

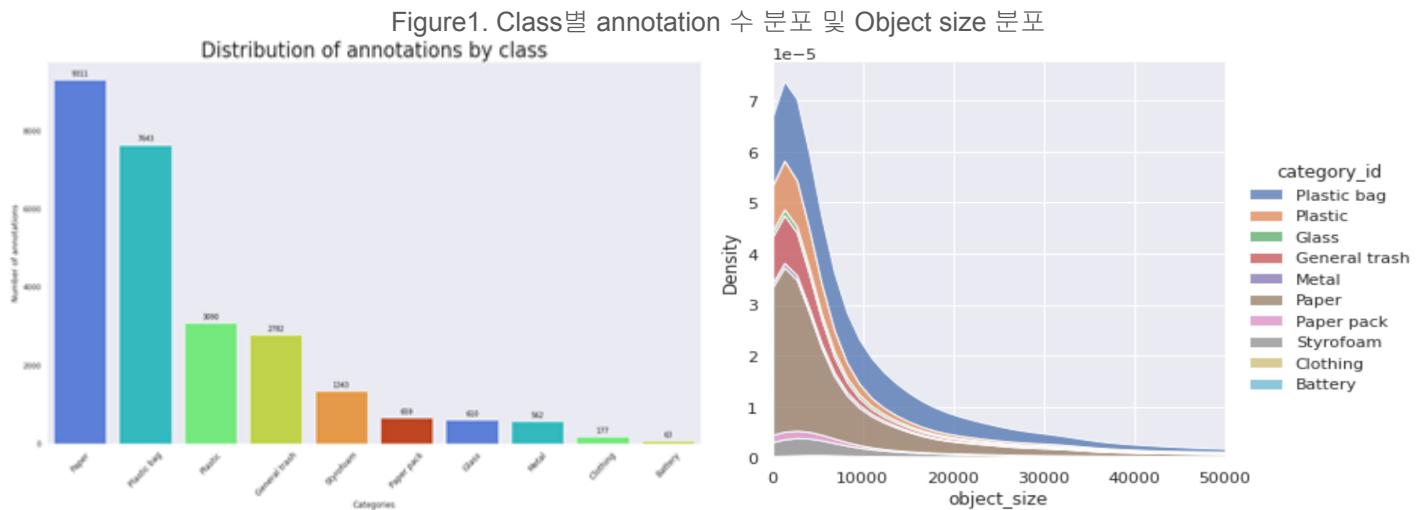
1. 프로젝트 개요

1) 프로젝트 주제

- 재활용 쓰레기 사진에 대하여 일반 쓰레기, 플라스틱, 종이, 유리 등의 10 종류의 재활용 품목으로 semantic segmentation을 수행함

2) 학습데이터 소개

- 전체 데이터 수 - 4091 장 (Size: 512x512)
 - Train - 3272 장
 - Test - 819 장
- 총 11개의 Class - ("BackGround", "General trash", "Paper", "Paper pack", "Metal", "Glass", "Plastic", "Styrofoam", "Plastic bag", "Battery", "Clothing")



3) 활용 장비 및 재료(개발 환경 등)

Development Environment	Reference Repository	Virtual Environment Library	
Language : Python 3.7.11 OS : Ubuntu 18.04.5 LTS GPU : V100	MMSegmentation	matplotlib 3.4.3 numpy 1.21.2 pillow 8.3.2 opencv-python 4.5.3.56	mmcv-full 1.3.16 pytorch 1.7.0 torchvision 0.8.0 mmsegmentation 0.18.0
	Segmentation Models	matplotlib 3.4.3 numpy 1.21.2 pillow 8.4.0 opencv-python 4.5.3.58	pytorch 1.7.0 torchvision 0.8.0 timm 0.4.12 smp 0.2.0

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. 얇은 물체 annotation



Figure 3. 종량제 봉투 내부 object annotation 되어있지 않음

① Perceptual distance를 이용한 val dataset 검증

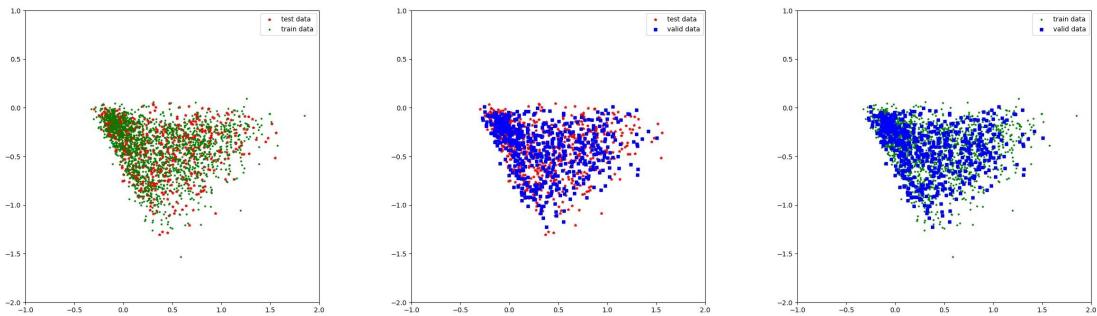


Figure 4. train, valid, test dataset의 Perceptual Distance Clustering [녹-train, 청-valid, 적-test]

- VGG19 Imagenet Pretrained Model을 이용하여 High-level feature map을 추출함
feature map은 High-level feature를 잘 뽑는다고 알려진 conv4_4 output을 사용함 (CartoonGAN, Perceptual loss 논문 ref)
- feature map 추출 이후 global average pooling을 이용하여 128 차원의 1d vector로 만든 후, autoencoder에 학습시켜 클러스터링 하였음
- High-level feature가 비슷할 경우, 분포하는 class가 유사할 것이라고 생각하여 진행함
- clustering 결과, train, test, valid set이 유사한 분포를 보였음
- 대신 train data(녹)의 경우 valid나 test에 비해 좀 더 넓은 분포를 가지고 있고, 몇몇 데이터는 분포에서 많이 멀어져있는 것을 확인함
- 해당 데이터들이 노이즈 형태로 학습에 악영향을 미칠 수 있다고 판단하였음

② 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

③ Class Imbalance

- Figure 1에서 처럼 Class 별 annotation 수의 불균형이 많이 나타남
- 배터리의 경우, 데이터가 63개로 다른 class에 비해 현저히 적음.

④ Various Dataset Environment

- Figure 2와 같이 다양한 환경에서 촬영된 이미지

2) Issue 및 성능 개선을 위한 시도

- ① **Data Augmentation** : Class Imbalance 및 Image의 촬영 환경 보완을 위한 다양한 Augmentation 기법 시도
→ Rotate, RandomResizedCrop, MotionBlur, GridDistortion, HueSaturationValue, RandomBrightnessContrast, ImageCompression, Hor/VerFlip
- ② **Model Selection** : 최적의 모델을 찾기 위해 다양한 모델로 실험
- ③ **Generalization** : 여러 Augmentation과 Noise를 넣어 시도
- ④ **Pseudo Labeling** : 학습한 모델로 test 데이터를 inference한 후, 그 결과로 추가 학습
- ⑤ **CRF(Conditional Random field)** : denseCRF 후처리를 통해 픽셀단위의 정확도 향상 도모
- ⑥ **Ensemble** : 여러 모델을 Ensemble(soft or hard voting) 함으로서 Robust한 모델 개선 시도
- ⑦ **YohanMix** : 클래스 불균형 해소를 위해 적은 개수의 클래스의 image를 기준 dataset에 CutMix와 같은 방식으로 이어붙이는 방식.
- ⑧ **TTA(Test Time Augmentation)** : 학습 때와 다른 input image를 통해 inference 하는 방법 / Multiscale

3) Model Select

적절한 모델 선정을 위해 다양한 모델 실험

Model	설명	mIOU (LB Score)
DeepLabV3++ (se-resnext)	CNN에서 semantic info를 더 detail하게 고려하기 위해 만든 모델	0.700
UPerNet (Swin)	Parts, Meterial, Scene, Objects, Textures 같이 다양한 visual concept을 파싱한 후, 이 정보들을 통합 이용하여 segmentation을 수행	0.767
OCRNet (HRNet)	객체와의 관계를 통해서 각 픽셀의 정보를 유추하는 OCR를 통해 pixel representations을 강화하고, HRNet을 사용해서 중요한 부분을 high resolution 정보를 계속 추가해주는 방식의 모델	0.705

* 주요 모델만 기재, 상세한 실험 결과 Appendix A 참고

4) 검증(Validation) 전략

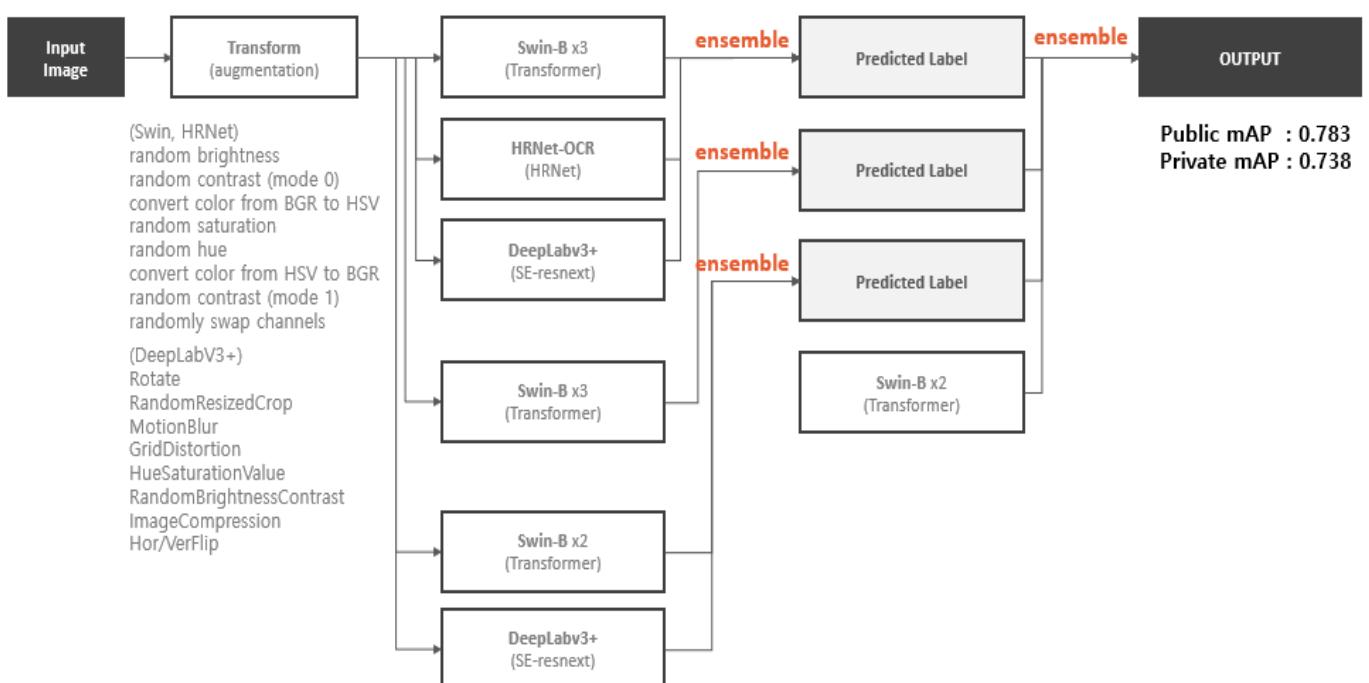
- Stratified validation과 Confusion Matrix를 통해 분류 모델 평가
- class별 mIOU를 validation set으로 확인하여 모델 평가

모든 모델이 General trash, Paper pack, Plastic Class에서 정확도가 떨어지는 경향성을 보임

Class	IoU	Acc
Background	97.2	98.83
General trash	53.91	60.37
Paper	85.84	92.89
Paper pack	59.49	68.19
Metal	74.27	80.28
Glass	68.35	77.26
Plastic	57.47	72.72
Styrofoam	82.42	87.15
Plastic bag	88.56	96.56
Battery	89.91	95.3
Clothing	67.49	75.29

aAcc	mIoU	mAcc
95.63	74.99	82.26

5) 최종 모델



다양한 시도 결과 가장 성능이 좋았던 모델들로 수도라벨링을 한 후, 이 모델들을 사용해서 양상을 시도

6) 시연 결과

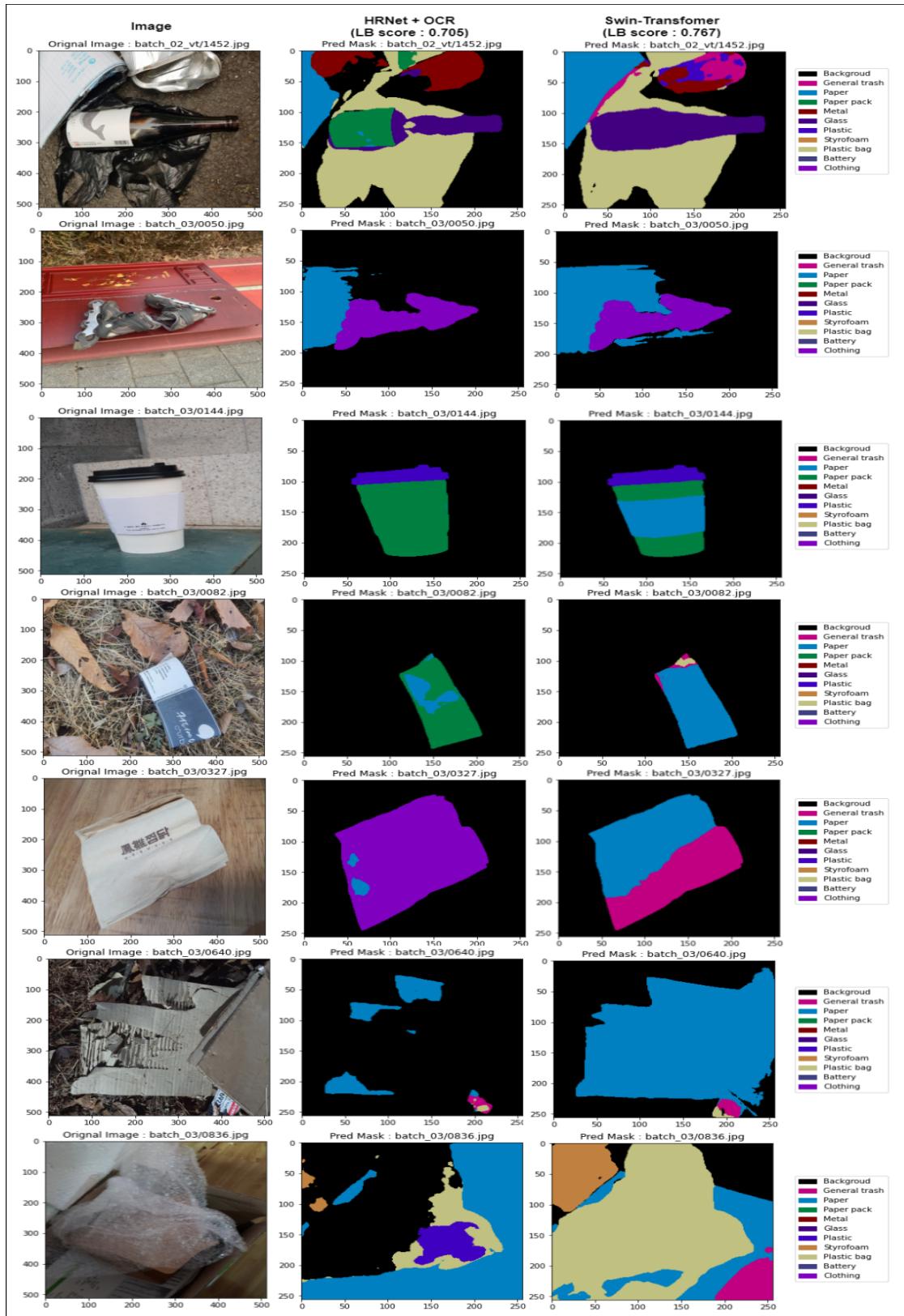


Figure4. 시연 결과

- 모델 별로 inference 결과가 다르게 나와서 ensemble 시도하였더니 성능이 더 좋아진 것으로 보임
- 전체적으로 OCR은 object의 모양과 texture 학습하는 경향이 있고, Swin은 image의 맥락을 학습하는 경향이 있는 것으로 사료됨

3. 자체 평가 의견

1) 팀내 자체 평가 및 아쉬운 점

- **mmsegmentation**과 관련된 다양한 라이브러리를 처음 활용함에 어려움이 있어서 내가 원하던 모델을 구현하는데 시간이 오래 걸림. 새롭게 배워간다기 보다 **mmsegmentation** 라이브러리를 통해 기계적으로 파라미터 튜닝을 하여 성능을 향상시킨다는 느낌이 강해서 아쉬웠음
- **test dataset**과 유사한 **validation dataset**을 찾기 어려워 연구가 진행될수록 **model**을 개선시키기 위한 뚜렷한 판단의 근거가 부족했음
- **SMP**에서 실험한 **augmentation test**가 **mmseg**에서는 적용 불가능하거나, 동일한 효과를 보이지 못해서 아쉬웠음
- **Pseudo labeling** 이외의 방법들은 모두 **public-private LB**가 많이 차이나서 **robust**한 모델을 만들지 못했다는 생각이 들어 아쉬웠음
- 성능평가 및 결과 분석, **EDA**를 통해 개선점을 찾아나가는 과정이 부족했다는 것이 아쉬움
- 개선 아이디어를 좁은 범위에서만, 이미 알려진 것에서만 찾았던 것 같음. (**augmentation**, **ensemble** 등) 재활용 데이터 셋 특징에 맞는 개선 아이디어를 찾지 못했던 것 같음. 딥러닝 범위 밖에서도 생각해보고, 적용가능성을 검토해봤어야 했음
- **SOTA** 모델이 무엇인지 같은 정보는 검색을 통해 찾아 볼 수 있었지만, 모델을 어떻게 운영할 지에 대한 경험이 부족해, 전체적인 상황을 보는 시각이 좁다는 느낌을 받았음
- 기존 대회와 다른 원가 기발한 방법을 적용시켜보고자 했으나, 크게 다르거나 획기적인 방법을 찾아내지 못했음.

2) 연구 방향 제시

- **Object**만 고려하기보다 **semantic info**를 함께 고려하는 **architecture**를 사용한다면 더 성능이 높아질 것이라 생각됨
- 전체적으로 **General trash**, **Paper pack**, **Plastic Class**에 대해서 정확도가 떨어졌으며 이에 대한 해결책으로 다양한 **augmentation**을 추가해보았으나 성능상 큰 효과를 나타내지 못하였고, 이를 해결하기 위한 다른 방안이 있다면 더 좋은 결과를 도출해낼 수 있을 것이라 생각됨.
- 애매한 **object**는 따로 **class**를 만들어 검출한 후 기존 **class**에 합친다면 효과가 올라갈 것으로 예상
- **Dense CRF** 시도시 예측한 마스크의 형태를 **Image**에 맞게 조정해주는 역할을 해줬는데, 예측한 **Mask**의 크기를 키운 후 **Dense CRF** 시도하면 성능 개선에 도움이 될것이라 생각됨

4. Appendix (실험결과 링크)

Architecture	Backbone Model	테스트 및 개선 시도	mIOU (LB)
DeepLabv3+	resnet 152	-	0.646
	se_resnet 152	-	0.611
	senet 154	-	0.671
	regnety	-	0.669
	dpn	-	0.637
	efficentnet-b7	-	0.652
	se_resnext 101	warmup cosine annealing , Augmentation	0.7
Upernet	Swin-B	기본	0.755
	Swin-B	dense-CRF iter 15회	0.755
	Swin-B	TTA (multi-scale)	0.764
	Swin-B	K-fold (k=5) hard-voting ensemble	0.762
	Swin-B	TTA (flip, multi-scale)	0.766
	Swin-B	pseudo labeling, TTA (flip, multi-scale)	0.776
	Swin-L	-	0.749
OCRNet	HRNetV2p-W48	기본	0.695
	HRNetV2p-W48	TTA (multi-scale)	0.705
UNet++	se_resnext 101	-	0.655
	HRNet	-	0.569
	resnet	-	0.472
FCN	resnet 101	cutmix 미사용	0.5624
	resnet 101	cutmix battery 추가	0.5571
	resnet 101	cutmix metal, glass, battery, clothing 추가	0.5688
PAN	resnet 101	-	0.585
	se_resnext 101	-	0.653