weight initialization, the meta learner can quickly find the optimal weight with very few gradient step. MAML can be applied easily to few shot learning for computer vision task and reinforcement learning of regardless of model architecture.

VI. Conclusion

In this paper, we provide a quick look over the meta learning research field. Due to the limit length requirement of the paper, we cannot provide all detail information. Readers who want to get more detailed information can refer to the reference list. Meta learning is emerging and gradually get more attention from researchers. The future research directions for meta is diversity. However, we think that improving meta generalization, multi-modality methods and reduce computational cost is a good way to go. We hope that the paper provides good general knowledge and terminology to help the reader approach meta learning and make your great research in the future.

References

[1] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. “Siamese neural networks for one-shot image recognition.” ICML Deep Learning Workshop. 2015.

[2] Oriol Vinyals, et al. “Matching networks for one shot learning.” NIPS. 2016.

[3] Adam Santoro, et al. “Meta-learning with memory-augmented neural networks.” ICML. 2016.

[4] Tsendsuren Munkhdalai and Hong Yu. “Meta Networks.” ICML. 2017

[5] Sachin Ravi and Hugo Larochelle. “Optimization as a Model for Few-Shot Learning.” ICLR. 2017.

[6] Chelsea Finn, Pieter Abbeel, and Sergey Levine. “Model-agnostic meta-learning for fast adaptation of deep networks.” ICML 2017

[7] Adam Santoro, A simple Neural Network module for relational reasoning, CoRR 2017

[8] Snell, Jake, Kevin Swersky, and Richard Zemel. "Prototypical networks for few-shot learning." Advances in neural information processing systems. 2017.

[9] Nichol, Alex, Joshua Achiam, and John Schulman. "On first-order meta-learning algorithms." arXiv preprint arXiv:1803.02999 (2018).

The Impact of Attention Mechanism in Fusion Step for Visual Question Answering

Trung X. Pham[1] and Chang D. Yoo[1]

[1] Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, {trungpx, cd_yoo}@kaist.ac.kr

Abstract

This paper reviews and investigates the effectiveness of the attention mechanism in the input embedding and output fusion step of a state-of-the-art deep learning architecture in the visual question answering (VQA) problems. In various computer vision tasks, attention techniques bring the huge improvements and interpretable vision by indicating the most important regions in the input that contribute to the discriminative features for classification and recognition. We focus on a recent study “Deep Modular Co-Attention Networks” that applied the self-attention with multi-head approach in both text and image feature domains, establishing the new state of the art in the challenging benchmark dataset VQA-v2. To this end, a proposal extension to the existing architecture with an attention block in the fusion step is experimentally conducted showing the improvements and superior capability in answering the visual questions over the original model.

Keywords— *VQA, Deep Learning, Self-Attention*

I. Introduction

One of the most challenging and interesting computer vision problems is the visual question answering (VQA) that requires both understanding in visual content of image and the textual information in the questions to infer the correct answers [1,2], Figure 1. VQA task is more difficult than other vision-language tasks such as visual captioning [3,4] and visual grounding [5] when VQA needs to attain well the understanding concurrently image and meaning of the text in question and predict a correct answer. To resolve this multimodal learning problem, Yu et. al [6] introduced an attention-based framework that can learn the interaction between the text and the image clues so that it can model the intramodal word to region to improve the deep reasoning for better final performance. This work is inspired by the self-attention (SA) in Transformer structure in the machine translation task [7], and modified the SA block to get another variant block with guided-attention (GA) to get more interaction between text and image modalities. At the end of each modality stream, the authors in [6]] used a fusion by addition or concatenation to get the final feature before classification. This conventional fusion methods in the [6], however, didn’t reflect the strong relation between the two modalities. We propose a simple version of a learnable fusion that can emphasize which modality is more important, and the more experiments show the effectiveness of the proposed fusion method by outperforming the original model with +0.12% accuracy in overall when evaluating on VQA-v2 dataset.

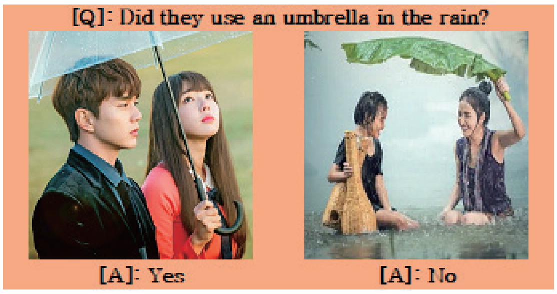


Fig. 1. Two examples of the visual question answering task. Inputs are an image and a corresponding question, output is one of the answers: yes or no?

II. Method

A. *Original network*

Each modality is fed into one separate learning stream and they are fused at the end of the process followed by a fully connected layer.

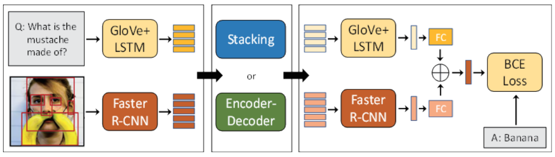


Fig. 2. The original framework for VQA task with the conventional fusion (add or concatenation) before the final classification layer.

B. *Attention Blocks Construction*

The multi-head attention has been applied showing the big impact in learning the presentation of the text data. With the query $q \in \mathbb{R}^{1\times d}$, a key matrix $K \in \mathbb{R}^{n\times d}$ and a value matrix $V \in \mathbb{R}^{n\times d}$, the attention weight over the V is determined by the following equation:

$$S = \text{Attention}(q, K, V) = \sigma\left(\frac{qK}{\sqrt{d}}\right).V \qquad (1)$$

Where σ is softmax or sigmoid function that outputs the value between 0 and 1. The multi-head composes of several paralleled heads which specified by the scaled dot-product function:

$$\text{head}_j = \text{Attention}(qW_j^Q, KW_j^K, VW_j^V) \qquad (2)$$

Where W is the matrix projected for the j^{th} head.

C. *Proposed Network Extension*

To manipulate the relation between the two stream outputs, a learnable alpha is constructed to fuse them (Figure 3). The original structure fused two output by conventional way $O = \sum_{i=1}^{2} FC_i(x_i)$, we extended this to an attention-based fusion:

$$O = \sum_{i=1}^{2} \alpha_i FC_i(x_i) \qquad (3)$$

where α is the learnable weight which can learn via the backpropagation process. FC stands for "fully connected" which indicates for the two outputs of the two streams corresponding to text and image modalities.

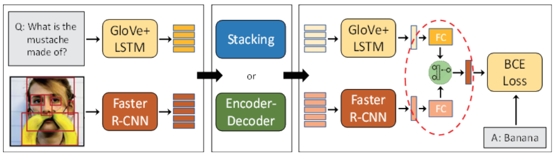


Fig. 3. The late attention-based fusion for the outputs of the two modalities text and image.

III. Experiments

A. *Dataset*

We evaluate the proposed method via VQA-v2 dataset, a challenging and most popularly used in VQA task. This dataset consists of question-answer pairs that were labeled by human with the images from MS-COCO dataset [8]. Each image has three questions and 10 answers, the training set: 80k images + 444k QA pairs, the validation set: 40k images + 214k QA pairs.

B. *Results*

The validation accuracy in the validation set has been reported in table 1. A simple yet effective fusion method shows the clear improvements in the question types and overall performance. It indicates that each modality (either text or image) contribute partly for understanding the task and it should not be treated as equally as the conventional fusion method in original paper. The experiments have been set to three individual random seeds trials. It took 2 days for training until converge.

C. *Table*

Method	All	Yes/No	Number	Other
MCAN-VQA [6]	67.1	84.8	49.4	58.4
Proposed (run 1)	**67.24**	**84.93**	49.35	**58.51**
Proposed (run 2)	**67.23**	**84.91**	**49.57**	58.34
Proposed (run 3)	**67.24**	**84.89**	49.35	**58.52**

Table 1. Comparison the current state-of-the-art VQA model and the proposed method in term of validation accuracy, running with 3 different trials of random seeds.

A visualization of the self-attention with multi-head attention has been show the effectiveness in Fig 4.

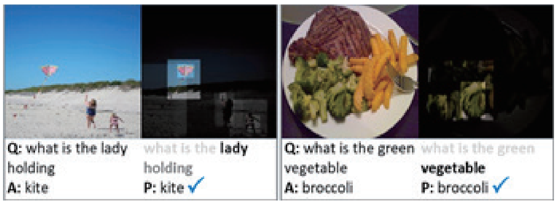

Q: what is the lady holding
A: kite
what is the lady holding
P: kite ✓
Q: what is the green vegetable
A: broccoli
what is the green vegetable
P: broccoli ✓

Fig. 4. Example of the attended part in questions and images that learned in the self-attention, multi-head, Figure from [6]

IV. References

[1] Stanislaw Antol and Aishwarya Agrawal. "VQA: Visual Question Answering" (ICCV 2015)

[2] Yin and Yang. "Balancing and Answering Binary Visual Questions" (CVPR 2016)

[3] Kelvin Xu. "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention", ICML 2015

[4] Simao Herdade, Armin Kappeler, "Image Captioning: Transforming Objects into Words", NeuIPS 2019

[5] Chaorui Deng. "Visual Grounding via Accumulated Attention", CVPR 2018

[6] Zhou Yu. "Deep Modular Co-Attention Networks for Visual Question Answering", CVPR 2019

[7] Ashish Vaswani. "Attention Is All You Need", NIPS 2017

[8] Tsung-Yi Lin, Michael Maire. "Microsoft COCO: Common Objects in Context", ECCV 2014.

Hindsight Goal Ranking on Replay Buffer for Sparse Reward Environment

Tung M. Luu[1] and Chang D. Yoo[1]

[1] Department of Electrical Engineering, KAIST
{tungluu2203, cd_yoo}@ kaist.ac.kr

Abstract

Reinforcement learning (RL) agents successively updates their parameters by way of recalling past experience via experience replay. It is a well-known fact that prioritizing the experience judiciously can improve sample efficiency. This paper considers a method for prioritizing the replay experience for off-policy RL referred to as Hindsight Goal Ranking (HGR) is proposed by addressing the limitation of Hindsight Experience Replay (HER) that generates hindsight goals based on uniform sampling. The proposed method combined with Deep Deterministic Policy Gradient (DDPG) accelerates learning significantly faster than that without any prioritization.

Keywords— *Reinforcement Learning, Multi-Goal Reinforcement Learning, Sparse Reward, Sample Efficiency, Hindsight Goal Ranking*

I. Introduction

Reinforcement Learning (RL) [7] is receiving escalating attention as a result of its successes in high profile tasks that include exceeding human-level performance in playing video games [3], defeating a Go master [6], and learning to accomplish simple robotic tasks autonomously [2].

Despite the many accomplishments, considerable challenges lie ahead in the transferring these success to complex real world tasks. An important challenge that must be addressed is to design a more sample efficient reinforcement learning algorithm, especially in sparse reward environments. To address this issue, Lillicrap et al. propose the Deep Deterministic Policy Gradient (DDPG) [2], which considers an agent that is capable of learning continuous control. Schaul et al. [4] develop the Universal Value Function Approximators (UVFAs), which allows value function approximation over both the states and any goal. Moreover, to make the agent learn faster in sparse reward environments, Andrychowicz et al. [1] introduce Hindsight Experience Replay (HER) that enables the agent to learn even from undesired outcomes. HER combined with DDPG lets the agent learn to accomplish more complex robotic tasks in a sparse reward environment.

In HER, the episodes and goals are uniformly sampled from the replay buffer. It would be more sample efficient if the episodes and goals are prioritized according to importance. The challenge now is to determine a criterion for measuring the importance of goals for replay. In this work, the significance of a goal is judged by the Temporal Difference (TD) error, which is an implicit way to measure learning progress [5]. Within an episode, a future visited state with high TD error will be labeled as hindsight goal more frequently. The significance of an episode is measured by the average TD error of the experience with its hindsight goal set as one of the visited states in the episode.

The proposed method is applicable to any robotic manipulation task that an off-policy multi-goal RL algorithm can be made use of. To evaluate the proposed method, experiments on pushing task were conducted. We also compare the sample efficiency of our method with baseline Vanilla HER [1], Energy-Based Prioritization (EBP) [8], and one-step prioritization experience [8].

II. Method

A. Two-step prioritizing hindsight goal

Instead of uniformly sampling future visited states as in HER for relabeling to hindsight goals, this paper improves sample efficiency by prioritizing the future visited states within an episode according to the magnitude of the TD error δ, computed by:

$$\delta = R(s,a,g) + \gamma Q_{\phi^-}(s', \mu_{\theta^-}(s',g),g) - Q(s,a,g)$$

This criterion has been considered as a proxy measure of the amount which the RL agent can learn from an experience: concretely, the TD error measures how far the value is from its next-step bootstrap estimate [5]. Specifically, in the replay buffer, the importance of an experience with a future visited state, which is probably become a hindsight goal, is ranked based on the magnitude of its TD error. Thus in the considered episode, the future visited state with larger magnitude TD error will be labeled as hindsight goal the following probability:

$$i \sim P'(j,i) = \frac{1}{Z'}|\delta_{ji}|^{\alpha'}, \;\; i \in \{j+1, \ldots, H\}$$

where the normalization function $Z' = \sum_{j=1}^{H-1} \sum_{i=j+1}^{H} |\delta_{ji}|^{\alpha'}$,

δ_{ji} is the TD error of the j^{th} experience and i^{th} visited state, H is horizon, i is index of future state, and j is index of sampled episode. Subsequently, the priority of the n^{th} episode in the replay buffer is defined as the average TD error by K such that $\delta^{(n)} = \frac{1}{K}\sum_k^K |\delta_k^{(n)}|$ where $\delta_k^{(n)}$ is the TD error of the k^{th} experience-goal combination from a total of K combination. Finally, the n^{th} episode is sampled with the probability as

$$n \sim P(n) = \frac{1}{Z}|\delta^{(n)}|^{\alpha}$$

where the normalization function $Z = \sum_n |\delta^{(n)}|^{\alpha}$. Here α determines how much prioritization should be incorporated.

B. *Prioritization and Bias trade-off*

The bias of Q-value estimation when using prioritization can be corrected by using importance-sampling (IS) weights. Specifically, when updating the parameter of approximated Q-value function by using the experiences in sampled episodes, the corresponding gradient is scaled by multiplying with the IS weight of the episode:

$$w_n = \left(\frac{1}{N_e}\cdot\frac{1}{P(n)}\right)^{\beta}$$

where N_e is the number of collected episodes in the replay buffer, β is hyper-parameter to control how much bias is corrected. Similarly, the bias induced by prioritizing episodes is compensated by:

$$w_{ji} = \left(\frac{2}{H(H+1)}\cdot\frac{1}{P'(j,i)}\right)^{\beta'}$$

where, H is horizon, β' plays a role similar with β, to control bias correction. The final IS weight to correct bias for an experience-goal is computed by

$$w_{ji}^{(n)} = w_n.w_{ji}$$

For new experience with an unknown TD error, the maximal priority, ($P_t = \max_{i<t} P_i$), is assigned to guarantee that all experience are replayed at least once. For more stable convergence, the IS weights are normalized by their maximum $1/\max w_{ji}^{(n)}$, thus the gradient is only scaled downwards.

III. Experiement

To verify the effectiveness of proposed method, the experiment in pushing task is conducted. In this task, a box is placed randomly on a table in front of the robot, and the robot arm attempts to move it to a target location on the table.

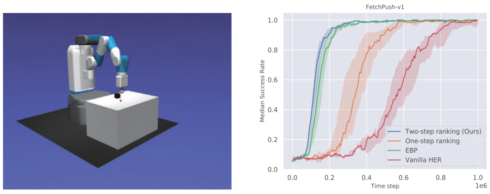


Fig. 1. The pushing environment (left) and its success rate (right).

From Fig 1, our method shows the convergence significantly faster compared to the others. Specifically, to achieves 50%, 75%, and 95% success rate, our method needs 115000, 145000, and 190000 samples, which is less than Vanilla HER, PER and EBP 5, 2.8, and 1.2 times, respectively. The final performance of the trained agent is shown in Table 1. From the result, our method is outperformed EBP with 0.15 percent point.

Table 1. The final mean test success rate.

	Vanilla	PER	EBP	Ours
Push	93.0%	99.36%	99.71%	**99.86%**

IV. Conclusion

In this paper, a prioritized replay method for multi-goal setting in the sparse rewards environment is considered. The proposed method divides the prioritized sampling into two steps: first, an episode is sampled according to the average TD error of experience with hindsight goals within the episode, then, experience with hindsight goals leading to larger TD error is sampled with higher probability. From the empirical results, the proposed method significantly improves sample efficiency.

References

[1] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *NeurIPS*, pages 5048–5058, 2017.

[2] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *ICML*, 2016.

[3] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[4] Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In *ICML*, pages 1312–1320, 2015.

[5] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *ICLR*, 2016.

[6] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *Nature*, 2017.

[7] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction.* MIT press, 2018.

[8] Rui Zhao and Volker Tresp. Energy-based hindsight experience prioritization. *CoRL*, 2018.

한국인공지능학회 2020 하계 및 추계 학술대회 논문집

추계학술대회 논문

ADHD Classification Based on Deep Learning Using MFCC Coefficients

Daeun Lee[1] and Joon S. Lim[2]

[1] Computer Engineering, Gachon University, Seongnam, South Korea, leede0418@likelion.org
[2] Computer Engineering, Gachon University, Seongnam, South Korea, jslim@gachon.ac.kr

Abstract

The current study focuses on classifying whether a subject is with ADHD or not by deep learning based on MFCC coefficients from visual coordinate data which is extracted from VR games. Distance data was extracted by obtaining the distance between two coordinates using the serially focused gaze coordinates. The distance data was divided into 10 windows, and a total of 14 MFCC coefficients were extracted using the divided data. The MFCC coefficients were figured as Mel Spectrogram images in CNN. As a result of this study, ADHD classification accuracy was higher when CNN was used than ANN. The ANN model showed about 55% accuracy, however, the CNN model produced about 71% and 75% accuracy for two experimental game datasets, respectively.

Keywords— *VR, MFCC, deep learning, CNN, Mel Spectrogram*

I. Introduction

ADHD(Attention Deficit Hyperactivity Disorder) is a disease found in people with distracted attention. It is not only the case with low concentration but also the case with excessively high concentration. The symptoms of a patient with ADHD can be relieved if the patient has some concentration training. This study converged in IT technologies, especially VR, was conducted to prove it. VR content from a first-person perspective has the advantage of providing immediate interaction and immersion. In this study, a subject was conducted to classify whether the subject is with ADHD or not based on visual coordinate data measured from VR games. The methods of this study were typically divided into three ways which were raw data preprocessing, ANN training and testing, and CNN training and testing. First, MFCC coefficients were extracted from the data manipulated from the raw data. Next, ANN model was trained and tested based on the MFCC coefficients. K-Fold Cross Validation was used to achieve a high performance with the little data. Last, CNN model was trained and tested based on Mel Spectrogram images extracted from the MFCC coefficients. By using a general Conv2D model, parameters like the number of epochs were tried to adjust to increase test accuracy. As a result, it was possible to know the probability of predicting the corresponding data result when new data came into ANN or CNN as input data with test accuracy. In other words, ADHD classification could be undergone with this procedure.

II. Relevant Studies

A. *An Innovative ADHD Assessment System Using Virtual Reality*

This study used a VR system to develop a brand-new ADHD assessment and diagnosis system. The study conducted listening test, CPT test, executive test, and visual memory test using VR technology instead of paper tests which are usually time-consuming. They conducted pilot tests before the VR is used in real situations and described the results. They successfully carried out the experiment based on the four tests mentioned. In the future, it was suggested that actual tests would be conducted on children with ADHD to further verify the medical effectiveness of the system [1].

B. *The development of tasks for discriminating ADHD tendencies using eye-tracker and neuropsychological attention tests*

This study examined if an experimental task using eye-tracker could be a useful tool to discriminate the ADHD tendency group from the normal group in adults. The study showed the ADHD tendency group had defects in effective disposition and movement of attention and response inhibition compared to the control group from “gaze-emotion task” and “exogenous-endogenous gaze-emotion task”. It was suggested that the experimental task based on the eye-tracking device was useful in discriminating the ADHD tendency group in adults [2].

C. The Word Frequency Effect on Reading in Typically Developing Children and Children with ADHD: An Eye-Tracking Study

This study observed the eye movement patterns of children with or without ADHD in natural reading tasks. The study proposed a new perspective on the language processing process of the children by measuring natural reading ways in real time by introducing eye movement tracking techniques. It was suggested that the children would have difficulties in high-level language processing, which analyzes and processes sentences based on linguistic knowledge, rather than lower-level language processing such as simple vocabulary [3].

III. Word Definitions

A. ADHD

ADHD stands for Attention Deficit Hyperactivity Disorder. It manifests due to lack of concentration and refers to a state of distraction, hyperactivity, and impulsiveness. There is no exact known cause of the disease until current time. However, experts believe that neurological and chemical factors, genetic and environmental factors have a complex correlation to interact with each other. The main symptoms of the disease are hyperactivity, attention deficit, impulsivity, and aggression.

B. MFCC

MFCC stands for Mel-Frequency Cepstral Coefficients. It is an algorithm that extracting features by dividing data into a certain section and analyzing the spectrum regarding the section. It is mainly used when extracting features from auditory data. In other words, it is used when vectorizing features from voice data.

IV. Study Methods

A. Raw Data Preprocessing

Each subject played different types of VR games, so raw data preprocessing was undergone with Game2 and Game3 which were played in common. The data was sliced based on the shortest amount of data because the total amount was different for each data. The distance between x and y coordinates was calculated with the data. The distance was divided into the size of window, which was 10, specified by timestamps to extract MFCC coefficients. The MFCC coefficients were extracted from "librosa" package. Each subject was obtained 14 MFCC coefficients in total with this way. Game score by date of each subject were checked to label the data. The score was divided into the control group if the data was less than 19 and the patient group if the data was above 19. The control group was labeled as 0 and the patient group was labeled as 1. The control group had much data than the patient group, so the patient group was appended three or four times more data to balance both groups.

B. ANN

ANN(Artificial Neural Network) based data training and testing were conducted. MFCC coefficients and labels were classified after loading the data preprocessed. The data was randomly shuffled just before classifying it. The data of the patient group was appended to the end of the entire data when preprocessing the data. This causes the problem that specific data is being biased to one side. The data was randomized with "np.random.shuffle()" to solve the problem. The MFCC coefficients and labels were converted to vectors after classifying them, respectively. This is because neural networks take vectors as input data. The ANN model was built with 1 input layer, 1 hidden layer, and 1 output layer. K-Fold Cross Validation was used considering the total of the data was small. The K was set to 4, so a total of four folds were performed. The 75% data was used for training and 25% data was used for testing during each fold. The number of epochs was set to 100. The ANN model showed about 56% test accuracy in Game2 and 54% test accuracy in Game3 as a result.

C. CNN

CNN(Convolutional Neural Network) based data training and testing were conducted. MFCC coefficients and labels were loaded and randomly shuffled. The reason why was the same as when experimenting with ANN. CNN takes images as input data, so Mel Spectrogram images were produced based on the MFCC coefficients. A Mel Spectrogram image was created by reflecting delta values obtained from the MFCC coefficients. It was difficult to use the whole image which had the whole data as input data. The image was divided into partial images per each timestamp to use as the input data. The labels for the partial images were vectorized by slicing only the labeling part from the initially loaded data. The CNN model was built with a general Conv2D model. CNN has "Flatten" process, which is additional, compared to ANN. This process transforms an input image into an 1D vector to insert into the neural network. The 60% data was used for training and 40% data was used for testing from the total data. The number of epochs was set to 800 and the batch size was set to 32 in Game2. On the other hand, the number of epochs was set to 850 and the batch size was set to 32 in Game3. The CNN model showed about 71% test accuracy in Game2 and 75% test accuracy in Game3 as a result.

V. Conclusion

Better results were came out in both Game2 and Game3 when CNN was used rather than ANN. In particular, the

	Game2	Game3
ANN	56%	54%
CNN	71%	75%

Table 1. ANN and CNN test accuracy regarding Game2, Game3

test accuracy was increased by 20%pt when CNN was used compared to ANN in the case of Game3. In other words, a test result, which is ADHD classification, would be correct with 50% probability and incorrect with 50% probability when using ANN. On the other hand, the test result would be correct with a relatively higher probability when using CNN. It is expected that the test accuracy can be further improved if VGG16 which is a pretrained model, which was not used in this study, is used instead of a general Conv2D model in CNN. The test accuracy would be increasing if fine tuning is undergone as well. However, it is necessary to give appropriate parameter options when building a model so that data is not to be overfitting when training the model.

REFERENCES

[1] Shih-Ching Yeh, Chia-Fen Tsai, Yao-Chung Fan, Pin-Chun Liu, and Albert Rizzo, An Innovative ADHD Assessment System Using Virtual Reality, 2012 IEEE EMBS International Conference on Biomedical Engineering and Sciences, pp.78-83 (2012).

[2] Sangil Lee, Mun-Seon Chang, Ho-Wan Kwak, The development of tasks for discriminating ADHD tendencies using eye-tracker and neuropsychological attention tests, Korean Journal of Psychology: General 2012, Vol. 31, No. 4, pp.1121-1230 (2012).

[3] So-Young Choi, The Word Frequency Effect on Reading in Typically Developing Children and Children with ADHD: An Eye-Tracking Study, Commun Sci Dis 2014, Vol. 19, No. 3, pp.307-319 (2014).

Gasoline engine misfire detection with machine learning

Kihyung Joo[1] and Jungseop Son[1]

[1] RD center, Hyundai Motors company, Korea, {steric5, jsson}@ hyundai.com

Abstract

As new technologies are applied to improve the performance of automobiles, the frequency of failures with the complexity is increasing. Misfire in a vehicle equipped with a gasoline engine is a typical failure occurring in an internal combustion engine. Misfire should not occur, but it is a difficult problem to solve because it is a problem caused by various reasons and parts. If misfires are detected exactly, it is possible to prevent fatal breakdowns of engines and other parts such as catalysts by notifying customers through engine warnings. The engineers try to develop misfire detection function with a lot of time and effort. The threshold value is determined in each area by analyzing the value of the factor representing the misfire in various driving areas. This work is time consuming process and is difficult to calibrate. Although the performance of misfire detection is very good, if the detection function misses fatal misfire, the engine or the catalyst to reduce harmful exhaust gases should be out of order. The purpose of this study is to alleviate the engineer's workload while the detection performance is similar or improved. Detection model is developed that can detect misfire through machine learning. This model proceeded with an interest in not only the detecting performance but also the embedding in the ECU should be possible.

Keywords— *Misfire, Random Forest, Ensemble, machine learning*

I. INTRODUCTION

As new technologies of automobiles are applied and systems are advanced, the complexity increases exponentially, and the frequency of failures is very high. Misfire refers to a phenomenon in which fuel is injected into the cylinder, but cannot be burned due to an abnormality in the ignition device or due to an abnormality in the injector or intake system. When misfire occurs, combustion stability is deteriorated, and it is a failure with a high proportion that can cause hardware failure such as catalysts. It is necessary to develop robust components to prevent misfires in advance, and to improve performance and failure-related calibration. As new technology is applied to the engine system, the calibration work load is rapidly increasing and the difficulty is multiplying. The conventional misfire detection measures the imbalance index of the engine cylinder and determines it regarding the threshold value. However, it is difficult to determine the threshold, and the engineers in the field feel difficult due to the long working hours. In order to alleviate the workload of engineers and improve the calibration accuracy, a predictive model that can replace the calibration that detects misfire with machine learning technology is devised. And finally, this study aims to implement this prediction model as an algorithm that can be embedded in the ECU. The main features of the false-facing detection model through this study are as follows.

- Selecting a model that is superior to the accuracy predicted by the existing calibration method
- Developing a model that can improve the workload of engineers while having similar or superior performance of machine learning models
- Hyper-parameter optimization
- Implementing an excellent algorithm for computation time and resource usage so that it can be embedded in ECU
- Engine / vehicle verification

The best machine learning is selected using various machine learning methods, including statistical regression and XGBoost, using the data obtained while generating a misfire in s/w in the vehicle as training data and test data.

II. MAIN SUBJECT

A. *misfire definition and mechanism*

Misfire means that the air mixed with fuel enters the cylinder of the engine, but no flame is generated from the spark plug. Misfire is caused by a variety of causes. Misfire may occur due to an inadequate ratio of air and fuel due

Items	Process
Misfire Window	Engine Roughness(ER) extraction
Cylinder correction for misfire detection	difference correction from occurrence cylinder and detection cylinder
Fuel Off learning	time correction regarding mechanical tolerance
Fuel On learning	ER deviation learning for combustion condition
Misfire monitoring diagnosis window	diagnosis window setting
Misfire detection threshold	detection threshold for each function

Table 1. development process for misfire detection

to an instantaneous insufficient amount of fuel mixed with air. In addition, an abnormality in the fuel injector is one of the reasons. In a gasoline engine, gas that has not been used for combustion is circulated back into the cylinder, which is called evaporate gas. Evaporate gas is collected in a storage device called a canister, and then goes in when air enters. The amount of evaporate gas is calculated and the actual amount of fuel is injected, but misfire occurs even if the amount is not adequate or does not evenly enter all cylinders. And when starting the engine for the first time, the coolant or the catalyst does not have enough temperature to perform its role, so more fuel is injected to increase the temperature. Due to the long mileage, the injector may actually fail due to poor durability. This can be seen as a problem in durability. Durable deterioration product failure is not covered here[4].

Misfire occurs little by little in all gasoline engines. When it occurs, control algorithm that detects it and prevents misfire are activated. However, if the occurrence of misfire increases, the stability of combustion may be broken or the exhaust gas purification device (catalyst) behind the engine may be damaged. When it gets worse, the catalyst is exposed to very high temperatures and even melts[2].

B. Introduction to calibration process

In general, the base calibration is performed in the engine or vehicle in a steady state. the term of calibration is a little like ”tuning” that people often say . Misfire monitoring calibration is tested and developed with the following items.

Misfire monitoring is developed with the process shown in Table 1. In this study, misfire detection threshold is set in the above process. For calibration of misfire detection threshold, the vehicle is driven in the chassis dynamometer under a specific driving condition, and the misfire is generated forcibly through s/w control. The threshold is set for this through this process. However, calibration for setting the threshold takes a minimum of 2 weeks to a maximum of 5 weeks. It may take longer to increase the accuracy. So, in order to solve these difficulties, this study is planned. In this study, the test data of the vehicle forcibly generating a misfire is used, and the machine learning(ML)'s work is compare to the engineer's work through may validation criteria. The calibration process of this study is shown in Figure 1 below.

C. Data preparation

Data from vehicle test is used as the data required for machine learning. As the measurement data, two vehicle test results are used. One data is used as training data, and the other data is used as test data. Usually, if data is not enough to train ML model, one data set is divided into 7:3, and 7 is used as training data and 3 is used as test data. However, if limited data is used in this way, the model trained with the characteristics of one data set may contain data of the same characteristics. For more reliability, the data created through the two vehicle tests are used with the same vehicle and under the same driving conditions, and the external conditions such as driver and temperature/humidity are different. The test vehicle is an hybrid with 1.6 Turbo GDI engine, and the driving conditions are made by adjusting engine parameters based on the engine speed(rpm) of five stages. The misfire is generated in each driving condition. A misfire is created by reducing injection amount of the injector or by cutting off the ignition. Misfire is generated by adjusting the s/w value of the ECU. Data are acquired as INCA's MDF format. Data are saved for each driving area. However, since the data has to be handled in python, the MDFReader library that can read MDF files in python is loaded into pandas DataFrame and temporarily saved in pickle format. The resolution of MDF data is different for each raster. The resolution is set to 10ms, the resolution of all data must be the same in DataFrame. Then, data measured in several driving areas are merged into one dataframe. Therefore, It is easy to handle data all driving areas.

D. Exploratory Data Analysis[EDA]

EDA must first be implemented to create data analysis and predictive models. EDA(Exploratory data analysis) is the process of observing and understanding data from various views. Through the process of understanding them from various views, various patterns that are not found in the problem definition stage can be discovered, and based on these, existing hypotheses can be modified or new hypotheses can be established. And when selecting a model, it is possible to define the model by viewing data features without having to try all the models.

It first goes through a process called data preprocessing. If there is missing data or data of a different format, it is deleted or converted. After pre-processing, the

feature	description
Misf_ER_0	total misfire engine roughness
Misf_ER_0_[0] [1,2,3]	first combustion cylinder of misfire engine roughness [second, third, fourth combustion cylinder]
Misf_ERC_0	total misfire engine roughness correction
Misf_ERC_0_[0] [1,2,3]	first combustion cylinder of misfire engine roughness correction [second, third, fourth combustion cylinder]
Misf_ERC_1	total misfire engine roughness difference
Misf_ERC_1_0 [1,2,3]	first combustion cylinder of misfire engine roughness difference [second, third, fourth combustion cylinder]
Thr_r	throttle position
Map_p	Map pressure sensor
Air_load	air mas flow
Eng_N	engine speed

Table 2. feature description

correlation between the data is checked. If multicollinearity can occur, it may affect the prediction model worse.

There are more than 200 variables in measured data, but the number of variables related to the misfire is limited to 23. Some variables are in Table 2. The correlation with 23 variables is confirmed. Considering the correlation, `Map_p` and `Air_load` showed a high positive correlation (0.96), and `Misf_ER_0_[0]` also showed a high correlation (0.92) with `Misfire_ERC_0_[0]`. If there are variables with high correlation, they have a negative effect on the prediction model when statistical regression, so may variables should be removed. As a method of removing multicollinearity, variables with high correlation are manually deleted, or high values are deleted using VIF (Variance Inflation Factor). In another way, PCA(Principle Component Analysis) between variables with high multicollinearity is used. In conclusion, these methods are used for logistic regression analysis, but no good results are found.

$$VIF = \frac{1}{1 - R_i^2} \tag{1}$$

R^2 is the coefficient of determination (2) and expresses the explanatory power of the regression model. It refers to the weight occupied by the value explained by the regression line, and the range is set at $0 \leq R^2 \leq 1$. The VIF equation is the same as (1), and if the coefficient of determination is 0.9 or more, it is considered as multicollinearity. variables usually are uses except for those with more than 10 in VIF, In Table 3, except for the top three variables, the remaining variables should be deleted, but they should be deleted while checking the performance.

VIF	features
3,413072	Misf_ER_0
3,837908	Misf_ERC_0
3,924367	Misf_ERC_1
15,43563	Misf_ERC_0_[3]
15.953009	Misf_ERC_0_[1]
19.826772	Thr_r
20.383632	Misf_ERC_0_[2]
22.890369	Misf_ER_0_[1]
22.917795	Misf_ERC_0_[0]
24.688023	Mis_ER_0_[3]

Table 3. Variance Inflation Factor

$$R^2 = \frac{\sum\,(\hat{y}_i - \bar{y})}{\sum\,(y_i - \bar{y})^2} \tag{2}$$

where $\hat{y}_i$ is predicted value, $\bar{y}$ is mean value of measure value, and y_i is measure value at i(i = 1,2,3,..., k)

The logistic regression model can take any value as x, but the output is always a value between 0 and 1. The model satisfies the requirements of the probability density function[3].

Checking whether training data and verification data have different data distributions(Fig.1) is important. Drawing a histogram based on the 23 explanatory variables and checking whether the distributions of the variables are unusually different are mandatory. This method is to check whether there is a peculiar pattern when misfire occurs for each variable.

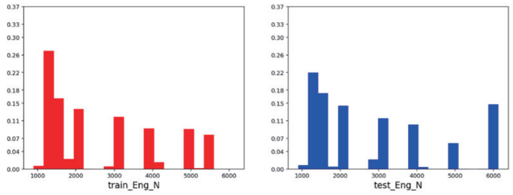


Fig. 1. comparison between distributions of train Eng N and test Eng N

E. *Model selection*

This study is a binary classification problem that classifies misfire/normal. The data contains actual misfire data. Using this, it can be classified as a supervised learning method. The binary classification problem uses traditional logistic regression methods using probabilities and machine learning methods such as Decision Tree using tree structure. In recent years, deep learning is also widely used for classification problems. However, deep learning is a black box model, and it is difficult to explain the results.

Since the final purpose of the model in the ECU is to be transplanted, deep learning, a black box model, is excluded from the candidate group. The various models such as logistic regression, decision tree, gradient boost model are used.

The study is conducted using the various models as a candidate group. Logistic Regression is the useful statistical method. It is easy to understand and is most often used for linear classification, but difficult to solve nonlinear problems. And because it needs to be regularized and is sensitive to multicollinearity, dimensional reduction and variable adjustments are required. The principle of the Logistic Regression model is simple.

$$\theta = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p \tag{3}$$

$$y = \frac{1}{1 + e^{-\theta}} \tag{4}$$

where β_p is multivariate coefficient and x_p is multivariate variable.

β_p is a coefficient that is determined to minimize errors, i.e., a direction to increase performance like a linear regression model. Binary classification is performed by sigmoid functioning (4) the determined in (3).

Decision Tree is a method classifying data in an if-then structure. After the classification model is completed, it has a tree structure and many branches

Tree-based machine learning tends to be insensitive to multicollinearity and normalization. In addition, it is easy to determine main factors(variables). However, if there are many branches and the depth gets deeper, it may overfit the learning data. So, it is generalized through a process called pruning. Since the variables are divided into two branches to minimize their value with Gini index (5) or entropy (6), the branches are divided in the direction to reduce the impurity to the maximum, so the main factors are determined through this.

$$GI = 1 - \sum_{i=1}^{k} p_i^2 \tag{5}$$

$$Entropy = -\sum_{k=1}^{m} p_k \log_2(p_k) \tag{6}$$

In this study, ensemble models that combines a decision tree and several classifiers are used. The ensemble models are Random Forest, Gradient Boost Classifier, XG Boost Classifier, and Support Vector Machine is also used. The ensemble model can be configured in several forms. The ensemble model shows better performance than individual classifiers by combining several classifiers to one classifier. This is a bagging model that extracts data and uses it in combination with several types of sub-models, and a boost method that increases accuracy by using data with large errors once more. Random Forest is a kind of bagging, and the sub models are composed of Decision Trees[1]. And, for each sub model, different training samples are extracted and used. Separate branches are made by calculating the impurity of the Gini index. Each sub classifier in Random Forest does not use the all feature variables, and the model is trained by randomly extracting features for each classifier. In the case of the classifier, the final result is determined through voting, and in the case of prediction, the more voted value is determined. The feature importance (factor importance) is also provided in the final result[5].

F. Model training and validation

1) How to train model

Two test data obtained from the vehicle are used as learning and test data. Data is pre-processed before use, and data characteristics are also checked. The data is used as training data and test data.

Model training and validation are conducted under various conditions. The purpose of this study is to select a model with high performance that prevents overfitting. So, I proceeded in various ways.

1. Data sampling method
   1) raw data
   2) under-sampling
   3) over-sampling
2. regularization
3. Comparison of model training results
4. PCA (Dimension Reduction)
5. Cross test
6. Hyper parameter tuning

The original data is used as it is with the data sampling method No.1. The performance of the under-sampling and over-sampling methods is worse than that of the raw data. Under-sampling has a high computational speed, but overfitting occurred. So, the original data is used as it is. No.2 Regularization is used only in Logistic Regression. The performance is different depending on the degree of variance of the variable, and the performance improved when normalized, but the decision tree and ensemble models of the tree structure has no correlation with the normalization. No.4 PCA did not improve performance in all models. Usually, variables with high multicollinearity are combined and used as predictive model variables, but this data has no effect. Cross-test No.5 prevented overfitting with a cross validation library. For the No.6 hyper parameter tuning, the number of sub classifiers is set as adjustment.

*2) How to validate model*The method of evaluating the performance of the model is carried out by comparing the confusion matrix with the classification report method. As shown in Table 4, the confusion matrix method compares Predict (the value predicted by the model) and True (the actual classification value). And the performance values are compared through the classification report (Table

	Predict[0]	Predict[1]
True[0]	469967	5139
True[1]	6209	21314

Table 4. confusion matrix

Classification Report	precision	recall	f1-score	support
0	0.99	0.99	0.99	475106
1	0.81	0.77	0.79	27523
accuracy			0.98	502629
macro avg	0.9	0.88	0.89	502629
weighted avg	0.98	0.98	0.98	502629

Table 5. Classification Report

5).

There are classification Report in Table 5. The higher the precision, recall, and f1-score are all, the better the performance. The calculation method is the same as (7), (8), and (9).

$$precision = \frac{TP}{TP+FP} = \frac{5139}{(5139+21314)} = 0.81 \qquad (7)$$

$$recall = \frac{TP}{P} = \frac{5139}{(6209+21314)} = 0.77 \qquad (8)$$

$$\text{f1-score} = 2\times\frac{recall\times precision}{(recall+precision)} =$$

$$2\times\frac{0.81\times 0.77}{(0.81+0.77)} = 0.79 \qquad (9)$$

G. Result

1) Comparison of model performance

Before comparing the performance between the models, it is necessary to compare them with the existing method of engineers. This is because it is meaningful only when the performance is the same as the existing method or when the performance is improved.

In Table 5, in the case of 0 (normal), precision, recall, and f1-score are higher than 1 (misfire). In all tests, the period of occurrence of misfire is short and there are much more normal cases, so the performance in normal cases is high. So, when comparing the data, the normal case is less considered, and the performance about the occurrence of misfire is more confirmed. The results of each model are as follows.

Random Forest has better performance than other models, but its computation speed is relatively long(Table 6). However, since the amount of data to be calculated is 478,672 rows, the calculation will be performed at a very high speed since the calculation is performed in a single row in the actual ECU. Each model is computed on the laptop. The laptop consists of intel i7 2.7GHz, 64G RAM, 64bit window7 OS.

	f1-score	computing speed
Base	0.79	
Decision Tree	0.59	145ms
Logistic Regression	0.70	33ms
Random Forest	0.85	2s
Gradient boost Model	0.76	698ms
XGBoost Model	0.76	943ms

Table 6. Comparison of f1-score and computing speed

Logistic Regression	precision	recall	f1-score
0(normal)	0.98	0.98	0.98
1(misfire)	0.73	0.68	0.70
accuracy			0.97
Random Forest	precision	recall	f1-score
0(normal)	0.99	0.99	0.99
1(misfire)	0.89	0.82	0.85
accuracy			0.98
Decision Tree	precision	recall	f1-score
0(normal)	0.98	0.96	0.97
1(misfire)	0.51	0.69	0.59
accuracy			0.95
Gradient Boost	precision	recall	f1-score
0(normal)	0.98	0.99	0.99
1(misfire)	0.86	0.69	0.76
accuracy			0.98
XG Boost	precision	recall	f1-score
0(normal)	0.98	0.99	0.99
1(misfire)	0.86	0.69	0.76
accuracy			0.98

Table 7. Comparison of models

As shown in Table 7 above, the random forest has the best performance. It can be seen that ensemble models perform better than Logistic Regression.

2) Optimizing the number of sub classifiers

In the case of Random Forest, it is a model made with 500 sub classifiers. Although this model has good performance, there are many sub classifiers to implement logic in the ECU. It is difficult to implement. So, in order to optimize the number of sub classifiers, the number of sub classifiers is optimized by the grid search method. The test results are compared by increasing the sub classifier every 2 from 2 to 500. In addition to the performance, the calculation speed is also compared.

As increasing the sub classifiers, the performance is greatly improved at first, but there is little change when it exceeds a certain level. However, the computational speed increased linearly.

In Fig.3, it can be seen that the operation speed increases

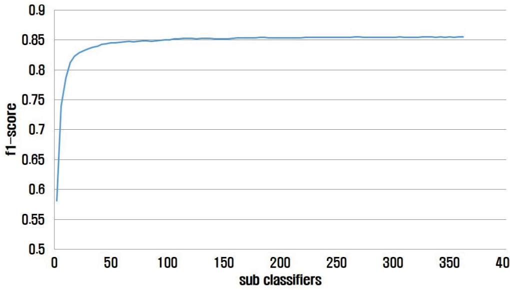


Fig. 2. f1-score regarding sub classifiers

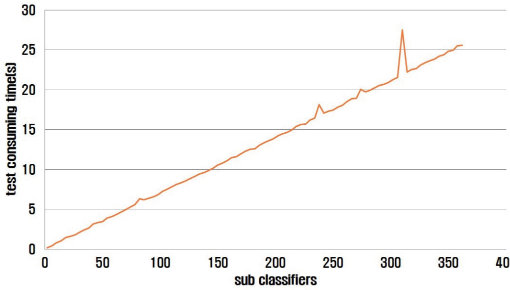


Fig. 3. test consuming time regarding sub classifiers

linearly as the number of sub classifiers increases. Using 500 sub classifiers took 35 seconds. Considering the performance and computation speed, 30 sub classifiers are selected(Fig.2). The f1-score is 0.84, and the computation time took 2s. When it is set to 500, the f1-score is 0.85, which is 0.01 difference, but the calculation speed is about 17 times, so 30 is set.

3) Key factor analysis

Among the 23 variables that are selected as input variables for the model, the main variables that greatly influenced the performance of the random forest model are analyzed.

Categorizing misfire and normal, 10 rankings are given(Table 8). Compared to other variables, `Misf_ER_0` is the most important variable, followed by `Misf_ERC_0` variable(Fig.4). The variable that affects the least is `AmbT_t` (outdoor temperature/environmental variable). Environmental variables have little effect on this model and variables. These are variables that affect the performance of the model. In the variables, they are all important variables related to the misfire in the conventional way. However, the bottom 13 out of 23 appear to have less impact. It is possible to check important variables through these machine learning programs. `Misf_ER_0` is the important feature.

rank	feature	importance
1	Misf_ER_0	0.295369
2	Misf_ERC_0	0.080958
3	Misf_ERC_0_[0]	0.071246
4	Misf_ERC_0_[2]	0.065415
5	Misf_dERC_1	0.046972
6	Eng_N	0.046670
7	Misf_ERC_0_[3]	0.044404
8	Misf_ERC_0_[1]	0.043937
9	Misf_ER_0_[2]	0.035462
10	Misf_ER_0_[0]	0.034913

Table 8. feature importance

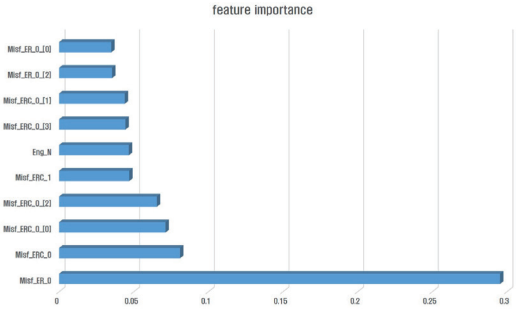


Fig. 4. feature importance

III. Conclusion

In this study, it is confirmed to reduce the development time with equivalent performance using machine learning for the calibration task in order to select a misfire detection threshold that takes a long time. When the engineers manually calibrate, the f1-score for the misfire is 0.81, but the performance is improved to 0.85 by the Random Forest. Moreover, the work that took at least 2 weeks can be shortened to 2 to 3 days. It takes more than two weeks to create a prediction model for this study. It takes a lot of time to convert MDF data into CSV format, data integration and pre-processing, model selection to increase predictive performance (f1-score), and processing of training data. All processes are implemented in one algorithm. Nevertheless, if data is prepared, it takes less than 10 minutes to train and verify the model. Since the process must be carried out with implementation in ECU, a lot of time is spent on logistic regression, but the f1-score of the misfire is worse than the existing method. However, the random forest has similar misfire detection performance, and the computational speed is good. However, it will not be easy to implement the logic with Random Forest. Since the sub classifiers have a tree structure, the code structure can be lengthened. Future study will be conducted to replace logic with Random Forest. The fields of application for this study are expected to be diverse. Although it is not easy to replace the logic based on the physical model, it is expected that it is possible to replace the part that sets the threshold. A predictive model such as this study can reduce workload and time of the engineers.

References

[1] Daniel Alvarez-Coello, Benjamin Klotz, Daniel Wilms, Sofien Fejji, Jorge Marx Gómez, and Raphaël Troncy. Modeling dangerous driving events based on in-vehicle data using random forest and recurrent neural network. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 165–170. IEEE, 2019.

[2] S Kevin Chen, Aditya Mandal, Li-Chun Chien, and Elliott Ortiz-Soto. Machine learning for misfire detection in a dynamic skip fire engine. *SAE International Journal of Engines*, 11(6):965–976, 2018.

[3] Stephan Dreiseitl and Lucila Ohno-Machado. Logistic regression and artificial neural network classification models: a methodology review. *Journal of biomedical informatics*, 35(5-6):352–359, 2002.

[4] Anson Lee, RNK Loh, and JZ Wu. Automotive engine misfire detection using kalman filtering. In *2003 IEEE 58th Vehicular Technology Conference. VTC 2003-Fall (IEEE Cat. No. 03CH37484)*, volume 5, pages 3377–3381. IEEE, 2003.

[5] Andy Liaw, Matthew Wiener, et al. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

한국인공지능학회 2020 하계 및 추계 학술대회 논문집

저자 색인