Object Detection Challenge Wrap-Up Report

CV-18조 The Visionaries

김준태_T5059

박재민_T5089

송인성_T5116

이지유_T5160

최홍록_T5220

목차

1. Project 개요

문제 정의

대량 생산, 대량 소비의 시대에 많은 제품이 생산되고 소비되면서 쓰레기 배출 현상이 심화됨에 따라 쓰레기 대란과 매립지 부족이 심각한 사회 문제로 제기되고 있다. 우리는 재활용 가능한 자원을 보다 효과적으로 회수하기 위해 분리수거 과정에서 object detection 기술을 활용해 쓰레기를 식별하고 쓰레기의 분리수거 정보를 파악하고자 한다. 이를 통해 object detection task에 대한 이해를 높이고 재활용 쓰레기 처리의 효율성을 높여 지속 가능한 환경을 조성하는데 기여할 수 있다.

팀 구성 및 역할

- 김준태 : EDA, MMDetection train baseline, 2 stage model (Cascade R-CNN) 기반 실험 및 결과 분석, k-fold & pseudo labeling 실험
- 박재민 : EDA, MMDetection train baseline, YOLO baseline code, 1 stage model (YOLO v8) 실험 진행, ensemble, data split code 작성, 실험 자동화 shell script 작성
- 송인성 : EDA, MMDetection config baseline, 1stage model (RetinaNet) & 2 stage model (Cascade R-CNN), data augmentation, data split, ensemble
- 이지유 : MMDetection config baseline, 2 stage model (Faster R-CNN, Cascade R-CNN) 학습 및 성능 개선 작업 수행, model ensemble 실험 진행
- 최홍록 : MMDetection config baseline, data cleaning 파일 생성, 1 stage model (YOLO v8) 실험

팀 목표

- 이유와 근거가 있는 실험을 체계적으로 진행하자.
- 여러가지 모델에 대해 공부하고 사용해보자.
- 새로운 팀원들과 협업하는 방법을 배우자.

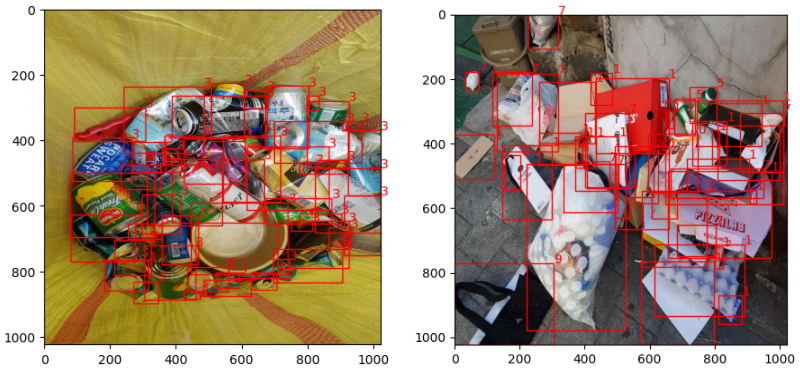
2. EDA & Data cleaning

EDA

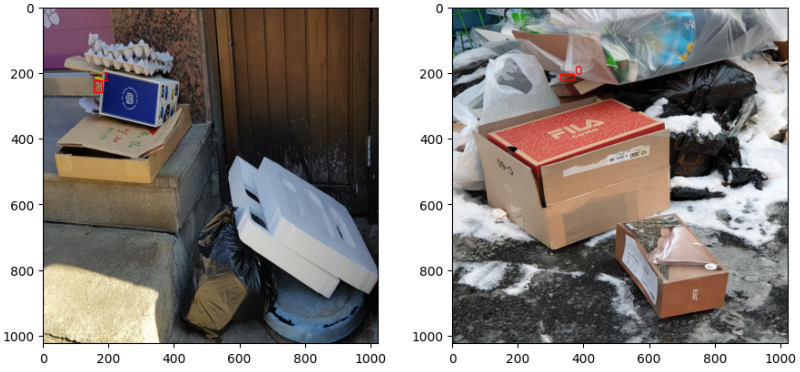
EDA를 통해 dataset에서 다음과 같은 문제점을 발견했다.

1. 극단적인 종횡비를 갖는 bounding box

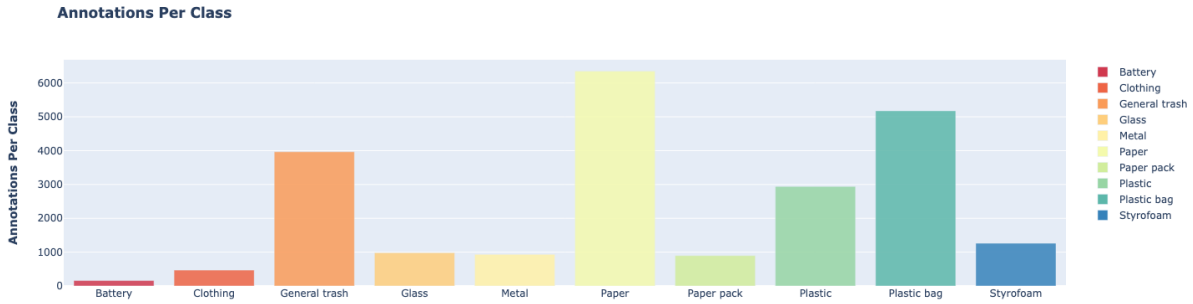
2. bounding box 수가 너무 많고 bounding box 간 겹침이 심한 image



3. 크기가 극단적으로 작은 bounding box



4. class 간 불균형



5. 명확하지 않은 class 선정 기준

Data Cleaning

EDA 결과로부터 발견한 문제점 중 1~4번을 해결하기 위해 다음과 같은 시도를 했다. 각 번호는 EDA의 문제 번호와 대응된다.

1. 극단적인 종횡비의 기준을 10으로 설정하고 이보다 큰 종횡비를 갖는 bounding box는 제거했다.
2. bounding box 개수가 35, 20, 10, 5, 2, 1 이상인 이미지를 차례대로 학습 dataset에서 제거해보았다.
3. 작은 bounding box 면적의 기준을 1048(i.e., 최대 이미지 면적의 1/1000)로 잡고 이보다 작은 면적을 갖는 bounding box는 제거했다.
4. CV strategy를 random split에서 stratified group k-fold로 바꿔 적용해보았다.

3. 실험 과정 및 결과

Object Detection의 대표적인 모델들을 공부해보고자 1 stage, 2 stage 모델로 나누어 실험했다.

1-Stage

Models

mmdetection의 YOLO v3[1]와 RetinaNet[2], ultralytics의 YOLO v8[3]로 실험을 진행했다.

Model	mAP50 (validation)
RetinaNet (ResNet101) RetinaNet (ResNeXt101)	0.4190 0.4200
YOLO v3 (512)	0.2880
YOLO v8 N YOLO v8 M YOLO v8 X	0.3535 0.4575 **0.4982**

비교 실험을 통해 확인한 결과 전반적으로 YOLO v8의 성능이 RetinaNet, YOLO v3 보다 좋았고 그중에서도 X 모델의 성능이 가장 높음을 확인할 수 있었다. 이후 YOLO v8을 fine-tuning하는 방향으로 실험을 진행했다.

Data

data cleaning에서 제안된 data들로 성능 변화를 측정하기 위한 실험을 진행했다. base model은 25 epoch동안 학습한 YOLO v8 N 모델이고 augmentation 기법을 적용하지 않고 학습했다.

Method	mAP50 (validation)
baseline	**0.1279**
종횡비 10 이상의 bounding box 제거	0.1255
bounding box 수가 35 초과인 image 제거 bounding box 수가 20 초과인 image 제거 bounding box 수가 10 초과인 image 제거 bounding box 수가 5 초과인 image 제거	0.1173 0.1194 0.1049 0.0937

[1] Redmon, Joseph, and Ali Farhadi. "YOLOv3: An Incremental Improvement." arXiv arXiv:1804.02767 (2018).
[2] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, Piotr Dollar “Focal Loss for Dense Object Detection” Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980-2988
[3] 'Ultralytics'. Accessed May 21, 2023. Available at: https://github.com/ultralytics/ultralytics.

bounding box 수가 2 초과인 image 제거 bounding box 수가 1 초과인 image 제거	0.0651 0.0469
면적이 1048 이하인 bounding box 제거	0.1209
class가 general trash인 bounding box만 유지	0.2206

실험 결과 전체적으로 base model보다 성능이 하락한 모습을 확인할 수 있었다. 때문에 모델을 학습할 때 data cleaning 기법은 사용하지 않기로 결정했다.

또 general trash 데이터를 잘 예측하지 못하는 점을 해결하기 위해 general trash만 존재하는 data를 만들어 학습을 진행했지만 별다른 성능 향상이 없었다.

Validation

YOLO v8 N 모델을 대상으로 baseline 모델과 validation 단계에서 nms[4](non-maximum suppression)를 적용한 모델과의 차이점을 실험했다. 실험은 25 epoch동안 진행되었으며 512x512 크기의 이미지를 입력으로 사용하고 augmentation 기법이 적용되지 않은 상태에서 하이퍼파라미터는 동일하게 설정했다.

Heuristic	mAP50 (validation)
baseline	0.1279
nms(iou: 0.7)	**0.3535**

nms를 적용한 모델에서 mAP50 점수가 0.2256점 상승하는 것을 확인했다. 이를 통해 validation시 nms의 적용이 성능에 미치는 긍정적인 영향을 확인할 수 있었다.

Augmentation

nms를 적용한 모델에 YOLO v8에서 적용할 수 있는 augmentation을 하나씩 적용해보며 성능변화를 측정했다.
실험은 YOLO v8 N 모델로 25 epoch동안 진행되었으며, 512x512 크기의 이미지를 입력으로 사용하고 하이퍼파라미터는 동일하게 설정했다.

Augmentation	mAP50 (validation)
nms(iou: 0.7)	0.3535
+ mosaic + translate(0.1) + scale(0.5) + hsv(0.015, 0.7, 0.4)	**0.4259** 0.3867 0.4150 0.3398

[4] Jan Hosang, Rodrigo Benenson, Bernt Schiele “Learning non-maximum suppression” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4507-4515.

hsv를 제외한 모든 augmentation 방법론에서 약 0.3~0.7의 mAP50 점수가 상승하는것을 확인했다. 이를 바탕으로 hsv를 제외한 augmentation 기법을 적용하기로 결정했다.

image rotation augmentation을 0°, 45°, 90°로 설정하여 성능변화를 측정했다. 90° 초과는 image의 상하가 바뀌어 학습에 악영향을 줄 것으로 생각해 실험에서 제외했다.

Degree	mAP50 (validation)
0°	0.5432
45° 90°	0.5608 **0.5742**

실험 결과 0°, 45°, 90°로 갈수록 성능이 향상되는 것을 확인할 수 있었다.

Epoch

지금까지 실험이 25 epoch을 기준으로 진행됐으므로 epoch에 따른 성능변화를 확인하기 위해 실험을 진행했다. 실험은 YOLO v8 N 모델로 진행했으며 25 epoch으로 학습한 모델과 augmentation을 적용하지 않은 150 epoch으로 학습한 모델, augmentation을 적용한 150 epoch으로 학습한 모델을 사용했다. 150 epoch으로 학습된 모델은 모두 early stop patient는 10으로 설정했다.

Epoch	mAP50 (validation)
25	0.3535
150 150 + augmentation	0.2982 **0.4168**

실험 결과 augmentation을 적용하지 않고 150 epoch 학습한 모델은 25 epoch으로 학습한 모델보다 mAP50점수가 0.0553 하락했다. 하지만 augmentation을 적용하고 150 epoch 학습한 모델은 25 epoch으로 학습한 모델보다 mAP50점수가 0.0633 상승했다. 이를 통해 augmentation 없이 150 epoch 학습한 모델은 train data에 overfitting되었지만 augmentation을 적용해 overfitting 문제를 해결하고 성능을 높일 수 있었다는 결론을 도출했다.

Resolution

입력 image resolution의 변화에 따른 성능변화를 확인하는 실험을 진행했다. 실험은 YOLO v8의 N 모델과 M모델, X 모델로 512x512, 620x620 (YOLO 기본 설정값), 1024x1024 3가지의 resolution으로 25 epoch 학습했다.

Resolution/Model	mAP50 (validation)		
	N	M	X
512	0.3535	0.4575	0.4982

620 1024	0.3580 **0.3781**	0.4741 **0.4893**	0.5003 **0.5179**

실험 결과 모델에 관계없이 image resolution이 커질수록 성능도 높아짐을 확인할 수 있었다.

최종 Model

위의 실험들로 다음 요소로 학습한 최종 model을 선정했다.

- model: YOLO v8 X
- validation strategy: nms(iou: 0.7)
- image resolution: 1024x1024
- augmentation: mosaic, scale, translate, rotation
- epoch: 150 + early stop patient 10

Model	mAP50 (validation)	mAP50 (test)
YOLO v8 X base	0.4982	-
YOLO v8 X best	**0.5742**	**0.4985**

최종 model은 base model보다 mAP50 점수가 0.076 상승했다.

2-Stage

Models

2-stage model의 모든 실험은 mmdetection library를 사용하여 진행하였다.

2-stage 기반의 model 중 기초가 되는 모델인 Faster R-CNN[5]과 세부 객체 검출에 더 유리한 Cascade R-CNN[6]의 성능을 비교했다.

Model	Method	mAP50 (validation)	mAP75 (validation)
Faster R-CNN	Resnet50 Resnet101 Resnet50 + PAFPN	0.402 0.416 0.406	0.219 0.241 0.226
Cascade R-CNN	Resnet50 Resnet101 Resnet50 + PAFPN	0.409 **0.423** 0.417	0.258 **0.284** 0.275

Faster R-CNN과 Cascade R-CNN model을 동일 조건에서 비교 실험을 진행하여 상대적으로 성능이 좋았던 Cascade R-CNN을 2-stage 실험의 base model로 결정하였다.

Base experiments

Cascade R-CNN을 기반으로 일반적으로 Object detection task에서 성능의 향상을 보여줬던 방법들을 적용하여 실험 해보면서 우리의 문제 상황에서도 성능 향상을 이뤄 낼 수 있는지 비교 실험을 진행하였다.

Method	mAP50 (validation)	mAP50 (test)
base base + pretrained weight	0.409 0.426	0.4016 0.4302
backbone Resnet50 backbone Resnet101 backbone SwinT-small backbone SwinT-base	0.409 0.423 **0.481** **0.481**	0.4016 0.4268 **0.4841** 0.4774
neck FPN neck PAFPN	0.409 0.417	0.4016 0.4071
NMS Soft NMS	0.409 0.412	0.4016 0.4057

[5] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440-1448.
[6] Zhaowei Cai UC San Diego, Nuno Vasconcelos UC San Diego “Cascade R-CNN: Delving into High Quality Object Detection” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 6154-6162

image scale 512 image scale 1024	0.409 0.436	0.4016 0.4487

실험 결과 모든 방법에 대해 성능 향상을 보였고 backbone과 image scale의 변화가 가장 큰 성능 향상을 보였다. 이 실험 결과를 참고하여 모델을 발전시키는 방향성을 잡았다.

Augmentation

데이터 수가 적은 문제와 같은 클래스 안에서 다양한 형태의 물체가 존재하기 때문에 augmentation을 사용하면 model이 robust 해지고 성능이 오를 것이라고 생각하여 mmdetection의 albumentation example에서 제공하는 augmentation들을 augmentation v1로 설정하여 base 모델(SwinT-small backbone)에 적용했다. 예상했던 것과 달리 성능이 떨어졌다. Image를 인식하기 어렵게 만드는 blur가 너무 많이 들어간 것의 영향이 클 것이라는 가설로 모든 blur를 제거해 실험하였고 성능이 상승한 결과를 보였다.
augmentation v1 : ShiftScaleRotate, RandomBrightnessContrast, IAAAffine, CLAHE, Equalize, RandomRotate90, InvertIng, GaussianBlur, MedianBlur, Blur

Method	mAP50 (validation)
base (SwinT-small) base + augmentation v1 base + augmentation v1 - Blur, MedianBlur, GaussianBlur	**0.480** 0.391 0.447

결과적으로 augmentation을 적용하면 적용하지 않았을 때보다 성능이 떨어졌다. validation data에 어떻게 표현되었는지 mmdetection의 browse_dataset 모듈을 통해 확인하였다. 성능 하락이 이상하지 않은 결과를 볼 수 있었다.

- 원본 & augmentation v1

다양한 크기의 물체를 효과적으로 탐지하기 위해 multi scale learning을 적용한 결과 성능이 향상되었다. 그리고 augmentation 방법을 하나씩 이미지에 출력해 보며 image의 영향을 최소화하면서 다양성을 확보할 수 있는 방법들을 선택해 augmentation v2를 구성했다. augmentation v2 적용 결과 성능 향상의 효과를 가져왔다.

augmentation v2 : RandomBrightnessContrast, CLAHE, RandomRotate90

Method	mAP50 (test)
base (SwinT-base) base + MultiScale learning base + MultiScale learning + augmentation v2	0.6155 0.6272 **0.6367**

최종 Model

Base Experiments 및 Augmentation 실험 결과를 바탕으로 성능 향상을 보인 변인들에 대해 아래와 같은 설정값을 적용한, 새로운 base model을 선정했다. 또한, 학습 속도 향상을 위해 mixed precision 기법을 도입했다.

- backbone: SwinT-base[7]
- neck: PAFPN
- box fusion strategy: soft-NMS
- image scale: 1024x1024
- multi-scale learning
- augmentation v2
- pretrained weights: cascade_rcnn_r50_fpn_1x_coco, swin_base_patch4_window7_224_22k

이후, 실험 시작 시점에 비해 모델의 크기와 표현력 다소 증가한 점에 집중해서 학습 dataset 구성의 변화에 따른 base model 성능 향상 추이를 살펴보는 실험을 진행했다.

Method	mAP50 (test)
base base + no validation base + pseudo labeling	0.6414 **0.6536** 0.6309

실험 결과, validation dataset을 생성하지 않고 학습 dataset을 주어진 train data 전부로 구성하는 no validation 전략은 성능 향상(+0.0122)을 보인 반면 base model로 라벨링된 test data를 학습 dataset에 추가하는 pseudo labeling 전략은 성능 하락(-0.0105)을 보였다. 이는 학습 dataset에 새롭게 추가되는 data가 no validation 전략은 기존 train data, 즉 annotation의 질이 우수한 data인 반면 pseudo labeling 전략은 mAP50 기준 0.6414 정도의 성능을 가진 model로 라벨링된 data, 즉 annotation의 질이 상대적으로 우수하지 못한 data이기 때문인 것으로 생각된다.

위와 같은 과정을 통해 성능이 가장 우수했던 base + no validation model을 2 stage의 최종 model로 선정했다.

[7] Ze Liu†* Yutong Lin†* Yue Cao* Han Hu*‡ Yixuan Wei† Zheng Zhang Stephen Lin Baining Guo. “Swin Transformer: Hierarchical Vision Transformer using Shifted Windows”. arXiv:2103.14030v2.

4. Ensemble

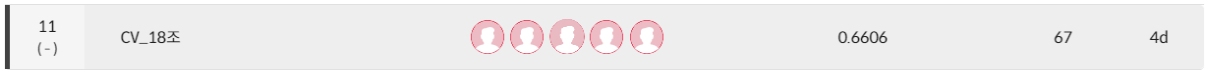
서로 다른 방법론과 모델로 학습한 결과를 ensemble을 하면 각 결과의 단점을 보완하며 성능이 향상될 것이라 생각했다. 모델의 특징, 학습 시 사용된 데이터를 고려하여 ensemble을 진행했다.

Method	mAP50 (test)
2 stage stratified group 4-fold	0.6421
2 stage best + general trash	0.6384
2 stage best + 1 stage best	0.6262
top 6 best submissions	**0.6606**

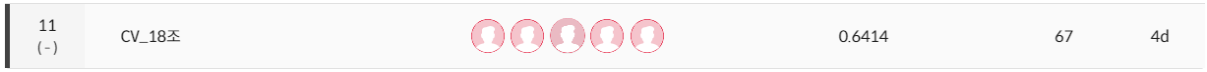
결과적으로 가장 많은 모델을 합친 ensemble 결과가 가장 좋은 결과를 보였고, 결과의 다양성이 높을수록 ensemble 시 긍정적인 영향을 가져다 준다고 생각했다.

5. Conclusion

public



private



잘했던 점

- 원하는 기능을 추가하기 위해 코드를 수정하며 라이브러리에 대한 이해를 높혔다.
- 많은 의견 공유를 통해 규칙을 정하여 실험을 계획하고 진행했다.
- 팀을 나누어 체계적인 파트 분배를 통해 실험을 진행했다.
- 근거 있는 실험을 진행했고, 결과에 대한 원인 분석을 했다.
- 새로운 팀으로 시작을 하면서 많은 회의를 하면서도 실험 시작후 가설 검증에 최선을 다했다.

아쉬웠던 점

- 초반 의견 조율에 시간을 많이 써서 진행하지 못한 실험이 있었다.
- 팀 간 소통이 부족했던 것 같다.
- 더 다양한 모델로 실험을 진행하지 못했다.
- 시간관계상 해보지 못한 실험이 있었고 변인통제를 잘 못한 부분이 있었다.
- 다양한 협업 툴을 사용하고 싶었지만 짧은 대회 기간 동안 전부 사용해 볼 수 없었다.
- 최신 기술들을 사용하지 못했던 것이 아쉬웠다.

대회 세부 사항은 팀 노션 페이지에서 확인할 수 있다.