

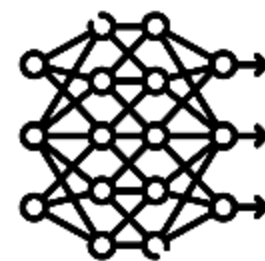
도전하는 머신러닝 개발자 백재훈입니다

2024 - PORTFOLIO

CONTACT

2hhunbaek@gmail.com

+82 10 4578 5164





도전하는 머신러닝 개발자 백재훈입니다

하리라(I will do it)와 우수함(Excellence)의 우를 합친 하리우라
Computer Vision에서 도전적으로 프로젝트를 우수하게 진행 할
능력이 있는 머신러닝 개발자입니다.

백재훈/JAEHUN BAEK

1995.07.01 / 경기도 시흥시

Tel. (+82) 010-4578-5164

Email. 2hhunbaek@gmail.com

github. hariura

GRADUATION

2013 은행고등학교 졸업

2021 명지대학교 수학과 졸업

2023 아주대학교 대학원 수학과
데이터사이언스 전공 졸업

SKILL

python	<div><div></div></div>	95
tensorflow	<div><div></div></div>	95
pytorch	<div><div></div></div>	80
git	<div><div></div></div>	80
docker	<div><div></div></div>	60

AWARDS

2022 아주대학교 데이터 경진대회 장려상

2022 안전한 자율주행을 위한 인공지능
알고리즘 개발 챌린지 장려상

PROJECT

2021 모바일 구강건강관리 어플리케이션에서
이상치 탐지

2022 안전한 자율주행을 위한 인공지능
알고리즘 개발 챌린지

2023 도로 위험물 탐지 객체 확대 연구

PROJECT.1

모바일 구강건강관리 어플리케이션 에서 이상치 탐지 알고리즘 개발

01

ABOUT PROJECT

활동기간 : 4월 2021- 3월 2022 (12개월)

소속 : 명지대학교 대학원, 아주대학교 대학원

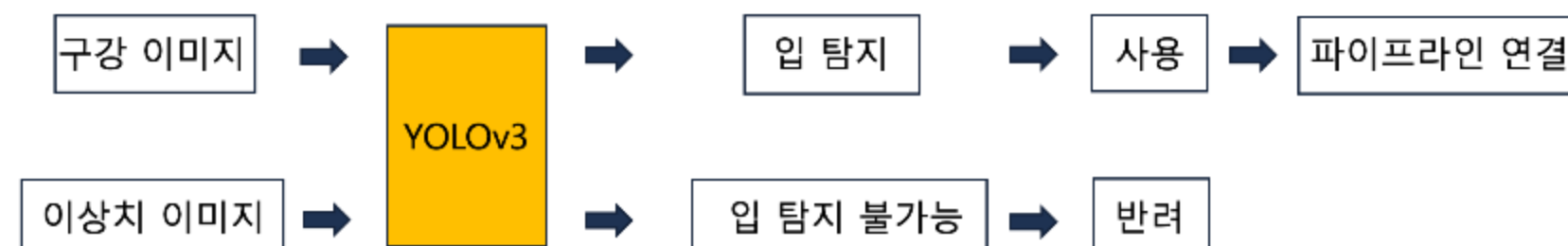
인원 : 5

출처 : 연구 과제

모바일 구강건강관리 어플리케이션에서 이상치 탐지 알고리즘 개발

연구 배경 :

1. 모바일 구강 건강 어플리케이션에 이상치 탐지 알고리즘을 적용,
2. 구강 이미지가 아닌 이미지에 대해서 구강 정보를 제공하지 않도록 필터링 작업



목표 : 이진 분류 모델과 오토 인코더 모델의 정확도(약 90%)보다 높은 성능 요구

역할 : 구강 이미지는 입을 포함하고, 구강 이미지가 아니면 입을 포함하지 않는 특성을 이용해, 입의 유무를 감지하는 객체 탐지 모델(YOLOv3)을 사용하여 이상치 탐지 구현.

배운점 : 머신러닝을 이용한 신뢰성 확보를 통해 사용자 경험의 중요성 체감

기여도



• 문제 해결을 위한 연구:

- 지도학습으로 모든 이상치에 대한 고려하는 것은 어렵기에,
적은 구강 이미지(340장)만 학습에 이용 (one class)
- Tensorflow2로 작성된 YOLOv3 모델을 이용해 transfer learning 진행

• 성능 향상을 위한 추가 연구: (뒷장 기재)

- 입에 대한 라벨링 기준 비교
- Loss 함수 수정
- 기초 통계량을 활용한 임계값 설정

• 성능 지표:

- 테스트 데이터 9962장 (normal: 5023, abnormal : 4939)를 사용
- 정확도 0.9980 달성 (4번째 lambda = 10, EL 모델)
- 객체 탐지 베이스 라인 대비 정확도에 대한 에러율을 14.55배 감소

• 성과

- 한국산업응용수학회에 2022년12월에 논문 게재

TABLE 7. Confusion matrices without the classification loss

		Predicted	
		Negative	Positive
$\lambda_{\text{conf}} = 1, \text{IL}$	Actual Negative	4902	37
	Actual Positive	253	4770

		Predicted	
		Negative	Positive
$\lambda_{\text{conf}} = 1, \text{EL}$	Actual Negative	4938	1
	Actual Positive	132	4891

		Predicted	
		Negative	Positive
$\lambda_{\text{conf}} = 10, \text{IL}$	Actual Negative	4910	29
	Actual Positive	71	4952

		Predicted	
		Negative	Positive
$\lambda_{\text{conf}} = 10, \text{EL}$	Actual Negative	4927	12
	Actual Positive	8	5015

TABLE 8. Accuracy, Precision, Recall and F1-score without the classification loss

	Accuracy	Precision	Recall	F1-score
$\lambda_{\text{conf}} = 1, \text{IL}$	0.9709 (▲0.0130)	0.9923 (▲0.0217)	0.9496 (▲0.0043)	0.9705 (▲0.0128)
$\lambda_{\text{conf}} = 1, \text{EL}$	0.9866 (▼0.0079)	0.9998 (▲0.0012)	0.9737 (▼0.0167)	0.9866 (▼0.0079)
$\lambda_{\text{conf}} = 10, \text{IL}$	0.9900 (▲0.0023)	0.9942 (▲0.0071)	0.9859 (▼0.0026)	0.9900 (▲0.0022)
$\lambda_{\text{conf}} = 10, \text{EL}$	0.9980 (▲0.0012)	0.9976 (▼0.0020)	0.9984 (▲0.0044)	0.9980 (▲0.0012)

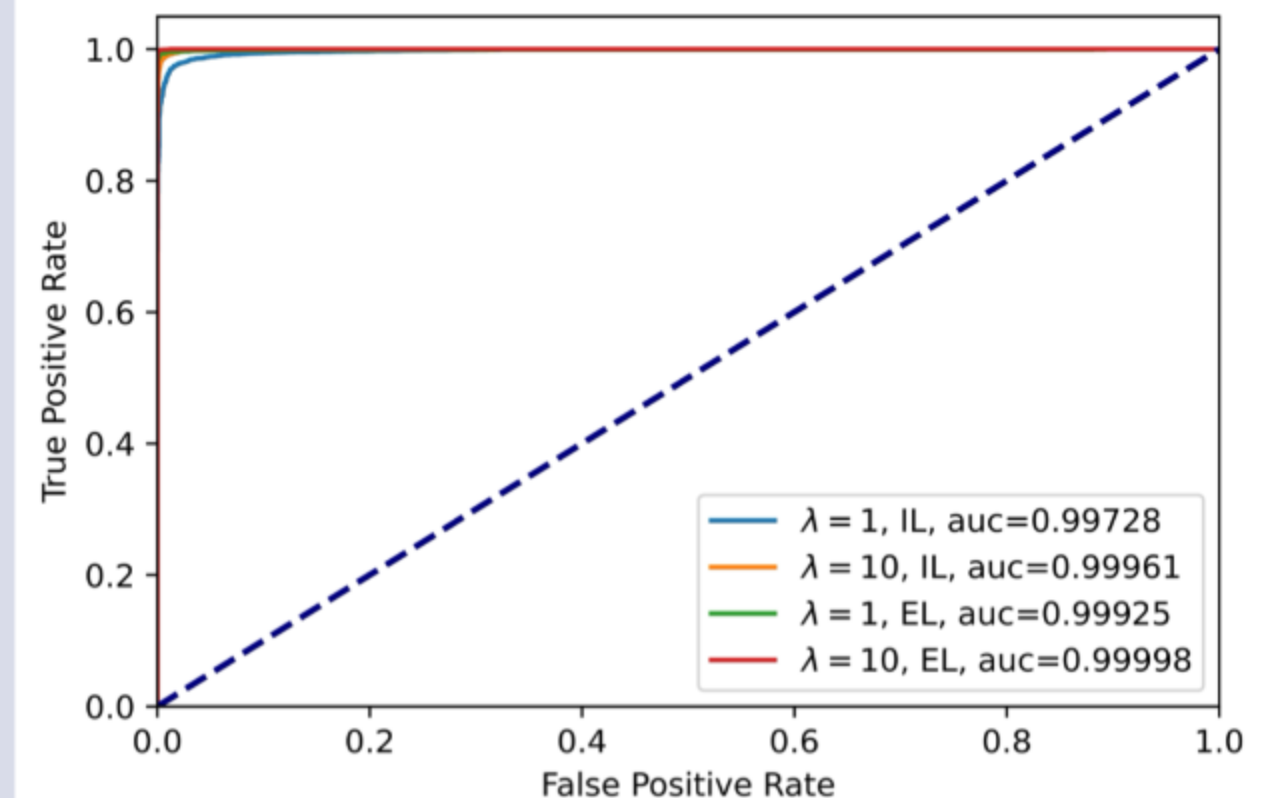


FIGURE 4. ROC curve and AUC for 4 cases without the classification loss

성능향상을 위한 연구

1. 입에 대한 라벨링 기준 비교

- 구강 이미지에서 찾을 객체인

입(입술 포함, 미포함)을 정의하여

학습 성능 비교

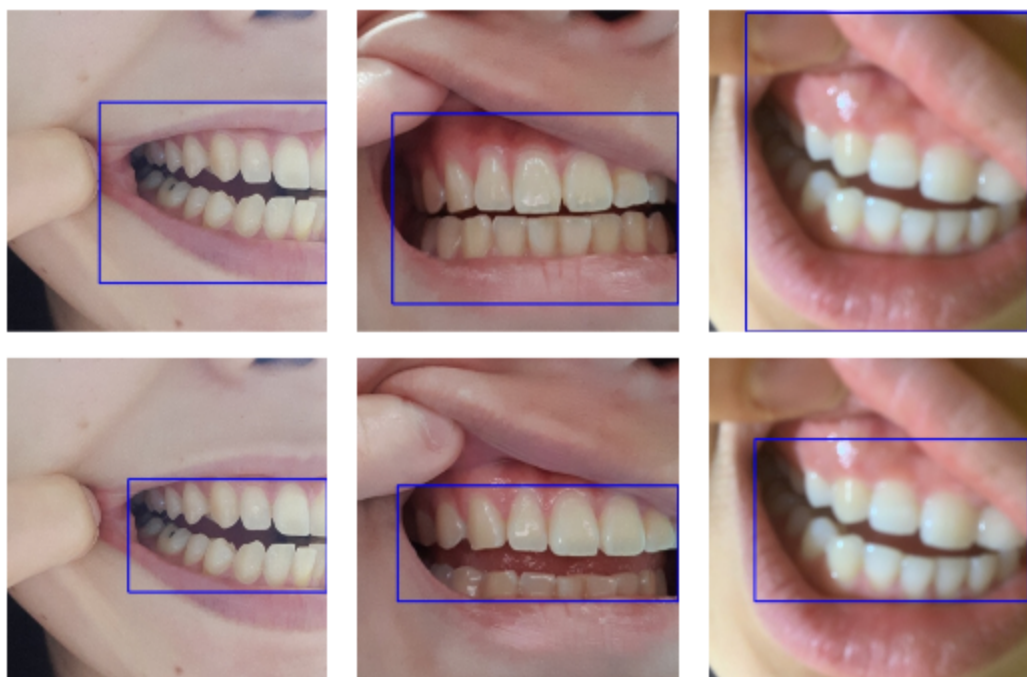


FIGURE 2. Bounding boxes including lips (top) and excluding lips (bottom).

2. Loss 함수 수정

- 적은 데이터를 통한 one class 객체 탐지를 위해

loss 수정 및 웨이트 조절

Sum Square Error loss

$$\begin{aligned} \text{Loss} = & \underbrace{\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right]}_{\text{Regression Loss (or Bounding box coordinate loss)}} \\ & + \underbrace{\sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2}_{\text{Confidence Loss}} \\ & + \underbrace{\sum_{i=0}^{S^2} \mathbb{I}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2}_{\text{Classification Loss}} \end{aligned} \quad (1)$$

Using GIoU, CE and Focal loss

$$\begin{aligned} \text{Loss} = & \underbrace{\sum_{i=0}^{S^2} \sum_{j=0}^B (1 - \text{GIoU}(X_{ij}, \hat{X}_{ij}))}_{\text{GIoU Loss (or Bounding box coordinate loss)}} \\ & + \underbrace{\lambda_{\text{conf}} \left[- \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{\text{obj}} \overbrace{(1 - \hat{C}_{ij})^2}^{\text{focal}} \log \hat{C}_{ij} - \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{\text{noobj}} \overbrace{\hat{C}_{ij}^2}^{\text{focal}} \log(1 - \hat{C}_{ij}) \right]}_{\text{Confidence Loss}} \\ & + \underbrace{\sum_{i=0}^{S^2} \mathbb{I}_i^{\text{obj}} (-\log \hat{p}_i(c))}_{\text{Classification Loss}} \end{aligned} \quad (3)$$

3. 기초 통계량을 활용한 임계값 설정

- 객체탐지에 대한 확률을 기준으로

임계값을 설정하여 이상치 탐지 정확도 향상

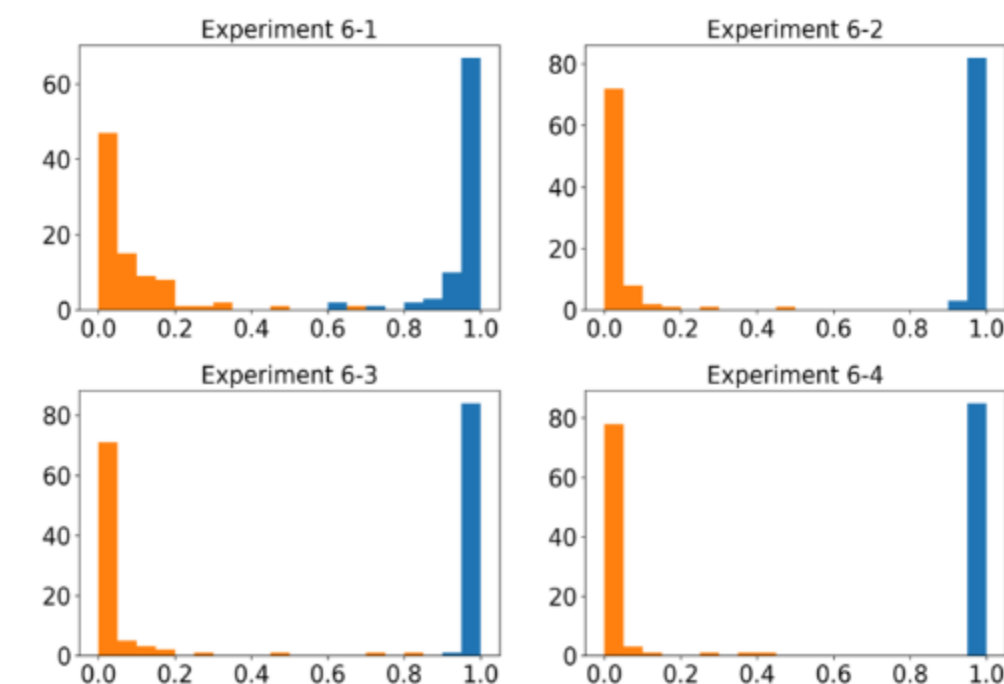


Figure 19: Histograms for Exp 6 ■ : Validation set ■ : Anomaly set

상단 표는 학위논문발췌

PROJECT.2

도로 위험물 탐지 객체 확대 연구

02

ABOUT PROJECT

활동기간 : 1월 2023 - 4월 2023 (4개월)

소속 : 아주대학교 대학원

인원 : 3

출처 : 연구 과제 (다리소프트)

도로 위험물 탐지 객체 확대 연구

연구 배경:

실시간 도로위험정보 서비스 시스템 구축 사업 마련을 위한 추진방향 및 전략 제시

목표 : 차량 주행 중 발생할 수 있는 위험물(공사, 보행자, 사고 차량)을 실시간으로 판단

역할:

1. 데이터 부재로 블랙박스 뷰로 되어 있는 사고 데이터 공수 및 라벨링
2. 객체 탐지 모델(YOLOv7)으로 모델 학습

배운점 : 사고 데이터의 획득 및 전처리에서의 어려움에도 목표 달성을 위한 방법 탐구

기여도

기획	<div><div></div></div>	55
개발	<div><div></div></div>	40



- 데이터 수집 및 모델 개발:

1. 데이터 수집: 유튜브의 차량 사고 영상에서 이미지 추출 및 라벨링
2. 작업 단축: 1차로 학습된 모델을 사용하여 추가 데이터에 대해서 전처리 분량 80% 단축

- 성능 측정:

- 공사와 보행자는 이미지 기반, 사고 차량은 시퀀스 기반으로 평가
- 이미지 기반: 하나의 프레임에서 어느 하나의 위험 요인을 정확히 탐지하면 예측으로

판단하는 accuracy metric

- 시퀀스 기반: 영상의 일정 구간에서 사고 장면을 하나라도 탐지하면 예측으로

판단하는 accuracy metric

- 모델 개선:

- 사고 차량 탐지에서 "전복-> 전복, 정상, 파손 차량"으로 클래스의 수를 변경해가며

정확도 향상

- 모델 성능:

위험물 판단 정확도 98.6% (285장) 사고판단 정확도 98.1% (54영상)

Sequence metric

Sequence detection: **False**



Sequence detection: **True**



PROJECT.3

안전한 자율주행을 위한 인공지능 알고리즘 개발 챌린지

LIDAR, 3D OBJECT DETECTION

03

ABOUT PROJECT

활동기간 : 10월 2022 – 11월 2022 (2개월)

소속 : 아주대학교 대학원

인원 : 3

주최 : 한국교통안전공단, (주)미디어그룹사람과숲

안전한 자율주행을 위한 인공지능 알고리즘 개발 챌린지

연구 배경:

자율주행을 위한 컴퓨터 비전 머신러닝 알고리즘 개발에 대학원생 팀 참여

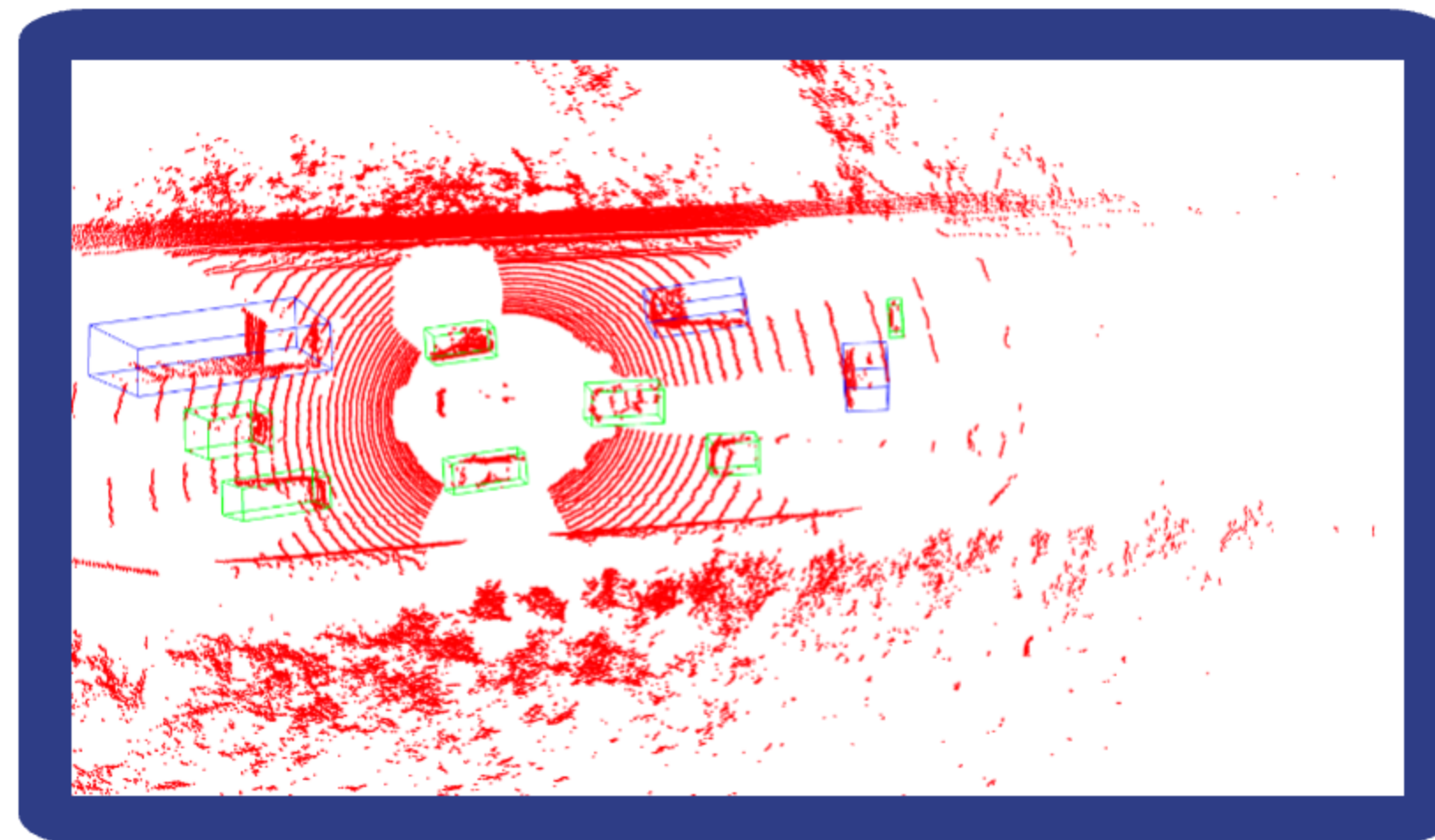
목표:

2D Object Detection(2D-OD), 2D Semantic Segmentation (2D-SS) and 3D Object Detection (3D-OD) 에서 높은 스코어 달성

역할:

새로운 도메인에 대한 도전으로 3D-OD 담당, 데이터 전처리, 모델 학습, 결과 도출

배운점 : 도메인 확장 경험을 통한 도전적인 문제를 이해 할 수 있는 시각 확장



기여도

기획	<div style="width: 50%;"></div>	33
개발	<div style="width: 50%;"></div>	33

안전한 자율주행을 위한 인공지능 알고리즘 개발 챌린지

2022 Autonomous Driving AI Challenge

대회 기간 : 2022.09.13(목) ~ 2022.11.18(금)

참가 대상 자율주행 기술개발 및 연구를 목적으로 하는 기업 기술 및 AI/ML, AI/ML 분야에 종사하는 연구자	지원처 2022.11.18(금)
진행 방식 객체 탐지, 2D/3D 객체 탐지, 2D/3D 객체 탐지 객체 탐지 및 2D/3D 객체 탐지	수상지 최대 2000명
진행 일정 2022.09.13(목) ~ 2022.10.29(목)	진행 일정 주제: 자율주행 대상: 2022년 대상: 2022년 대상: 2022년
신청 기간 2022.10.11(수) ~ 2022.10.21(수)	신청 방법 1. 홈페이지에서 신청 2. 이메일로 신청 www.aiaa2022.com

• 데이터 수집 및 모델 개발:

- 데이터 : 대회에서 PCD 형식의 LiDAR 데이터를 제공
- 모델 : openPCDet 프레임워크의 PV-RCNN 이용해 custom data 학습

• 전처리 문제해결 :

- 구글링으로 명확히 이해가 되지 않았던 heading angle 부분은 현직자 커뮤니티 통해 해결
- KITTI 데이터 학습 방법을 참고해 npy파일 변환후 학습이 안되는 상황 해결
- 학습의 용이성을 위해 제시된 8개의 클래스 중 클래스의 분포가 높은 3개를 선별해서 사용

• 주요성과

- 스코어 mAP 60 달성
- 종합된 결과로 장려상(3위) 수상

```
[53]: # add intensity 1, dtype 32
pcd = o3d.io.read_point_cloud(file)
array = np.asarray(pcd.points, dtype=np.float32)
points = np.hstack([array, np.full((array.shape[0], 1), 1, dtype=np.float32)])
np.save("/Users/jh/Desktop/GitHub/3D_project/000160_1" + ".npy", points)

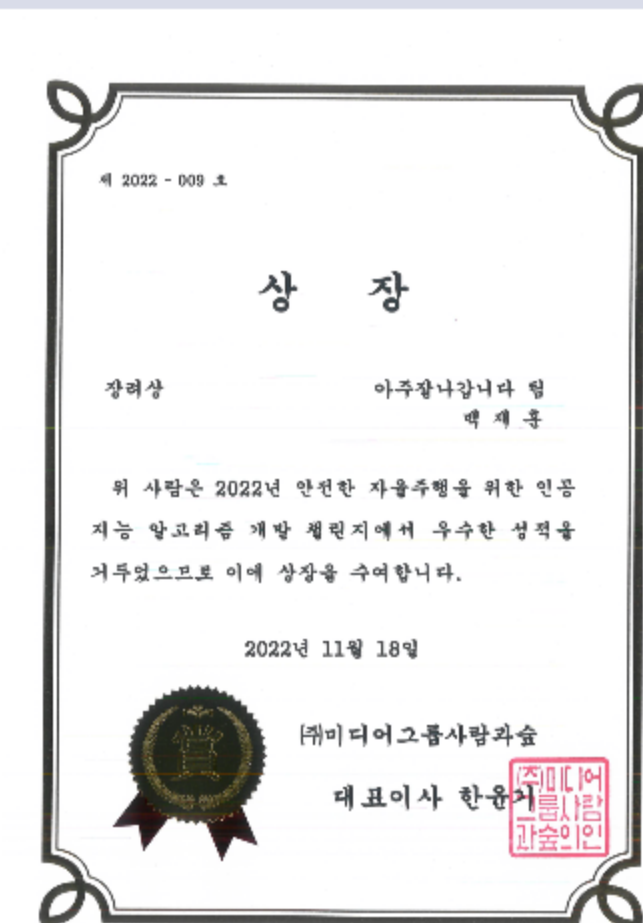
[54]: # add intensity 0, dtype 32
pcd = o3d.io.read_point_cloud(file)
array = np.asarray(pcd.points, dtype=np.float32)
points = np.hstack([array, np.full((array.shape[0], 1), 0, dtype=np.float32)])
np.save("/Users/jh/Desktop/GitHub/3D_project/000160_2" + ".npy", points)

[55]: # add intensity 1, dtype 64
pcd = o3d.io.read_point_cloud(file)
array = np.asarray(pcd.points)
points = np.hstack([array, np.full((array.shape[0], 1), 1)])
np.save("/Users/jh/Desktop/GitHub/3D_project/000160_3" + ".npy", points)

[56]: # add intensity 0, dtype 64
pcd = o3d.io.read_point_cloud(file)
array = np.asarray(pcd.points)
points = np.hstack([array, np.full((array.shape[0], 1), 0)])
np.save("/Users/jh/Desktop/GitHub/3D_project/000160_4" + ".npy", points)

[57]: # add intensity 0, dtype 32
pcd = o3d.io.read_point_cloud(file)
array = np.asarray(pcd.points, dtype=np.float32)
points = np.hstack([array, np.full((array.shape[0], 1), 0, dtype=np.float32)])
points = points.round(3)
np.save("/Users/jh/Desktop/GitHub/3D_project/000160_5" + ".npy", points)

[58]: # add intensity 1, dtype 32
pcd = o3d.io.read_point_cloud(file)
array = np.asarray(pcd.points, dtype=np.float32)
points = np.hstack([array, np.full((array.shape[0], 1), 1, dtype=np.float32)])
points = points.round(3)
np.save("/Users/jh/Desktop/GitHub/3D_project/000160_6" + ".npy", points)
```



감사합니다. 잘부탁드립니다!

2024 - JAEHUN BAEK

CONTACT

2hhunbaek@gmail.com

+82 10 4578 5164

