
Object Detection for Recycling Item Classification

Wrap-up Report Using SOTA Model

Hyeonwoo Jung¹ Jinsuk Kim¹ Minseok Choi¹
Jaehoon Choi¹ Dongyeong Kim¹ Jungwoo Yoon¹

1. 프로젝트 개요

1.1. 개요

오늘날 대량 생산과 소비가 일상화되면서 폐기물 관리 문제는 더욱 심각해지고 있다. 잘못 분리배출된 폐기물은 매립지 부족과 환경 오염을 유발하며, 이에 따라 분리수거는 자원을 재활용하여 환경 부담을 줄이는 중요한 역할을 한다. 하지만 정확한 분리수거는 여전히 많은 어려움을 수반한다. 이 문제를 해결하기 위해 본 프로젝트에서는 사진 속 쓰레기 객체를 정확히 검출하고, 이를 10 가지 카테고리(일반 쓰레기, 플라스틱, 종이, 유리 등)로 분류하는 Object Detection 모델을 개발하고자 한다.

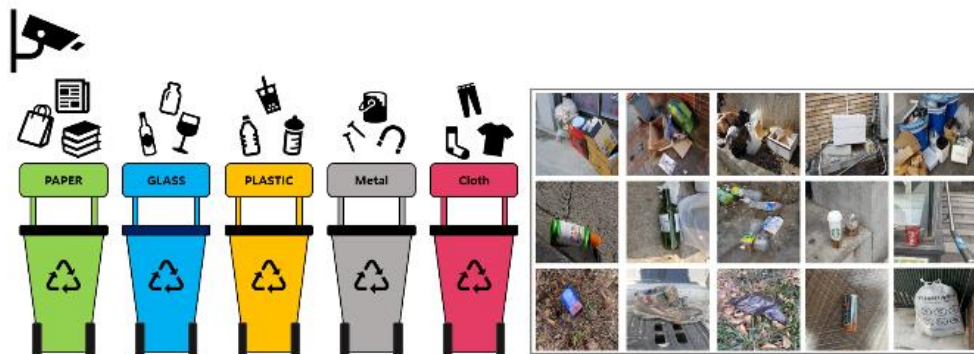


Fig 1. 재활용 분류 그림과 실제 데이터셋에 있는 사진들

이 보고서에서는 다음과 같은 가설을 세워 모델의 성능을 향상하기 위한 전략을 검토할 예정이다.

1. 클래스 간 불균형 문제가 존재하여, K-Fold 를 활용하면 해결될 것이다.
2. mmdetection 의 기본 설정이 우리 데이터셋의 특성을 충분히 반영하지 못 할 것이다.
3. Augmentation 기법을 통해 이미지에 다양한 변형을 적용하면, 모델이 여러 상황에서의 객체 특성을 학습할 수 있어 일반화 성능이 향상될 수 있을 것이다.
4. 제공받은 학습 이미지가 총 4,883 장으로 모델 학습이 충분하지 못 할 것이다.
5. 동일한 모델이라도 파라미터 설정이 달라지면 각 모델이 데이터를 해석하는 방식과 관점이 달라질 것이다.

1.2. 평가 방법

이번 대회 평가 방식은 mAP_{50} (Mean Average Precision)으로 평가한다.

mAP_{50} 은 Object Detection에서 사용하는 대표적인 성능 측정 방법이며 Ground Truth 박스와 Prediction 박스 간 IoU(Intersection Over Union, Detector의 정확도를 평가하는 지표)가 50이 넘는 예측에 대해 True라고 판단한다.

mAP 은 Precision(정밀도)와 Recall(재현율)의 조합을 기반으로 계산된다. mAP 은 여러 임계점에서 Precision을 종합하여 계산하기 때문에 단순히 FP가 늘어난다고 해서 항상 성능이 나빠지는 것은 아니다. 특히, 객체를 더 많이 탐지하는 경우 Recall이 향상될 수 있어 mAP 은 증가할 수 있다.

결론적으로, 이번 대회에서는 mAP_{50} 지표를 최대화하는 것이 중요하며, 이는 정확한 탐지뿐만 아니라 더 많은 객체를 탐지하는 것이 성능에 큰 영향을 미칠 수 있음을 의미한다.

1.3. 개발 환경

- (팀 구성 및 컴퓨팅 환경) 6인 1팀, 총 4개의 서버를 VSCode와 SSH로 연결하여 사용
각 서버 사양:

CPU: Intel(R) Xeon(R) Gold 5120 CPU @ 2.20GHz

GPU: Tesla V100-SXM2-32GB

- (협업 및 의사소통 환경) Notion, Excel, Github, WandB, ZOOM, Slack, Discord

2. 프로젝트 팀 구성 및 역할

전체	모델별 성능 table 작성, 모델 성능 실험, Wrap up report 작성, EDA
정현우	전체적인 실험 설계, Ensemble, Hyperparameter 최적화, TTA
김진석	MMDetection 사용법, Ensemble, Hyperparameter 최적화, TTA
최민석	MMDetection Baseline 작성, Pseudo Labeling 실험, wandb 연결 테스트
최재훈	Code refactoring, Git 정리, 실험 관리, EDA, Data preprocessing, Ultralytics 사용법, Ensemble
김동영	Base model 실험, Cosine Annealing scheduler 실험, Resize 실험, Ensemble
윤정우	Data preprocessing (Multilabel Stratified K Fold), Cosine Annealing scheduler 실험, Ensemble

3. 프로젝트 수행 절차 및 방법

내용	10월																							
	2	3	새로고침 데이						9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
프로젝트 계획 수립 및 베이스 라인 코드 분석																								
EDA, Preprocessing																								
Modeling																								
Hyperparameter 실험																								
Data augmentation (Pseudo-labeling, etc.)																								
TTA 및 앙상블																								

Fig 2. 프로젝트 수행 절차 간트차트

3.1. 프로젝트 파이프라인



Fig 3. 프로젝트 파이프라인

3.2. 협업 방식

- Core-time 에 지속적인 소통으로 무엇을 해야 하는지 토론하고 각자에게 역할 분배
- Notion 을 통해 아이디어 수립 및 가설 정리
- 서버 사용자 현황을 작성해 서버 현황 관리
- Spread sheet 에 사용한 모델, Data Augmentation, Configuration, Ensemble 들을 기록
- WandB 를 통해 모델 훈련 과정을 기록
- Google Drive 를 통해 훈련한 모델을 공유

4. 프로젝트 수행 결과

4.1. EDA & Preprocessing

EDA(탐색적 데이터 분석) 및 전처리 과정에서, 데이터셋의 클래스 불균형 문제를 해결하기 위해 Stratified K-Fold 방식을 사용하여, 클래스 불균형으로 인한 예측 성능 저하를 개선할 수 있을 것으로 기대했다[1].

Annotations Per Class

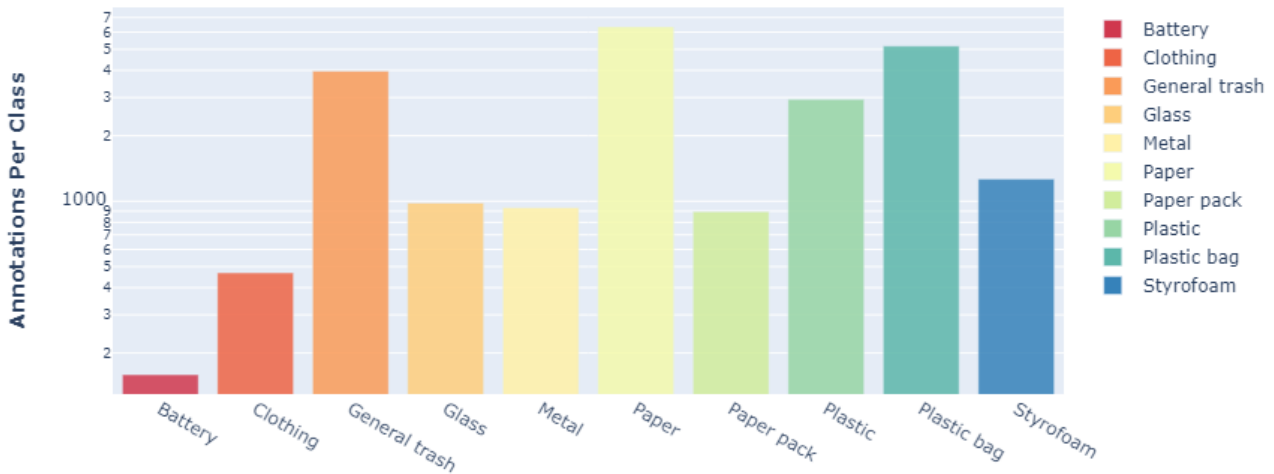


Fig 4. 클래스 당 Annotation 개수 그래프

EDA 결과 Fig 4 를 보면 배터리의 Annotation 개수가 다른 클래스의 Annotation 개수에 비해 매우 적다. 단순히 Random split 해서 train, val 데이터를 나누면 클래스 불균형 문제로 특정 클래스에 편향된 학습이나 과적합 문제가 생길 것이라고 예상된다[2]. 그리고 부정확한 validation set 으로 모델의 성능을 판단하는 데 문제가 생길 수 있다.

Train Split 4 - Class Distribution

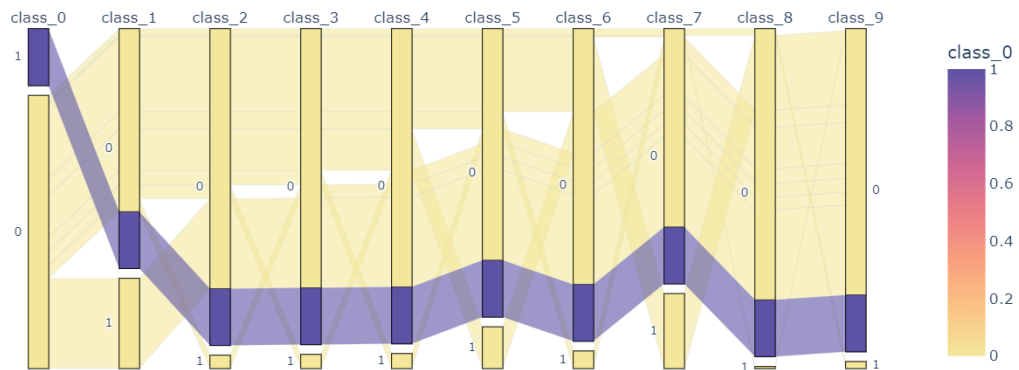


Fig 5. Multi-label Stratified K Fold 결과 클래스별 Train-Val data split 그래프

하나의 이미지에 여러 가지 label 이 존재하니 Multi-label Stratified K Fold 를 사용해서 라벨의 비율을 일정하게 나눠서 Annotation 클래스의 비율을 train 과 validation 이 같게 맞춰주면 배터리의 클래스 불균형 문제의 해결과 좀 더 정확한 validation set 으로 쓸 수 있다.

4.2. Model Selection

Inductive Bias 관점으로 관찰했을 때, CNN 은 local receptive field 를 가지기 때문에 복잡한 객체 간의 상관관계를 학습하는데 한계점이 존재한다. 또한 쓰레기 데이터셋의 특징상 여러 객체가 복잡하게 배치되어 있거나 겹쳐 있는 이미지가 많기 때문에 CNN 은 부적합하다 판단했고, 이를 검증하기 위해 Transformer 기반 모델과 CNN 기반 모델을 선택하여 비교하는 실험을 거쳤다 [3]. 또한 Transformer 기반 모델들을 파악하기 위해 Paper with Codes 를

참조하여 COCO 데이터 셋 기반의 Object Detection 리더보드의 모델들을 조사 후 다양한 환경에서 성능 테스트를 진행한 뒤 비교하는 과정을 거쳤다 [4].

모델명	Library	Backbone	#epochs	mAP ₅₀
Yolov10s	Ultralytics	CSPDarknet +	100	0.3953
Faster_RCNN	MMdetection	R50	12	0.4139
Yolo11x	Ultralytics	CSPDarknet +	100	0.4620
Deformable DETR++	MMdetection	R50	50	0.5028
Diffusion Det	MMdetection	R50	60	0.5050
Co-DETR	MMdetection	R50	12	0.5791
Co-DETR	MMDetection	Swin-L	12	0.6240
Co-DETR	MMdetection	Swin-L	36	0.7139

Table 1. 모델 실험 결과 표

다양한 모델을 테스트한 결과를 토대로 성능 비교표(Table 1)를 작성했으며, Co-DETR (Swin-L) 모델이 우수한 성능을 기록한 것을 확인할 수 있었다[5].

4.3. Hyperparameter

쓰레기 데이터셋의 특성이 COCO 데이터셋과 다르기 때문에, COCO 데이터셋에 최적화된 Co-DETR의 기존 하이퍼파라미터는 적합하지 않다고 판단했다. 이에 따라, 쓰레기 데이터셋의 특성과 양에 맞게 하이퍼파라미터를 조정하는 전략을 수립했다.

4.3.1 milestone

기존 Multi-Step LR Scheduler에서 Learning Rate를 한번 감소시키길 때, mAP가 크게 상승하는 것을 관찰했다. 이를 기반으로 Learning Rate를 mAP가 감소했을 때 Learning Rate를 감소시켜 학습이 Local Minima에 빠졌을 때 효과적으로 벗어나 global minima에 도달할 것으로 판단했다 [6].

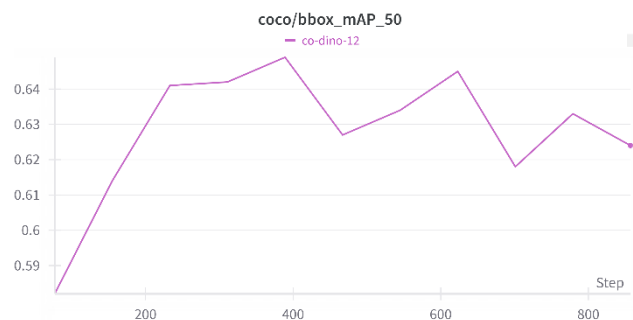
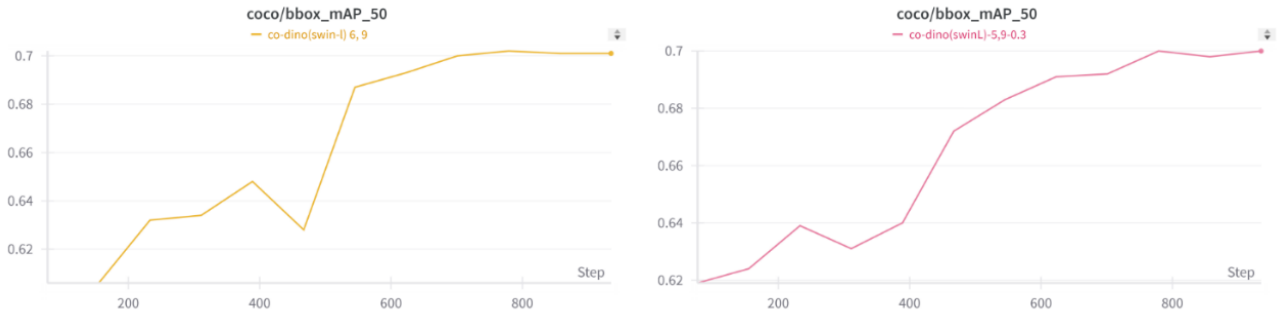


Fig 6. Co-DETR (Swin-L) 12 epoch 실험 val mAP₅₀ 결과

기존 Multi-Step LR 스케줄러에서 학습률을 1번 감소시켰으나, 모델 학습 과정 중 mAP 기록을 분석한 결과, 성능 향상을 위해 학습률을 조정할 필요가 있다고 판단했다. 특히, 학습 초기에 mAP가 많이 감소하는 구간과 성능이 눈에 띄게 개선되지 않는 구간에서 학습률을 추가로 감소시켰다.

Fig 7. Co-DETR (Swin-L) Milestone 변경 실험 val mAP₅₀ 결과

학습률을 두 번 낮췄을 때의 기록을 분석한 결과, 학습률을 낮춘 이후 모델의 학습 성능이 향상되는 것을 확인했다. 그러나 기본적으로 설정된 Gamma 값이 0.1로, 학습률을 두 번 낮춘 후 후반부에 학습률이 지나치게 작아지면서, 두 번째 학습률 감소 이후 제대로 학습되지 않는 그래프 패턴을 보여서 이를 해결하기 위해 Gamma 값을 0.3으로 조정했다. 또한, 초기 학습에서 mAP가 급격히 감소하는 현상이 어느 정도 완화되었으나, 학습률이 떨어지는 구간이 명확하게 분리되지 않아, 초기 학습률을 낮추는 시점을 더 앞당겨 조정했다.

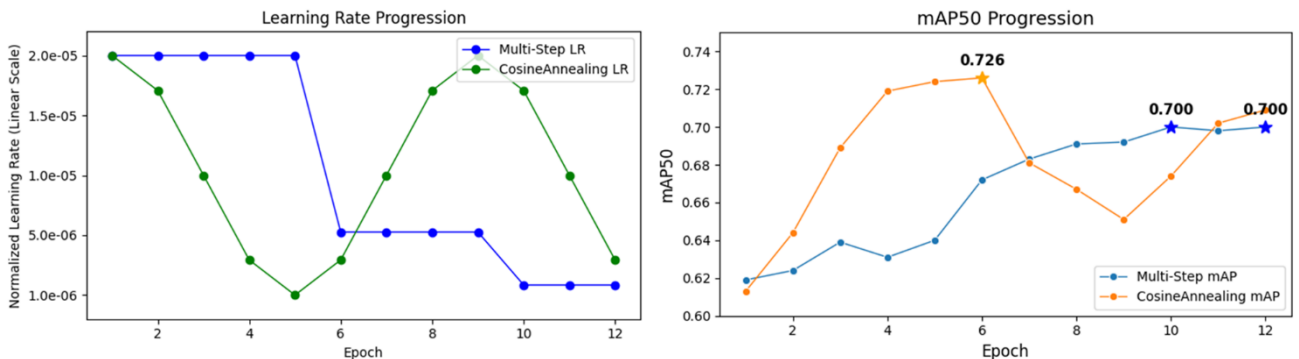
최종적으로 12 Epoch로 모델을 훈련했을 때 milestone을 5와 9 구간에 설정하고 Gamma를 0.3으로 설정하여 쓰레기 데이터에 대해 모델이 Local Mimima에 빠지지 않고 가장 안정적으로 학습하는 결과에 도달했다.

Model	Milestone	Gamma	mAP ₅₀
CO-DETR	11	0.1	0.6240
	6, 9	0.1	0.6998
	5, 9	0.3	0.7028

Table 2. MileStone 변경 실험

4.3.2 cosine Scheduler

Multi-Step LR Scheduler의 근본적인 한계를 극복하기 위해, Cosine Annealing LR 방식을 도입했다. 이는 Cosine 함수 그래프와 warm restart를 결합한 스케줄링 방식이 CIFAR-100 데이터셋에서 기존 스케줄러 대비 최대 0.5-1%의 성능 향상을 끌어낸 실험 결과에 기반했다. [7]에서 해당 실험은 CNN 기반의 WRN 모델에서 이뤄졌으나, transformer 기반에서도 Multi-Step 스케줄러보다 효과적인 학습을 할 수 있을 것이라 가정하고 실험을 진행했다.

Fig 8. 스케줄러에 따른 lr과 mAP₅₀ 비교 실험

실험 환경은 Co-DETR로 모델을 고정한 상태에서 12 epoch, T_{max} (학습률 주기) = 4, eta_{min} (최소 학습률) = 1e-6으로 설정하여 진행했다. 실험 결과, Multi-Step 스케줄러는 학습률 감소에 비례해 mAP₅₀이 점진적으로 상승하여

10 epoch 에서 0.7 의 최고치를 기록했다. 반면, Cosine Annealing 스케줄러는 4 epoch 에서 mAP_{50} 0.719 에 도달하며 Multi-Step 스케줄러의 최대 mAP_{50} 을 초과했고, 6 epoch 에서 mAP_{50} 최고점 0.726 에 도달했다. 이는 Cosine Annealing 스케줄러가 기존 스케줄러 대비 적은 epoch 수로 더 높은 성능을 끌어낼 수 있음을 시사한다.

다만 전반적인 성능 향상이 Cosine Annealing scheduler 만의 결과인지는 확인할 수 없었다. 위 그래프에서 mAP_{50} 은 Multi-Step 과 Cosine Annealing 모두 학습률 하락에 비례하여 상승하는 경향을 보였다. 반면 Cosine Annealing 실험에서 학습률 상승 시, mAP_{50} 은 하락세를 보였다. 이는 주기적인 학습률 상승으로 saddle point 나 학습 정체 구간을 벗어나 성능을 개선하고자 하는 Cosine Annealing 의 의도와는 상반되는 결과이며, 12 epoch 밖에 진행하지 않기 때문에 주기성을 확인하기 힘들다. 그 때문에 위 실험 결과는 Cosine Annealing 의 동적 학습률 상승-하강 메커니즘으로 인한 성능 향상보다는, 학습률 감소로 인한 local minimum 수렴의 결과로 해석하는 것이 타당할 듯하다.

4.4. Data Augmentation

쓰레기 이미지 데이터는 실생활에서 촬영된 이미지가 상당히 큰 비중을 차지한다. 이는 실제 환경에서 발생하는 다양한 상황과 조건을 반영하여 모델의 현실 적용성을 높이는 데 중요한 역할을 한다. 이러한 문제를 해결하고 모델의 전반적인 성능을 향상하기 위해, 다양한 Data augmentation 기법을 적용하는 방법을 고려한다. 이러한 기법들은 학습 데이터의 다양성을 증가시키고, 모델이 다양한 상황에 더 잘 대응할 수 있도록 돕는다

4.4.1 Resize

이미지 해상도를 조절하는 Resize 를 적용해 여러 객체의 디테일한 정보를 추가하고 Edge 정보들을 강화한다. 특히 작은 객체에 대한 추가적인 정보들을 확보할 수 있다. 실험은 Co-DETR 과 Yolo11x 를 사용하여 각각 이미지 크기를 720, 1024, 1380 순으로 진행했고, Table 3 은 그 결과를 나타낸다. 하지만 이미지 해상도를 키우는 것은 모델의 연산속도를 저하하는 단점이 있으므로 적당한 값으로 조정한다.

모델명	이미지 크기	mAP_{50}
Yolo11x	(720, 720)	0.3978
Yolo11x	(1024, 1024)	0.4186
Co-DETR	(1024, 1024)	0.7028
Co-DETR	(1380, 1380)	0.7207

Table 3. 이미지 크기에 따른 비교 실험

이를 통하여 모델의 성능은 이미지 Resize 크기에 따라 Co-DETR 기준 이미지 해상도를 Resize 로 1024 에서 1380 으로 변경하여 0.1 가량 성능을 향상했다.

4.4.2 CopyPaste

Co-DETR 모델에서 두 가지 데이터 증강 기법을 실험했다. 첫 번째는 DETR 논문에서 제시한 방식이고 [5], 두 번째는 Large Scale Jittering(LSJ) 기반의 Copy-Paste 방식이다[8]. 이 두 증강 기법의 성능을 비교하기 위해 실험을 진행했다.

DETR 방식의 증강은 입력 이미지를 여러 크기의 스케일로 조정하여, 각 주석(Annotation) 단위로 잘라내는(Crop) 방법을 사용한다. 반면 LSJ 증강은 입력 이미지에서 Annotation 단위로 객체를 복사한 뒤, 이를 큰 범위의 스케일로 조정하여 대상 이미지에 붙여 넣는 방식이다.

실험 결과, 표에서 확인할 수 있듯이 두 증강 방법 간에 mAP_{50} (mean Average Precision) 기준으로 약 0.05 의 차이가 나타났다.

모델명	특징	mAP ₅₀
Co-DETR	No LSJ	0.6648
	LSJ	0.7028

Table 4. LSJ Augmentation 비교 실험

이 결과를 통해, LSJ 증강 기법을 적용했을 때 모델 성능이 크게 향상됨을 확인할 수 있었다. Table 4는 두 가지 증강 기법을 적용했을 때의 성능 향상이 얼마나 두드러지는지를 잘 보여준다.

4.5. Ensemble

동일한 모델을 사용하더라도, 하이퍼파라미터나 여러 Data Augmentation 등을 달리하여 모델이 데이터에 대해 학습하는 관점과 예측하는 방식이 달라질 것이다. 이러한 차이점을 통해 모델 예측 성능에 변화를 불러오며, 이를 통해 다양한 관점을 통합해 주는 앙상블 기법이 효과적일 수 있다. 아래 Table 5에서 다양한 하이퍼파라미터 설정과 Data Augmentation 등을 사용하여 모델의 다양성을 확보하고 앙상블의 여러 전략을 고려하여 실험을 진행했다.

Ensemble	Method					Threshold	mAP ₅₀
	Epoch 36	Milestone 59	Resize 1380	Cosine	Pseudo		
Soft-NMS	O	O	O			0.6	0.4213
WBF	O	O	O			0.6	0.6671
NMS	O	O	O			0.6	0.735
NMW	O	O	O			0.6	0.735
	O	O	O			0.6	0.7397
	O	O	O	O		0.6	0.7412
	O	O	O	O	O	0.65	0.7422

Table 5. Ensemble 실험

위 실험을 통하여 Non-Maximum Suppress(NMS), Soft-NMS, Non-Maximum Weights(NMW), Weighted Box Fusion(WBF) 등을 사용했다[9][10][11]. Co-DETR 단일 모델로 각각 Baseline(36Epoch), Milestone(5,9), Resize(1380)등을 앙상블 하였다. 이때 NMS와 NMW는 public score 0.735를 달성했고 Soft-NMS와 WBF는 각각 0.4213, 0.6671을 달성하였다.

Ensemble Pipeline

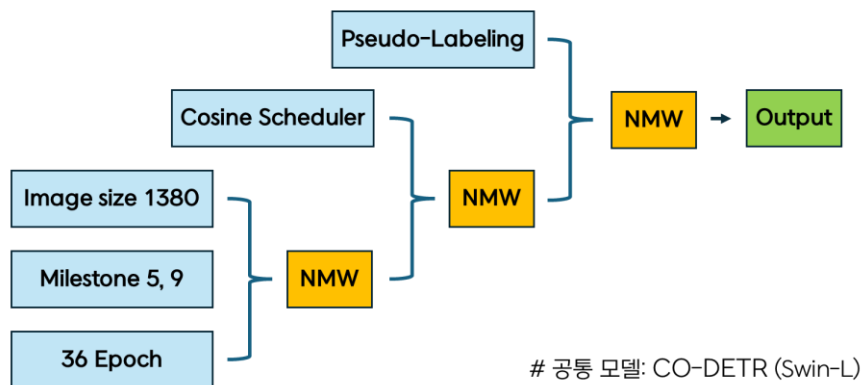


Fig 9. 최종 Ensemble Pipeline

위 실험 결과를 통하여 NMS 와 NMW 는 Threshold 값을 기준으로 불필요 한 박스를 제거하므로 단일 모델에서 중첩되는 Bbox 는 줄이면서 다른 관점으로 예측한 Bbox 를 남기는 것이 가능하다. 또한 가중치를 적용한 NMW 를 통해 정확한 모델에 가중치를 부여해 정확한 결과를 도출할 수 있었다. 반면 WBF 는 Bbox 를 평균화해 통합하기 때문에 예측이 비슷하게 되면 오히려 정확한 박스 위치를 흐리게 만드는 결과를 초래했다. 즉, 단일 모델 예측의 다양성이 부족했기 때문에 WBF의 장점을 충분히 활용하지 못했다. 결론적으로 단일 모델에서는 NMW나 NMS가 효과적임을 알 수 있었다. 이를 통해 추후 여러 실험을 통해 Insight 가 있는 모델들을 NMW 방법으로 Fig 9 와 같이 앙상블해서 최종 public score 0.7422, private score 0.7344 를 도달했다.

5. 자체 평가 의견

A. 잘했던 점

1) 다양한 최신 기술과 도구를 활용하여 문제 해결에 접근한 점

MMDetection 과 Ultralytics 라이브러리를 사용하여 CO-DINO, DiffusionDet, YOLO 등 다양한 최신 모델을 실험해 본 뒤 특성과 성능을 비교 분석할 수 있었고, 문제에 가장 적합한 모델을 선택하는 데 도움이 되었다.

2) 데이터에 대한 이해를 바탕으로 접근한 점

EDA 결과를 토대로 가설을 먼저 수립한 후, 이를 검증하기 위해 Data Augmentation, Hyperparameter 조정, Ensemble 등의 기법을 적용한 것도 좋은 접근 방식이다. 이러한 방법론은 단순히 여러 기법을 무작위로 시도하는 것보다 훨씬 효율적이고 효과적인 결과를 얻을 수 있게 해줬다.

3) 모델 훈련 과정에서 WandB 를 활용하여 실험 결과를 꼼꼼히 기록하고, 예측 결과물을 시각화하여 분석한 점

기록과 시각화를 통해 모델의 성능 변화를 실시간으로 모니터링하고, 문제점을 신속하게 파악할 수 있었다. 이러한 분석을 바탕으로 새로운 가설을 수립하고, 성능 향상을 위한 다양한 방안을 시도해 봤다.

B. 시도 했으나 잘 되지 않았던 것들

1) Data Augmentation

Auto Augmentation, AugMix, Random Augmentation, Color Augmentation 을 적용해 보았지만, 훈련된 모델이 사용했던 Augmentation 을 그대로 사용하는 것보다 성능이 오히려 더 낮게 나왔다. 이러한 결과는 예상과는 달리, 추가적인 데이터 증강이 모델의 일반화 능력을 저하한 것 같다. 특정 증강 기법이 데이터셋의 특성과 잘 맞지 않을 경우, 오히려 성능 저하를 초래할 수 있다는 점을 확인했다.

2) Test Time Augmentation

TTA 기법을 이용해 부족한 데이터셋을 보완하여 성능을 향상하려 했다. Augmentation 을 통해 변환된 이미지를 훈련한 모델이 예측한 후, 예측 결과들을 Ensemble 하는 기법이지만 결과적으로 mAP 가 매우 낮게 나왔다. 최종 결과를 시각화한 결과 bounding box 의 위치가 객체와 다른 위치에 그려지고 있었다. 이미지를 변환한 후 예측한 bounding box 의 위치가 역변환을 적용해 최종적으로 box 의 좌표가 틀어지지 않도록 해야 하는데, 역변환이 제대로 적용되지 않은 것 같다는 분석을 했다. 시간이 촉박했기에 문제를 해결하진 못했다.

3) Pseudo Labeling

Pseudo Labeling 기법을 통해 기존 모델이 Unlabel Dataset 인 Test Dataset 을 추론하여 나온 결과를 Confidence 값을 Threshold 미만인 것들을 Filtering 하여 다시 새로운 모델에 학습 데이터와 같이 학습시켰다[12]. 이를 통해

부족한 학습 데이터를 증강할 수 있을 것이다. 또한, Class 불균형 문제가 크게 드러난 배터리, 의류 등을 보완할 수 있어 모델 성능을 크게 끌어낼 것으로 생각했다. 이를 통하여 기존 앙상블을 통해 mAP 0.7412 가 나온 모델과 앙상블을 적용하여 0.001 정도 향상했다. 하지만 mAP 를 크게 향상할 수 있을 거라는 의도와는 다르게 Train 데이터셋과 모델의 추론한 Test 데이터 셋의 Label Noise 가 더해져서 학습에 방해가 되었다는 점을 알게 되었다.

C. 아쉬웠던 점들

1) Github Issue 사용

이번 프로젝트를 진행하면서 GitHub 이슈 기능을 충분히 활용하지 못한 점이 아쉬웠다. 이슈를 통해 작업 상황을 더 명확히 공유하고, 문제를 체계적으로 관리했다면 팀원들과의 커뮤니케이션이 더 원활했을 것으로 생각한다. 다음 프로젝트에서는 GitHub 이슈를 적극적으로 활용할 계획이다. 초기 기획 단계에서 이슈를 생성해 작업 항목을 구체적으로 나누고, 각 이슈에 우선순위를 부여하여 작업의 흐름을 명확히 할 예정이다. 또한, 이슈 템플릿을 사용하여 일관된 형식으로 문제를 기록하고, 주기적으로 이슈를 점검하며 해결 상태를 업데이트함으로써 프로젝트 관리의 효율성을 높이려고 한다.

2) Docker 를 사용한 서버 설정

이번 대회에서 최신 버전의 Mmdetection 을 사용하면서 여러 문제를 겪었다. 특히 라이브러리 종속성 issue 로 인하여 서버가 자주 다운되는 현상을 겪었으며, 만약 Docker 를 사용해 라이브러리 최적화 후 배포한다면 환경에 종속 받지 않는 서버 세팅이 가능하다.

3) 다양한 모델과의 앙상블 실험 미흡

프로젝트를 진행하면서 모델 선정 과정에서 리더보드의 SOTA 모델들에만 집중하여 실험을 진행했던 것이 아쉬웠다. 특히 Yolo, InternImage H, ATSS(SwinL) 등의 모델들은 초기 성능이 기대에 미치지 못하거나 실험이 실패하자 시간적 제약으로 인해 추가적인 실험을 중단했다. 하지만 돌아보면, 이러한 다양한 특성을 가진 모델들을 좀 더 깊이 있게 실험하고 서로 다른 성격의 모델들 간의 앙상블을 시도했다면 더 좋은 결과를 낼 수 있었을거라 생각한다.

D. 프로젝트를 통해 배운 점 또는 시사점

1) 최신 라이브러리 사용법

본 대회를 통해 MMDetection 의 최신버전 라이브러리를 사용하는 법과 Config 를 수정하여 모델과 세부적인 파라미터(Scheduler, Hyperparameter 등)를 수정하는 법을 알 수 있었다.

2) WandB 를 통한 모델 훈련 기록

모델의 학습 과정을 WandB 를 통해 기록하면 다양한 실험 결과와 로그를 자동으로 저장할 수 있어 실험 결과를 기록하는 데 걸리는 시간을 절약할 수 있었다.

3) 가설과 검증 과정

미리 가설을 세우고 실험을 진행하니 방향성이 명확해지고 체계적으로 진행할 수 있었다. 이를 통해 결과 분석 후 새로운 가설을 세워 실험의 효율성을 높일 수 있었다.

4) 실험 테이블 작성

학습된 모델의 결과를 실험에 사용된 설정과 함께 기록하여, Hyperparameter, Scheduler 등이 성능에 미치는 영향을 쉽게 비교 분석할 수 있었다.

[Reference]

- [1] Sechidis, Konstantinos, Grigorios Tsoumakas, and Ioannis Vlahavas. "On the stratification of multi-label data." *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011, Proceedings, Part III* 22. Springer Berlin Heidelberg, 2011.
- [2] Kemal Oksuz, et al. "Imbalance Problems in Object Detection: A Review", *IEEE transactions on pattern analysis and machine intelligence*, 2020
- [3] Alexey Dosovitskiy, et al. "AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE", *International Conference on Learning Representations*, 2020
- [4] Paper with Codes, <https://paperswithcode.com/sota/object-detection-on-coco>
- [5] Zhuofan Zong, Guanglu Song, and Yu Liu. "DETRs with Collaborative Hybrid Assignments Training", *Proceedings of the IEEE international conference on computer vision workshops*, 2023
- [6] Yanzhao Wu, et al. "Demystifying Learning Rate Policies for High Accuracy Training of Deep Neural Networks", *Proceedings of the IEEE international conference on Big Data*, 2019
- [7] Ilya Loshchilov, et al. "SGDR: Stochastic Gradient Descent with Warm Restarts", *International Conference on Learning Representations*, 2016
- [8] G Ghiasi, et al. "Simple Copy-Paste is a Strong Data Augmentation Method for Instance Segmentation", *IEEE international conference on computer vision workshops*, 2021
- [9] Bodla, Navaneeth, et al. "Soft-NMS--improving object detection with one line of code." *Proceedings of the IEEE international conference on computer vision*. 2017.
- [10] Zhou, Huajun, et al. "Cad: Scale invariant framework for real-time object detection." *Proceedings of the IEEE international conference on computer vision workshops*. 2017.
- [11] Solovyev, Roman, Weimin Wang, and Tatiana Gabruseva. "Weighted boxes fusion: Ensembling boxes from different object detection models." *Image and Vision Computing* 107 (2021): 104117.
- [12] Eric Arazo, et al. "Pseudo-Labeling and Confirmation Bias in Deep Semi-Supervised Learning", *International joint conference on neural networks(IJCNN)*, 2020