Object Detection Wrap Up Report (재제출용)

1. Introduction

1-1. Problem Statement

'쓰레기 대란', '매립지 부족'과 같은 사회적 문제의 심각성이 날이 갈수록 높아지고 있다. 이를 해결하기 위해 사진을 통해 **쓰레기를 검출하여 분류**하는 모델을 탐구하여 올바를 분리수거 환경을 조성하는 것을 제안한다. 이와 더불어 우수한 성능의 모델은 쓰레기장에 설치되어 정확한 분리수거를 돕거나, 어린아이들의 분리수거 교육 등에 사용될 수 있다는 이점이 있다.

1-2. Competition Specifics

- Dataset
  - format : COCO format dataset
  - 전체 이미지 개수 : 9754 장
  - 10 class : General trash, Paper, Paper pack, Metal, Glass, Plastic, Styrofoam, Plastic bag, Battery, Clothing
  - 이미지 크기 : (1024, 1024)
- Evaluation metric : mAP50
- Experiment environment : GPU V100 server

2. Methods

2-1. Model

	Description
InternImage[1]	현 task에서는 비슷한 모양의 물건이 다양한 시각으로 나타나거나 같은 소재도 빛에 따라 다르게 나타난다. 따라서 Deformable Convolution 모델이자 현 시점 Object Detection 분야 SOTA인 InternImage를 시도할 가치가 있다고 판단했다.
UniverseNet[2]	이미지를 다양한 해상도와 크기로 치환하여 학습하여 다양한 크기의 쓰레기를 잘 검출할 것이라고 기대하여 선택했다.
Cascade R-CNN[3]	Cascade R-CNN은 낮은 IoU로 학습된 detector의 출력값으로 좀 더 높은 IoU를 설정해 detector을 학습시키며 stage가 진행될수록 좀 더 정확한 proposal로 학습한다. 난이도가 높은 이번 task의 특성 상 최대한 false negative가 작게 하는 이 모델이 좋을 것이라 판단했다.
YOLOv6[4]	YOLO v8, v7 이후에 출시된 모델로 yolo 계열 모델 중에 성능이 가장 높은 모델이다. 모델 다양성

	을 위해 1 stage detector도 포함하고자 선택했다.

2-2. Supporting Techniques

- 이전 대회 참가자들의 솔루션을 탐구한 결과 해상도 조절과 mosaic기법을 많이 활용하였다.
- 또한 대회에서만 통용되는 방법이지만, pseudo-labeling과 ensemble도 활용하였다.

3. Experiment

3-1. Data

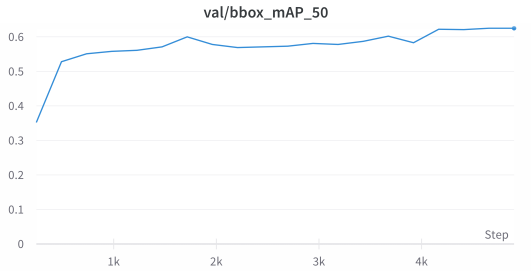
Dataset	Descriptions
train	제공된 전체 학습 데이터 (validation set을 나누지 않음)
train0 / valid0	`StratifiedGroupKFold`로 train / valid를 나눈 후 train 데이터에서 area 1024 미만 annotation 및 1024 이상 9216 미만 중 부분만 나오는 annotation을 삭제한 데이터
train1 / valid1	area가 1000 미만인 annotation을 전체 삭제하고 `StratifiedGroupKFold`로 train / valid를 나눈 데이터

EDA 결과 area가 작고 이미지에 일부만 나오는 object의 annotation에서 라벨이 잘못된 경우가 많았다. 이를 기반으로 annotation을 적절히 제거하여 학습에 사용하였다.

3-2. Model

Model	Descriptons	Public score
RetinaNet (SwinTransformer)	MMDetection, epoch 74, FocalLoss, L1Loss	0.4521
FasterRCNN (Resnet152)	MMDetection, epoch 20, CrossEntropyLoss, L1Loss	0.4763
Cascade R-CNN (InternImage)	epoch 14, CrossEntropyLoss, SmoothL1Loss	0.5864
UniverseNet	epoch 20, DistributionFocalLoss, GIoULoss	0.6140
YOLO v6	epoch 35, SIou, GIou	0.5903
Cascade R-CNN (swim-l)	epoch 45, CrossEntropyLoss, SmoothL1Loss	0.6276

Faster RCNN과 RetinaNet의 backbone 변경 및 epoch 증가 등 다양한 실험을 수행하였지만 public score가 0.5 이상 나오지 않았다. 모델 자체 성능의 문제임을 확인하고 PapersWithCode에서 성능이 검증된 SOTA 모델을 리서치하였다.

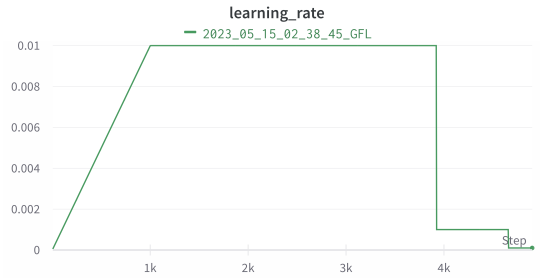


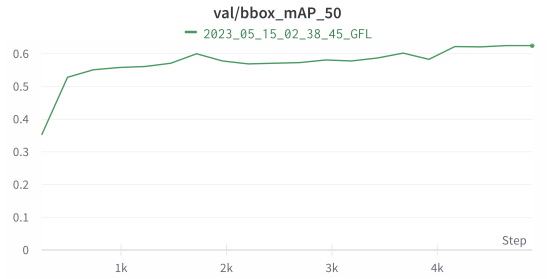
UniverseNet의 mAP50 validation 그래프

UniverseNet은 validation 기준 에폭 1회만에 mAP50이 0.5를 넘는 것을 확인할 수 있다. 즉 자체 성능이 높은 모델을 선택하는 것이 이번 task에 중요하게 작용했음을 확인할 수 있다.

3-3. Learning rate scheduler

다음은 universenet을 learning rate 최고점 0.01, 최저점 0.0001로 학습시킨 결과이다.





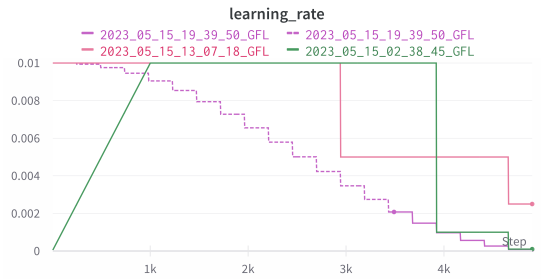
놀랍게도 l**earning rate의 변화폭이 클 때마다 성능 향상의 폭도 큰 것**을 알 수 있다. 초반 warmup 구간에서 mAP50이 가파르게 오르고, 후반 step scheduler가 적용될 때 정체되던 mAP50이 상승기조로 바뀌었다. 반대로 학습 중반 부분에서 학습률이 유지되니 mAP50의 상승 폭이 점차 정체되면서 하락하는 것을 확인할 수 있다.

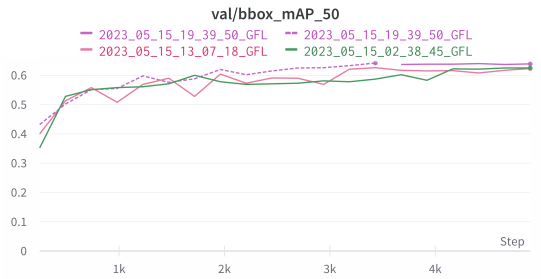
즉 **learning rate의 변화폭=성능 향상**이라는 추측을 할 수 있었다. 그래서 총 세 가지 learning rate를 비교했다. 나머지 환경은 동일하게 구성한 채, learning rate scheduler만 변경하였다.

초록색: step lr(gamma=0.1, warmup, 짧은 step)

핑크색: step lr(gamma=0.5, 긴 step)

보라색: cosine annealing





결론부터 이야기하면 validation 기준 cosine annealing의 최종 성능이 가장 좋았다. 즉 **학습률에 꾸준한 변화 폭을 줄 수록 학습을 안정적으로 수행할 수 있음**을 확인할 수 있다.

또한 cosine annealing은 가장 빠르게 에폭 14회만에 최적점에 도달하였다. 즉 **학습률에 주기적인 변화를 줄 수록 모델이 빠르게 최적화가 된다는 점**도 알아낼 수 있었다.

그러나 이처럼 validation에서 좋은 성능을 보인 모델이 test data에서는 성능이 감소하였다. 따라서 이후 모델 학습에는 원래대로 step lr을 활용하였지만, overfitting이나 잘못된 validation set 문제를 의심해볼 수 있는 만큼 일반화 기법과 함께 cosine annealing을 적용해본다면 모델 최적화에 효과를 볼 수 있을 것이라 추측한다.

3-4. Augmentations

1. Mosaic

	Data	Model	Score (map50)
Mosaic (o)	train0/valid0	Faster R-CNN	0.4300
Mosaic (x)	train0/valid0	Faster R-CNN	0.4034

모자이크를 하는 것이 성능 향상에 도움 됐다.

2. data cleansing

	Data	Model	Score (map50)
size 1000 이하 제거	train1/valid1	Faster R-CNN	0.4063
size 2000 이하 제거	train1/valid1	Faster R-CNN	0.3534
size 3000 이하 제거	train1/valid1	Faster R-CNN	0.3522

3. Pseudo Labeling

	Data	Model	Score (map50)
Original Model	train	Faster R-CNN	0.4665
Original Model + pseudo labeling(5 epochs)	train + pseudo labeled test dataset	Faster R-CNN	0.4188

Pseudo Labeling으로 5 epochs 진행하였을 때 오히려 성능이 떨어졌다. 기존 모델의 성능이 안 좋아서 이런 결과가 나왔다고 짐작한다.

3-5. Ensemble

Model	Public score	Stages	Data	Confidence Threshold	NMS IOU Threshold
Cascade RCNN (Swin-L)	0.6276	2	train	0.03	0.45
Cascade RCNN (Swin-L) + Mosaic	X	2	train0	0.03	0.45
YOLO v6	0.5893	1	train1	0.4	X
YOLO v6	0.5903	1	train0	0.2	X
UniverseNet	0.6140	1	train	0.4	X

각 모델의 예측 결과에서 중복 예측을 제거하고 오분류를 없애기 위해 NMS와 confidence score를 기준으로 예측된 box를 삭제하였다. 그 이후 IOU threshold 0.45인 Weighted boxes fusion 방식으로 위 표의 다섯 개 모델을 ensemble하여 publis score를 0.6775까지 향상시켰다.

	Public	Private
최종 제출 (5 model ensemble)	0.6775 (8위)	0.6652 (7위)

Private 점수 공개 결과 8위에서 7위로 올랐다.

4. Conclusion

이번 Object Detection 대회는 다양한 모델을 접할 수 있는 좋은 기회였다. InternImage, Universenet 등 수업에서 언급하지 않았던 모델을 리서치하여 적용하는 과정에서 시야를 넓힐 수 있었다. 팀원간 실험 공유도 노션과 슬랙을 이용하여 효율적으로 진행했다.

그러나 아쉬운 점도 많았다. 우선 팀원 모두 MMDetection을 처음 접해봐서 baseline 코드를 제대로 파악하는 데도 시간이 많이 소요됐다. 또한 train/valid으로 데이터를 나누는 과정에서 팀원 간에 일관된 데이터로 실험하지 못하여 정확한 비교가 어려웠다.

이 프로젝트를 통해 대회에서 mAP가 높은 모델이 서비스적으로는 꼭 그렇진 않을 수 있다는 것을 알게되었다. 또한 대회에서 ensemble 기법을 활용할 때 각 모델의 결과를 비교하여 시너지를 잘 낼 수 있는 조합을 찾는 것이 중요하다고 느꼈다.

5. References

[1] Wang, Dai, et al. “InternImage: Exploring Large-Scale Vision Foundation Models with Deformable Convolutions” arXiv preprint arXiv:2211.05778 (2022)

[2] Shinya, “USB: Universal-Scale Object Detection Benchmark” arXiv preprint arXiv:2103.14027 (2021)

[3] Cai, Vasconcelos, “Cascade R-CNN: Delving into High Quality Object Detection” arXiv preprint arXiv:1712.00726 (2017)

[4] Li, Chuyi, et al. "YOLOv6: A single-stage object detection framework for industrial applications." arXiv preprint arXiv:2209.02976 (2022)