

---

# **BoostCamp AI Tech**

## **Project2: Object Detection**

### **Wrap-up Report**

CV-23

**Member:** 김세연, 안지현, 김상유, 김태욱, 김윤서

---

## 목차

1. 프로젝트 개요.....	3
2. 프로젝트 팀 구성 및 역할.....	3
3. 프로젝트 수행절차 및 방법.....	3
3.1 데이터 분석 streamlit.....	3
3.2 Glt Hub 규칙 및 컨벤션.....	5
3.3 Wandb.....	5
4 프로젝트 수행 결과.....	7
4.1 Validation Data.....	7
4.2 모델 실험.....	7
4.2.1 GFL.....	7
4.2.2. Co-DETR.....	8
4.2.3 PVT.....	9
4.2.4 DyHead.....	11
4.2.5. ETC.....	12
4.3 Model Ensemble.....	12
5. 팀 회고.....	13
6. 참고문헌.....	14
7. 개인회고.....	15

## 1. 프로젝트 개요

대량 생산과 대량 소비의 시대에서 환경 부담을 줄일 수 있는 분리수거의 중요성이 더욱 강조되고 있다. 잘못 분리배출 되면 그대로 폐기물로 분류되어 매립 또는 소각되기 때문이다. 따라서 이번 프로젝트에서는 올바른 분리배출을 위해 쓰레기를 정확히 탐지하는 Object Detection 모델 제작을 목표로 한다.

데이터 셋은 일반 쓰레기, 플라스틱, 종이, 유리 등 10종류의 쓰레기가 찍힌 사진을 사용한다. 학습 데이터 4,883장, 평가 데이터 4,871장으로 구성되어 있으며 이미지는 모두 (1024, 1024)크기로 제공된다.

## 2. 프로젝트 팀 구성 및 역할

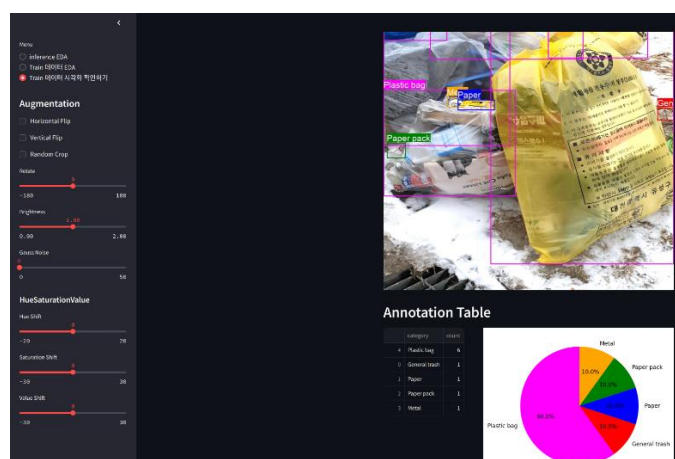
이름	역할
김세연	Streamlit EDA, GIT Template, 모델 실험 및 평가, AugFPN 마이그레이션
안지현	Wandb 등 실험 관리 도구, 모델 실험, 학습 결과 분석
김상유	모델 실험, Val_data, 모델 평가
김태욱	Ensemble 구현 및 실험, Detection 라이브러리 분석
김윤서	모델 실험, 데이터 증강, Val_data

## 3. 프로젝트 수행절차 및 방법

### 3.1 데이터 분석 streamlit

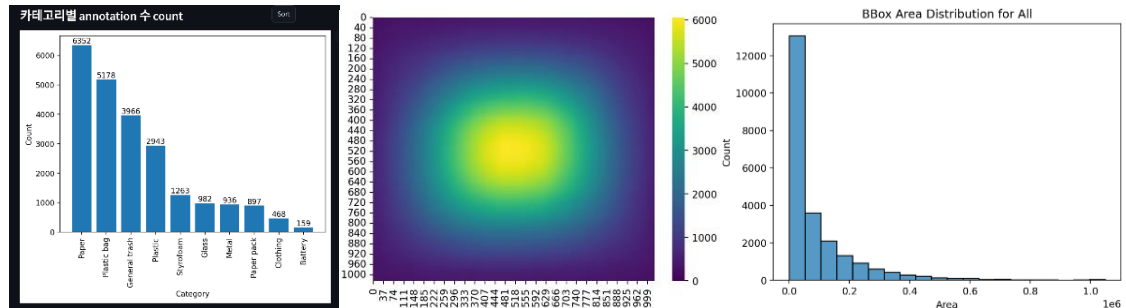
효율적인 EDA를 위해 Streamlit을 통해 데이터 분석 및 시각화 기능을 구현하였다. 또한, Train 데이터 시각화, Train 데이터 EDA, inference 및 validation EDA로 기능과 목적에 따라 세부 페이지를 구성하였다.

#### 3.1.1. Train 데이터 시각화



위 사진처럼 bbox와 이미지를 시각화 했고, Select box를 통해 이미지를 선택할 수 있게 기능을 구현했다. 또한, pie plot와 Table을 통해 현재 bbox의 카테고리과 숫자를 쉽게 파악할 수 있게 배치하였으며 다양한 이미지 증강 기법을 적용하여 합리적인 증강 기법을 정할 수 있도록 시각화 하였다.

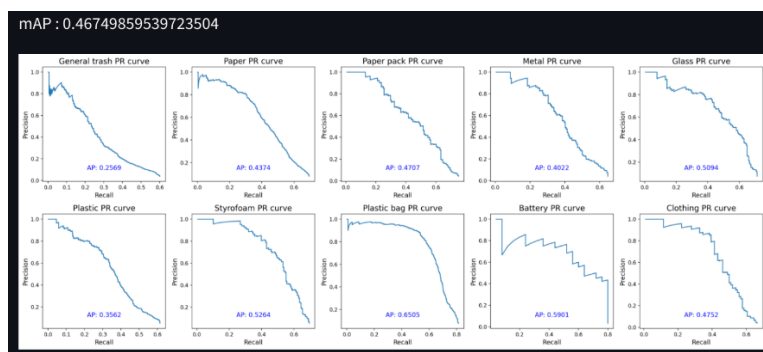
### 3.1.2. Train 데이터 EDA



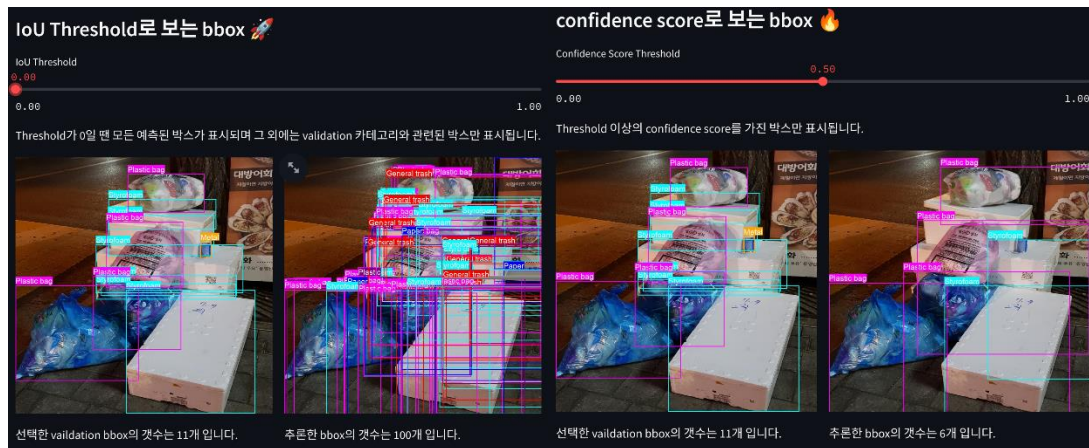
카테고리 별 annotation은 왼쪽 사진과 같다. Paper가 6352개로 가장 많고 Battery가 159개로 가장 적으며 카테고리 간에 불균형이 심하다. 또한, bbox 위치도 가운데에 집중된 것을 볼 수 있으며, area가 50000이하인 bbox가 많은 것을 볼 수 있다. Slider와 타이핑으로 그래프의 x축 bbox 크기를 쉽게 지정할 수 있도록 구현하였다.

### 3.1.3. Inference 데이터 EDA

Inference 혹은 Validation EDA 및 시각화를 위한 페이지이다. 대회 대시보드 metric인 mAP 50와 각 카테고리의 AP를 시각화하였다.



또한, inference한 bbox와 GT bbox를 비교하기 위한 시각화를 진행하였다. IoU, Confidence Score Threshold를 Slider를 통해 쉽게 변경하며 비교할 수 있게 구현하였다. 카테고리 별 bbox heatmap, 크기 분포는 Streamlit을 실행하여 볼 수 있다.



## 3.2 Glt Hub 규칙 및 컨벤션

효율적인 Git 협업을 위해 Git 컨벤션을 정했으며, commit message 가독성 향상을 위해 Gitmoji를 도입하였다. 상세한 내용은 다음과 같다.

아이콘	코드	Tag Name	Description
🌟	:sparkle:	Feat	새로운 기능을 추가
🐛	:bug:	Fix	버그 수정
🔥	:fire:	Remove	파일을 삭제하는 작업만 수행한 경우
🚑	:ambulance:	:ambulance: !HOTFIX	급하게 치명적인 버그를 고쳐야하는 경우
💄	:lipstick:	Style	코드 포맷 변경, 세미 콜론 누락, 코드 수정이 없는 경우
♻️	:recycle:	Refactor	프로덕션 코드 리팩토링
💡	:bulb:	Comment	필요한 주석 추가 및 변경
📝	:memo:	Docs	문서 수정
🧪	:test_tube:	Test	테스트 코드, 리팩토링 테스트 코드 추가, Production Code(실제로 사용하는 코드) 변경 없음
💩	:poop:	Chore	그 외 잡다한 일 (코드 기능에 영향 x)
🍺	:beers:		술 취해서 쓴 코드

Issue Template은 구현하고자 하는 기능에 따라 크게 feature, refactoring, bug Templates으로 분류하였으며, PR Template은 스크린샷과 설명, 그리고 리뷰어에 대한 전달 내용을 구성하여 다른 팀원들의 리뷰 시 이해하기 쉽도록 제작하였다.

## 3.3 Wandb

- 이전 프로젝트에서 학습된 모델 관리를 위해 구글 공유 드라이브를 사용했을 때 다음과 같은 문제점이 있어 이를 보완하기 위해 실험 관리 도구 플랫폼 도입
  - a. 무료 플랜 사용 시 용량 부족 및 수동적으로 실험 모델 파일을 업로드
  - b. 모델 지표를 따로 추적하지 않아 각 모델 간의 성능 차이를 한 눈에 보기 어

려움 따라서 팀원 간 실험 결과를 공유하기 어려워 체계적인 실험관리가 되지 않음

- 위 문제점을 해결하기 위해 MLflow Experiments 사용

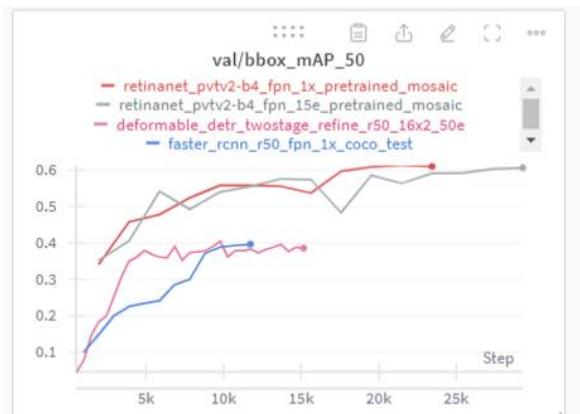


- 한 눈에 각 실험 간의 결과를 비교할 수 있고 기록된 실험 구성을 통해 재현 가능한 실험 환경을 만들 수 있어 체계적인 실험 관리가 가능해짐
- 학습 중의 성능 지표를 실시간으로 성능 저하를 즉시 확인하고 학습 코드를 검토할 수 있었음

- MLflow 서버 구축에 관련한 초기 비용 문제와 서버의 불안정성 문제로 실험관리 도구를 Wandb tracking으로 교체

▼ Summary metrics: {} 98 keys

acc0: 87.5  
base\_lr: 0.00002  
coco/bbox\_mAP: 0.603  
coco/bbox\_mAP\_50: 0.697  
coco/bbox\_mAP\_75: 0.625  
coco/bbox\_mAP\_l: 0.684  
coco/bbox\_mAP\_m: 0.238  
coco/bbox\_mAP\_s: 0.048



- 복잡한 실험 구성과 성능 지표를 수동으로 설정할 필요없이 자동으로 기록하여 반복 작업을 최소화할 수 있어 시간 절약
- SaaS 서비스로 초기 서버를 구축할 필요가 없고, 인프라 관리를 직접 할 필요가 없으므로 시간 절약에 더욱 유리함
- 무료 플랜으로도 100GB 지원으로 용량 걱정없이 안정적인 실험 기록 가능

## 4 프로젝트 수행 결과

### 4.1 Validation Data

Object Detection에서 Test Data와 유사한 Val Data를 만드는 것은 모델 평가에 중요하다. 총 4가지 방법으로 Val Data를 구성했고 Val Data과 리더보드 Public Data의 mAP\_50 성능 차이가 가장 적은 Data로 팀원 모두가 공유할 Validation Data를 지정했다.

기준	Val_mAP50	Public_mAP50
random	0.3890	0.3956
Box_area	0.4026	0.4101
Class	0.4287	0.4120
Box + Class	0.4215	0.4035

결론: Random으로 생성한 Validation Set이 차이가 가장 적다. 이후에 나오는 모든 실험은 특별한 언급이 없다면 모두 Random으로 생성한 Train, Val data로 구성된다.

### 4.2 모델 실험

#### 4.2.1 GFL<sup>[1]</sup>

문제점

- one-stage 모델을 통한 실험이 필요 + 실험과 동시에 높은 성능을 기대
- 프로젝트 시간이 부족하여 많은 실험이 불가능함

해결방안

- One-stage 모델 중 성능이 높고 비교적 학습 속도가 빠른 GFL을 실험 모델 선택
- 실험 성능이 좋으면 바로 최종 앙상블에 투입 가능한 장점

모델 특징

- Bounding Box의 위치 분포를 예측하는 Loss 사용
- 연속적인 label을 처리하는 Focal Loss 사용
- Box localization과 class classification 사이의 상관 관계를 학습과정에 추가

실험 환경

- 12 epochs + SGD+ stepLR (backbone Swin의 경우는 AdamW)
- 사전학습: 이미지넷으로 backbone 학습

이미지 증강 = (512, 512) + base coco\_detection.py

- Model = r-101 + dcn + FPN + GFLHead

#### 실험 결과

변경점	mAP_50	mAP_75	mAP_m	mAP_L
None	0.433	0.296	0.024	0.352
NasFpn	0.427	0.327	0.022	0.393
Load weight	0.477	0.377	0.032	0.445
Swin_t	0.500	0.351	0.045	0.403
Swin_L_1024	0.600	None	None	None

#### 결과 분석

NasFpn을 사용한 경우 큰 성능 향상이 일어나지 않았으며 백본의 크기에 따라 성능이 크게 달라지는 것을 확인했다. Swin\_L을 활용한 모델을 최종 앙상블에 사용했다.

#### 4.2.2. Co-DETR<sup>[2]</sup>

##### 문제점

- 학습한 모델들의 mAP가 기존의 베이스라인에서 큰 성능 개선을 보이지 않음
- deformable DETR<sup>[3]</sup> 및 augFPN 등 학습한 모델에서 공통적으로 배경과 혼동하는 모습을 보임

##### 해결 방안

- COCO detection dataset에서 SOTA를 기록한 모델 실험 및 해당 SOTA 모델 구조를 분석함으로써 성능 향상을 위한 아이디어 탐색
- 객체의 특징 추출 면에서 충분한 학습이 필요한 것 같아 정답에 대한 정보를 양적으로 더 학습할 수 있는 Co-DETR 사용

##### 모델 특징

- 보조 헤드 K개를 활용하여 기존 DETR 구조에서 1:1 매핑에서의 문제점인 부족한 positive sample에 대한 정보를 양적으로 추가

##### 실험 환경



- 12 epochs + AdamW + Multi-step
- 사전학습: COCODataset으로 모델 전체
- Model = backbone + (detr encoder + dino decoder) + auxiliary head(ATSS head + FasterRCNN head)

## 실험 결과

Backbone	mAP_50	mAP_75	mAP_m	mAP_L
Resnet50	0.596	0.511	0.144	0.573
Swin - L	0.697	0.625	0.238	0.684

## 결과 분석

- 작은 박스에 대한 mAP가 모두 작게 나왔으나 전체 데이터셋에 COCO Metric에서 small box와 medium box 기준에 속하는 bbox가 많이 없으므로 큰 의미가 없다고 생각
- 이전 deformable DETR 실험 시와 비슷하게 General trash, Paper Pack, Metal, Plastic와 같이 분류를 못하는 클래스가 여전했고 오류 분석 시 배경과 다른 클래스와의 혼동이 가장 많았음
- Swin-L로 backbone을 변경한 후의 성능 변화가 드라마틱하게 올랐기에 전체 이미지의 특징 추출에 대한 backbone의 충분한 학습이 성능에 영향을 미친다고 분석

### 4.2.3 PVT<sup>[4]</sup>

#### 문제점

- Focal Loss를 추가한 모델들의 성능이 매우 낮게 나타난다.
- Focal Loss의 클래스 불균형 문제 해결을 통한 배경 인식 성능을 가져오고 싶다.

#### 해결 방안

- RetinaNet에서 backBone만 변경하여 높은 성능을 낸 PVT 모델을 사용하고자 한다.

#### 모델 특징

- Transformer 모델을 CNN와 유사하게 다양한 해상도의 Feature map 추출
- 연산량 감소 + Positional 임베딩을 DWConv로 변경 + Patch 임베딩 개선

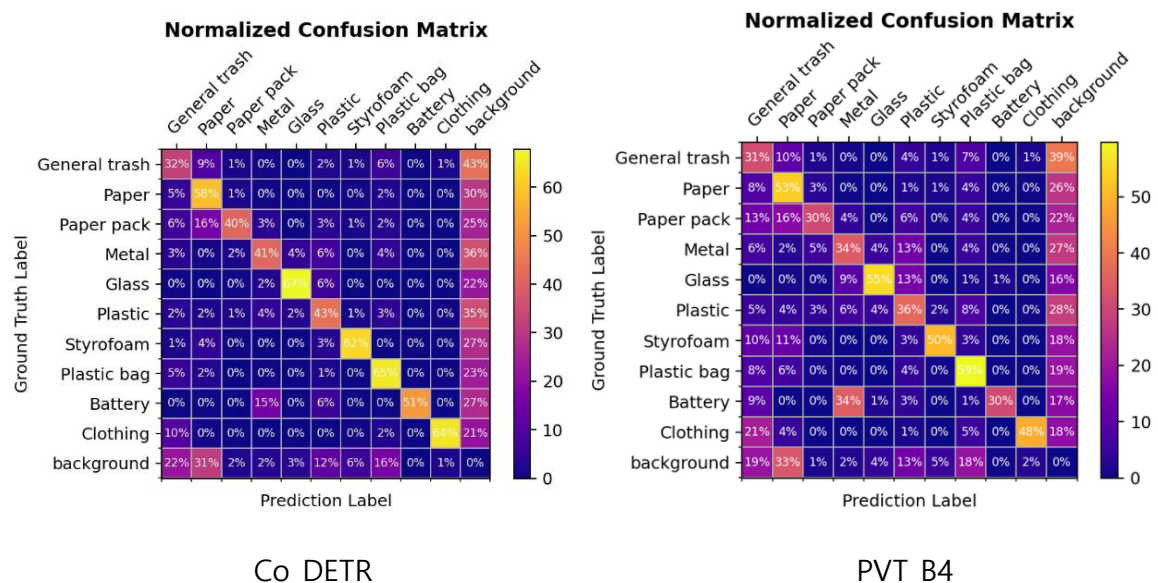
## 실험 환경

- 12 epochs + AdamW + stepLR
- 사전학습: COCODataset으로 모델 전체
- 이미지 증강 = (1024, 1024) + base coco\_detection.py
- Model= PVTv2-B4 + FPN + RetinaHead
- 추가 데이터 증강 실험: PVT\_B4 + mosaic

## 실험 결과

	mAP_50	mAP_75	mAP_m	mAP_L
PVT_B4	0.607	0.477	0.121	0.535
+ mosaic	0.611	0.472	0.125	0.536

## 결과 분석



MAP50 성능이 유사한 Co\_DETR 모델과 비교를 해보겠다. 배경을 객체로 잘못 인지한 경우는 **Co\_DETR = 평균 9.5%, PVT\_B4 = 평균 9.7%**로 거의 동일한 성능을 보였다. 객체를 배경으로 잘못 인식한 경우는 **Co\_DETR = 평균 28.9%, PVT\_B4 = 평균 23.0%**로 PVT\_B4에서 성능이 훨씬 좋았다. Focal Loss의 장점인 positive sample과 negative sample 문제를 해결하여 객체를 배경으로 인지하는 문제를 해결한 것이다.

추가적인 증강 실험 결과, mosaic가 faster rcnn의 전반적인 mAP\_50을 0.02 향상시켰던 것과 달리 PVT\_B4에서는 mAP50이 0.004 향상되는 것에 그쳤다. 그러나 mAP\_m의 수치가 평균적으로 향상되고 public 점수가 전체 데이터를 학습한 PVT\_B4모델보다 0.023정도

(0.5976 -> 0.6207) 높아진 것을 볼 때, mosaic가 보다 복잡하고 작은 객체를 가진 이미지를 생성함으로써 성능을 향상시키는 효과가 있다고 판단했다.

#### 4.2.4 DyHead<sup>[5]</sup>

##### 문제점

- 모델의 neck, head를 다양하게 조합하는 과정에서 backbone을 제외한 나머지 부분이 사전학습이 되지 않는 문제가 발생
- 모델 전체를 학습한 가중치를 가져오는 경우와 성능차이가 심함

##### 해결 방안

- 모델 전체의 가중치를 가져올 수 있는 모델 중 뛰어난 성능의 모델을 사용
- Model zoo 기준 box AP 56.2를 달성한 DyHead 모델을 사용

##### 모델 특징

- ATSS<sup>[6]</sup>를 활용한 label assingment으로 class imbalance 해결
- 객체의 크기, 형태, 표현 방법을 모두 고려한 DyHead 사용

##### 실험 환경

- 12 epochs + AdamW + stepLR
- 사전학습: COCODataset으로 모델 전체
- 이미지 증강 = (1024, 1024) + base coco\_detection.py
- Model: Swin\_Transformer\_L + FPN + DyHead + ATSSHead

##### 실험 결과

Val\_dataset을 사용하지 않고 모든 데이터로 학습을 했기에 다른 지표는 존재하지 않음

DyHead_pretrained_swin_L_1024			0.6475	2024.10.20 22:14
DyHead_pretr..._1024				

##### DyHead 적용 추가 실험

모델	mAP_50	mAP_75	mAP_m	mAP_L
Faster_RCNN	0.210	0.127	0.009	0.149
GFL(3epoch)	0.037	0.017	0.00	0.024

## 분석

DyHead는 연산량이 많은 구조로 사전 학습이 되어 있는 경우에는 높은 성능을 보였지만 다른 모델에 추가해서 사용하기에는 다소 무리가 있는 것으로 보인다.

초기에 지적인 문제점과 같이 사전 학습된 Neck과 Head 사용이 중요하다고 생각한다.

### 4.2.5. ETC

- [AUGFPN \[7\]](#)

- [Swin Transformer](#)

- [EfficientDet \[8\]](#)

## 4.3 Model Ensemble [9]

앙상블은 모델 학습에 있어 예측 결과를 결합하여, 예측이 불안정하거나 잘못된 결과를 방지하기 위해 사용했다. 실험에 사용된 앙상블 기법은 NMS, WBF, Soft-NMS입니다.

### 1. NMS (DyHead\_1024 + DyHead\_512 + GFL)

처음 실험한 NMS의 경우 단일 모델에 비해 성능 상승이 있었다.

### 2. WBF (DyHead\_1024 + DyHead\_512 + GFL)

WBF의 경우 결합을 통해 비용이 많이 들지만 이번 프로젝트에 주어진 데이터 셋의 경우 겹친 object가 많기에 실험의 가치가 있다고 생각했다.

결과로는 NMS보다 높은 점수를 가져왔다.

-> NMS 기법의 경우는 bbox를 제거하면서 일부 정보를 잃어 위 결과 보다 점수가 낮게 나온 것으로 추정된다.

-> 또한, 단일 모델들이 찾아낸 각각 찾아낸 결과(box)가 결합되어 좀 더 정확한 결과가 나왔다.

### 3. Soft-NMS (DyHead\_1024 + DyHead\_512 + GFL)

Soft-NMS의 경우는 최종 앙상블 작업에서는 낮은 결과가 나왔다.

### 4. Soft-NMS + WBF (DyHead\_1024 + DyHead\_512 + GFL)

추론과정에서 soft-nms를 사용하면 성능이 개선되었기 때문에 각 모델의 추론과정에 soft nms를 도입한 후 결과물을 WBF 앙상블 하였더니 준수한 성능이 나왔다.

## 5. Soft-NMS + WBF (DyHead\_1024 + GFL + CO\_DETR + PVT)

성능이 좋은 모델을 추가하여 앙상블을 진행했더니 mAP가 0.7을 넘겼다.

## 5. WBF (NMS, WBF, Soft-NMS + WBF)

최종적으로는 NMS, Soft\_NMS+WBF, WBF를 사용하여 만든 결과를 한 번 더 WBF 기법을 통해 앙상블을 진행했다.

위와 같은 결과를 확인 후, NMS의 특징과 WBF의 특징으로 찾아낸 BOX를 다시 결합하여 좋은 성능을 낼 수 있다는 결론을 내었다.

Ensemble	Score
NMS	0.6506
WBF	0.6916
Soft-NMS	0.4713
Soft-NMS + WBF	0.6909
Soft-NMS + WBF (+ Co_DETR + PVT)	0.7180
WBF(NMS, WBF, Soft-NMS + WBF)	0.7249

## 5. 팀 회고

이번 프로젝트에서 새로 시도했던 점

- Wandb를 활용하여 모델 실험 정리 및 통계화 활용을 높여 팀 협업에 도움을 줌
- GitHub Issue를 적극적으로 활용하여 작업과 문제에 취약했던 점을 개선함
- Streamlit으로 구현된 EDA의 시각화 작업으로 모델 선정에 도움이 됨

이번 프로젝트에서 아쉬웠던 점

- 각 파트의 작업에 집중되어, 타 파트의 작업을 이해하는데 부족하였음
- > 이후 프로젝트에서 코드 구현 및 실험 결과를 리뷰 및 토론을 통해 개선하려 함
- 사전 작업을 위해 시간 소모가 커 모델 실험을 촉박하게 진행
- > 해당 프로젝트로 얻은 경험을 통해 이후 프로젝트 사전 작업 속도를 향상시켜 개선하려 함

이번 프로젝트는 지난 프로젝트의 부족했던 점인 협업에 대해 개선을 하려 노력했다. GitHub와 Notion활용으로 팀원들의 의견을 합쳐 성능 개선에 도움이 되었고, Streamlit, Wandb 등의 기능들로 작업의 편리성과 정리에 큰 도움이 되었다.

## 6. 참고문헌

- [1] [Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection](#) - Xiang Li, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang, Jian Yang
- [2] [DETRs with Collaborative Hybrid Assignments Training](#) - Zhuofan Zong, Guanglu Song, Yu Liu
- [3] [Deformable DETR: Deformable Transformers for End-to-End Object Detection](#) - Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, Jifeng Dai
- [4] [Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions](#) - Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, Ling Shao
- [5] [Dynamic Head: Unifying Object Detection Heads with Attentions](#) - Xiyang Dai, Yinpeng Chen, Bin Xiao, Dongdong Chen, Mengchen Liu, Lu Yuan, Lei Zhang
- [6] [Bridging the Gap Between Anchor-based and Anchor-free Detection via Adaptive Training Sample Selection](#) - Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, Stan Z. Li
- [7] [AugFPN: Improving Multi-scale Feature Learning for Object Detection](#) - Chaoxu Guo, Bin Fan, Qian Zhang, Shiming Xiang, Chunhong Pan
- [8] [EfficientDet: Scalable and Efficient Object Detection](#) - Mingxing Tan, Ruoming Pang, Quoc V. Le
- [9] [Weighted boxes fusion: Ensembling boxes from different object detection models](#) - Roman Solovyev, Weimin Wang, Tatiana Gabruseva

## 7. 개인회고

### 김상유

#### 1. 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

기계적으로 실험을 반복하지 않기 위해 가설을 세우고 실험을 통해 증명하는 것이 팀 전체의 목표였습니다. 해당 목표를 달성하기 위해 모델을 선정한 간단한 이유와 실험 결과 분석 글을 노선에 정리하고 성능이 좋았던 모델은 논문을 읽어보며 원리를 이해하였습니다.

#### 2. 나는 어떤 방식으로 모델을 개선했는가?

학습이 완료된 모델들의 문제점을 찾고 해당 문제점을 해결할 수 있는 방안을 수색했습니다. 서로의 단점을 상쇄할 수 있는 모델리스트를 만들고 공통적인 개선책으로 이미지 해상도를 올리고 train데이터 모두 학습시키는 방식을 적용하여 최종 모델을 학습하는 방식으로 성능을 올렸습니다.

#### 3. 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?

모델이 가지고 있는 장점을 실험을 통해 검증해볼 수 있었습니다. 단순히 최종 성능이 좋다고 넘어가는 것이 아닌 모델이 가지는 특징을 판별할 수 있는 평가지표를 활용하여 정말 해당 부분이 다른 모델에 비해 개선되었는지 확인할 수 있었습니다. 이 과정에서 변인 통제의 중요성을 느꼈고 가설을 뒷받침하기 위해서는 한 두번의 실험이 아닌 변인 통제가 완벽하게 이루어진 조건에서 여러 번의 교차검증이 필요하다는 것을 깨달았습니다.

#### 4. 전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

이전 프로젝트와 다르게 MLflow와 Wandb를 도입하여 실험 관리를 진행했습니다. 실험 결과를 수기로 구글문서에 정리하던 번거로움이 사라졌고 실시간으로 팀원들이 어떤 모델을 학습하고 있는지 알 수 있어서 편리했습니다. 또한 Git을 적극적으로 도입하여 팀원 모두가 코드 공유를 수행하였습니다. Git을 통해 팀원이 개발한 도구들을 간단하게 사용할 수 있어 협업 효율성이 증가했습니다.

#### 5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- 모델과 실험 기록이 저장된 서버가 터져 정보가 사라진 점
- 데이터 증강을 많이 시도해보지 못한 점
- 실험 평가 방법을 일관성 있게 세우지 못한 점

#### **6. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?**

다음 프로젝트에서는 데이터 분석과 증강을 중심으로 진행해보고자 합니다. 각자 진행한 실험을 평가할 수 있는 일관성 있는 평가 지표를 공유하고 실험 결과를 정리할 수 있는 템플릿 작성을 필수로 하고자 합니다. 매일 지정된 시간에 팀원 모두 실험 결과를 읽어보고 리뷰를 하는 방식을 도입하고자 합니다.



## 안지현

### 1. 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

성능 향상을 위해 무의미하게 모델을 다수 선택하여 학습을 돌리기보다 해당 모델 선택에 대한 근거를 기반으로 가설 설정 후 모델 학습 결과를 분석하는 실험 파이프라인을 구축하려 했습니다. 이를 위해 논문을 통해 핵심 아이디어를 조사하고 이를 기반으로 모델을 선택하여 학습을 진행했습니다. 또한 결과 분석에 초점을 두고 다음 가설 설정을 시도했습니다.

### 2. 나는 어떤 방식으로 모델을 개선했는가?

기존 베이스라인 모델인 faster RCNN이 COCO 데이터 셋으로 사전 학습했다는 점을 확인하고, COCO 데이터 셋에 대해 조사하여 COCO 데이터 셋과 사용 데이터 셋 간에 분류 기준이 모호하다는 공통점이 있다는 것을 확인했습니다.

따라서 학습 시, COCO 데이터 셋에서의 사전학습 된 가중치를 사용하였으며, COCO 데이터 셋에서 SOTA를 기록하고 있는 Co-DETR 모델을 사용하여 성능을 개선하였습니다.

또한 학습한 후 오류 분석과 혼동행렬을 통해 해당 모델이 객체에 대한 충분한 특징 추출이 이루어지지 않고 있음을 제시하였습니다.

### 3. 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?

EDA와 학습한 모델의 결과를 분석하여 비교하며 가설을 검증 및 재설정해가며 EDA와 결과 분석의 중요성을 알게 되었고 모델 및 데이터셋에 대한 분석은 한 번에 끝내는 것이 아닌 반복적으로 수행하는 것이 중요하다는 것을 깨달았습니다.

### 4. 전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

팀원 간의 손쉬운 협업을 위해 mlflow 및 wandb를 통한 실험 관리 플랫폼을 도입했습니다. 이를 통해 팀원 간에 실험 결과 공유가 이전 프로젝트에 비해 잘 이루어진 것 같습니다. 또한 노선을 적극 활용하여 진행한 실험에 대한 개요와 결과를 공유함으로써 협업이 순조롭게 진행되었습니다.

## 5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

mlflow 는 생각보다 초기 비용이 많이 드는 작업이었고 불안정한 요소가 너무 많아 예상치 못한 상황이 너무 많이 발생하므로 팀원들의 열심히 실험한 자료들이 사라져서 실험 관리 도구를 도입할 때는 충분한 고려가 필요하다는 사실을 깨달았습니다.

또한 고정된 파이프라인이나 통제 변인 없이 실험이 이루어져 모델 별 결과 분석에 있어서 어려움을 겪었습니다.

EDA 및 실험 관리 도구 구축으로 인해 프로젝트 초반에 시간이 부족해 많은 실험을 하지 못했습니다.

## 6. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

베이스라인과 EDA를 통해 초반 가설 설정 및 통제 변인을 최대한 빠르게 정하고 팀원간의 결과 분석 논의를 활발히 하여 체계적인 실험 파이프라인을 경험해보고 싶습니다.

# 김윤서

## 1. 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

지난 프로젝트에서는 데이터 증강에 집중해서 실험하다 보니 다른 요인들이 어떤 효과를 보이는지 실험해보지 못한 점이 아쉬웠다. 그래서 이번 프로젝트에서는 강의에서 배운 데이터 증강부터 모델 구조까지 다양한 요인을 바꿔보면서 모델 실험을 진행했다. 팀에서는 공통적으로 협업 툴의 활용도를 높이자는 의견이 나와 git 규칙에 따라 github를 관리하고 이슈를 통해 작업 과정을 공유하였다.

## 2. 나는 어떤 방식으로 모델을 개선했는가?

베이스라인을 기반으로 구성 요소를 하나씩 바꿔보면서 데이터셋에 해당 기법이 효과적인지 확인하는 방식으로 실험하였다. Transformer 기반 backbone의 성능을 확인하기 위해 swin transformer로 바꿔보거나 neck 구조를 fpn에서 PAfpn, Nasfpn으로 바꾼 후 기존 모델과의 성능을 비교하였다. 또한, 이미지에 존재하는 흔들림이나 작은 객체를 잘 탐지하는 못하는 특징을 고려하여 motionblur, mosaic 같은 데이터 증강 기법을 시도하였다.

## 3. 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?

베이스라인에 mosaic로 데이터 증강 기법을 시도해보고 성능 개선 효과를 발견했다. 이를 팀원이 실험한 다른 모델에도 적용하여 성능 개선 효과를 확인할 수 있었다. 하지만 같은 실험을 여러 번 해보는 과정에서 같은 조건이어도 학습마다 같은 효과를 보장할 수 없고 변동성이 존재한다는 것을 알 수 있었다. 단일 실험만으로 결과를 판단하기는 어렵고 평가지표를 종합적으로 고려해야 함을 알 수 있었다.

## 4. 전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

함께 정한 git 규칙에 따라 할 일을 이슈로 생성하고 그에 따라 작업하였으며, pull request를 통해 피드백을 주고받았다. 내가 하고자 하는 작업을 좀 더 확실히 할 수 있었고, 작업 과정에서 알게 된 내용을 서로 공유할 수 있었다는 점에서 협업이 발전하였다고 생각한다. 또한, 실험관리를 위해 mlflow, wandb를 사용하면서 결과 기록과 실험 간 비교가 보다 쉽게 이뤄졌다.

## 5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

실험 결과에 대한 분석이 부족했다. 분석 방법이 한정적이었으며, 원인 분석 후 가설을 세우는 절차가 잘 이뤄지지 않았다.

모델 리서치가 부족했다. 베이스라인을 기반으로 강의에 나온 모델 구조 위주로 시도하다 보니 소극적인 실험이 이뤄졌다. 성능이 좋은 모델을 찾고 그 특징을 공부해보며 새로운 지식을 얻어가는 방향으로 나아가지 못한 점이 아쉬웠다.

## 6. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

- 다음 프로젝트에서는 결과 분석과 가설 설정을 구체적으로 하고 노션에 기록하기
- 주어진 eda 코드 이상으로 필요한 데이터 분석 찾아서 하기
- 실험 관리가 잘 이뤄질 수 있도록 통일된 metric 찾기

# 김태욱

## 1. 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

이전 프로젝트와 마찬가지로 Detection 분야의 모델 이해도를 중점으로 목표로 두었고, 앙상블을 추가적인 목표로 두었다. 먼저 MMDetectionV2과 Detectron2의 모델 구조에 대해 알아보고 추가로 Ultralytics의 YOLO11 모델과 MMDetection3 라이브러리를 사용해 보며 One Stage 모델과 최신 버전의 라이브러리의 환경을 구성해보았다. 앙상블의 경우는 Ensemble\_boxes 라이브러리를 통해 NMS, WBF, Soft-NMS, NMW 기법들의 구조를 살펴보고 실험을 통해 앙상블의 이해도를 높이려고 노력하였다.

## 2. 나는 어떤 방식으로 모델을 개선했는가?

WBF 앙상블 기법으로 프로젝트 데이터에 맞는 설정 값들을 실험을 통해 찾고 해당 값을 중심으로 여러 모델들을 결합하여 정확도 상승에 도움을 주었다. 서로 다른 앙상블을 다시 앙상블 작업을 통해 기법들 서로의 문제점을 결합으로 정확도를 추가로 높일 수 있는 결과로 모델을 개선하였다.

## 3. 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?

앙상블의 결과로 팀 목표 점수인 0.70점 이상을 달성하였고, 추가 실험으로 최종적으로 0.72 점 이상을 달성하였다. 이번 프로젝트를 통해 Detection 앙상블 기법에 대한 이해도와 모델 구조에 대해 알게 되었다

## 4. 전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

이전 프로젝트의 경우 첫 프로젝트 경험이기도 하며 모델 구조에 대해 이해하는데 어려움이 있었으나, 모델의 논문과 실험결과를 통해 이해하려 노력하였다. 노력의 결과로는 라이브러리 내부 구성 파일들을 살펴보며 모델 개선에 도움을 줄 수 있다 생각하였다.

## 5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

단일 모델 개선에 있어 Config 파일 외에 부분은 시도를 하기에는 이해의 어려움이 있어 많은 실험을 하지 못하였다. 해당 문제가 모델 이해도가 낮은 이유라 생각하고 추가로 원인을 찾아보지 못했던 점이 아쉽다.

## **6. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?**

다음 프로젝트에서는 이전 프로젝트들에서 시도하지 못했던 부분의 이해도를 높이는 목적으로 진행하고싶다. 데이터 엔지니어링에 대한 이해도가 낮기에 다음 프로젝트 시작 전 기초 지식을 복습하고 아이디어를 조사할 생각이다.

# 김세연

## 1. 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

효율적인 협업과 데이터 분석을 위해 깃 컨벤션 및 규칙, Issue, PR template을 제작하였으며, Streamlit을 통해 EDA를 위한 페이지를 만들었습니다. 또한, 저번 프로젝트에서 모델의 성능에만 집중했던 것과 달리 이번 프로젝트는 EDA와 Inference 결과, 실험 log 등을 분석하며 정량적, 정성적 평가를 통해 모델을 결정하는 과정을 거쳤습니다. 나아가, AugFPN을 마이그레이션하면서 mmdetection의 구조에 대해 보다 깊은 이해를 쌓는 경험을 할 수 있었습니다.

## 2. 나는 어떤 방식으로 모델을 개선했는가?

데이터를 분석한 결과, 작은 bbox를 가지는 객체가 많았습니다. 따라서 작은 bbox에 특화된 모델을 찾고자 했고, AugFPN이 Feature map에 따라 가중치를 다르게 주어 ROI의 크기에 강건할 것이라는 가설을 세우고, 이 모델을 학습했습니다. 또한, EfficientDet 역시 BiFPN의 Weighted Feature Fusion으로 크기가 불균형한 bbox 데이터에 적합하다고 생각하여 모델을 결정했습니다. 이를 통해 basecode보다 0.9 높은 mAP를 달성할 수 있었습니다.

## 3. 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?

이전에 성능에 집중하여 무작정 크고 SOTA를 달성한 모델을 사용했다면, 데이터와 log 등에 기반한 근거를 가지고 모델을 결정했을 때, 예상과 다르게 결과가 좋지 않더라도 그 원인을 보다 쉽게 파악할 수 있었습니다. 특히, 원인을 해결하기 위한 다음 스텝을 정하는 데 훨씬 수월함을 느꼈습니다.

## 4. 전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

깃 컨벤션 및 이슈, Issue, PR template을 통해 팀원들과 이전에 비해 수월한 협업을 이뤘습니다. 서로 코드와 성능을 리뷰할 수 있었고, 다른 팀원들이 어떤 업무를 하고 있는지 파악할 수 있었습니다. 또한, 단순히 주피터 노트북에서 시도하던 EDA를 Streamlit을 통해 구현함으로써 좀 더 효율적인 데이터 분석을 할 수 있었습니다. 또한, MLflow, Wandb 등의 MLOps 툴을 사용하여 실험에 효율성을 가질 수 있었습니다.

## 5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

Mmdetection, Detectron2 등 라이브러리에 대한 이해가 필요했고, 모델마다의 config가 달라 모델마다의 설정을 이해하는 시간을 많이 할애하여 많은 실험을 하지 못했던 것 같습니다. 또한, 팀원들 간에 실험 결과와 분석이 잘 공유되지 않는다고 느꼈고, 모델 평

가에 대한 통일성이 부족하다고 느꼈습니다.

**6. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?**

매일 하루마다 발표 형식으로 작업물 공유의 시간을 가졌으면 좋겠습니다. 또한, 모델 평가에 대한 template을 만들어 통일성을 가지면 좋을 것 같습니다.