

도메인 프로젝트 WRAP-UP 리포트

CV-08 떡볶이

I. 프로젝트 개요

주제	재활용 품목 분류를 위한 Object Detection
기간	2025.12.03 10:00~2025.12.11 19:00
최종 성적	9 등, mAP: 6764
문제 정의	쓰레기들의 사진을 입력으로 받아 사진 속 포함되어 있는 객체들을 10 개 항목으로 분류 및 탐지하는 모델을 개발 하는 것이다.
데이터셋 개요	쓰레기 판별 모델 학습을 위한 쓰레기 이미지 9754 장, 이미지에 포함되는 쓰레기는 10 class 를 가지고 있으며 사진과 객체 정보를 담는 annotation file 은 coco format 으로 이루어져 있다.
기대 효과	환경 부담을 줄이는 효과를 기대할 수 있다. 분리수거를 돕거나 분리 수거 교육에 사용될 수 있다.
팀 구성원	T8058 김해찬, T8067 문재영, T8124 오연서, T8188 정회성

II. 프로젝트 수행 절차 및 내용

절차	수행 내용	기간
1	실험 준비	12/1~12/4
	강의 수강	12/1~12/3
	팀 협업 환경 구축	12/1~12/4
	개발 환경 세팅	12/3~12/4
	EDA	12/3~12/4
	실험 베이스라인 코드 구축 및 탐색	12/3~12/4
2	단일 모델 실험	12/5~12/10
	데이터셋 이상치 제거 및 검증용 데이터셋 구축	12/5~12/7
	모델 가설 설정 및 성능 실험	12/5~12/10
	전 처리 가설 설정 및 성능 실험	12/5~12/10
3	앙상블 실험	12/10~12/11
	앙상블 가설 설정 및 성능 실험	12/10~11

i. 실험 준비 단계

A. 팀 협업 환경 구축

토의를 통해 결정된 팀 협업 환경과 규칙은 다음과 같다.

1. Notion

Notion 에 팀 워크스페이스 환경을 구축하여 실험 목표, 결과, 일정 등을 통합적으로 관리한다.

링크 : [도메인 프로젝트 1 cv08 Notion 워크스페이스](#)

2. Github

팀 Github 을 이용해 팀원간 비슷한 환경을 유지해 실험 결과의 신뢰도를 높이고 실험 가설의 빠른 적용을

링크 : [도메인 프로젝트 1 cv08 Github](#)

3. WanB

팀 WandB 프로젝트 환경을 구축해 팀원이 수행하는 모든 실험의 실시간 추적, 성적의 시각화를 이용하여 다음 전략을 세운다.

링크: [도메인 프로젝트 1 cv08 WandB](#)

B. 개발 환경

대회에서 제공된 baseline 을 참고하여 토의를 통해 결정된 개발 환경은 다음과 같다.

	사용자	GPU
서버 1	T8067 문재영	Tesla V100-SXM2-32GB
서버 2	T8124 오연서	
서버 3	T8058 김해찬,,T8188 정회성	
주요 패키지 requirements	torch 2.1.0, mmcv 2.1.0, mmengine 0.10.7, mmdet 3.3.0, Ultralytics	

config 재현성, 다양한 모델 지원을 이유로 주 사용 패키지를 MMDetectin 으로 결정하였다.

C. EDA

i. 군집 데이터 유무

데이터셋의 Object detection task 수행에 필요한 정보들이 담겨 있는 annotation 파일은 coco format 으로 이루어져 있다. 해당 format 에서 데이터 중 대규모 객체 그룹을 나타내는 iscrowd 속성이 있다. 이러한 이미지 들은 개별 인스턴스를 구별하거나 정밀 처리하기 어려워 종종 전체 그룹에 대해 하나의 큰 경계상자를 사용하는 경우가 있다. 이러한 데이터는 학습 전 단계에서 모델 마다 다른 전처리가 필요할 수 있다.

프로젝트에서 사용되는 데이터의 군집 데이터 유무를 확인하였고 이 데이터셋에서는 **군집 데이터가 존재하지 않는다**는 것을 확인하였다. 이는 학습할 모델의 목표를 개별 인스턴스를 고유한 경계 상자로 탐지하는 것임을 알 수 있다.

ii. Image Resolution

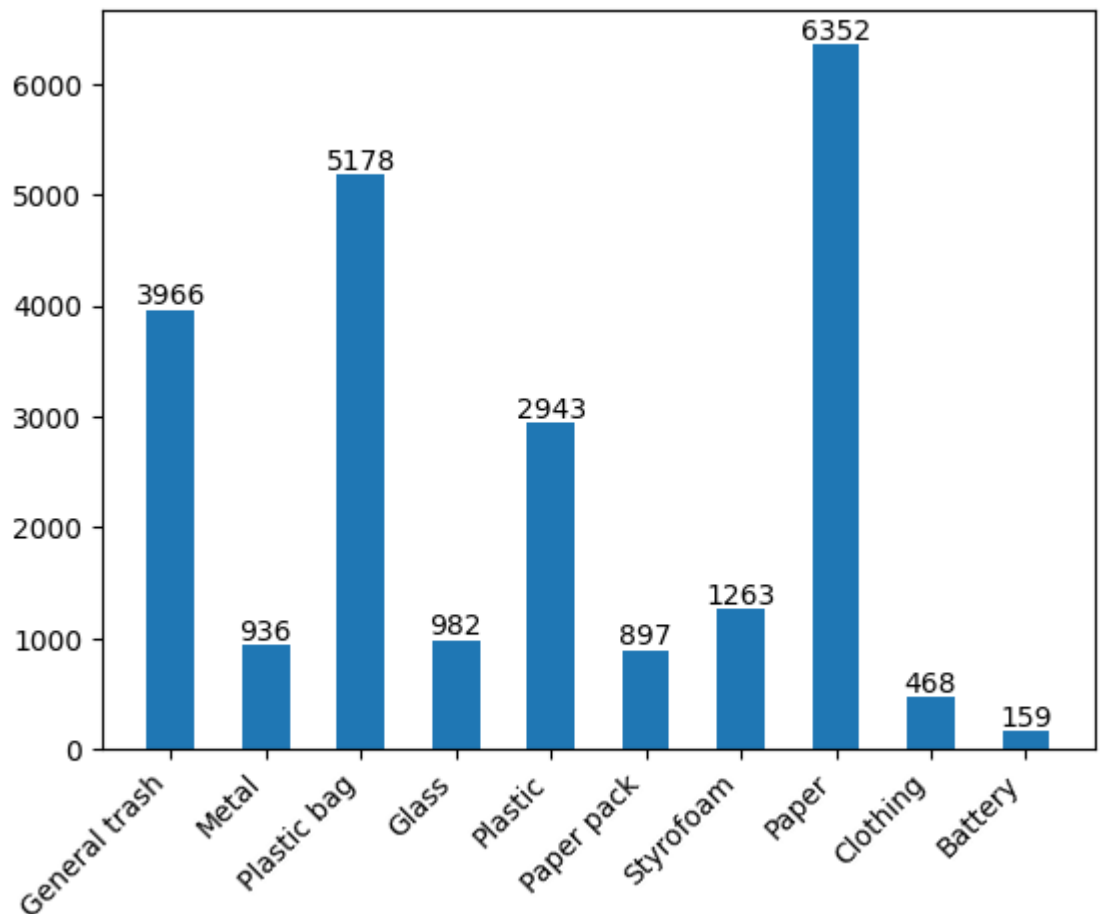
데이터셋의 이미지들의 해상도는 모두 1024x1024 이다. 이는 모델이 항상 동일한 비율과 크기로 스케일링된 데이터를 학습하기 때문에, 이미지 크기나 종횡비 변화로 인해 발생할 수 있는 성능 불균일을 줄이고 더 안정적인 특징 추출을 기대할 수 있다.

iii. Class Counts and Distribution

10 개 항목당 데이터셋에 얼마나 존재하는지 분석한 결과는 다음과 같다.

label	Name
0	General trash
1	Paper
2	Paper pack
3	Metal
4	Glass
5	Plastic
6	Styrofoam
7	Plastic bag
8	Battery
9	Clothing

항목	0	1	2	3	4	5	6	7	8	9	계
수	3966	6352	897	936	982	2943	1263	5178	159	468	23,144



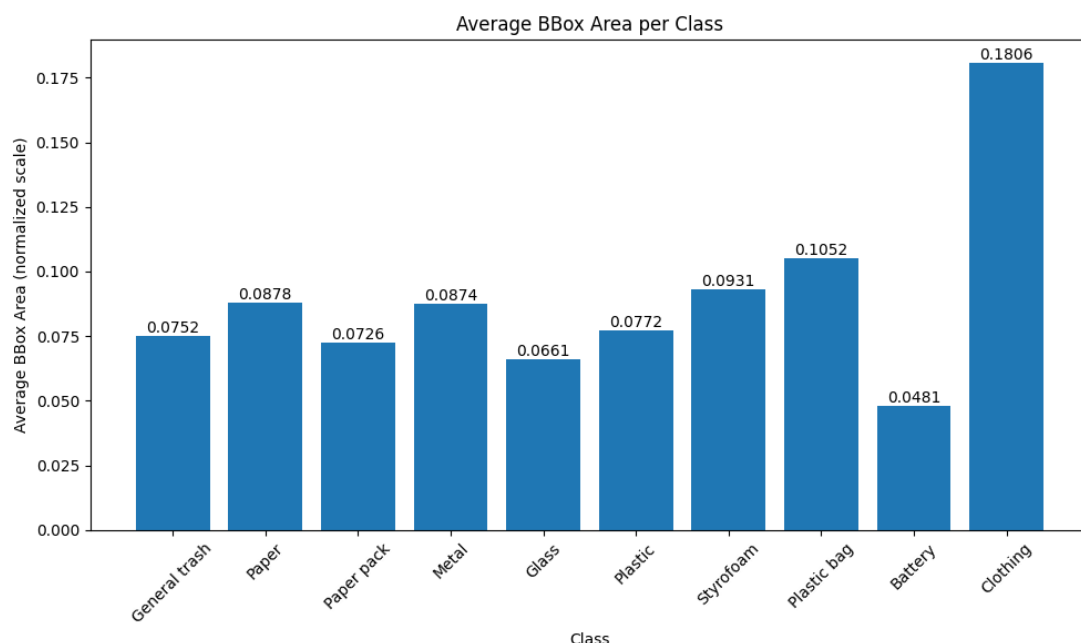
이미지 내 객체들의 숫자를 세고 클래스 마다 개수를 정리해보았을 때 극단적인 경우 한 클래스의 객체들의 수가 다른 클래스의 객체의 수보다 40 배 가까이 차이남을 확인하였다. 이는 데이터셋에 **Class Imbalance** 가 존재함을 의미한다.

이러한 특징은 모델 훈련에 있어 성능 편향, 소수 클래스 탐지 성능 저하 등의 문제가 있을 수 있기에 Oversampling, Augmentation, Focal loss 등 균형을 맞추는 전략이 필요하다.

iv. Bbox Size Distribution

1. Average BBox Area ratio per class

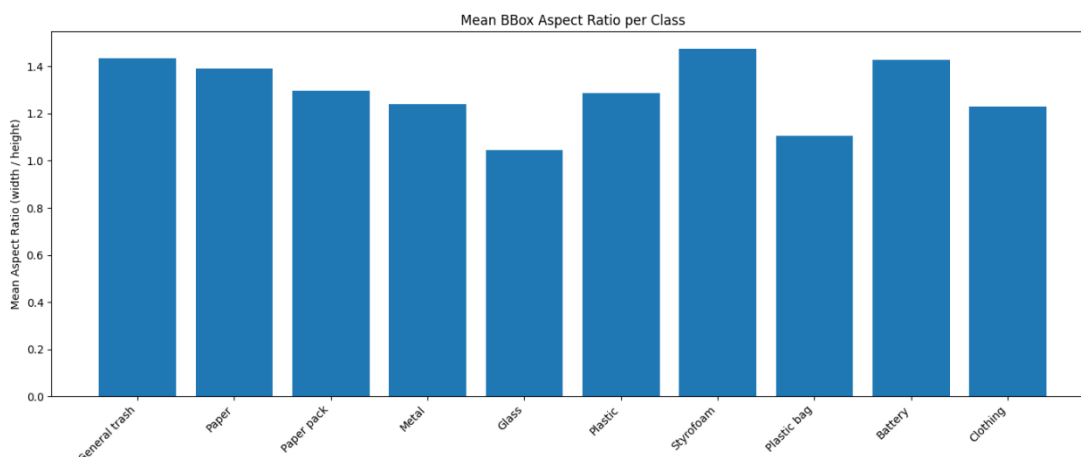
10 개 클래스 항목이 이미지에 평균적으로 얼마나 많은 부분을 차지하고 있는지 분석하였고 결과는 다음과 같다.



이미지의 18%를 차지하는 클래스와 5%를 차지하는 클래스는 크기에서 상당한 차이가 있으며, 이는 모델의 작은 객체 탐지 능력에 직접적인 영향을 줄 수 있다. 이를 해결할 수 있는 다양한 해상도를 이용, 작은 객체 이미지의 증강, FPN 활용, TTA의 멀티 스케일 추론 등의 전략이 필요하다.

2. Mean BBox Aspect ratio per class

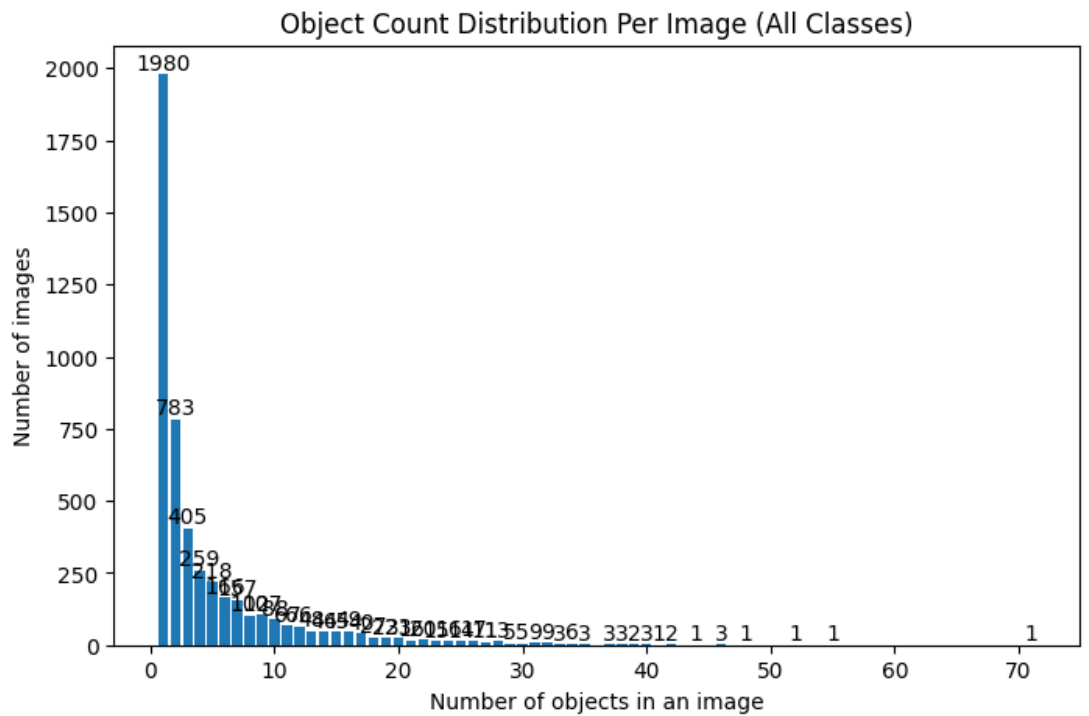
각 클래스 별로 평균적인 bbox의 가로세로 비율을 분석한 결과이다.



v. Object Density

1. Object count Distribution per image

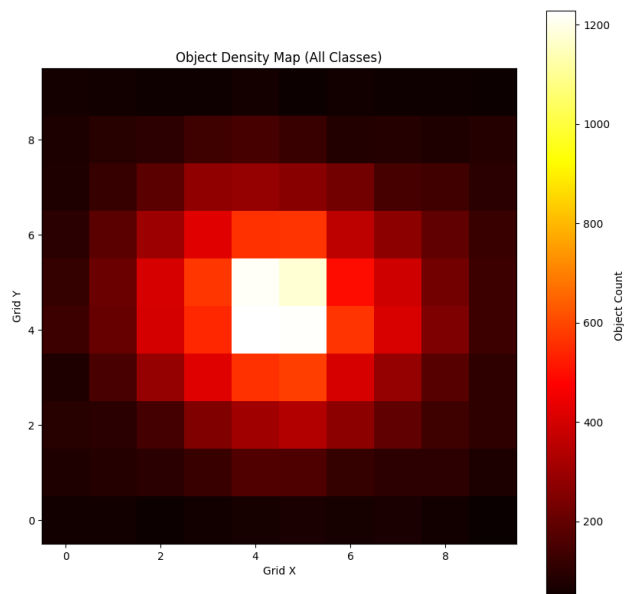
한 이미지당 포함하고 있는 object 의 수를 분석한 결과이다.



한 이미지 내에 존재하는 객체 수의 편차가 크고, 매우 밀집된 이미지가 소수 존재한다. 이미지 중 5%가 객체 18 개 이상을 포함하고 있다. 이는 객체 탐지 모델 훈련에 쉬운 샘플 오버피팅의 문제의 가능성이 있다. 그러나 프로젝트 성능 지표 mAP 인 경우 과밀집한 이미지에서의 성능 저하를 명확히 보여주지 못할 수도 있으니 주의하여야 한다.

2. Object Density Map

이미지 내 object density map 을 출력한 결과는 다음과 같다.



이미지 내 객체들이 대부분 중앙 부분에 위치해 있다. 이는 소수 클래스 객체 증강 전략에서 copy paste 전략을 택할 시 다른 객체 가림 혹은 밀집을 피하기 위해 중앙에서 벗어난 위치에 수행할 필요를 보여준다.

D. 실험 베이스라인 구축 및 탐색

다음과 같이 실험 베이스라인 구축함으로써 실험 관리가 용이하도록 하였다.

프로젝트 구조

```
recycling_project/
├── configs/                                # Config 파일들
│   ├── _base_/                            # 공통 설정
│   │   ├── dataset.py                    # 데이터셋 설정
│   │   ├── runtime.py                   # Runtime 설정
│   │   └── schedule_*.py                # * epoch 학습 스케줄
│   └──
│       ├── models/                      # 모델별 Config
│       │   ├── dino_swinl_1024_6e.py
│       │   └── cascade_rcnn_swinl_1024_6e.py
│       └──
├── tools/                                # 실행 스크립트들
│   ├── train.py                          # 학습 스크립트
│   ├── prepare_dataset.py               # 데이터셋 split 생성
│   └── run_experiments.py               # 여러 실험 실행
├──
└── work_dirs/                            # 학습 결과 (자동 생성)
    ├── dino_swinl_1024_6e/
    └── cascade_rcnn_swinl_1024_6e /

Experiments/                             # Yolo 실험환경
├── run_grid.py                           # 여러 실험 실행
└── train_single.py                       # 학습 스크립트
```

ii. 단일 모델 실험 준비 및 가설 설정

A. 데이터셋 이상치 제거 및 검증용 데이터셋 구축

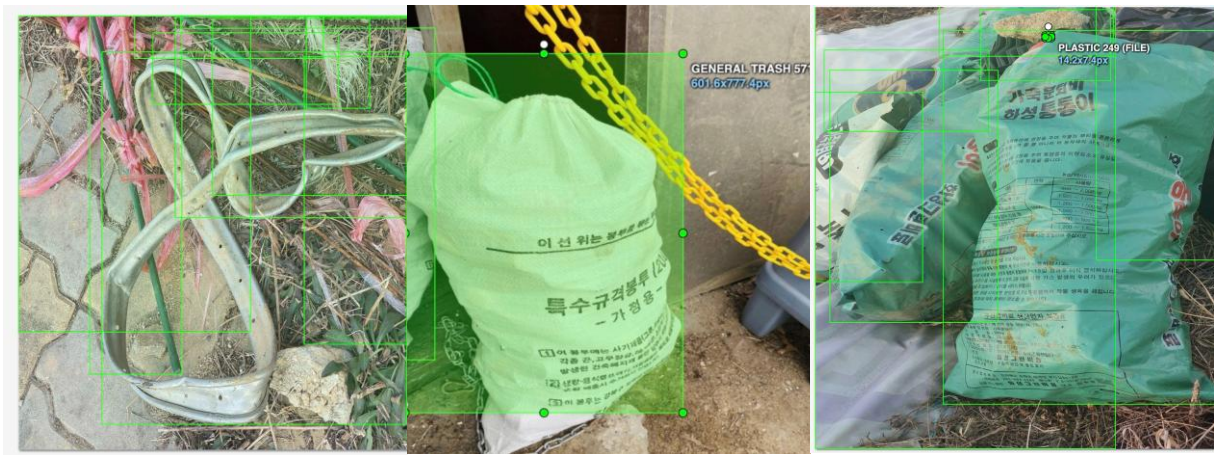
i. 데이터셋 이상치 탐색 및 재라벨링

데이터셋 이미지 탐색 중 적지 않은 이미지에서 bbox가 올바르게 못한 위치에 생성되었음을 확인하였다. Bbox 제거 혹은 재라벨링의 필요를 느꼈고 제거와 재라벨링 대상이 될 이미지를 다음과 같은 기준으로 선별하였다.

선별기준

1. **bbox_count** : 한 이미지 내 bbox의 개수가 과도하게 많은 경우
 - 객체 포함 수 상위 1~5% 구간
 - 여러 객체가 겹쳐 보이는 경우
 - 작은 bbox로 잘게 분리된 경우
2. **bbox_area** : 일부 bbox가 객체보다 과하게 크거나 작은 경우
 - bbox 크기 하위 5% 구간
 - 잘못된 부분을 crop하거나 객체의 일부만 포함된 경우
3. **aspect_ratio** : bbox의 ratio가 비정상적으로 길쭉한 경우
 - $\log(h/w) \notin [-1.5, 1.5]$ 구간
 - 객체의 일부만 잡힌 경우
 - 객체 외의 배경 영역을 포함한 경우
 - 두 개 이상의 객체가 합쳐진 경우
4. **wrong_class** : 클래스 오류

위 기준에 선별된 이미지 예시는 다음과 같다.



객체의 경계가 불명확하거나
애매한 경우

객체의 크기를 온전히 포함하지 특정 크기 이하의 잘못된 라벨링
못한 경우

선별된 이미지에서 삭제 및 수정을 수행했으면 수행 기준은 다음과 같다.

수행	기준
삭제	(1) bbox_count 가 비정상적으로 많은 경우 + 배경이 복잡한 경우
	(2) area 가 너무 작거나 라벨 자체가 무의미한 경우
	(3) aspect_ratio 가 심각하게 깨짐 + 객체가 잘려 있음
재라벨링	(1) bbox_count 는 많지만 대부분 정상적인 쓰레기가 분포하는 경우
	(2) aspect_ratio 이상치인데 실제로는 객체 두 개가 붙어 있음
	(3) area-small 이지만 실제로는 작은 object 가 맞는 경우 (예: 담배꽂초)
유지	위 기준을 충족하지 못한 경우

위 기준에 따라 bbox 257 개 제거, 15 개의 수정이 수행되었다.

ii. 검증용 데이터셋 구축

제출 기회가 제한되어 있으므로 public score 경향을 잘 따라가는 검증 데이터셋이 필요하다고 판단하여 원본 데이터의 분포와 최대한 유사하게 10% 샘플링하여 검증 데이터셋을 구성하였다. 비교 기준이 된 분포는 클래스별 분포, 클래스별 object 크기, 이미지 별 object 수, object density map 이다.

검증데이터셋이 유용한지 판단하기 위해 3 가지 다른 조건으로 실험을 수행하고 검증데이터셋의 metric 경향과 public score 의 경향이 잘 맞는지 확인하였다.



	Public score
실험 1	0.3539
실험 2	0.2129
실험 3	0.3549

검증데이터에 대한 mAP50 양상과 public score 양상 모두 두 실험이 비슷한 점수를, 나머지 하나가 낮은 점수를 기록하는 것으로 보아 구성한 검증 데이터셋이 잘 구성되었다.

B. 모델 가설 설정

i. 모델 선정 기준

1. MMDetection configs 기반 mAP 벤치마크 참고
 - 해당 프로젝트 평가 지표 특성상 연산량보다는 mAP 성능 위주로 관찰
2. 아키텍처 다양성 확보 (1-stage, 2-stage, transformer)
 - 양상불 고려, 데이터 특성상 작은 객체나 불규칙 패턴에 대한 강점이 다르기에 모두 비교 진행

3. YOLO 는 최신 + 대형 모델 우선

- YOLO11/YOLO12 는 최근 모델이므로 baseline performance 확인 목적

C. 전처리 가설 설정

i. Mosaic, Copy-Paste 등 강력한 전처리의 필요

EDA 결과와 절대적인 학습 이미지 양이 부족하다는 의견을 종합해 Mosaic 과 copy-paste 를 활용한 strong augmentation 을 적용했을 때 다음과 같은 이점을 얻을 것이라 가설을 세웠다.

- A. 모델의 일반화 성능 향상
- B. 데이터 희소성 문제 완화
- C. 작은 객체 탐지 성능 향상

ii. Stage 별 transform 조절의 필요

강력한 전처리를 적용할 때 모델 구조나 사전 학습된 데이터 분포에 따라 오히려 학습 성능 하락의 결과를 볼 수 있다는 문제가 있다. 그래서 모델 학습 초반에는 강한 증강을 적용하다 후반에는 최대한 원본과 비슷한 입력데이터를 받아 학습한다면 강력한 전처리로서 얻을 수 있는 이점뿐 아니라 원본 데이터 그대로 학습하여 성능 하락도 막을 수 있다는 가설을 세웠다.

iii. 앙상블 모델 실험 준비 및 가설 설정

A. 앙상블 가설 설정

단일 모델 학습 후 좋은 성능을 가진 모델들끼리 앙상블 모델을 만든다면 전체적인 일반화 성능을 향상시켜 좋은 public score 을 얻을 수 있다 판단했다. 선택한 앙상블 기법과 기대 효과는 다음과 같다.

1. WBF

WBF(Weighted Boxes Fusion) 앙상블 기법은 여러 모델이 예측한 BBox 들을 가중치 기반으로 병합하여 최종 결과를 도출하는 방식이다.

이 기법을 통해 개별 모델들의 오탐지(FP)는 줄이고, 탐지된 객체의 위치 정확도를 높여 전반적인 객체 탐지 성능(mAP)을 향상을 기대한다.

2. Soft-NMS

Soft-NMS(Soft Non-Maximum Suppression)는 객체 탐지 결과 중 BBox 가 겹치는 정도에 따라 점수를 점진적으로 낮추는 방식이다.

이를 통해 객체가 밀집되어 있거나 서로 겹쳐 있는 상황에서 불필요하게 탐지 결과가 제거되는 것을 방지하여 탐지 성능의 향상을 기대한다.

III. 프로젝트 수행 결과

A. 1 차실험 (스모크 테스트)

목적

- 빠른 시간 내 모델 구도 스캔하기 위해 수렴 여부 확인하는 단계 : 6 epoch, 50% 데이터로 수렴 여부 테스트
- 빠르게 모델 컷오프 (수렴이 안 되는 모델은 파라미터 조정을 해도 대부분 결과가 낮을 가능성이 커서 제외)

실험 대상

- Faster R-CNN: baseline 수준 확인

- Cascade R-CNN: small object 에 강함
- VFNet: vari-focal loss 가 imbalance 상황에서 장점
- DINO: transformer 중 안정적 구조
- RTMDet-X: 1-stage 중 강력한 모델
- DDQ-DETR: DETR 계열 중 성능 최상위
- Deformable DETR: multi-scale 특징 활용 강점

결과 요약

Model	backbone	dataset	val score	public score
FasterRCNN	Resnet	raw		0.4177
Cascade R-CNN	ConvNeXt	50%	0.333	0.3297
Cascade R-CNN	Swin-L	50%	0.351	
DINO	Swin-L	50%	0.288	0.2304
DDQ-DETR	Swin-L	50%	0.428	0.3656
RTMDet-X		50%	0.011	
Deformable DETR	Swin-L	50%	0.152	

- 수렴한 모델들(Cascade R-CNN, DINO, DDQ-DETR) 6 epoch 로는 판단 불충분 → 데이터/epoch 확장 필요
- 수렴 실패한 모델들
 - RTMDet-X : 강한 mosaic 기반 학습을 전제로 하는데 이번 스모크 테스트에서는 hard aug 를 사용하지 않아 학습 진행이 거의 안 된 것으로 예상 (일단 제외하고 추후 필요 시 추가 실험)
 - Deformable DETR : warmup 단계가 필요해서 스모크 테스트에서는 특성이 제대로 발현되지 않은 것으로 예상

B. 2 차실험 (epoch 수, dataset 확장)

환경

- 데이터 100%, epoch 12 → 실제 모델 성능 확인
- YOLOv11/12 는 Ultralytics 로 별도 50% 테스트

결과요약

Model	backbone	dataset	val score	public score
Cascade R-CNN	Swin-L (swin-large, patch4, window7)	100%		0.4642
VFNet	ResNeXt-101-64x4d	100%		0.3751
YOLO11x		50%	0.4266	0.3396
YOLO12x		50%	0.361	
DDQ-DETR	Swin-L	100%	0.596	0.5986
Cascade R-CNN	ConvNeXt	100%	0.549	0.532

DINO	Swin-L	100%	0.465	
TOOD		100%		0.2686
Sparse R-CNN		100%		0.2873

- **DDQ-DETR (Swin-L)**
 1. val/public 모두 상위권
 2. augmentation 이 적어도 성능이 잘 나오는 편
- **Cascade R-CNN (ConvNeXt/Swin-L)**
 1. small/mid object 에 강해 baseline 이 좋음
 2. 다만 transformer 대비 상위 한계가 명확해 보임
 3. 2-stage 라 augmentation 민감도가 비교적 낮고 안정적
- **YOLO11**
 1. val 은 높았지만 public 은 낮음 → ****overfitting 의심**** → 추가 실험
 2. YOLO12x 는 명확히 성능 불안정 → 라이브러리에서도 현재는 불안정한 모델이라고 명시
- **TOOD / Sparse R-CNN**
 1. 실험 당시 상당히 낮은 성능
 2. 현재 데이터 특성(작고 변칙적 객체)과의 호환성이 낮다고 판단

C. 3 차실험

모델별 튜닝, 전처리 전략

- Cascade R-CNN: 12 epoch, 데이터셋 100% 풀 학습,
- YOLOv11x: 120 epoch, Ultralytics 기본 스케줄 사용, 데이터셋 100% 풀 학습
- DDQ-DETR:
 - i. Epoch: 18 (시간 제약으로 24 대신 18 로 결정)
 - ii. LR 스케줄: LR 감소를 epoch 12, 16 에 적용
 - iii. Soft (Transformer 계열은 학습 안정성 유지를 위해 해당 증강만 활용)
 - RandomFlip: 50% (상하, 좌우)
 - RandomResize: $\pm 10\%$ (scale)

실험결과

Model	backbone	dataset	public score
YOLO11x		100% (clean)	0.4681
DDQ-DETR	Swin-L	100%	0.6185
Cascade R-CNN	Swin-L	100% (clean)	0.5320

D. 4 차실험

실험 전략

- **Cascade R-CNN + Hard augmentation**
 - Mosaic 섞은 hard augmentation 적용
- **YOLOv11x + 2 stage augmentation**
 - epoch 80
 - 2-stage augmentation 도입
 - Stage1 (1~60): mosaic, mixup, copy-paste 포함

- Stage2 (60~80): soft aug
- **CO-DETR + Hard Augmentation**
 - 선택 이유: projects 모델 중 상위 성능
 - 적용 aug
 - Mosaic
 - RandAugment(Color/Contrast/Sharpness 등)

실험 결과

실험	Public score	비고
CO-DETR+ Hard Augmentation	0.6910	단일 모델 최고 성능
Cascade R-CNN + Hard Augmentation	0.5058	
YOLO11x + +2 stage Augmentation		검증 score 가 낮아 제출 X

E. 앙상블 실험

앙상블 대상 모델은 단일 모델 실험 중 높은 성능을 기록했던 모델로 한다.

대상 모델들

1. CO-DETR + Hard Augmentation
2. DDQ-DETR
3. Cascade R-CNN + Hard augmentation

실험 결과

앙상블 방식	Public score
WBF	0.3921
NMS	0.6138

결과 해석

둘 다 단일 모델 성적에 비해 낮은 성적을 기록했지만 WBF 를 사용했을 때 보다 NMS 만 사용했을 때 성적이 좋았다. 이는 단일 모델의 "가장 자신 있는 박스"를 살려주는 방향이 유리했다고 볼 수 있고 또 성적이 제일 뛰어난 모델에 비해 다른 모델들의 성능이 많이 떨어진다고 해석할 수 있다.

F. 최종 제출

- i. 단일 모델: CO-DETR + Hard Augmentation
- ii. NMS 기반 앙상블 모델
- iii. 최종 성적

	최종 public score
CO-DETR + Hard Augmentation	0.6764
NMS 기반 앙상블 모델	0.6031

IV. 자체 평가 및 회고

본 프로젝트를 통해 객체 탐지 전반에 대한 데이터 구축-모델 실험-앙상블-추론 파이프라인까지의 전체 흐름을 실습 중심으로 경험할 수 있었다. 특히 단순히 모델 성능을 높이는 데 그치지 않고,

데이터 품질 관리와 실험 운영 방식이 결과에 미치는 영향을 팀 차원에서 인식하게 되었다는 점에서 의미 있는 프로젝트였다.

A. 성과 및 긍정적인 점

우선, **CVAT**를 활용한 **라벨링 및 재라벨링 경험**을 통해 실제 데이터 구축 과정의 중요성을 체감하였다. 단순히 주어진 데이터를 사용하는 것이 아니라, 클래스 오류, bounding box 품질 저하 등 오류 유형을 정의하고 이를 수정하는 과정에서 데이터 품질이 모델 성능에 직결된다는 점을 명확히 인식할 수 있었다.

또한, 데이터 전처리의 stage 별로 다르게 적용하는 증강전략을 직접 구현하면서 모델 실험 단계에서 다양한 전략이 쓰일 수 있음을 인지했다. 이는 향후 프로젝트에서도 데이터 전처리 및 품질 관리의 우선순위를 설정하는 데 중요한 기준이 될 것으로 판단된다.

또한 다양한 객체 탐지 모델(Cascade R-CNN, VFNet, Deformable DETR, TOOD, Sparse R-CNN, RTMDet, CODetr 등)을 실제 데이터에 적용해 보며 **모델 구조별 특성과 장단점을 경험적으로 이해**할 수 있었다. 특히 inference 과정에서 발생한 오류(DETA 관련 이슈)를 단순 결과 문제로 넘기지 않고, label 분포, meta-info, score 분포 등을 단계적으로 점검하며 원인을 분석함으로써 **mmdetection 기반 추론 파이프라인에 대한 이해도를 크게 향상**시킬 수 있었다.

아울러 submission 단위에서 동작하는 **앙상블 유틸리티(NMS/WBF)**를 직접 구현하여, 프레임워크에 의존하지 않고 다양한 결과를 결합할 수 있는 재사용 가능한 파이프라인을 구축한 점도 팀의 기술적 성과 중 하나였다.

B. 한계 및 아쉬운 점

반면, 프로젝트 전반에서 **사전 기획과 실험 전략 설계의 부족**이 주요 한계로 드러났다. 모델 포트폴리오가 명확히 정의되지 않은 상태에서 실험이 확장되다 보니, 시간 대비 효율이 낮은 시도들이 다수 발생하였고 실험 우선순위 관리가 어려웠다. anchor 기반, transformer 기반, one-stage 계열 모델을 역할별로 나누어 계획적으로 배치했다면 더 안정적인 성능 조합을 구성할 수 있었을 것으로 판단된다.

또한 validation 기반의 **정량적·정성적 에러 분석이 충분히 이루어지지 못한 점**도 아쉬움으로 남는다. Public score 를 잘 추종하는 validation dataset 을 만들어 냈지만 실험과정에서 완벽히 사용되지 못했다. 결국 public leaderboard 중심의 판단으로 인해 per-class mAP, 객체 크기별 성능, 어려운 이미지 유형에 대한 분석까지는 도달하지 못했으며, 이로 인해 어떤 클래스에서 앙상블이 효과적인지에 대한 명확한 결론을 도출하지 못했다.

앙상블 및 후처리 전략 역시 사전에 충분히 설계되지 못해, 모델 간 예측 분포 차이로 인해 WBF 등 기법의 효과가 제한적으로 나타났다. 강력한 단일 모델(CODetr)을 기반으로 보다 공격적인 조합을 탐색할 여지가 있었으나, 시간 제약과 리스크 관리로 인해 탐색 범위가 제한되었다.

C. 향후 개선 방향

이번 프로젝트를 통해 향후에는 실험 시작 전 **모델 포트폴리오 및 앙상블 전략을 사전에 설계**하고 모델별 실험 템플릿 및 로그 기준을 표준화하여 **프로젝트 운영 효율을 높이며** validation

기반의 체계적인 에러 분석을 통해 **성능 개선의 근거를 명확히 하는** 방식으로 접근할 필요성을 확인하였다.

종합적으로 본 프로젝트는 기술적 성과뿐 아니라, **실험 관리와 팀 협업 방식의 중요성을 인식하는 계기**가 되었으며, 해당 경험은 향후 대회 및 실무 프로젝트에서 보다 성숙한 전략 수립으로 이어질 것으로 기대된다.