

Object Detection Challenge Wrap-Up Report

CV-18조 The Visionaries

김준태_T5059

박재민_T5089

송인성_T5116

이지유_T5160

최홍록_T5220

목차

1. Project 개요
2. EDA & Data cleaning
3. 실험 과정 및 결과
4. Ensemble
5. Conclusion

1. Project 개요

문제 정의

대량 생산, 대량 소비의 시대에 많은 제품이 생산되고 소비되면서 쓰레기 배출 현상이 심화됨에 따라 쓰레기 대란과 매립지 부족이 심각한 사회 문제로 제기되고 있다. 우리는 재활용 가능한 자원을 보다 효과적으로 회수하기 위해 분리수거 과정에서 **object detection** 기술을 활용해 쓰레기를 식별하고 쓰레기의 분리수거 정보를 파악하고자 한다. 이를 통해 **object detection task**에 대한 이해를 높이고 재활용 쓰레기 처리의 효율성을 높여 지속 가능한 환경을 조성하는데 기여할 수 있다.

팀 구성 및 역할

- 김준태 : EDA, 2 stage model (Cascade R-CNN) 기반 실험 및 결과 분석, k-fold & pseudo labeling 실험
- 박재민 : MMDetection train 코드 작성, YOLO baseline code 작성, YOLO v8 실험 진행, bug 수정, ensemble과 data split code 작성, 자동화를 위한 shell script 작성
- 송인성 : EDA, modeling - RetinaNet & Cascade R-CNN, focal loss 적용, data augmentation, data split, ensemble
- 이지유 : MMDetection config baseline 구현, 2 stage model(Faster R-CNN, Cascade R-CNN) 학습 및 성능 개선 작업 수행, model ensemble 실험 진행
- 최홍록 : base config 파일 작성, data cleaning 파일 생성, YOLO v8 실험

팀 목표

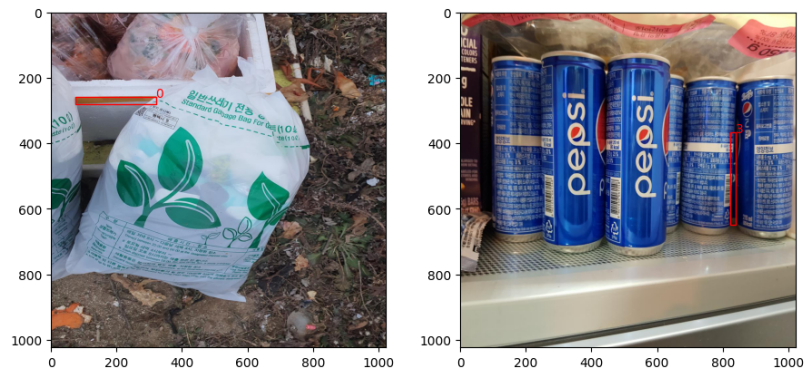
- 이유와 근거가 있는 실험을 체계적으로 진행하자.
- 여러가지 모델에 대해 공부하고 사용해보자.
- 새로운 팀원들과 협업하는 방법을 배우자.

2. EDA & Data cleaning

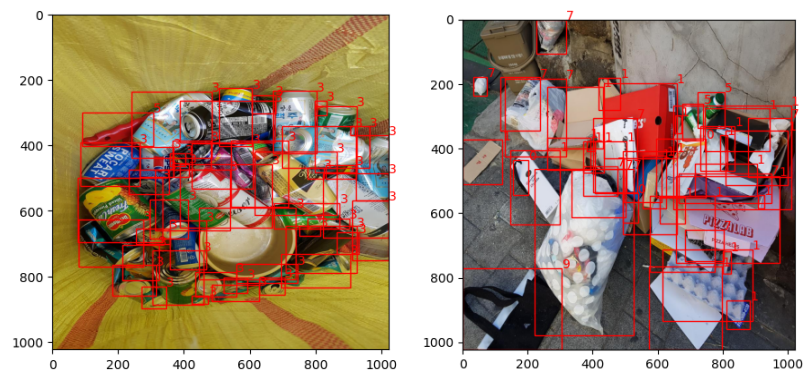
EDA

EDA를 통해 dataset에서 다음과 같은 문제점을 발견했다.

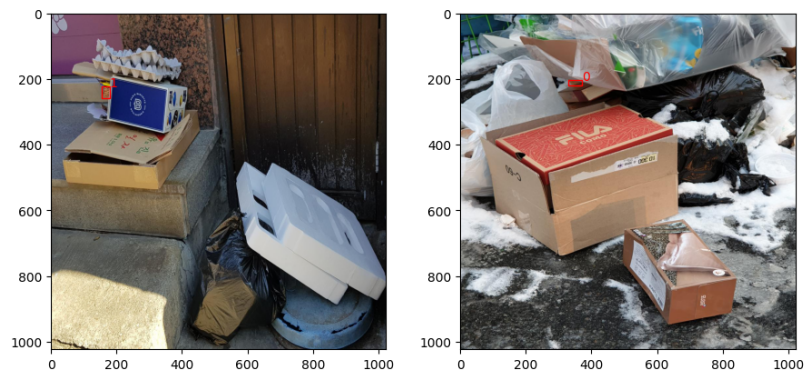
1. 극단적인 종횡비를 갖는 bounding box



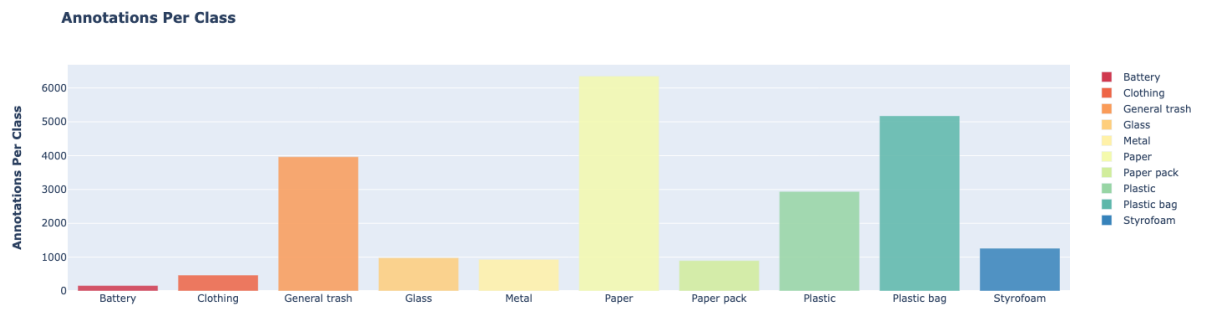
2. bounding box 수가 너무 많고 bounding box 간 겹침이 심한 image



3. 크기가 극단적으로 작은 bounding box



4. class 간 불균형



5. 명확하지 않은 class 선정 기준

Data Cleaning

EDA 결과로부터 발견한 문제점 중 1~4번을 해결하기 위해 다음과 같은 시도를 했다. 각 번호는 EDA의 문제 번호와 대응된다.

1. 극단적인 종횡비의 기준을 10으로 설정하고 이보다 큰 종횡비를 갖는 bounding box는 제거했다.
2. bounding box 개수가 35, 20, 10, 5, 2, 1 이상인 이미지를 차례대로 학습 dataset에서 제거해보았다.
3. 작은 bounding box 면적의 기준을 1048(i.e., 최대 이미지 면적의 1/1000)로 잡고 이보다 작은 면적을 갖는 bounding box는 제거했다.
4. CV strategy를 random split에서 stratified group k-fold로 바꿔 적용해보았다.

3. 실험 과정 및 결과

Object Detection의 대표적인 모델들을 공부해보고자 1 stage, 2 stage 모델로 나누어 실험했다.

1-Stage

Models

mmdetection의 YOLO v3¹와 RetinaNet², ultralytics의 YOLO v8³로 실험을 진행했다.

Model	mAP50 (validation)
RetinaNet (ResNet101)	0.4190
RetinaNet (ResNeXt101)	0.4200
YOLO v3 (512)	0.2880
YOLO v8 N	0.3535
YOLO v8 M	0.4575
YOLO v8 X	0.4982

비교 실험을 통해 확인한 결과 전반적으로 YOLO v8의 성능이 RetinaNet, YOLO v3 보다 좋았고 그중에서도 X 모델의 성능이 가장 높음을 확인할 수 있었다. 이후 YOLO v8을 fine-tuning하는 방향으로 실험을 진행했다.

Data

data cleaning에서 제안된 data들로 성능 변화를 측정하기 위한 실험을 진행했다. base model은 25 epoch동안 학습한 YOLO v8 N 모델이고 augmentation 기법을 적용하지 않고 학습했다.

Method	mAP50 (validation)
baseline	0.1279
종횡비 10 이상의 bounding box 제거	0.1255
bounding box 수가 35 초과인 image 제거	0.1173
bounding box 수가 20 초과인 image 제거	0.1194
bounding box 수가 10 초과인 image 제거	0.1049
bounding box 수가 5 초과인 image 제거	0.0937

¹ Redmon, Joseph, and Ali Farhadi. "YOLOv3: An Incremental Improvement." arXiv arXiv:1804.02767 (2018).

² Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, Piotr Dollar "Focal Loss for Dense Object Detection" Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980-2988

³ 'Ultralytics'. Accessed May 21, 2023. Available at: <https://github.com/ultralytics/ultralytics>.

bounding box 수가 2 초과인 image 제거	0.0651
bounding box 수가 1 초과인 image 제거	0.0469
면적이 1048 이하인 bounding box 제거	0.1209
class가 general trash인 bounding box만 유지	0.2206

실험 결과 전체적으로 base model보다 성능이 하락한 모습을 확인할 수 있었다. 때문에 모델을 학습할 때 data cleaning 기법은 사용하지 않기로 결정했다.

또 general trash 데이터를 잘 예측하지 못하는 점을 해결하기 위해 general trash만 존재하는 data를 만들어 학습을 진행했지만 별다른 성능 향상이 없었다.

Validation

YOLO v8 N 모델을 대상으로 baseline 모델과 validation 단계에서 nms⁴(non-maximum suppression)를 적용한 모델과의 차이점을 실험했다. 실험은 25 epoch동안 진행되었으며 512x512 크기의 이미지를 입력으로 사용하고 augmentation 기법이 적용되지 않은 상태에서 하이퍼파라미터는 동일하게 설정했다.

Heuristic	mAP50 (validation)
baseline	0.1279
nms(iou: 0.7)	0.3535

nms를 적용한 모델에서 mAP50 점수가 0.2256점 상승하는 것을 확인했다. 이를 통해 validation시 nms의 적용이 성능에 미치는 긍정적인 영향을 확인할 수 있었다.

Augmentation

nms를 적용한 모델에 YOLO v8에서 적용할 수 있는 augmentation을 하나씩 적용해보며 성능변화를 측정했다.

실험은 YOLO v8 N 모델로 25 epoch동안 진행되었으며, 512x512 크기의 이미지를 입력으로 사용하고 하이퍼파라미터는 동일하게 설정했다.

Augmentation	mAP50 (validation)
nms(iou: 0.7)	0.3535
+ mosaic	0.4259
+ translate(0.1)	0.3867
+ scale(0.5)	0.4150
+ hsv(0.015, 0.7, 0.4)	0.3398

⁴ Jan Hosang, Rodrigo Benenson, Bernt Schiele "Learning non-maximum suppression" Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4507-4515.

hsv를 제외한 모든 augmentation 방법론에서 약 0.3~0.7의 mAP50 점수가 상승하는것을 확인했다. 이를 바탕으로 hsv를 제외한 augmentation 기법을 적용하기로 결정했다.

image rotation augmentation을 0°, 45°, 90°로 설정하여 성능변화를 측정했다. 90° 초과는 image의 상하가 바뀌어 학습에 악영향을 줄 것으로 생각해 실험에서 제외했다.

Degree	mAP50 (validation)
0°	0.5432
45° 90°	0.5608 0.5742

실험 결과 0°, 45°, 90°로 갈수록 성능이 향상되는 것을 확인할 수 있었다.

Epoch

지금까지 실험이 25 epoch을 기준으로 진행됐으므로 epoch에 따른 성능변화를 확인하기 위해 실험을 진행했다. 실험은 YOLO v8 N 모델로 진행했으며 25 epoch으로 학습한 모델과 augmentation을 적용하지 않은 150 epoch으로 학습한 모델, augmentation을 적용한 150 epoch으로 학습한 모델을 사용했다. 150 epoch으로 학습된 모델은 모두 early stop patient는 10으로 설정했다.

Epoch	mAP50 (validation)
25	0.3535
150 150 + augmentation	0.2982 0.4168

실험 결과 augmentation을 적용하지 않고 150 epoch 학습한 모델은 25 epoch으로 학습한 모델보다 mAP50점수가 0.0553 하락했다. 하지만 augmentation을 적용하고 150 epoch 학습한 모델은 25 epoch으로 학습한 모델보다 mAP50점수가 0.0633 상승했다. 이를 통해 augmentation 없이 150 epoch 학습한 모델은 train data에 overfitting되었지만 augmentation을 적용해 overfitting 문제를 해결하고 성능을 높일 수 있었다는 결론을 도출했다.

Resolution

입력 image resolution의 변화에 따른 성능변화를 확인하는 실험을 진행했다. 실험은 YOLO v8의 N 모델과 M모델, X 모델로 512x512, 620x620 (YOLO 기본 설정값), 1024x1024 3가지의 resolution으로 25 epoch 학습했다.

Resolution/Model	mAP50 (validation)		
	N	M	X
512	0.3535	0.4575	0.4982

620 1024	0.3580 0.3781	0.4741 0.4893	0.5003 0.5179
-------------	-------------------------	-------------------------	-------------------------

실험 결과 모델에 관계없이 image resolution이 커질수록 성능도 높아짐을 확인할 수 있었다.

최종 Model

위의 실험들로 다음 요소로 학습한 최종 model을 선정했다.

- model: YOLO v8 X
- validation strategy: nms(iou: 0.7)
- image resolution: 1024x1024
- augmentation: mosaic, scale, translate, rotation
- epoch: 150 + early stop patient 10

Model	mAP50 (validation)	mAP50 (test)
YOLO v8 X base	0.4982	-
YOLO v8 X best	0.5742	0.4985

최종 model은 base model보다 mAP50 점수가 0.076 상승했다.

2-Stage

Models

2-stage model의 모든 실험은 mmdetection library를 사용하여 진행하였다.

2-stage 기반의 model 중 기초가 되는 모델인 Faster R-CNN⁵과 세부 객체 검출에 더 유리한 Cascade R-CNN⁶의 성능을 비교했다.

Model	Method	mAP50 (validation)	mAP75 (validation)
Faster R-CNN	Resnet50	0.402	0.219
	Resnet101	0.416	0.241
	Resnet50 + PAFPN	0.406	0.226
Cascade R-CNN	Resnet50	0.409	0.258
	Resnet101	0.423	0.284
	Resnet50 + PAFPN	0.417	0.275

Faster R-CNN과 Cascade R-CNN model을 동일 조건에서 비교 실험을 진행하여 상대적으로 성능이 좋았던 Cascade R-CNN을 2-stage 실험의 base model로 결정하였다.

Base experiments

Cascade R-CNN을 기반으로 일반적으로 Object detection task에서 성능의 향상을 보여줬던 방법들을 적용하여 실험 해보면서 우리의 문제 상황에서도 성능 향상을 이뤄 낼 수 있는지 비교 실험을 진행하였다.

Method	mAP50 (validation)	mAP50 (test)
base	0.409	0.4016
base + pretrained weight	0.426	0.4302
backbone Resnet50	0.409	0.4016
backbone Resnet101	0.423	0.4268
backbone SwinT-small	0.481	0.4841
backbone SwinT-base	0.481	0.4774
neck FPN	0.409	0.4016
neck PAFPN	0.417	0.4071
NMS	0.409	0.4016
Soft NMS	0.412	0.4057

⁵ Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440-1448.

⁶ Zhaowei Cai UC San Diego, Nuno Vasconcelos UC San Diego "Cascade R-CNN: Delving into High Quality Object Detection" Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 6154-6162

image scale 512	0.409	0.4016
image scale 1024	0.436	0.4487

실험 결과 모든 방법에 대해 성능 향상을 보였고 **backbone**과 **image scale**의 변화가 가장 큰 성능 향상을 보였다. 이 실험 결과를 참고하여 모델을 발전시키는 방향성을 잡았다.

Augmentation

데이터 수가 적은 문제와 같은 클래스 안에서 다양한 형태의 물체가 존재하기 때문에 **augmentation**을 사용하면 **model**이 **robust** 해지고 성능이 오를 것이라고 생각하여 **mmdetection**의 **alumentation example**에서 제공하는 **augmentation**들을 **augmentation v1**로 설정하여 **base** 모델(**SwinT-small backbone**)에 적용했다. 예상했던 것과 달리 성능이 떨어졌다. **Image**를 인식하기 어렵게 만드는 **blur**가 너무 많이 들어간 것의 영향이 클 것이라는 가설로 모든 **blur**를 제거해 실험하였고 성능이 상승한 결과를 보였다.

augmentation v1 : ShiftScaleRotate, RandomBrightnessContrast, IAARffine, CLAHE, Equalize, RandomRotate90, InvertImg, GaussianBlur, MedianBlur, Blur

Method	mAP50 (validation)
base (SwinT-small)	0.480
base + augmentation v1	0.391
base + augmentation v1 - Blur, MedianBlur, GaussianBlur	0.447

결과적으로 **augmentation**을 적용하면 적용하지 않았을 때보다 성능이 떨어졌다. **validation data**에 어떻게 표현되었는지 **mmdetection**의 **browse_dataset** 모듈을 통해 확인하였다. 성능 하락이 이상하지 않은 결과를 볼 수 있었다.

- 원본 & augmentation v1



다양한 크기의 물체를 효과적으로 탐지하기 위해 **multi scale learning**을 적용한 결과 성능이 향상되었다. 그리고 **augmentation** 방법을 하나씩 이미지에 출력해 보며 **image**의 영향을 최소화하면서 다양성을 확보할 수 있는 방법들을 선택해 **augmentation v2**를 구성했다. **augmentation v2** 적용 결과 성능 향상의 효과를 가져왔다.

augmentation v2 : RandomBrightnessContrast, CLAHE, RandomRotate90

Method	mAP50 (test)
base (SwinT-base)	0.6155
base + MultiScale learning	0.6272
base + MultiScale learning + augmentation v2	0.6367

최종 Model

Base Experiments 및 Augmentation 실험 결과를 바탕으로 성능 향상을 보인 변인들에 대해 아래와 같은 설정값을 적용한, 새로운 **base model**을 선정했다. 또한, 학습 속도 향상을 위해 **mixed precision** 기법을 도입했다.

- backbone: SwinT-base⁷
- neck: PAFPN
- box fusion strategy: soft-NMS
- image scale: 1024x1024
- multi-scale learning
- augmentation v2
- pretrained weights: cascade_rcnn_r50_fpn_1x_coco,
swin_base_patch4_window7_224_22k

이후, 실험 시작 시점에 비해 모델의 크기와 표현력 다소 증가한 점에 집중해서 학습 **dataset** 구성의 변화에 따른 **base model** 성능 향상 추이를 살펴보는 실험을 진행했다.

Method	mAP50 (test)
base	0.6414
base + no validation	0.6536
base + pseudo labeling	0.6309

실험 결과, **validation dataset**을 생성하지 않고 학습 **dataset**을 주어진 **train data** 전부로 구성하는 **no validation** 전략은 성능 향상(+0.0122)을 보인 반면 **base model**로 라벨링된 **test data**를 학습 **dataset**에 추가하는 **pseudo labeling** 전략은 성능 하락(-0.0105)을 보였다. 이는 학습 **dataset**에 새롭게 추가되는 **data**가 **no validation** 전략은 기존 **train data**, 즉 **annotation**의 질이 우수한 **data**인 반면 **pseudo labeling** 전략은 **mAP50** 기준 0.6414 정도의 성능을 가진 **model**로 라벨링된 **data**, 즉 **annotation**의 질이 상대적으로 우수하지 못한 **data**이기 때문인 것으로 생각된다.

위와 같은 과정을 통해 성능이 가장 우수했던 **base + no validation model**을 2 stage의 최종 **model**로 선정했다.

⁷ Ze Liu[†]* Yutong Lin[†]* Yue Cao^{*} Han Hu^{*‡} Yixuan Wei[†] Zheng Zhang Stephen Lin Baining Guo. "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows". arXiv:2103.14030v2.

4. Ensemble

서로 다른 방법론과 모델로 학습한 결과를 **ensemble**을 하면 각 결과의 단점을 보완하며 성능이 향상될 것이라 생각했다. 모델의 특징, 학습 시 사용된 데이터를 고려하여 **ensemble**을 진행했다.

Method	mAP50 (test)
2 stage stratified group 4-fold	0.6421
2 stage best + general trash	0.6384
2 stage best + 1 stage best	0.6262
top 6 best submissions	0.6606

결과적으로 가장 많은 모델을 합친 **ensemble** 결과가 가장 좋은 결과를 보였고, 결과의 다양성이 높을수록 **ensemble** 시 긍정적인 영향을 가져다 준다고 생각했다.

5. Conclusion

public

11 (-)	CV_18조		0.6606	67	4d
-----------	--------	--	--------	----	----

private

11 (-)	CV_18조		0.6414	67	4d
-----------	--------	---	--------	----	----

잘했던 점

- 원하는 기능을 추가하기 위해 코드를 수정하며 라이브러리에 대한 이해를 높였다.
- 많은 의견 공유를 통해 규칙을 정하여 실험을 계획하고 진행했다.
- 팀을 나누어 체계적인 파트 분배를 통해 실험을 진행했다.
- 근거 있는 실험을 진행했고, 결과에 대한 원인 분석을 했다.
- 새로운 팀으로 시작을 하면서 많은 회의를 하면서도 실험 시작후 가설 검증에 최선을 다했다.

아쉬웠던 점

- 초반 의견 조율에 시간을 많이 써서 진행하지 못한 실험이 있었다.
- 팀 간 소통이 부족했던 것 같다.
- 더 다양한 모델로 실험을 진행하지 못했다.
- 시간관계상 해보지 못한 실험이 있었고 변인통제를 잘 못한 부분이 있었다.
- 다양한 협업 툴을 사용하고 싶었지만 짧은 대회 기간 동안 전부 사용해 볼 수 없었다.
- 최신 기술들을 사용하지 못했던 것이 아쉬웠다.

대회 세부 사항은 [팀 노션 페이지](#)에서 확인할 수 있다.

프로젝트 학습 목표

- 새로운 팀원들을 만나 바로 진행된 프로젝트였기 때문에, 팀원들에 대한 이해도를 높이고 활발한 의사소통을 통해 팀 프로젝트 방향성을 함께 잡아가고자 했습니다.
- **Object detection** 분야에 대한 경험이 없었기 때문에, 가설을 세우고 실험으로 검증해보는 과정을 통해 얻은 인사이트로 **Object detection**에 대한 배경 지식과 문제해결 능력을 키우고자 했습니다.

전과 비교해서, 내가 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

- 이전에는 실험을 체계적으로 하지 않아서 실험을 통해 얻은 인사이트가 명확하지 않았고 이후의 파생되는 실험을 발전시키는 과정에서도 어려움을 겪었습니다. 이번에는 기초적인 실험부터 진행하고 정리된 인사이트를 이후의 실험에도 적용할 수 있게 함으로써 실험의 효율성을 높였습니다.
- 팀원마다 실험 진행 방식이나 집중하는 문제가 다를 수 있다고 생각했기 때문에 많은 회의를 통해 역할을 분담 하고 컨벤션을 통일하는 방식으로 협업을 진행하여 팀원이 실험한 내용을 팀 입장에서 수용하기 원활하게 했습니다.

나는 내 학습 목표를 달성하기 위해 무엇을 어떻게 했는가?

- 아직은 잘 모르는 팀원들에 대해 많이 알아가고 싶어서 다양한 주제에 대해 많은 토론을 해보기를 원했고, 저의 생각도 적극적으로 이야기를 하며 팀과 저의 생각을 많이 맞춰보려고 노력하였습니다.
- **Object detection** 분야에서 일반적으로 좋다고 알려진 방법론에 대해 실험을 우선적으로 진행하였고 실험 결과를 바탕으로 분석을 하면서 왜 이런 결과가 나왔는지에 대해 고민을 많이 했습니다.

나는 어떤 방식으로 모델을 개선했는가?

- **Cascade-RCNN** 모델의 성능을 올리기 위해 강의 시간에 다뤘던 **backbone**, **neck**, **image scale**, **suppression algorithm**에 대한 변화를 실험했습니다. 특히 **image scale**의 경우 전체 이미지 **scale**이 같아도 객체의 **scale**은 서로 다르기 때문에, 이미지를 **resize**시 객체마다 정보 손실이 다르게 일어날 수 있다고 생각했습니다. 이를 **multi scale learning**을 통해 다양한 **scale**의 객체를 학습 시켜 극복하려 하였고, 실험 결과 작은 크기의 객체에 대해 성능 향상이 있었습니다.
- 모델 분석 결과 **General trash**에 대한 성능이 가장 좋지 않았고, **EDA** 과정에서 생각했던 것처럼 클래스를 구분하는 기준이 명확하지 않기에 시각적 정보만으로 **General trash**를 구분하기가 어렵다고 생각했습니다. 이를 해결하기 위해 **General trash** 클래스만 학습에 사용한 모델의 결과와 기존 모델의 결과를 앙상블 하면 성능 향상이 오를 것이라 기대했습니다. 결과는 기존 모델을 사용하는 것보다 오히려 낮아졌고 **General trash**에 대한 오탐지율이 높았던게 문제였다고 생각했습니다.
- 학습 데이터를 전부 이용하고, **robust**한 모델을 만들고자 **K-fold** 기법을 이용했습니다. 눈에 띄는 성능 향상은 없었지만, **private** 점수에서 기존 모델보다 좋은 점수를 기록한 것으로 보아 **robust**한 결과를 보여줬다고 생각했습니다.
- **pseudo labeling**을 사용하여 더 많은 데이터를 모델에 학습 시키고자 하였습니다. 오히려 모델의 성능이 조금 하락했고 **pseudo labeling**시 더 고품질의 데이터를 만들어낸다면 개선이 될 것이라 생각했습니다.

마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- 시간 관리를 제대로 하지 못했습니다. 라이브러리를 이해하는 시간이 오래 걸려서 실험 시작이 늦었고, 모델을 커스터마이징 하는 것도 라이브러리가 어느 정도 이해된 후반 시점에 조금 밖에 해보지 못했습니다.
- **Object detection** 분야가 생소하다 보니 최신 트렌드의 방법론과 모델을 사용하기 보다 강의 시간에 다뤄서 알고 있는 기본적인 모델만 사용하게 되었던 것이 아쉽습니다.

내가 해본 시도 중 어떠한 실패를 경험했는가? 실패의 과정에서 어떠한 교훈을 얻었는가?

- 시간 관리를 잘 하지 못하여 제대로 증명해보지 못한 실험들이 존재했고 효율적인 시간 사용을 위해 기초적인 실험들은 학습 시간이 적게 걸리는 모델을 활용하여 검증해 보면 좋다고 생각했습니다.
- **Object detection** 분야에서 일반적으로 좋다고 알려진 기법들에 대해서는 성능 향상이 있었지만, 저의 직관을 통해 설계한 데이터 클리닝, 모델 구조 변경 실험에 대해서는 대부분 성능 향상이 없었습니다. 아직 **Computer vision** 분야에 대한 이해도가 부족하다고 느꼈고, **Kaggle**이나 논문을 참고하여 사용된 방법론을 더 많이 알아보고 시야를 넓히고 방법론 사용 시 충분한 이해를 동반해야겠다는 생각이 들었습니다.

협업 과정에서 잘된 점/ 아쉬웠던 점은 어떤 점이 있는가?

잘았던 점

- 팀원과 오프라인으로 만나 친해져서 편한 분위기가 형성되었고 많은 의견 교류가 이뤄져서 좋았습니다.
- 많은 회의를 통해 정해진 컨벤션과 규칙을 기반으로 개인의 실험 내용의 기록과 공유가 잘 이루어진 것 같아서 좋았고, 나중에 결과물을 합치는 과정도 원활하게 진행되었던 것 같습니다.

아쉬운 점

- 많은 의사소통을 하는 것은 좋았지만, 효율성 측면에서 조금은 개선점이 필요해 보였습니다. 팀 활동 시간이 길어지다 보니 개인의 시간이 부족해져서 개인의 능력과 잠재력을 전부 발휘하지 못했던 것 같은 모습이 보였습니다.
- 저는 아직은 협업 툴 사용에 있어서 어색한 느낌이 있었습니다. 팀원 모두가 협업 툴을 사용하는 방식이 조금은 달랐고, 익숙하지 않은 툴도 있어서 그랬던 것 같습니다. 대회라는 짧은 기간의 프로세스에 맞는 협업 툴을 사용해야 할 것 같습니다.

한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점은 무엇인가?

- 이론적인 지식에 대한 충분한 이해를 바탕으로 최신 트렌드의 방법론이나 모델을 적용하여 도전적인 방식으로 주어진 문제를 풀어보며 더 성장하고 싶습니다.
- 저의 개인적인 이해와 실험에 대한 내용을 충분히 정리하고, 팀원들이 잘 이해할 수 있게 충분히 고민 한 다음 공유하는 협업 방식을 통해 팀에 긍정적인 기여를 하고 싶습니다.

나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

- 나는 다음과 같은 목표를 가지고 프로젝트를 진행했다
 1. Object Detection 분야에 대한 이해도 높이기
 2. MMDetection과 YOLO등 Object Detection 라이브러리에 대한 이해도 높이기
 3. 전 프로젝트에서 Github 관리나 실험관리 등 잘 했던 요소들은 그대로 이어가고 잘 기록하는 습관, 메세지 큐를 이용한 자동 분산처리등 개선하고 싶었던 점을 개선하기
- 이를 위해 다음과 같은 규칙을 세우고 지키기 위해 노력했다
 1. 수업을 들으며 논문 찾아보고 정리하기, 모르는 내용 질문하기
 2. 코드를 자세히 살펴보고 어떤 구조로 되어있는지 찾아보기, 어떻게 구현되어있는지 확인해보기
 3. 개인 노선에 내가 한 실험을 모두 정리하고 아이디어나 생각들을 바로바로 메모하기

나는 어떤 방식으로 모델을 개선했는가?

1. 기존 train data를 random split 또는 stratified group k-fold로 train, valid data로 나누는 코드를 작성
2. 다른 캠퍼와 함께 MMDetection의 train.py 코드를 수정
3. MMDetection의 내부 코드를 수정해 wandb에 image가 올라가지 않던 문제를 해결, class별 mAP를 class별 mAP50으로 출력하도록 변경
4. json 파일을 입력받아 YOLO format의 data로 변경하고 학습을 진행하는 YOLO v8 baseline code를 작성
5. MMDetection과 YOLO 각각 config 폴더를 읽어 자동으로 학습시키는 shell 스크립트 작성
6. 모델들의 submission.csv 파일을 입력받는 앙상블 코드 작성
7. 다른 캠퍼와 함께 YOLO 모델 고도화
 - a. 모델별 성능 차이를 확인하기 위한 실험
YOLO v8 N, X에 대한 실험을 진행한 후 멘토님의 조언에 따라 M 모델을 실험했고 점수가 $N < M < X$ 임을 확인한 후 L까지 실험을 진행했다. 이후 가장 성능이 좋았던 X 모델을 사용하기로 결정했다
 - b. 클리닝된 데이터를 이용해 학습을 진행하는 실험
25 epoch 학습결과 뚜렷한 성능향상을 확인하지 못해 폐기처분하였다
시간상 150 epoch에서 학습해보지 못해 아쉬웠다
 - c. 성능에 영향을 미치는 요소(augmentation, nms iou, epoch, resolution)를 찾아보기 위한 실험
다른 augmentation 기법들은 같은 조건 하에서 실험했지만 rotation은 늦게 확인해 모델실험이 어느정도 된 후에 실험을 진행했다
 - d. 기존 submission에 Soft-NMS나 WBF를 적용해 제출하고 파라미터를 조절하는 실험

제출기회가 많지 않아 각각의 파라미터를 수정했을 때 어떤 결과가 나오는지 제대로 확인하지 못해 아쉬웠다

내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?

1. 지금까지 프로젝트를 할 때 모델을 제외한 부분을 직접 구현해 사용하고 있었는데 **High Level** 라이브러리를 처음 사용해보면서 라이브러리로 프로젝트를 어떻게 진행해야하는지 배울 수 있었다
2. **Object Detection** 라이브러리의 사용법을 배웠고 원하는 부분을 수정하려면 어떤 방식으로 접근해야 할지, 어떤 부분을 수정해야 자동화를 쉽게 할 수 있을지 배울 수 있었다
3. 캐글과 데이콘에서 다른 사람들의 솔루션을 보면서 다른 사람들은 어떤 식으로 코드를 짜는지 어떤 식으로 문제 해결을 위해 접근하는지를 찾아볼 수 있는 능력을 기를 수 있었다
4. 실험 내용과 왜 내가 그런선택을 했는지 그때그때 메모해두었는데 랩업리포트나 팀간의 발표자료를 만들 때 큰 도움이 되어 전 프로젝트보다 좀 더 성장한 것 같아 좋았다

마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

1. 팀원들의 의견을 잘 수용하지 못한 점을 저번 프로젝트에서도 문제점으로 썼었는데 이번에도 잘 고쳐지지 않은것같아 아쉬웠고 팀원들에게 미안했다
2. 위와 연결되는 내용인데 혼자 말을 너무 많이 하고 다른 사람의 얘기는 잘 듣지 못한것 같아 아쉬웠다
3. 저번 프로젝트에서 이어가고 싶었던 요소들을 많이 적용하지 못했고 개선해야 할 점도 기록하는 습관 외에는 개선하지 못해 딱히 성장한 느낌을 느끼지 못했고 이점이 아쉬웠다
4. 랩업 리포트를 쓰다보니 전체적인 실험이 하이퍼 파라미터 튜닝과 다름바가 없는것 같아 아쉬웠다
5. 실험중 변인 통제를 하지 못한 **resolution**이나 **epoch** 실험 등 실험들이 몇개 있어 그부분이 아쉬웠다
6. 다른 팀들은 앙상블 파라미터 분석이나 **submission** 파일을 분석해본 팀이 많았는데 우리는 앙상블은 제출기회가 별로 없어 못했고 **submission**은 분석하겠다는 생각을 못했다. 분석을 해봤으면 모델 특성을 좀 더 잘 알 수 있었을 것 같아 아쉬웠다
7. 실험을 돌려놓고 남는 시간이 있었는데 그 시간을 잘 활용하지 못해서 아쉬웠다

한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점은 무엇인가?

1. **Github** 사용하기
2. 메세지 큐를 이용해 서버를 묶어 관리하기
3. 새로운 방법론 적용해보기
4. 실험을 좀 더 신중하게 설계하기
5. 다른 팀들과의 교류를 통해 문제를 보는 다양한 시각 가져보기
6. 말을 줄이고 다른 팀원들의 말을 들어보기
7. 사용한 방법론은 최대한 빨리 공부하고 문서화하기

송인성_T5116 개인회고

나는 내 학습 목표 달성을 위해 무엇을 어떻게 했는가?

- 초반 1주일간 많은 회의로 인해 부족했던 강의 수강 시간으로 강의 내용을 주말동안 모두 수강 완료하며 빠르게 습득하려고 노력했다. 미션을 다 하지는 못했지만 챌린지의 목표들을 달성하기 위해 노력해서 완료했다.

나는 어떤 방식으로 모델을 개선했는가?

- 나는 **eda** 결과 엄청나게 작은 크기의 물체를 보고 작은 크기의 물체를 탐지 하는 것에 집중해봤다. 사이즈 범위별로 개수를 파악하고 이상치로 처리해도 되는 범위의 사이즈 만큼 제거 해보자는 분석을 해봤다. 성능을 높이진 못했지만 근거가 있었던 실험이라고 생각한다. 그리고 **stratified group kfold** 를 구현하며 **train set** 과 **validation set** 의 비율을 원본 데이터와 유지하며 나눠 학습과 검증이 유의미하게 이뤄지게 하였다. 그리고 **confusion matrix** 를 통해 백그라운드의 영향이 줄어들지 않는것을 보고 **2 stage detector** 의 **bbox head** 중 **class loss** 를 **focal loss** 로 바꿔 실험 해봤지만 결과가 좋지는 않았다. **Augmentation** 방법들을 팀원들과 눈으로 하나씩 살펴보고, 만들어서 적용한 조합이 성능 향상을 이뤄내서 뿌듯했었다.

내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?

- **Augmentation** 실험은 방법마다 눈으로, 실험으로 비교해 보며 어떤것을 조합할지 시도 해야한다는 것을 알게 되었다. 또한 **focal loss** 를 적용한다고 **2stage detector** 의 성능이 오히려 떨어지는 것으로 보아 **confusion matrix** 에서 FN 으로 **background** 의 퍼센트가 높은 것은 탐지하지 못한 남은 퍼센트 값을 **background**로 처리했기 때문이고 **focal loss** 를 적용하니 배경의 영향이 줄어들며 다른 작은 물체들을 오탐지했을 가능성이 높다고 생각한다. **Mmdetection library** 의 다양한 기능들을 알게 되었고, **mmdetection v3**의 사용법 또한 챌린지 종료 후 시도해본 결과 사용할 수 있게 되었다.

마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- 강의 미션 내용에 있는 **efficientdet** 관련 코드를 늦게 보고 **mmdetection v3** 에 있는 **efficientdet** 을 사용 해보려 하다가 실패하여 주말 대부분을 날렸다.. 이것이 가장 아쉬운 점이다. **mmdetection v2**의 **mmcv** 버전 문제로 해결이 안되는 것을 알게되었고 팀원들에게 공유 했지만 대회 마지막날 다른 팀에서 가상환경에서의 **mmdetection v3** 설치 방법을 공유하였다. 명백한 나의 한계였다. 잘못된 정보를 물어보지 않고 나의 시도에서 버전 문제로 나왔던 부분을 전부라고 생각하고 다른 팀원들의 시도를 뺀 것 같다.

한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

- 챌린지 종료 당일 새벽 시간 까지 **mmdetection v3** 사용을 위해 에러를 고치며 끝까지 시도 했다. 그 결과 **efficientdet d3** 가 작동하였고 학습 완료를 했다. 하지만 성능이 원하는 만큼 나오지 않았다. 아무런 고도화 작업 없이 진행했던 것 때문이었다. 다음 프로젝트에서 시도해볼 것은 내가 도전적인 업무를 주도해서 맡기 보다 서포트 역할을 하면서 많은 도움을 주는 것이다. **Conda** 가상환경에 설치하는 법을 배워서 다른 실험들을 진행 할때 도움이 될수 있도록 할 것이다. 그것이 성공적으로 진행되면 다시 도전적인 부분을 맡고싶다. 앞으로 질문을 끊임없이 하며 내가 아는것도 다시한번씩 체크하고 검증 받아야겠다.

학습 목표를 달성하기 위해 무엇을 어떻게 했는가? 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

본 챌린지의 개인 학습 목표는 효율적인 팀 실험 시스템을 체계적이고 올바른 방법으로 구축해보는 것이었다. 이로서 직전 대회에서 안 좋은 실험 시스템으로 인해 제한 시간 내 신뢰 가능한 실험 데이터를 확보하지 못하여 유의미한 결론을 도출하기 어려웠던 한계점을 개선해보고자 하였다. 개인 학습 목표를 달성하기 위해 새롭게 시도한 주요 변화들은 아래와 같다:

- 네이밍 컨벤션
 - 세부 내용: 파일 이름, 디렉토리 이름, WandB 런 이름 등 실험 관련 각종 요소의 명명 관련 팀 컨벤션을 구축하였다.
 - 효과: 팀원 간 의사소통이 수월해졌다. 또한, 팀원의 작업물을 별도의 설명 없이도 이해할 수 있게 되었다.
 - 고도화 전략: 코드 수준(.e.g, 변수 이름, 함수 이름, 주석, etc.)으로 확장하여 도입할 수 있겠다.
- 실험 스케줄러
 - 세부 내용: 쉘 스크립트 기반 단일 큐 실험 스케줄러⁸를 사용하였다.
 - 효과: 실험 가능 시간을 추가로 확보할 수 있었다. 실험 설계와 코드 수정 작업에 집중할 수 있었다.
 - 한계점: UI/UX가 불편하였다. 스케줄링 대상의 상태(e.g., 정상 종료, 비정상 종료, etc.)를 수동으로 파악해야 되었다. 스크립트 실행 과정에서 부가적으로 발생하는 산출물(e.g., nohup.out 파일)을 수동으로 관리해야 되었다.
 - 개선 방안: 스케줄링 대상의 상태를 세분화하여 자동으로 분류하도록 구현할 수 있겠다. 스크립트 실행 과정에서 부가적으로 발생하는 산출물(e.g., nohup.out 파일)의 관리를 자동화할 수 있겠다.
 - 고도화 전략: 큐를 여러 서버가 공유할 수 있도록 확장하여 구현할 수 있겠다.
- Ablation Study
 - 세부 내용: 각 실험 변인마다 **ablation study**를 철저하게 진행했다.
 - 효과: 실험 변인이 모델 성능에 준 영향을 보다 정확하게 분석할 수 있게 되었다.
 - 한계점: 소요 시간이 길어서 한정된 자원으로 다양한 실험을 병행하기 어려웠다.
 - 고민: 제한된 시간과 자원으로 실험의 다양성을 유지하면서 완벽한 **Ablation Study**를 진행하기 어렵다고 느껴 향후 방향성이 고민된다.

마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가? 한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점은 무엇인가?

실험 시스템 구축에 집중한 나머지 실험의 질에 신경쓰지 못한 점이 가장 아쉽다. 시간적 여유가 없다 보니 객체 탐지 태스크에서 범용적으로 사용되는 모델 성능 개선 기법을 중심으로 실험하게 되어 주어진 데이터셋만의 고유 문제점을 겨냥한 실험은 미처 설계해보지 못하였다. 또한, 처음 접하는 라이브러리에 단시간 안에 적응해야 되다 보니 기본적인 기능을 익히느라 객체 탐지 태스크의 최신 기술은 사용해볼 수 없었다. 따라서, 다음 대회에서는 주어진 데이터셋만의 고유 문제점을 파악한 후 하나의 문제점을 최신 기술로 깊이 있게 탐색하는, 좁고 깊은 방식의 실험을 시도해보고 싶다.

⁸ 소스코드 출처: T5089_박재민

최홍록_T5220 개인회고

- 나는 내 학습목표를 달성하기 위해서 무엇을 어떻게 했는가?

Level1 실험 당시 실험을 너무 중구난방으로 한 경험이 있어, 이번 실험에서는 실험에 근거를 가지고 실험하는 것을 목표로 했으며, 조 안에서 나뉜 팀원들과 논의를 하며, 근거를 가진 실험을 했으며, 예상치 못한 결과가 나왔다면 왜 그랬는지 분석하려 노력했다.

또한, 이런 라이브러리를 사용한 학습은 처음이어서, 시작은 막막했지만, 이 라이브러리들의 사용법을 익히는 것을 목표로 잡기도 했다. 결과적으로, 라이브러리의 구성과 모델 변경 방법 등, 여러 측면에서 라이브러리 학습을 성공적으로 할 수 있었다고 생각한다.

- 전과 비교해서, 내가 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

Level1 실험과 달리 데이터 자체가 가진 문제점을 파악하려고 여러 방향으로 **eda**를 진행했으며, **bbox** 개수, 크기, 중형비 등을 고려하여 **data cleaning**을 하는 시도를 했고, 결과적으로는 모델 향상이 일어나지 않았지만, 개인적인 학습 측면에서 시야를 넓힐 수 있는 기회라고 생각한다.

- 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

나는 이번 실험에서 **mmdetection**과 **yolov8(ultralytics)**, 두 개의 라이브러리를 모두 다뤄볼 기회가 있었다. 라이브러리들을 이용한 실험은 편리하지만, 반대급부로 실험자의 자유도가 별로 없는 환경이었다. 좀 더 자유도가 있었다면, 다양한 시도를 해볼 수 있었을 것 같다고 생각한다.

또한, 이번 실험에서는 초반에 많은 것을 정하려고 했고, 의견 조율 과정에서 프로젝트 시작 시기가 늦어지게 되어, 본격적인 실험을 한 기간은 주말을 포함해 1주일 정도밖에 없었고, 실제로 계획대로 진행되지 않은 점도 있었다.

- 한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점은 무엇인가?

이번 실험에는 의견 조율에 많은 시간이 걸려 하고싶은 실험을 못해본 부분도 있었다. 다음 실험부터는 너무 많은 것을 정하지 말고 개략적인 것만 정하고 일단 시작해보는 것도 나쁘지 않다고 생각한다. 많은 시간이 확보된다면, 좀더 다양한 측면에서 모델을 구상해볼 수 있을 것이며, 다양한 실험을 해볼 수 있을 것이다.

- 나는 어떤 방식으로 모델을 개선했는가?

내가 가진 지식 안에서 좋다고 알려진 것은 거의 다 해본 것 같다. **Augmentation**을 넣어 보기도 하고, **epoch**을 늘려보기도 하는 등 여러 실험을 했으며, 어떤 **augmentation**이 특히 효과가 있는지도 분석하여, 효율적으로 모델을 개선했다고 생각한다.

- 내가 해본 시도 중 어떠한 실패를 경험했는가? 실패의 과정에서 어떤 교훈을 얻었는가?

마지막 날 앙상블을 했는데, 하나의 앙상블 말고는 다 성능이 떨어지는 것을 목격했다. 이후 진행된 스페셜 피어세션에서 나눈 의견을 바탕으로, 앙상블은 어떤 경향성이 있는 것이 아닌 거의 무작위의 **hyperparameter tuning**이라는 것을 느꼈다. 이번 실험에 시작이

늦어지면서 낭비된 제출 기회가 많았는데, 다음 실험에는 이를 활용하여 앙상블 방법을 고민해보는 것도 좋을 것 같다.