

Level-2 P-Stage : Object Detection

Wrap-UP Report by Team AI-TEEN(CV18)

1. 프로젝트 개요

1-1. 프로젝트 주제

코로나로 인한 비대면 식품 소비가 늘면서 포장재와 일회용품 사용 비율이 늘고 쓰레기 대란, 매립지 부족 문제가 부각되었다. 높은 분리수거 참여 비율에 비해 실 분리 수거율이 낮은 것이 문제의 원인이며, 이에 우리는 카메라 센서를 통해 입력되는 이미지 내에서 재활용품을 선별해 분리배출에 도움이 되는 솔루션을 만들고자 한다.

1-2. 활용 장비 및 협업 도구

- AI Stages 서버 내 V100, [Github](#), [Notion](#), WandB

1-3. 프로젝트 구조

- 학습 데이터 입력 및 출력
 - 입력 : 쓰레기 객체가 담긴 1024×1024 크기 이미지와 COCO format인 bbox 정보(좌표, 카테고리)
 - 출력 : bbox 좌표, 카테고리, score 값
- 사용한 최종 모델 구조
 - Cascade R-CNN, Yolov5x6 모델을 fine-tuning 후 출력값을 WBF로 앙상블
- 솔루션 적용 대상 및 적용 예시
 - 적용 대상 : 분리배출 교육 대상
 - 적용 예시 : 스마트폰과 같은 Edge device에 모델을 탑재

1-4. 기대 효과

쓰레기 선별작업과 분리배출 홍보에 드는 대다수의 비용은 인건비이다. 만약 이미지에서 분리배출 대상을 검출할 수 있다면 인적 자원 비용을 줄일 수 있을 것이다. 또한, 가정 및 사업장에서 이를 분리배출 교육용으로 사용해 분리수거 재활용 비율을 높일 수도 있을 것이다. 추가로 재활용 선별처리 공정의 자동화 과정에 기여하거나, 쓰레기 배출구성 지표를 얻을 수 있을 것으로 기대한다.

2. 프로젝트 팀 구성 및 역할

- 김서기 : 모델 개발(Yolov5, Cascade R-CNN), 모델 앙상블
- 김승훈 : EDA, 모델 개발(Yolov5, RetinaNet, CenterNet, Yolof_r50), 모델 앙상블
- 배민한 : 모델 개발(Swin, Yolov5, Cascade R-CNN), Yolov5), 모델 앙상블
- 손지아 : EDA, 모델 개발(Faster R-CNN, EfficientDet, Yolov5)
- 이상은 : EDA, 모델 개발(Cascade R-CNN, DetectoRS, Swin, Yolov5)
- 조익수 : 모델 개발(Cascade R-CNN)

3. 프로젝트 수행 절차 및 방법

3-1. 프로젝트 사전 기획 및 문제 정의

우리는 스마트폰과 같은 Edge device에 모델을 탑재해 분리수거를 도와주는 쓰레기 탐지 애플리케이션을 대회를 진행하면서 기획했고 이에 수행 속도가 빠르고 모델을 export하기 쉬운 Yolov5 위주로 실험을 하였다.

- 학습 데이터와 평가 데이터는 각각 이미지 4,883장과 4,871장이며 무작위로 선정한다.

- 이미지 내 쓰레기 종류는 총 10가지, 각각 0 ~ 9으로 라벨링되어있다. (General trash, Paper, Paper pack, Metal, Glass, Plastic, Styrofoam, Plastic bag, Battery, Clothing).
- 평가 지표는 mAP 50으로 모든 이미지의 클래스별 AP를 계산 후, 평균내어 최종 점수 산출한다.

3-2. 프로젝트 수행 절차 및 과정

1주차, 2주차는 데이터 EDA 및 MMDetection, EfficientDet으로 Object detection을 수행하였고 3주차에는 YOLOv5를 집중적으로 사용하고 mAP 값이 높은 여러 모델들을 앙상블하였다.

4. 프로젝트 수행 결과

4-1. 탐색적 자료 분석(EDA) 및 전처리

▼ bbox 시각화

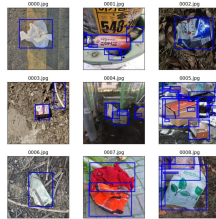


그림1. 재활용 쓰레기 이미지 예시

▼ 검출해야 하는 객체의 종류는 10가지이고 불균형이 심하다.

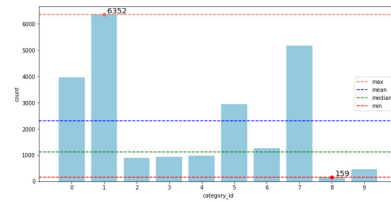


그림2. 클래스별 카운팅을 해봄.

▼ bbox의 비율(가로:세로) 분포

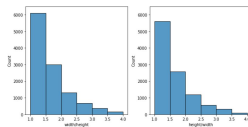


그림3. 비율이 4.0 이상인 bbox는 500개 미만으로 전체 bbox의 개수 23,000개 비하면 아주 적었다.

▼ 이미지별 bbox 1개인 경우가 많다.

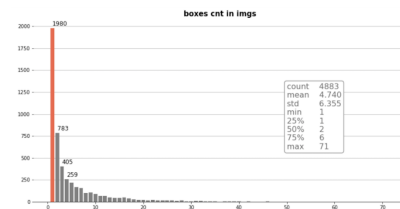


그림4. 이미지별 box 개수 counts

4-2. 모델 평가 및 개선 과정

- 모델 선정 : One-stage detector 중 학습 속도와 성능이 좋은 YOLOv5와, two-stage detector 중 Swin based model 선정
- 데이터 전처리 및 증강 : Test-time augmentation (TTA) in YOLOv5
- 하이퍼파라미터 튜닝 : Cosine annealing learning rate scheduler
- 모델 구조 : mAP 성능이 높은 모델들을 앙상블

4-3. 최종 선정 모형

- MMDetection의 Cascade R-CNN과 YOLOv5의 YOLOv5x6 모델을 fine-tuning, 이후 WBF로 앙상블했다.

1) Models

Architecture/backbone/Neck	Val_mAP50	LB_mAP50	optimizer	loss (class, bbox)	batch, epochs	Aa ETC
Cascade R-CNN / ResNet50 / FPN	0.715	0.521	SGD, CosineAnnealing, lr=1e-3 ~ 5e-6	CE SmoothL1Loss	4, 36	[1]
Cascade R-CNN / ResNet50 / RFP+SAC (DetectoRS)	0.810	0.559	SGD, CosineAnnealing, lr=1e-3 ~ 5e-6	CE SmoothL1Loss	4, 36	[1]

Architecture/backbone/Neck	Val_mAP50	LB_mAP50	optimizer	loss (class, bbox)	batch, epochs	ETC
Cascade R-CNN / Swin base / FPN	0.620 (12 epochs)	0.540 (12 epochs)	AdamW, CosineAnnealingWarmRestarts, lr=1e-5 ~ 5e-6	FocalLoss DIoULoss	4, 36*	[1].[2]
Yolov5 (Ensemble)	0.7197	0.547	SGD	BCEwithLogitsLoss, GloU	4, (10,20)	[4]
Yolov5 (s,m6,l6) / img_weights / Yolo_ensemble	0.7315	0.541	SGD	BCEwithLogitsLoss, GloU	4, (300,20,20)	Untitled

[1] Data argument

- RandomRotate90, GaussNoise(p=0.5), Resize(1024, 1024), RandomFlip(p=0.5), Normalize
- Blur, MedianBLur
- RandomBrightnessContrast, CLAHE, RandomGamma
- HueSaturationValue, RGBShift

[2] OutOfMemory

- soft_nms 적용
- 36epoch으로 설정했으나, OutOfMemory로 12 epoch 학습

[3] Image_weight

- Yolov5의 Image_weight 사용

[4] Details

- Data augmentation은 Yolov5에서 기본으로 제공하는 모듈 사용 (mosaic 등 포함)
- Yolov5x6 10 Epoch 돌린 모델과 20 Epoch 돌린 모델을 앙상블해서 탐지함
- Detection 과정에서 Yolov5모듈에서 제공하는 TTA 사용

2) Ensemble

- 위 단일 모델별로 얻어낸 `submission.csv` 들을 앙상블하여 최종 제출 파일을 생성
 - Public LB `0.610` / Private LB `0.590`
- 앙상블 방식으로는 `WBF` 를 사용했고 파라미터 설정은 아래와 같다.
 - `weight=None` , `iou_thr=0.5` , `skip_box_thr=0.0001`

5. 자체 평가 의견

- 김서기 : mmdetection이 Yolov5에 비해서 라이브러리 활용이 어렵게 되어있어서 고생을 많이 했다. 모델을 만드는 것에 그치는 게 아니라, 이를 활용해서 다른 애플리케이션에 적용해보고 싶다는 생각했다.
- 김승훈 : naive model 구현을 못 해 아쉽지만 다양한 library를 활용해 실생활 문제를 풀려고 시도했다는 점에서 나아졌다.
- 배민한 : Object detection의 다양한 프레임워크를 활용할 수 있어서 알 수 있었다.
- 손지아 : 다양한 프레임워크를 사용할 수 있었다. 많은 실험을 했지만 학습시간이 오래 걸려서 더 많은 실험을 하지 못한 점이 아쉽다. 팀원 모두 GPU와 메모리가 더 있었으면 좋겠다는 의견을 많이 나누었다.
- 이상은 : 서버가 한 개였기에 불편했다. 공부용 하나, 대회용 하나 해서 두 개였으면 더 좋았을 듯싶다.
- 조익수 : 모델 학습 시간이 길어 다양한 모델을 실험할 시간이 부족했다. 학습 데이터 불균형을 보완할 데이터 증강기법을 더 적용하지 못한 점이 아쉽다.