

Object Detection Wrap-up Report

CV-13조 (SMiLE)

김영일, 안세희, 유한준, 윤일호, 이재혁

1. 프로젝트 개요

가. 프로젝트 주제

현대 사회는 대량 생산, 대량 소비의 시대로서, 많은 물건이 대량으로 소비되고 소비되는 시대를 살아가고 있다. 하지만, 이런 문화는 ‘쓰레기 대란’, ‘매립지 부족’과 같은 여러 사회 문제를 낳고 있다.

분리수거는 이러한 환경 부담을 줄일 수 있는 방법 중 하나다. 잘 분리배출 된 쓰레기는 자원으로 가치를 인정받아 재활용 가능하지만, 잘못 분리배출 된 쓰레기의 경우 그대로 폐기물로 분류되어 매립 또는 소각되기 때문에 환경 문제를 야기할 수 있다. 따라서 우리는 사진에서 쓰레기를 잘 탐지할 수 있는 딥러닝 모델을 개발하여 이런 문제점을 해결해보고자 한다.

나. 활용 장비 및 재료

(1) 서버

- CPU: Intel® Xeon® Gold 5120
- GPU: V100 32GB 1EA

(2) 의사소통

- 데일리 스크럼 및 피어세션 시간에 Zoom을 활용하여 실험 진행 방향 공유
- Slack기반 실험 결과 공유

(3) 협업

- Git를 활용하여 협업 진행
- Team Wandb 기반 실험 진행상황 공유

2. 프로젝트 팀 구성 및 역할

가. 팀 소개

저희 팀의 이름은 ‘SMiLE’로, 어떤 일을 하게 되든 항상 웃으면서 포기하는 인원없이 최종 프로젝트까지 다같이 열심히 달려보자는 포부를 가지고 있음.

나. 역할

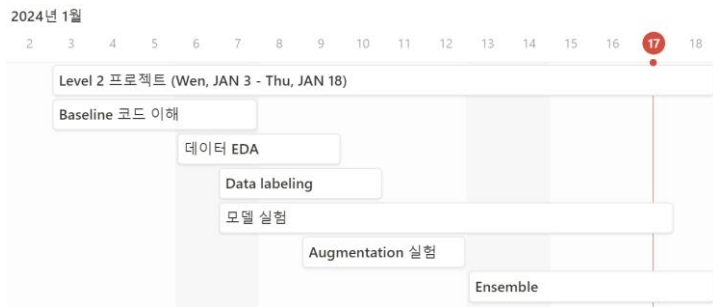
이름	역할
김영일 (팀장)	데이터 labeling 준비, EDA, one-stage 모델 실험, transformer 모델 실험, Random으로 object oversampling 후 모델 실험
안세희 (팀원)	데이터 labeling, 데이터셋 split, 추론 이미지 시각화, two-stage 모델 실험
유한준 (팀원)	데이터 labeling, two-stage 모델 실험, 업샘플링, 앙상블 및 멀티스케일 훈련, Confusion matrix 구현
윤일호 (팀원)	데이터 labeling, Faster R-CNN pytorch 구현 및 hyper parameter tuning
이재혁 (팀원)	Git 협업 준비, 데이터 레이블링, Augmentation 실험, two-stage모델 실험

3. 프로젝트 수행 절차 및 방법

가. 프로젝트 공동 목표

- 1주차: Baseline code 이해 및 EDA
- 2주차: 데이터 labeling, 여러가지 모델 및 augmentation 실험
- 3주차: 최적의 Ensemble 모델 탐색

나. 타임라인

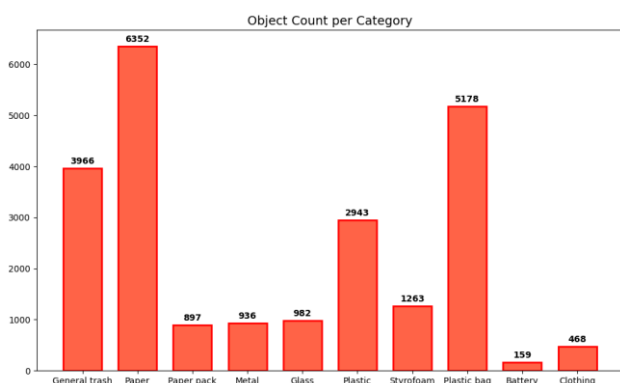


다. 협업 Rule

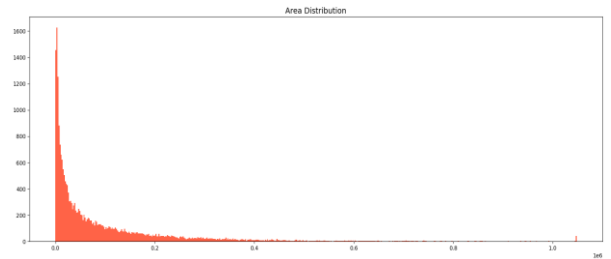
- 1 Issue, 1 branch, 1 PR을 기준으로 진행
- default branch를 develop으로 생각하여 작업을 진행하며, feature branch를 생성하고 작업이 완료되면 develop에 PR 진행
- conflict 방지를 위해 각 실험마다 별도의 폴더에 config 파일을 만들어 진행
- 데일리 스크럼시 당일 할 작업에 대해 공유하고, 진행 상황을 피어세션 때 공유

4. 프로젝트 수행 절차 및 방법

가. EDA (Exploratory Data Analysis)



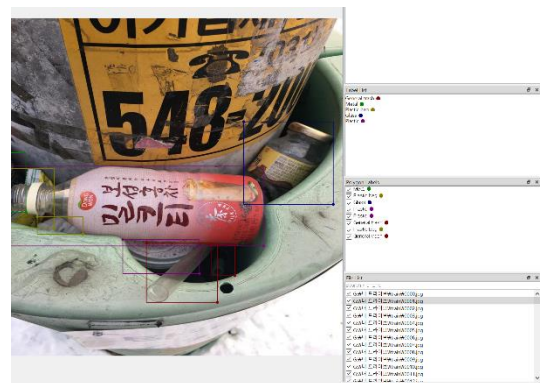
- 예측해야 할 클래스가 총 10개이며, 전체 데이터셋에 있는 object 수를 시각화 해본 결과 imbalance하다는 결과를 얻게 됨
- paper와 paper pack class의 경우 모델이 혼동해서 예측할 가능성이 높다고 생각함
- General Trash class의 경우 다른 class에 비해 가장 광범위하다고 생각했으며 시각화 해 볼 필요성을 느낌



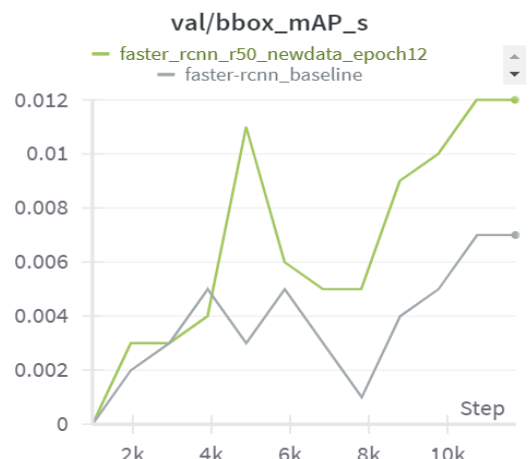
- object의 box면적 분포를 바탕으로 너무 작거나 큰 object의 경우 학습에 방해될 것이라는 생각이 들게 됨
- 면적이 40만보다 크거나 1000보다 작은 object를 제거하고 실험해본 결과 YOLOv8x 모델을 기준으로 mAP50이 0.4703 -> 0.4466으로 떨어지는 결과를 얻음

나. 데이터 전처리

(1) 데이터 레이블링



- Image Annotation Tool인 'Labelme'를 활용하여 Dataset labeling을 진행하였으며, 이를 통해 object class가 틀린 label을 수정하고 labeling 되지 않은 object를 추가함
- 담배꽂초, 전단지, 영수증 object는 General Trash class로 처리함



- YOLOv8x, Faster R-CNN 모델에 새로운 데이터셋으로 학습시킨 결과 test mAP-50가 0.0002 상승함
- 새로운 데이터셋 기반의 모델은 mAP_s 수치가 높으며, 원본 데이터셋 기반 모델은 mAP_m이 높아 둘을 앙상블 했을 때 0.02가 오르는 결과를 얻게 됨

(2) 배터리 클래스 Upsampling

- EDA 결과를 바탕으로 상대적으로 class수가 적은 배터리 class를 모델이 잘 예측하지 못할 것이라 생각했으며, confusion matrix 결과 또한 동일한 경향성을 보임
- 배터리 class에 이미지 증강 기법을 적용하여 데이터 수를 늘리는 방법을 시도하였으며, mAP50이 0.45 -> 0.5로 상승함

(3) Object OverSampling



- EDA결과 이미지당 object를 1개만 가지고 있는 이미지가 40% 이상인 것을 보고 해당 이미지에 random으로 object를 추가한 후 학습을 진행하고자 함
- object수가 부족한 class(배터리, 옷, 공병 등)를 추출하여 이미지로 저장한 후 object를 1개만 가지고 있는 이미지에 랜덤으로 추가하여 학습 진행
- YOLOv8x k-fold로 학습하여 WBF 앙상블한 결과로 비교해봤을 때 mAP50이 0.5608 -> 0.5506으로 감소하는 결과를 얻음

다. 모델 실험

Model (backbone)	Validation mAP	LB score
Faster R-CNN (ResNet-50)	0.414	0.3877
Faster R-CNN (ResNet-101)	0.417	N/A
RetinaNet (Swin T)	0.388	0.3532
YOLOv8n (x)	0.4102	0.3861
YOLOv8x (x)	0.5303	0.4703
Cascade RPN (ResNet-50)	0.402	0.3936
Cascade R-CNN (ResNext)	0.426	0.3944
Cascade R-CNN (ResNext-101)	N/A	0.4413
Cascade R-CNN (convnext)	0.159	N/A
Cascade R-CNN (Swin T)	0.522	0.5121
Cascade R-CNN (Swin T Large)	0.64	0.6367
ATSS (Swin T Large)	N/A	0.5764
RTMDet (Swin T)	0.382	0.3715
DINO (ResNet-50)	0.472	N/A
DINO (Swin T Large)	0.628	0.6005

라. 개선 기법

(1) Augmentation 실험

Augmentation	Validation mAP
None	0.4160
add mixup	0.3760
add mosaic	0.4240
add alumentation	0.4400

- Baseline으로 제공해준 모델인 Faster R-CNN을 바탕으로 augmentation 실험을 진행한 결과, alumentation을 적용했을 때 validation mAP score가 가장 높게 나타남

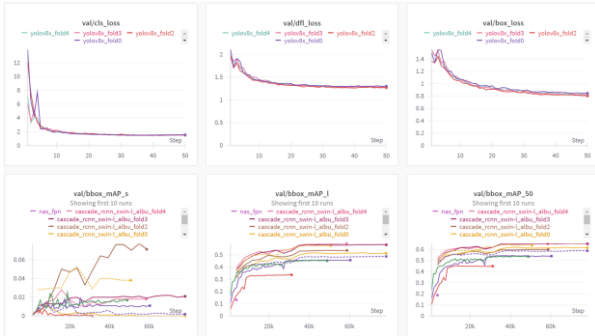
(2) Multi-scale Training

- 큰 해상도 이미지에서는 작은 객체를 잘 잡으며, 작은 해상도 이미지에서는 큰 객체를 잘 잡는 경향성을 활용하여 학습할 때 여러 scale을 입력으로 활용해보고자 함

- multi-scale을 적용하지 않았을 때에 비하여 mAP_s와 mAP_l이 모두 높아졌으며, mAP가 0.39 -> 0.45로 상승하는 결과를 얻음

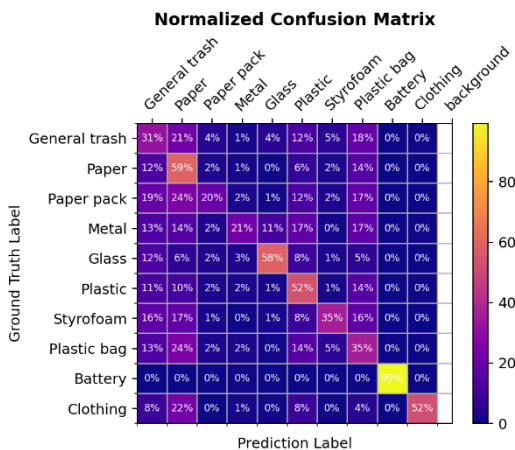
마. 실험결과 로깅 및 시각화

(1) Wandb



- Team 전용 Wandb를 설정하여 실험을 모니터링하였으며, 이를 통해 현재 모델의 성능을 가늠하고 다음 실험해볼 모델을 선정하는데 큰 역할을 함

(2) Confusion Matrix



- 학습한 모델 파일과 config를 이용하여 confusion matrix를 그려주는 코드를 구현함
- 모델이 배경과 물체를 잘 구분해내지 못한다는 인사이트를 얻었으며, 이를 통해 데이터셋 레이블링을 진행하게 됨
- 각 학습시킨 모델이 예측하는 경향성을 파악하여 최종 앙상블 시 사용할 모델을 선정하는데 도움이 됨

바. 앙상블

(1) 모델 후보

① YOLOv8x	kfold0: 0.5109
	kfold1: 0.52
	kfold2: 0.4959
	kfold3: 0.5194
	kfold4: 0.532
NMS Ensemble: 0.5402 (LB score)	
WBF Ensemble: 0.5608 (LB score)	

② ATSS (Swin T large)	0.5764 (LB score)
--------------------------	-------------------

③ Cascade R-CNN (ResNext-101)	kfold0: 0.565
	kfold1: 0.559
	kfold2: 0.559
	kfold3: 0.573
	kfold4: 0.543
WBF Ensemble: 0.5610 (LB score)	

④ Cascade R-CNN (Swin T)	kfold0: 0.522
	kfold1: 0.564
	kfold2: 0.517
	kfold3: 0.553
	kfold4: 0.538
WBF Ensemble: 0.5628 (LB score)	

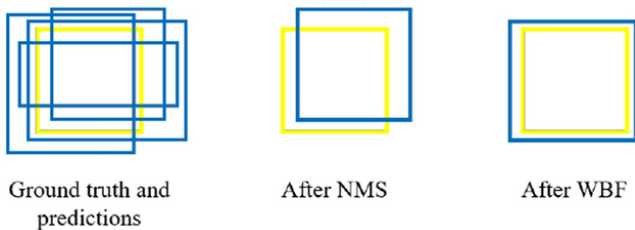
⑤ Cascade R-CNN (Swin T large)	kfold0: 0.617
	kfold1: 0.642
	kfold2: 0.598
	kfold3: 0.651
	kfold4: 0.649
NMS Ensemble: 0.6624 (LB score)	
WBF Ensemble: 0.6881 (LB score)	

⑥ DINO	0.6005 (LB score)
⑦ DINO (no val)	0.6364 (LB score)
NMS Ensemble: N/A	

(2) 앙상블 결과

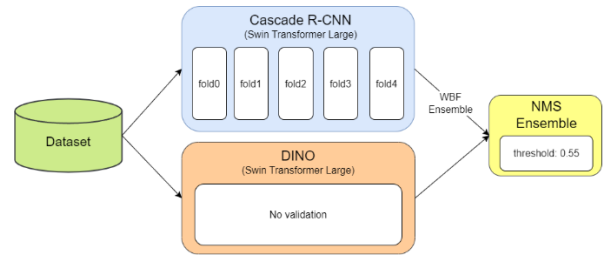
Candidates	LB score
① (nms) + ②	0.5642 (wbf)
① (nms) + ② + ③ (nms)	0.6041 (wbf)
① (nms) + ② + ③ (nms) + ④ (nms) + ⑤ (fold1)	0.6491 (wbf)
① (nms) + ② + ③ (nms) + ④ (nms) + ⑤ (fold1)	0.6443 (wbf)
① (nms) + ② + ③ (nms) + ⑤ (fold1)	0.6484 (wbf)
① (nms) + ③ (nms) + ⑤ (nms)	0.6501 (wbf)
① (wbf) + ③ (wbf) + ⑤ (wbf)	0.6688 (wbf) 0.6727 (nms)
⑤ (fold0, ..., fold4) + ⑥	0.6059 (wbf)
① (fold0) + ⑤ (wbf)	0.6878 (nms)
⑤ (wbf) + ⑥ (fold1)	0.6930 (nms)
⑤ (wbf) + ⑦	0.7013 (nms)
⑤ (wbf) + ⑦	0.7011 (nms, th=0.7)
⑤ (wbf) + ⑥, ⑦ (nms)	0.6980 (nms)

(3) NMS/WBF 앙상블



- kfold하지 않은 단일 모델의 결과를 다른 모델과 앙상블 할 경우 NMS보다 WBF를 활용하는 것이 score가 좋음
- kfold 앙상블 결과를 다른 단일 모델과 합칠 경우 NMS 앙상블을 진행한 kfold 모델이 더 좋은 성능을 보임

(4) 최종 선택 모델



- Cascade R-CNN 모델의 backbone을 Swin Transformer Large로 변경하여 학습을 진행하였으며, GroupStratifiedKfold한 원본 데이터셋을 활용하여 학습을 진행함. 각 kfold에서 나온 결과물을 WBF ensemble하여 LB score로 0.6884점을 받음
- Transformer모델인 DINO의 backbone을 Swin Transformer Large로 변경하여 전체 데이터셋으로 학습시킨 모델로 LB score 0.6364점을 받음
- 최종적으로 Cascade R-CNN과 DINO 두 모델을 NMS ensemble하였으며, LB score 0.7013점을 받음

바. 자체 평가 의견

(1) 잘했던 점

- 데이터를 보고 mAP를 어떻게 올릴 수 있을지 팀 전체가 토의하고 가설을 세운 후 그에 따라 실험을 진행한 점
- 데이터 labeling을 진행하면서 데이터 하나하나 뜯어보고 고칠 수 있었고, 성능 향상도 이루어졌던 점이 좋았음
- 모델별로 가지고 있는 특징을 파악하고, 다른 특징을 가진 모델들을 서로 앙상블 하여 모델의 성능을 올리는 과정이 좋았음
- 여러 Metric(mAP50, mAP75, mAP_s, mAP_m, mAP_l)을 보고 각 모델의 특징을 파악하려 했던 점이 좋았음

적응하는데 오래 걸려 진행이 부진했음

(4) 프로젝트를 통해 배운 점

- 어떻게 mmdetection을 사용해 객체 탐지 모델을 학습하고 추론할 수 있는지 배움
- 팀원들과 Git, Notion, Slack을 이용해 협업하는 방법을 배움
- Object detection task에서 고려해야 할 부분에 대해 좀 더 알게 되었고, 유효한 가설들을 경험적으로 배울 수 있었음
- COCO dataset 형식과 데이터셋 전처리 및 시각화 방법을 배웠음
- 데이터 레이블링을 진행하며 규칙을 정하고 이를 바탕으로 데이터 클리닝을 진행하는 과정을 배웠음

- 하나의 실험마다 하나의 branch를 파서 각각 구현한 파일들을 Git에 올리며 협업했던 시간들이 좋았음
- 다른 캠퍼님들에게 도움이 되고자 AI Stage 게시물에 YOLOv8 학습을 위한 데이터 전처리 과정 및 랜덤 시각화 코드를 공유했던 점이 좋았으며, 다음 프로젝트에서도 공유할 수 있도록 노력할 예정임

(2) 시도했지만 잘 되지 않았던 실험

- mmdetection 라이브러리를 이용해 더 많은 실험을 시도해보지 못한 점이 어려웠음
- YOLO 모델을 학습시킬 때 이미지 scale을 키워서 학습시키면 손실이 적어 결과가 좋을 것이라 생각했으나 결과가 좋지 않았음
- class수가 적은 object를 oversampling 했을 경우 imbalance 문제를 해결하여 성능이 올라갈 것이라 생각했으나, 성능이 저하됨
- 작은 객체만 잡기 위해 box가 작은 객체들만 따로 학습시켜 ensemble한 결과는 오히려 성능이 저하됨
- 성능이 가장 좋았던 모델에 Nas-FPN을 적용했으나, 성능이 저하됨
- mmdetection의 hook 기능을 이용해 confusion matrix와 submission 파일을 만들려고 했지만 잘 되지 않았음

(3) 아쉬웠던 점

- 모델을 한 번 학습하는데 시간이 너무 오래 걸려 많은 결과를 확인하기 어려웠음
- 새로운 데이터셋과 기존 데이터셋을 앙상블해서 더 좋은 결과를 내고 싶었지만 시간이 부족해 실험하지 못했던 점이 아쉬움
- SOTA 모델을 충분히 실험하지 못한 부분이 매우 아쉬움
- 초반에 baseline code를 이해하기 어려웠고