

# Level-2 P-Stage : Semantic Segmentation

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

## 1. 프로젝트 개요

### 1-1. 프로젝트 주제

우리는 재활용 쓰레기 분리배출을 잘 하는 국가로 알려져 있으나 지난 15년 동안 재활용 시장을 위한 투자나 인프라 구축을 거의 하지 않았기 때문에 분리배출, 선별, 재활용 등이 제대로 운영되고 있지 않은 실정이다. 재활용되지 않고 버려지는 수많은 쓰레기로 인해 지구온난화, 바다로 유입되는 플라스틱, 야생 동물 생태계 위협 등 많은 문제가 발생한다. 이를 해결하기 위해 우리는 Computer vision AI 기술을 사용하고자 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  $512 \times 512$  크기 이미지와 COCO format 정보(마스크, 카테고리)
  - 출력 : 예측 클래스 및 마스크 정보를  $256 \times 256$  Pixel 좌표로 표현
- 사용한 최종 모델 구조
  - Decoder : FPN
  - Encoder : timm-efficientnet-b6 (pre-trained weight: 'noisy-student'.)

### 1-4. 기대 효과

Semantic Segmentation 모델은 객체 테두리를 더 정교하게 식별할 수 있어서 단순한 Detection 모델보다 재활용 쓰레기 분리배출 공정을 자동화하는 데 유리하다. 그렇기에 쓰레기 선별작업과 분리배출에 드는 인적 자원을 줄일 수 있을 것이다. 또한 고도로 정확한 모델을 개발할 수 있다면 쓰레기 수거 드론, 쓰레기 배출 방지 비디오 감시, 사람의 쓰레기 분류를 돕는 AR 기술에 적용할 수 있을 것이다.

## 2. 프로젝트 팀 구성 및 역할

- 김서기 : Research(*HRNet*, *MMSeg* library), Pseudo Labeling, TTA
- 김승훈 : Find Augmentations, Code Refactoring
- 배민한 : Research(*smp* library, loss), Model Enhancement, K-Fold, Ensemble
- 손지아 : Research(*smp* library, loss), Model Enhancement, MLK-Fold
- 이상은 : Research(*HRNet*, optimizer, loss), Pseudo Labeling, Augmix
- 조익수 : Research(*MMseg* library)

## 3. 프로젝트 수행 절차 및 방법

### 3-1. 프로젝트 사전 기획 및 문제 정의

강의에서 수강한 Semantic Segmentation Method 이론을 바탕으로 Torchvision, Segmentation-models-pytorch의 Segmentation model을 사용하여 SOTA 에서 검증된 모델을 기준으로 모델을 선정했으며, Augmentation의 경우 Fold를 적용하지 않은 가벼운 모델(MobileNet)을 이용하여 입력 데이터에 적용 가능한 사항들을 확인했다.

- 전체 이미지는 4,091장으로 3,272장은 학습 데이터로 나머지 819장은 테스트 데이터로 이용되었다.

- 이미지 내 쓰레기 종류는 총 11가지, 각각 0 ~ 10으로 라벨링 되어 있다. ( `Background`, `General trash`, `Paper`, `Paper pack`, `Metal`, `Glass`, `Plastic`, `Styrofoam`, `Plastic bag`, `Battery`, `Clothing` ).
- 평가 지표는 mIoU로, 모든 이미지의 클래스별 IoU를 계산 후 이를 평균 내어 최종 점수를 산출한다.

## 3-2. 프로젝트 수행 절차 및 과정

- 1주차 : FCN, LRASPP, DeepLabV3, FPN, UNet, UNet++ 등 다양한 디코더로 실험
- 2주차 : 가장 성능이 좋게 나온 FPN 디코더를 사용하고 하이퍼 파라미터를 변경해가며 실험
- 3주차 : 코드 모듈화, 모델 파라미터 수정 및 Wrap-up report 작성

## 4. 프로젝트 수행 결과

### 4-1. 모델 평가 및 개선 과정

#### 1) 모델 선정

학습 속도와 성능이 좋은 FPN 디코더 및 파라미터 수가 많으면서 V100 상에서 모델을 2개 돌리면서 실험할 수 있을 정도의 파라미터를 가진 timm-efficientnet-b6 인코더 선정

#### 2) Augmentation 실험

Augmentation (Fix 24 epochs)	# mIoU	# mIoU derivation
<code>None</code>	0.571	0
<code>GridDistortion</code>	0.583	0.012
<code>RandomGridShuffle</code>	0.585	0.014
<code>GridDropout</code>	0.587	0.016
<code>ElasticTransform</code>	0.598	0.027
<code>RandomResizeCrop</code>	0.619	0.048

#### 3) Optimizer

Tan et al. 2019에 의하면 Adam이 EfficientNet에 기반한 모델을 잘 학습시키지 못하는 경향이 있어 Adam 대신 General-purpose optimizer인 **MADGRAD** 선택

#### 4) Test-time augmentation(TTA)

horizontal flip, vertical flip 등  $f^{-1} = f$  를 만족하는 augmentation만 실제 적용

### 4-2. 최종 선정 모형

#### 1) Models

- Encoder : **timm-efficientnet-b6**
  - Weight : noisy-student
- Decoder : **FPN**
  - In channel : 3, Classes : 11
- Optimizer : MADGRAD
- Learning rate : 2e-4
- LR\_scheduler : CosineAnnealingWarmRestarts(max\_lr=2e-4)
- Loss : SoftCrossEntropy loss
- Batch size : 16
- Epoch : 30 epochs

- KFold : Random split 5-Fold
- TTA : Horizontal flip, Vertical flip

## 2) Data argument

- Compose
  - RandomRotate90(p = 1.0)
- Oneof
  - A.RandomResizeCrop(height = 512, width = 512, p = 1.0)
  - ElasticTransform(p=1.0)
  - HorizontalFlip(p = 1.0), VerticalFlip(p = 1.0)
  - RandomGridShuffle(p = 1.0), GridDistortion(p = 1.0), GridDropout()

## 4-3. 대회 결과

Val 0.9079 / Public LB 0.781 / Private LB 0.717

- 최종 제출 모델은 Multi-Label Stratified K-Fold를 적용한 것이 아니기에 class 별 균등한 분류가 되지 않았으며, 이를 적용한 다 하더라도 특정 클래스의 개수가 매우 적어 Data imbalance 문제를 해결하지 못했다.
  - Public set의 Lucky Seed로 인해 LB Score가 높았으나, Private LB에서는 현저히 낮은 score를 얻었다.
  - Data imbalance 문제를 1주차부터 해결했어야 했는데 아쉽다.
  - Seed ensemble을 통한 모델 성능의 일반화 작업이 추가해보면 좋을 것으로 보인다.
  - 다양한 디코더를 앙상블 했으면 일반화 성능이 올라갔을 수도 있었을 텐데 아쉽게도 FPN을 제외한 디코더의 성능이 좋지 못하여 디코더 앙상블을 구현하지 못했다.

## 5. 자체 평가 의견

- 김서기 : 초반에 SOTA에서 가장 좋은 모델인 HR-Net을 사용하려고 했지만, 생각보다 성능이 나오지 않아서 아쉬웠다. 왜 성능이 안 나오는 것에 대해서 분석을 해야 했는데 결과로만 확인했던 게 너무 아쉬웠다. 그리고 최종 결과물에 pseudo labeling을 적용하지 못한 게 아쉬웠다.
- 김승훈 : 무지성으로 사용한 MobileNet 에 대해 논문도 읽어봤지만 어떤 custom을 해보면 좋을지 알지 못해 아쉬웠고 체계적인 실험 계획 수립과 실험 후 결과 정리 정돈을 잘 해두는 것이 큰 도움이 된다는 것을 알게 되었다.
- 배민한 : 실험 내역 및 근거를 명확히 작성 후 대회 기간에 맞추어 시간 계획을 수립했다면 조금 더 좋은 효율을 얻을 수 있었을 것 같다.
- 손지아 : 지난 Object detection 대회 때 보다 더 PyTorch에 능숙해졌고 그 덕분에 많은 실험을 했다. 그러나 많은 실험을 한 것과 별개로 리더보드 상에서 좋은 점수를 얻지 못하였는데, 1주차부터 논문을 읽고 논문 재현 실험부터 했었으면 더 좋은 성적을 거두지 않았을까 싶은데 그렇지 못한 점이 아쉽다.
- 이상은 : 3주 중 2주는 온전하게 부스트캠프에 집중하지 못해 개인적으로 아쉬움이 많았던 대회였다.
- 조익수 : 단순히 segmentation 도메인 지식뿐만 아니라 개발에 필요한 여러 지식이 부족했는데, 한정된 시간 내에 중요도를 선별하여 성능 향상에 도움이 되는 작업부터 공부했다면 팀 성과에 보탬이 됐을 것 같다.