

## 1. 프로젝트 개요

### 1) 프로젝트 주제

- 사진에서 일반 쓰레기, 플라스틱, 종이, 유리 등 10 종류의 쓰레기를 Detection하는 모델을 만들어 쓰레기 분리배출에 도움이 되고자 함

### 2) 학습데이터 소개

- 전체 데이터 수 - 9,754장 (Size: 1024×1024)
  - Train - 4,883 장
  - Test - 4,871 장 (public 데이터는 평가 데이터의 약 50%)
- 총 10개의 Class - (“General trash”, “Paper”, “Paper pack”, “Metal”, “Glass”, “Plastic”, “Styrofoam”, “Plastic bag”, “Battery”, “Clothing”)

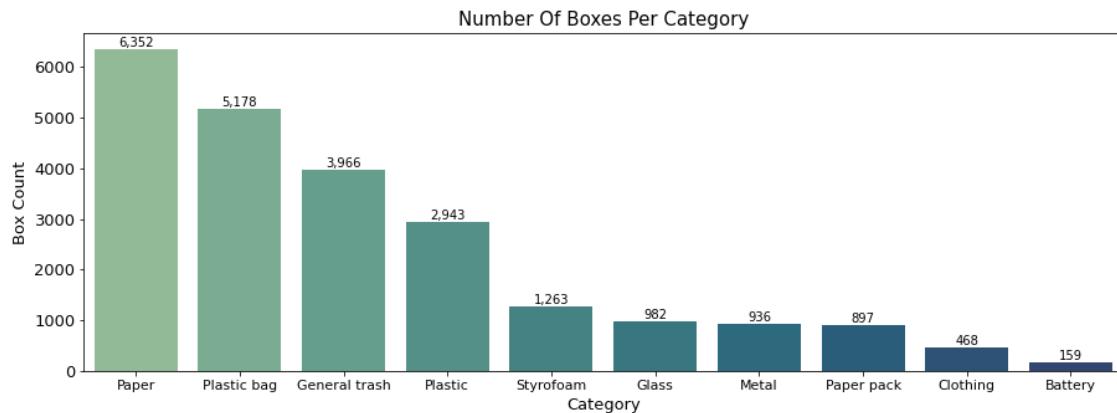


Figure1. Class별 Bounding Box 수 분포

### 3) 활용 장비 및 재료(개발 환경 등)

Development Environment	Reference Repository	Virtual Environment Library	
Language : Python 3.8.5 OS : Ubuntu 18.04.5 LTS Server : V100	<a href="#">MMDetection</a>	matplotlib 3.4.3 numpy 1.21.2 pillow 8.3.2 opencv-python 4.5.3.56	mmcv-full 1.3.14 mmdet 2.15.1 openmim 0.1.5
	<a href="#">YOLO v5</a>	matplotlib 3.4.3 numpy 1.21.2 pillow 8.3.2 opencv-python 4.5.3.56	pandas 1.3.3 scipy 1.7.1 pytorch 1.9.1 torchvision 0.10.1
	<a href="#">convert2 Yolo</a>	matplotlib 2.2.2 numpy 1.14.3 Pillow 7.2.0	

### 4) 기대 효과

쓰레기장에 설치되어 정확한 분리수거를 도와 인적자원의 낭비와 환경 부담을 줄이고, 어린아이들의 분리수거 교육 등에 활용할 수 있다는 기대효과가 있음.

## 2. 프로젝트 내용

### 1) EDA 탐색적 데이터 분석



Figure 2. 다양한 환경에서 촬영된 이미지



Figure 3. 불분명한 annotation

#### ① Class Dependency

- 전단지의 경우 일반 쓰레기와 종이 두 가지로 annotation되어 있음
- 유리와 투명 플라스틱이 매끈한 표면, 투명함 등 이미지상에서 유사한 특징을 보임
- 얇은 물체(노끈이나 줄 등)에 대한 background Error가 아주 높은 경향을 보임

#### ② Class Imbalance

- Figure 1에서처럼 Class별 bbox 수의 불균형이 심함
- 배터리의 경우, 데이터 수가 159개(평균 2,314 개)뿐

#### ③ Various Dataset Environment

- Figure 2 와 같이 다양한 환경에서 촬영된 이미지

### 2) Issue 및 성능 개선을 위한 시도

- ① Data Cleansing : Figure 3과 같은 train image의 잘못된 labeling이나 annotation을 수정해 성능 향상을 요함
- ② Data Augmentation : Class Imbalance 및 Image의 촬영 환경 보완을 위한 다양한 Augmentation 기법 시도  
→ Randomfog, Blur, RandomBrightness, Cutmix, Mixup, Mosaic, Resize, normalization, MultiScale, RandomFlip
- ③ Model Selection : Inductive bias를 최소화하기 위해 다양한 backbone model을 사용하여 학습
- ④ Generalization : 여러 Augmentation과 Noise를 넣어 시도, TTA시도
- ⑤ Pseudo Labeling: 학습한 모델로 test 데이터를 inference한 후, 그 결과로 추가 학습
- ⑥ Ensemble : 1-stage model 과 2-stage model을 Ensemble 함으로서 Robust한 모델 개선 시도
- ⑦ Binary Classification : 각각의 single class를 binary classification로 학습

### 3) Model Select

적절한 모델 선정을 위해 다양한 모델 실험

Model	설명	Baseline mAP	mAP
UniverseNet101	RetinaNet based, Res2Net-50-v1b + FPN + SEPC	0.520	<b>0.624</b>
Swin-S	각 patch를 window size로 나누어 해당 window안에서만 self-attention을 수행하고 window를 shift하여 다시 self-attention을 수행한 모델	0.523	<b>0.586</b>
Yolo-V5	CSP-Darknet backbone 기반, 크기 별로 모델 구성(s, m, l, x)	<b>0.586</b>	0.583
PVTv2-B3	기존 ViT 모델에서 self-attention의 연산량이 높아 high resolution image를 처리하기 어려워 이를 CNN으로 해결하고자 시도한 모델	0.541	<b>0.573</b>

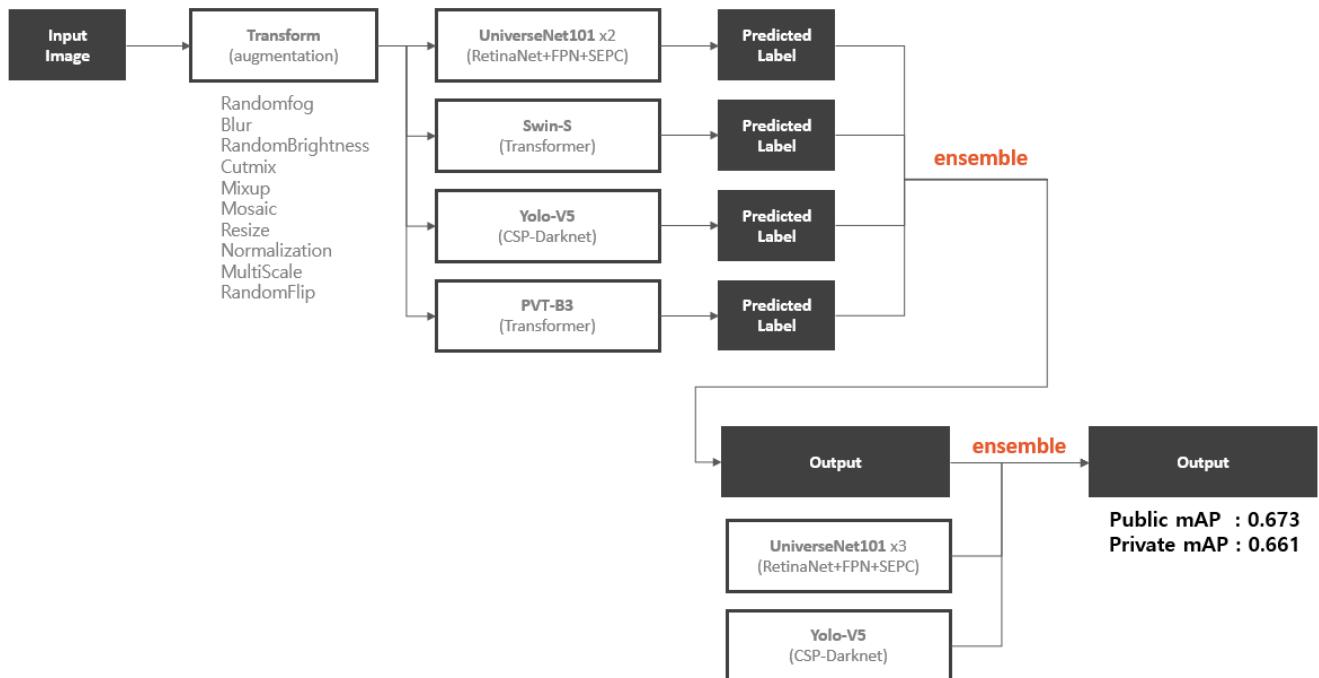
\* 상세한 실험 결과 Appendix A 참고

### 4) 검증(Validation) 전략

- Stratified validation과 Confusion Matrix를 통해 분류 모델 평가
- class별로 AP를 확인하면서 어떤 유형을 틀리고 있는지 파악

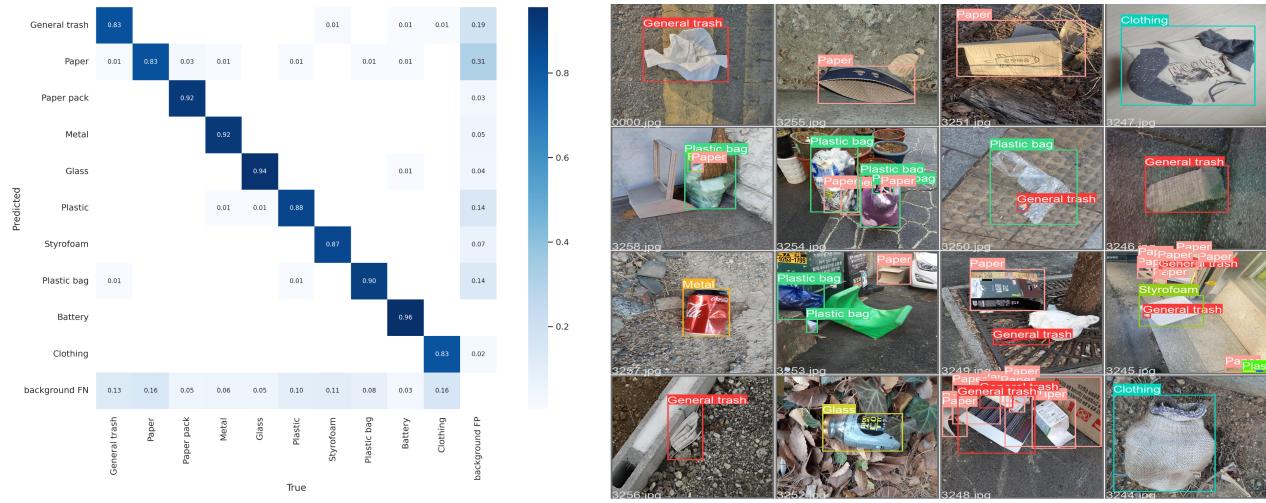
category	AP	category	AP	category	AP
General trash	0.543	Paper	0.508	Paper pack	0.724
Metal	0.698	Glass	0.656	Plastic	0.596
Styrofoam	0.593	Plastic bag	0.697	Battery	0.818
Clothing	0.740	None	None	None	None

### 5) 최종 모델



다양한 시도 결과 가장 성능이 좋았던 5가지 모델을 사용해서 양상을 시도

## 6) 시연 결과



Confusion Matrix

Inference Image

Figure 4. Confusion Matrix 및 시연 결과

Data 수가 적어 걱정했던 battery는 오히려 대부분이 clean data라 잘 검출해내는 반면, paper, general trash는 data의 수는 많지만 background와 겹치거나 annotation 자체가 잘못된 경우가 많아 model 성능이 제대로 나오지 않았음. 이에 따라 data cleansing을 시도했으나 효과가 없었던 걸로 보아 test dataset이 train과 품질 차이가 없을 것으로 생각됨. 하지만 다양한 augmentation과 ensemble의 경우 효과가 있어 generalization이 중요한 요소였을 것으로 예상됨.

## 3. 자체 평가 의견

### 1) 팀내 자체 평가 및 아쉬운 점

- mm detection과 관련된 다양한 라이브러리를 처음 활용함에 어려움이 있어서 내가 원하던 모델을 구현하는데 시간이 오래 걸림. 새롭게 배워간다기 보다 mm detection 라이브러리를 통해 기계적으로 파라미터 튜닝을 하여 성능을 향상시킨다는 느낌이 강해서 아쉬웠음
- object detection을 처음 입문함에 있어서 다양한 테스트를 해봐야 하는데 결과를 보기까지의 시간이 너무 오래 걸렸고, 이로 인해 다양한 실험을 기간 내에 해결하지 못함
- test dataset과 유사한 validation dataset을 찾기 어려워 연구가 진행될수록 model을 개선시키기 위한 뚜렷한 판단의 근거가 부족했음

### 2) 연구 방향 제시

- 겹쳐진 이미지 및 박스에 담겨져 있는 이미지들은 bounding box로 검출해 내는데 한계가 존재함 때문에 Segmentation task로 풀어 나가면 좋을 듯 함
- 재활용 쓰레기 데이터셋의 경우 box가 겹쳐있는 경우가 많아 Adaptive NMS 구현하여 적용시 성능 향상 예상됨

- Appendix A

Model	개선 시도	mAP
UniverseNet101	Mixup	0.573
	Mosaic / Adam / batch_size 16 / LR 0.0001 / Cutmix Dataset	0.419
	Mosaic / Adam / batch_size 16 / LR 0.0001	0.451
	HueStauration / Adam / batch_size 16 / LR 0.0001	0.619
	RandomFog / batch_size 16 / LR 0.0001	0.623
	RandomBrightness / batch_size 16 / LR 0.002	<b>0.624</b>
	Blur, RandomFog, RandomBrightness, Mixup / Adam / batch_size 16 / LR 0.0001 /Data Cleaning	0.613
	Data cleaning / RandomBrightness / batch_size 16 / LR 0.002	0.590
Swin-T,S	Swin-T / Adam / batch_size 4 / LR 0.0001	0.523
	Swin-T / MultiScale / Adam / batch_size 4 / LR 0.0001	0.550
	Swin-S / MultiScale / AdamW / batch_size 4 / LR 0.0001	0.561
	Swin-S / MultiScale, Mixup / AdamW / batch_size 4 / LR 0.0001	0.571
	Swin-S / Randomfog, RandomBrightnessContrast, ShiftScaleRotate, MultiScale, Mixup / AdamW / batch_size 4 / LR 0.0001	<b>0.586</b>
Yolo-V5	baseline (Yolo v5x6 default) / 300 epoch	<b>0.586</b>
	label smoothing (T=0.05) / 300 epoch	0.583
	single class / 300 epoch	0.008
	random fog / 50 epoch	0.573
	pseudo labeling (conf_threshold=0.6) / train data(10epoch x 3), pseudo data(10epoch x 2) 번갈아가며 학습 / pseudo data 학습시 obj loss 제외 / 50 epoch	0.559
PVTv2-B3	baseline epoch 30 / batch size 4 / scheduler step / LR 0.002 / fp16	0.541
	batch size 3 / LR 0.0001 / Adam / score_thr = 0	<b>0.573</b>
	batch size 2 / LR 0.0001/ mixup, RandomFog, blur, randbright / Adam / Data cleaning	0.526