

Object Detection Wrap-up Report

CV-03

김승기 김준영 신우진 전형우 천지은

1. 프로젝트 개요

요즘 쓰레기 관련 문제들이 항상 문제로 떠오르고 있습니다. 이러한 문제를 해결할 수 있는 방법 중 하나는 분리수거를 잘 하는 것입니다. 잘 분리 배출된 쓰레기는 자원으로써 가치를 인정받아 재활용되지만, 잘못 분리 배출되면 그대로 폐기물로 분류되어 매립 또는 소각되기 때문입니다. 따라서 분리 배출을 잘 하기 위해서 사진에서 쓰레기를 detection하는 모델을 만들어 보려고 합니다. 쓰레기를 잘 detection하는 모델을 만든다면 쓰레기장에 설치되어 정확한 분리수거를 도울 수 있고 어린 아이들의 분리수거 교육에 사용될 수도 있을 것입니다.

2. 프로젝트 팀 구성 및 역할

팀원	역할
전체	EDA, Dataset Split, Online augmentation
김승기	offline augmentation, git convention
김준영	2-stage detector, neck, optimizer, Mmdetection baseline
신우진	swin based model
전형우	1-stage detector, YoloV8
천지은	offline augmentation, optimizer, pseudo labeling, ensemble

3. 프로젝트 목표

3-1. 팀 목표

저희 팀의 목표는 협업입니다. 개발이라는 것도 코드 작성만이 아니라 사람을 대하면서 일하는 것이기 때문에 저희는 협업을 하면서 soft skill을 기르는 것을 주목표로 정했습니다. 여기서 말하는 soft skill은 원활한 커뮤니케이션을 넘어서 팀원의 의견에 비판적인 견해를 제시할 수 있는 능력까지 말하고 있습니다.

저희 팀의 협업 환경은 기본적으로 Weight and Bias의 팀 repository와 Notion을 통한 실험 공유, Git flow를 이용한 브랜치 전략, Udacity commit convention을 기반으로 convention을 정해서 프로젝트를 진행했습니다.

3-2. 협업 환경 구성

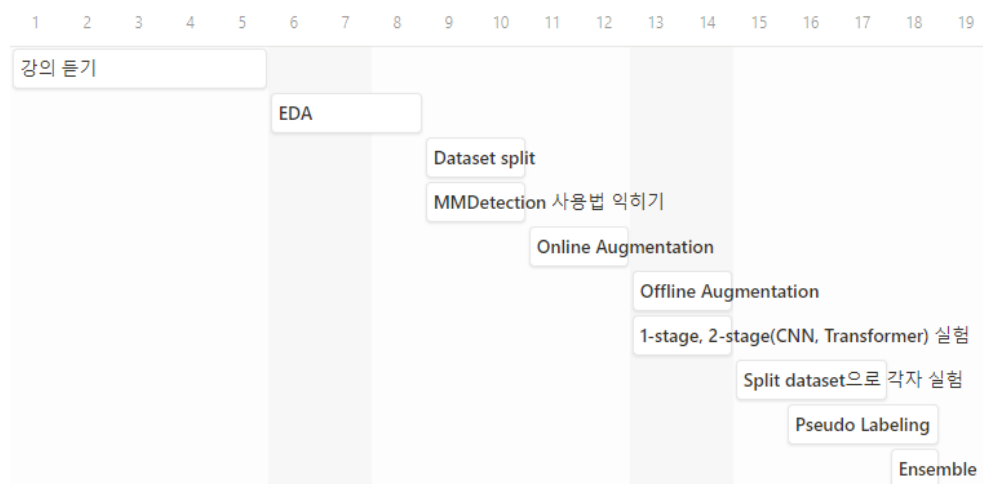
저희 팀의 협업 환경은 기본적으로 Weight and Bias의 팀 repository와 Notion을 통한 실험 공유, Git flow를 이용한 브랜치 전략, Udacity convention을 기반으로 commit convention을 정해서 프로젝트를 진행했습니다.

Mindmap: https://miro.com/app/board/uXjVMNN4fwE=

Notion: <https://www.notion.so/Object-Detection-Project-e5b6579064ff43539b22a832cf31e1c6?pvs=4>

Github: https://github.com/boostcampaitech5/level2_objectdetection-cv-03

4. 프로젝트 수행 절차



프로젝트 타임라인

1. EDA

- 데이터를 파악해서 현재 우리가 수행하는 task에 영향을 줄 수 있는 문제점들을 분석하고, 파악한 내용을 바탕으로 가설을 세웠습니다.

2. Dataset Split

- Validation set과 training set을 나눌 기준을 팀원 각자 생각해와서, aistages mAP50이 제일 높은 데이터셋을 기초 실험에 대한 데이터셋으로 선택했습니다.

3. Augmentation

- 팀원들에게 augmentation 기법을 할당하여 실험을 진행했습니다.

4. Model

- 크게 1-stage detector, 2-stage detector 그리고 2-stage detector 중에서도 Swin transformer를 backbone으로 사용한 모델을 실험했습니다.

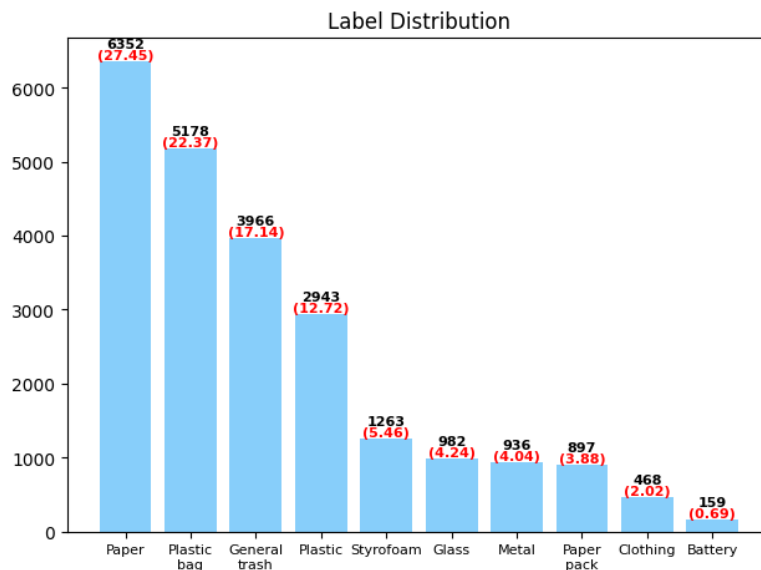
5. 프로젝트 수행 결과

5-1. EDA

데이터셋은 train과 test로 각각 4883, 4871장의 1024x1024 해상도 이미지로 주어졌습니다. 데이터셋에 대한 annotation은 Coco format으로 JSON 파일로 주어졌습니다. 클래스는 Paper, Plastic bag, General trash, Plastic, Styrofoam, Glass, Metal, Paper pack, Clothing, Battery 총 10개의 클래스가 존재하며, 배경에 대한 클래스는 주어지지 않았습니다.

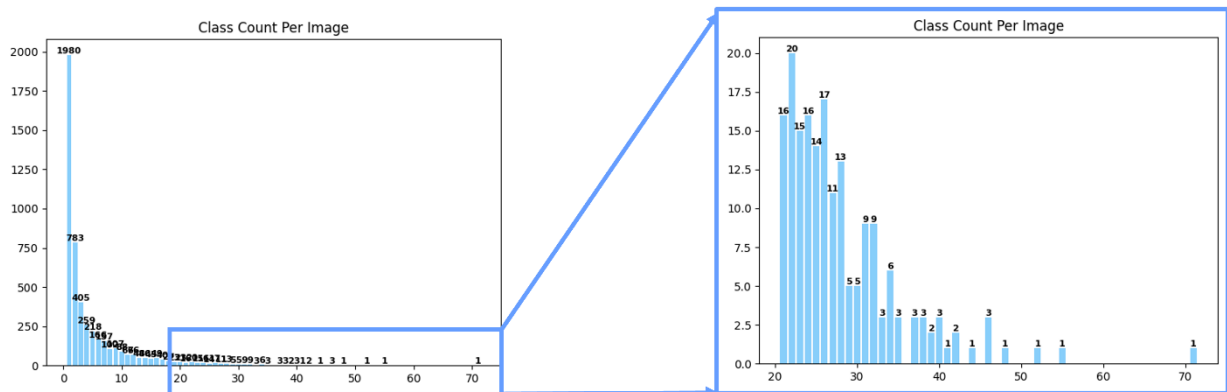
5-1-1. 클래스 불균형

- Paper, Plastic bag만 합쳐도 전체 bounding box 수의 절반 가량을 차지합니다. General trash, Plastic까지 더하면 79.68%로 대부분 bounding box를 차지하고 있음을 알 수 있습니다.
- 나머지 클래스는 전체 bounding box에서 비중이 적으며, 특히 Battery는 매우 적은 수(0.69%)의 bbox가 존재하는 것을 알 수 있었습니다.
- 확인된 Battery의 종류는 Coin Battery, AAA Battery, C Battery, D Battery, FC-1 Battery, Portable Battery, 육안으로 확인하기 힘든 Battery 등이 존재했습니다.



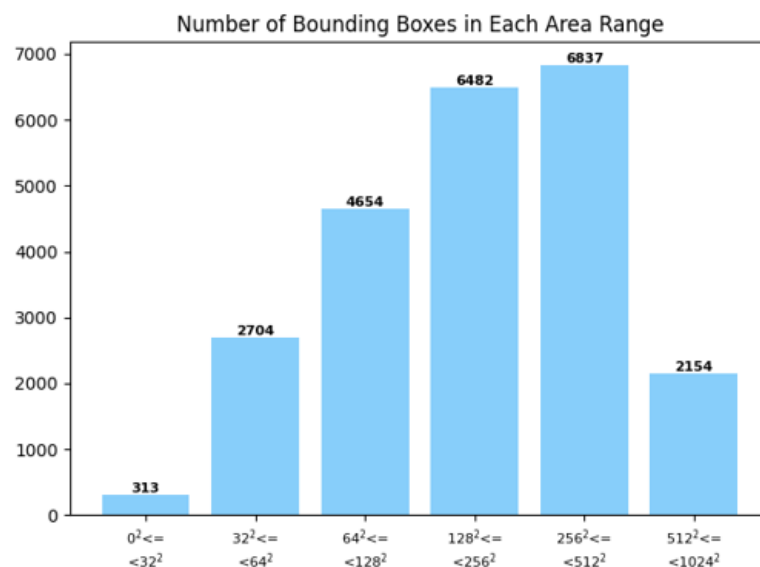
5-1-2. 하나의 이미지에 존재하는 Bounding box의 수

- 하나의 이미지내에 존재하는 bounding box의 수에 대한 분포를 확인했습니다.
- 확인 한 결과 하나의 이미지에는 최대 71개의 bounding box가 존재하는 경우가 있고, 대부분 이미지는 20개내로 존재하는 것을 확인할 수 있었습니다.
- 많은 bounding box가 존재하는 경우를 살펴보면, 많은 object들이 좁은 간격으로 모여 있는 경우가 많은 것을 확인할 수 있었습니다.



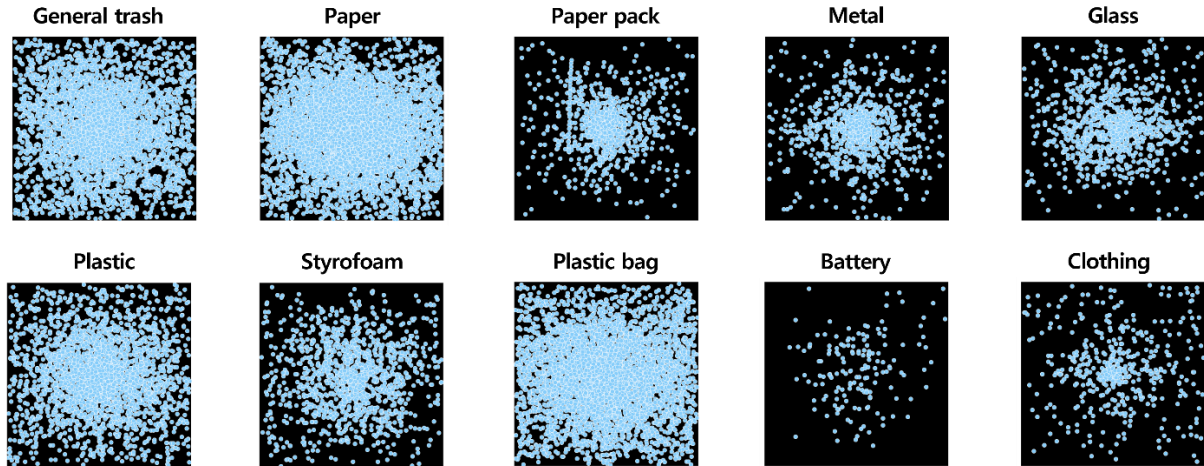
5-1-3. Bounding box의 넓이 범위 별 분포

- Bounding box의 넓이 별 분포도 불균형한 것을 확인할 수 있었습니다.
- 생각보다 넓이가 작은 bounding box의 수가 많다는 것을 확인할 수 있었습니다.



5-1-4. Bounding box의 Center Point 분포

- 클래스가 부족한 object의 경우 Bounding box들의 center point가 대부분 중심에 있다는 것을 확인할 수 있었습니다.



5-2. 문제점과 가설

1. Bounding box의 클래스 불균형이 존재합니다.
2. 작은 bounding box가 많이 존재하며, 검출이 어려울 것이라고 예상됩니다.
3. Center point들이 중심에 몰려 있는 데이터가 너무 많을 경우 bounding box가 외곽에 존재하는 데이터의 예측에 대한 어려움이 우려됩니다.
4. 이미지들에 대한 배경이 다양하게 존재합니다.

클래스 불균형의 경우, offline augmentation으로 up-sampling 해서 해결할 수 있을 것으로 예상했습니다. 작은 bounding box 검출의 경우, 작은 box를 찾을 수 있는 2-stage detector에 대한 실험, 이미지의 해상도를 변경하면서 학습하는 multi scale training으로 해결하려고 했습니다. Center point가 중심에 몰려 있는 경우에 대해서는 geometric transform을 이용한 augmentation을 사용하면 다양한 bounding box의 위치에 대해 학습시킬 수 있을 것이라고 예상했습니다. 다양한 배경의 경우, 일반적으로 배경이 다양한 상황에서도 예측할 수 있는 모델을 만드는 것이 일반화 측면에서 바람직하다고 생각되어서, 일부러 배경을 노이즈 취급하는 접근은 제외했습니다.

5-3. Dataset Split

- Dataset은 validation set이 train set과 클래스 분포가 비슷하면서, 최대한 bounding box의 넓이 별 분포가 고르게 분포하게 구성하는 경우 aistages의 mAP50이 제일 높게 나오는 것을 확인하고 baseline dataset으로 사용했습니다.

5-4. Augmentation

5-4-1. Online Augmentation

- Online augmentation의 경우, 단순 mAP50 수치로는 RandomFlip, Mosaic, ColorTransform이 높았습니다.
- 단순 수치로 augmentation을 정하기에는 한계가 있다고 생각해서, 실제로 나올 법한 real world data를 예상하고, 거기에 초점을 맞추어서 augmentation을 적용했습니다. 가령, RandomFlip과 Rotate과 같은 geometric transform은 여러가지 viewpoint에서 찍힐 수 있는 이미지에 대해 모델이 학습할 수 있도록 도울 것이라고 예상했습니다. Color, Brightness, Contrast Transform은 다양한 밝기나 색에 대한 이미지를 학습할 수 있을 거로 예상했습니다.

5-4-2. Offline Augmentation

부족한 클래스에 대한 데이터를 증강시켜서 클래스별 불균형 문제 해소를 기대하여 다음의 3가지 방법을 진행했습니다.

1. Battery Up-sampling

- 매우 부족한 Battery 클래스가 포함된 이미지에 대해 몇 가지 geometric transform과 RandomBrightnessContrast를 적용해서 up-sampling을 했습니다.

2. Mosaic

- 부족한 클래스인 Paper pack, Metal, Glass, Styrofoam, Clothing, Battery만 포함된 이미지 중 4개를 랜덤하게 뽑아서 mosaic를 진행했습니다.

3. Cutmix

- Mosaic과 마찬가지로 부족한 클래스만 포함된 이미지를 target 이미지로 정하고, 부족한 클래스에 포함된 bounding box를 source 이미지에서 copy해서 target 이미지로 paste 하는 방식으로 새로운 이미지를 생성했습니다.
- 학습 과정에서 발견한 Cutmix의 문제점: Bounding box를 단순 copy paste 방식으로 새로운 이미지를 만들어내는 방식이라서 비슷한 이미지가 존재하는 문제점이 있었습니다. 또한, paste한 bounding box의 경계가 너무 뚜렷해서 validation에서 네모 모양인 경우 object로 예측하는 경우가 생겼습니다.

5-5. Model

1. 1-stage detector

- 2-stage 보다 낮은 성능을 보였습니다.
- YoloV8도 aistages mAP50이 0.4637로, 2-stage에 비해 낮은 점수를 보였습니다.
- 양상블을 위해 YoloV8을 계속 고려했습니다.

2. 2-stage detector

- Neck의 경우 PAFPN과 FPN 비슷한 성능을 보였습니다.
- Swin transformer를 backbone으로 사용한 경우 Faster RCNN이 Cascade RCNN보다 medium box에 대한 성능은 좋지만 small box에 대한 성능은 떨어져서 나중에 양상블에서 고려하기로 했습니다.



FasterRCNN_Swin_V3: Backbone - Swin Large, Detector- Faster RCNN

CascadeRCNN_SwinL_V3: Backbone - Swin Large, Detector- Cascade RCNN

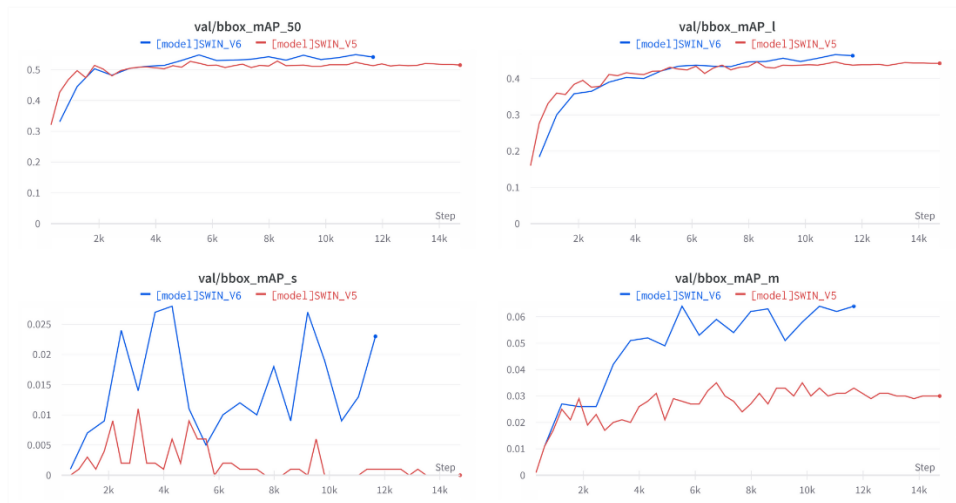


Faster RCNN V5: Neck - FPN

Faster RCNN V6: Neck - PAFPN

3. 이미지 크기(512 vs 1024)

- 이미지의 크기를 키우니 small과 medium bounding box에 대한 성능이 올라간 것을 확인할 수 있었습니다.
- Multi scale training의 경우에도 small과 medium bounding box에 대한 성능이 올라간 것을 확인할 수 있었습니다.



SWINV5 – 512x512

SWINV6 – 1024x1024

4. TTA

- Mmdetection의 경우 TTA가 NMS 또는 WBF 방식으로 적용될 것으로 예상했습니다. 이 때 TTA에서 test 데이터가 너무 현실과 동떨어진 데이터가 나오면 안 되기 때문에, multi-scale resize 또는 flip 정도가 적용되도록 했습니다.

5. Ensemble

- Small, Medium, Large box에 대해서 각각 성능이 좋은 모델 3개씩 WBF 방식으로 앙상블을 진행 후 각 3개의 앙상블 결과를 다시 WBF로 앙상블을 진행했습니다.
➔ Public mAP50: 0.6634
- Test dataset에 대한 mAP50이 높은 모델 3개와 YOLOv8을 앙상블을 진행했습니다.
➔ Public mAP50: 0.6793(제출)
- Test dataset에 대한 mAP50이 높은 모델 5개와 YOLOv8을 앙상블을 진행했습니다.
➔ Public mAP50: 0.6835(제출)

5-6. 결과

Public

순위	팀 이름	팀 멤버	mAP 
6 (-)	CV_03조	    	0.6835

Private

순위	팀 이름	팀 멤버	mAP 
6 (-)	CV_03조	    	0.6713

6. 자체 평가 의견

잘했던 점

- 꾸준한 커뮤니케이션과 협업을 통해서 프로젝트를 진행했습니다.
- git flow를 이용한 branch 관리, PR을 통한 코드 리뷰, commit convention을 잘 지켰습니다.
- 전부 적극적으로 커뮤니케이션에 참여했습니다.

시도했으나 잘 되지 않았던 점

- MMdetection에서 구현되지 않은 SOTA 모델을 사용하려고 했으나 잘 되지 않았습니다.
- PR을 진행할 때, 꼼꼼함이 부족했습니다.
- Issue에 대한 관리가 부족했습니다.

아쉬웠던 점

- 비슷한 task에 대해 먼저 찾아보고 프로젝트를 진행하지 않은 것이 아쉽습니다.
- 모델 하나하나가 학습시간이 오래 걸려서 많은 것을 시도해보지 못했습니다.
- 실험 방향성에 대한 리마인드가 부족했습니다.
- 중간에 다시 EDA를 통해 문제점을 파악하지 못했습니다.
- 실험에 진행함에 있어서 괜찮은 baseline 모델을 정하지 못한 것 같아서 아쉽습니다.
- 프레임워크 사용법을 익히는데 너무 많은 시간을 이용한 것이 아쉽습니다.
- 모델에 대한 정확한 이해도 없이 성능만으로 선택한 것이 아쉽습니다.

프로젝트를 통해 배운 점 또는 시사점

- 문제점과 해결방안을 정의할 때 조금 더 논리적으로 접근할 필요성을 느꼈습니다.
- 중간에 리뷰시간을 가져서 가설을 확인해보고 방향성을 다시 정할 필요가 있는 것 같습니다.
- 협업에 있어서 반대 의견을 낼 줄 아는 것이 중요한 것 같습니다.
- git flow를 통한 협업을 배울 수 있었습니다.
- 여러가지 soft skill을 익힐 수 있었습니다.
- MMdetection 프레임워크에 대한 사용법을 익힐 수 있었습니다.

다음 프로젝트에서 시도할 것

- 모든 실험에 충분한 근거를 가지고 실험하겠습니다.
- 강의 내용에 대한 충분한 이해를 가지고 프로젝트 진행하겠습니다. (공부한 강의 내용 공유하기)
- EDA를 통해 정의한 가설을 실험을 통해서 검증하겠습니다.
- 기존의 연구를 찾아보고 프로젝트 진행하겠습니다.
- 팀원이 제시한 의견에 대한 비판적 사고를 가지고 의사소통 하겠습니다.

개인 회고: 김승기

내 개인적인 학습 목표

이번 프로젝트를 통해 git flow 를 통한 협업을 경험해보고 커뮤니케이션 능력과 같은 소프트 스킬을 기르는 것이 첫 번째 목표였습니다. 팀원들과 협업을 위한 규칙을 정하고, 프로젝트를 진행하면서 어떻게 하면 효과적으로 의견 교환을 할 수 있는지 많은 고민을 했습니다. 두 번째 목표는 object detection 을 실전에서 경험해보면서 이론과 실전의 차이를 인지하고, MMDetection 프레임워크에 익숙해지는 것이었습니다.

내 학습 목표를 달성하기 위해 한 일

1. Git flow, Udacity convention

협업을 위해서 git flow 방식의 branch 전략을 사용했습니다. Udacity convention 을 이용한 commit template 을 작성해서 팀의 commit convention 으로 이용했습니다. Develop branch 로 merge 하기 전, 항상 PR 에서 코드 리뷰를 하고 진행했습니다. 또한, Github issues 에서 기능의 사용법이나 코드의 변경 사항에 대한 공지를 올렸습니다.

2. Baseline code(MMDetection) 이해 및 customize

저는 configuration 기반의 프레임워크인 MMDetection 의 pipeline 을 이해했습니다. 실험의 편의성을 위해서 MMDetection 의 train.py 를 저희의 프로젝트 구조에 맞게 수정하고, 실험의 자동화를 위해 shell script 를 작성했습니다.

3. EDA

데이터를 파악하기 위해 클래스 분포와 bounding box 넓이 범위 별 분포를 시각적으로 확인했습니다. 또한 데이터를 직접 육안으로 확인하면서 특성을 확인하고, 특정 이미지 transform 을 적용할 경우의 전후 결과를 확인해서 그 차이를 파악하려고 노력했습니다.

4. Offline Cutmix 구현

부족한 클래스를 보충하기 위해서 offline cutmix augmentation 을 구현했습니다. Dataset 을 순차적으로 돌면서 target_label 이라는 리스트에 포함된 label 만을 가진 bounding box 를 source 이미지에서 복사했습니다. 그 후 오로지 target label 만을 가진 이미지를 target image 로 정해서, 특정 IoU threshold 를 넘여가지 않은 범위에서 target image 위에 paste 하는 코드를 구현했습니다.

학습 목표를 달성하기 위한 과정 중 깨달은 점

Git flow를 통해서 branch를 관리하고, 정확한 convention을 지켜가면서 협업을 하면서 프로젝트를 효과적으로 진행할 수 있어서, 이런 규칙을 정하고 협업을 하는 것이 생산성에 굉장히 큰 요소라는 것을 느꼈습니다. 또한 프로젝트를 위한 실험을 진행하면서, CV task를 위한 기초적인 지식이 탄탄해야 할 필요성을 느꼈습니다. 가설 검증이나 실험 방향성을 정하는 과정에서 모델에 대한 이해, 데이터에 대한 이해, 통계학적 지식이 바탕이 되어야 확장할 수 있는 시야가 넓어진다는 것을 느꼈습니다.

마주한 한계

1. 구현 능력 부족

특정 기능을 구현하는 과정에서 많은 어려움을 느꼈습니다. 원하는 기능을 Github에서 찾고 사용할 수도 있겠지만, 정말 필요한 기능을 특수하게 구현하는 경우 많은 시간을 소비해야 했습니다.

2. MMDetection customize에 대한 어려움

MMDetection에서 TTA(Test Time Augmentation)가 적용되는 원리를 찾고 싶어서 documentation이나 Github를 살펴보았지만 찾지 못했습니다. 또한, 찾더라도 내가 원하는 기능을 customize할 수 있을 것이라는 자신감이 없었습니다.

3. 이론

모델에 대한 실험을 진행하면서, 이론에 대한 지식을 실전에서 적용하는 것이 어려웠습니다.

아쉬웠던 점

시간 부족

정말 시도해보고 싶은 실험들이 많았지만, 시간이 부족해서 아쉬웠습니다.

다음 프로젝트에서 시도해보고 싶은 것들

1. Issue Tracking

Issue를 올리고, issue number를 branch와 PR 관리에 제대로 사용해 볼 것입니다.

2. GPU scheduler

GPU scheduler에 대한 사용법을 익혀보고 싶습니다.

개인 회고: 김준영

나는 내 학습목표를 달성하기 위해 무엇을 어떻게 했는가?

Object detection 에 대해 공부를 했습니다. 모델을 사용하면서 그 모델의 특징들을 모르고 성능만 올리는 것들은 의미가 없다고 판단해서 이번에 사용한 모델에 대해서 공부를 했습니다. 특히 2 stage 에 집중을 했고 공부한 이론이 어떻게 코드로 구현이 되어있는지 연결을 시켜가며 공부를 했습니다.

- 그 중에서도 Classification 대회에서도 사용한 Swin Transformer 에 대해서 논문을 직접 읽으며 공부를 했던 점이 좋았던 것 같습니다.

전과 비교해서, 내가 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

- Github 을 이전 대회에 비해서 협업의 툴로서 더 잘 활용한 것 같습니다. 특히 Git flow 전략을 잘 지킨 것 같고 PR 과 Issue 활용을 잘한 것 같습니다. 특히 PR 과 Issue 를 활용하다 보니 어떤 부분이 코드에서 변경되고 어떤 기능이 추가되고 어떤 문제들이 현재 있는지 파악하기 수월했습니다.

- 이전 대회에서는 Notion 에 실험을 각자 기록하다 보니 팀원분들이 어떤 실험을 하고 있는지 제대로 공유가 안되는 부분이 있었는데 Notion 페이지에 다같이 실험을 기록하니 이후 실험을 진행해도 될지 아니면 조금 더 시간이 필요할지, 다음엔 어떤 실험이 필요한지 알 수 있었던 것 같습니다.

마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- Object Detection 이라는 task 를 처음 접하다 보니 이전에는 놓치지 않고 했던 부분을 이번 대회를 진행하면서 놓친 것이 아쉬웠습니다. 이전 대회에서는 실험을 진행하고 실험의 결과를 다음 실험에 반영하려는 노력들을 했었는데 이번 대회에서는 시간의 부족, object detection 이라는 분야의 생소함 때문에 이런 부분들을 놓쳤던 것 같습니다.

- 이전 대회에서 EDA 의 결과를 가지고 가설을 설정하고 가설을 검증하는 부분을 놓쳤어 서 이번 대회에서는 이러한 부분을 놓치지 않고 하고 싶었지만 그러지 못한 점이 너무 아쉽습니다. 조금만 더 일찍 실험들을 진행했다면 마음의 여유를 가지고 할 수 있지 않았을까 하는 아쉬움이 남습니다.

- 무작정 모델을 선정했던 부분이 마음에 걸리는 것 같습니다. Swin Transformer 를 선택한다면 선택한 이유가 무엇인지 명확히 했었으면 더 좋지 않았을까 싶습니다. 단순히 성능이 잘 나와서 선택한다면 저에게는 발전이 없지 않을까 생각합니다. 그리고 CNN 과 Transformer 에 어쩔 수 없는 차이 때문에 transformer 계열을 backbone 으로 선택했다면 transformer 에서도 왜 Swin Transformer 를 선택했는지 또는 실험을 통해 어떤 부분에서 다른 transformer 계열의 모델이 아닌 Swin Transformer 를 선택했는지 알아냈어야 했다고 생각하는데 이 부분을 못한 점이 아쉬운 것 같습니다.

한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇일까?

- 가설 설정 → 실험 하나 진행 → 가설 검증의 순서를 꼭 지켜서 프로젝트를 진행하고 싶습니다.
- 이후 재사용을 위해 EDA 를 위한 코드 또한 협업하고 싶습니다.
- 프로젝트 진행 전 관련 연구들 찾아보려고 합니다.
 1. SOTA 모델은 어떤 것들이 있는지 찾아보려고 합니다.
 2. EDA 를 통해 찾은 문제들을 어떻게 해결했는지 알아보려고 합니다.
- 모델을 먼저 선정하고 실험들 진행하려고 합니다.
- 모든 선택에 이유를 만들고자 합니다.

개인 회고: 신우진

나는 내 학습목표를 달성하기 위해 무엇을 어떻게 했는가?

이번 대회에 학습 목표로 설정한 것은 두가지입니다. 첫번째는 협업을 배우는 것이고 두번째는 object detection 에 대한 이해를 높이는 것입니다.

첫번째 학습 목표를 협업으로 한 이유는 다음과 같습니다. 이전 팀과 level 1을 진행할 때 협업이 잘 이루어지지 않았기 때문입니다. baseline 부터 시작해서 실험 결과 등등 공유했으면 좋았을 정보들도 잘 공유되지 않았습니다. 저 뿐만 아니라 기존 팀원들도 대부분 이런 대회가 처음이었기 때문에 각자 대회 자체에 적응하는데 시간이 많이 걸려서 협업이 잘 되지 않았던 것 같습니다. 저번 대회에서 얻지 못했던 경험을 얻고 싶기도 하고 저번 대회에서 좋은 성적을 냈던 팀들은 대부분 협업을 잘 했었기 때문에 첫번째 목표를 협업으로 설정했습니다.

두번째 학습 목표를 object detection model 에 대한 이해로 한 이유는 다음과 같습니다. 저번 대회에서는 모델에 대한 이해를 충분히 가지지 않은 상태에서 대회를 진행했습니다. 해보고 싶었던 실험을 먼저 진행하고 싶었기 때문입니다. 제가 생각했던 실험을 충분히 해 볼 수 있다는 점에서는 좋았지만 대회를 통해 모델에 대한 이해가 깊어지지 않는 않았습니다. 따라서 이번 대회는 우선 object detection model 에 대한 이해를 충분히 가지고 난 뒤에 대회에 참여하고 싶었습니다.

전과 비교해서 새롭게 시도한 변화, 효과

저번 대회와 비교하여 이번 대회에서는 notion 을 적극적으로 활용했습니다. 저번 대회에서도 notion 을 활용하긴 했으나 팀 페이지 안에 개인 페이지를 만들고 각자의 스타일대로 작성했기 때문에 사실상 notion 을 협업 툴로서 제대로 활용하지는 않았습니다. 이번 대회에서는 각자의 페이지를 만들기 보다는 하나의 포맷 안에 서로의 내용을 공유하는 페이지를 많이 만들었습니다. notion 을 활용하는 것이 익숙하지 않았고, 실험 결과를 공유하는 과정이 번거로웠지만 그래도 팀원들에게 공유해야 하기 때문에 구체적이고 논리적으로 실험 결과를 공유하려고 노력했습니다. 이런 과정을 통해 팀원들의 실험 결과를 손쉽게 확인할 수 있을 뿐만 아니라 제가 진행하는 실험이 기존보다 조금 더 체계적으로 변했음을 느꼈습니다. 이전 대회에서는 했던 실험 결과를 제대로 정리해 놓지 않아서 실험의 방향이 불분명했는데 이번 대회에서는 남들이 알아볼 수 있을 정도로 실험 결과를 정리해야만 했기 때문에 다음 실험 계획을 세울 때도 조금 더 체계적으로 세울 수 있었습니다. 저번 대회에서는 강의를 충분히 듣지 않은 상태에서 프로젝트를 진행했었습니다. 그래서 강의를 통해 쉽게 얻을 수 있는 정보를 너무 어렵게 얻는 경우가 많았습니다. 따라서 이번 대회를 진행할 때는 우선 강의를 먼저 듣고 object detection model 에 대한 이해를 어느정도 갖춘 뒤 프로젝트에 참여했습니다.

마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

마주했던 한계 첫번째 한계점은 object detection 에 대한 전반적인 이해 부족입니다. object detection model 을 먼저 충분히 이해한 뒤 프로젝트에 참여하는 것이 목표였지만 이 부분이 잘 이루어지지 않았습니다. 대회에 참여하기 위해 object detection model 을 이해하는 것 이외에도 알아야 할 부분들이 많이 있었습니다. 특히 MMDetection 라이브러리를 이해하는데 시간이 많이 걸렸습니다. 또한 object detection model 을 모두 충분히 이해한 뒤 프로젝트를 참여하기에는 시간이 너무 부족했습니다. 프로젝트 기간이 짧기도 했지만 object detection model 을 이해하는 것이 생각보다 더 어려웠고 시간이 많이 걸렸기 때문입니다. object detection 을 이해하기 쉽지 않았고 라이브러리를 자유롭게 사용하기 까지 시간이 많이 걸렸기 때문에 더 다양하게 실험을 진행하지 못한 것 같습니다.

두번째 한계점은 프로젝트에서 해결해야 할 문제점을 잘 정의하지 못한 것 입니다. EDA 를 하면서 해결해야 할 문제점을 찾을 때 충분히 검증하지 않고 문제를 정의했습니다. 예를 들어 사진 모퉁이에 있는 객체가 많지 않다는 점을 데이터를 통해 확인한 뒤 바로 모델이 모퉁이에 있는 객체를 잘 잡지 못할 것이라고 문제를 정의했습니다. 모델이 정말 모퉁이에 있는 물체를 잘 잡을지 아닐지는 직접 확인해 봐야 아는 문제인데 단순히 상대적으로 데이터가 적다는 이유로 이를 문제점이라고 생각했습니다.

한계, 교훈을 바탕으로 다음 프로젝트에 시도해보고 싶은 점들

다음 프로젝트에서는 문제를 정의하는 시간을 충분히 가지고자 합니다. 프로젝트에서 풀어야 하는 문제를 정의할 때 데이터를 보고 이런 문제가 있을 것이다 추측하는 것이 아니라 정말 그 문제가 있는지 실험을 통해 확인해야 한다는 것을 느꼈습니다. 따라서 다음 프로젝트에서는 EDA 의 범위를 모델 실험까지 넓히려고 합니다.

실험을 정리하고 공유하는 습관은 계속해서 이어가고자 합니다. 남에게 보여 줄 수 있을 만큼 체계적으로 정리해야 저 스스로도 충분히 정리가 되는 것을 느꼈기 때문입니다. 실험 뿐만 아니라 공부한 논문에 대해서도 포스팅을 할 수 있을 만큼 정리된 글을 쓸 수 있어야 한다는 것을 느꼈습니다. 여러 모델에 대한 이해를 더 높이기 위해 정리하는 습관을 더 기르려 합니다.

개인 회고: 전형우

나는 내 학습목표를 달성하기 위해 무엇을 어떻게 했는가?

Object Detection 이라는 task 를 실제로 도전하는 것은 처음이었기 때문에 생소한 점이 많았습니다. 그래서 우선 1-stage Detector, 2-stage Detector 의 이론에 대해 다시 복습을 하며 task 에 대한 이해를 하려고 노력했습니다. 그리고 베이스라인 코드가 기존과 다르게 Object Detection 에 특화된 오픈소스 라이브러리를 통해 이루어져 있었고, config 형식의 파이썬 스크립트 파일로 이루어져 있어 매우 생소했기에 이 라이브러리의 구조에 대해 이해하려고 노력했던 것 같습니다.

모델 개선을 위해 어떤 방식을 시도해봤는가?

1. hyper parameter 변경

YOLOv8 모델의 학습 과정에서 Loss 에 box loss, class loss, dfl loss 가 있는데 이 loss 들에 대한 가중치를 조절하는 파라미터를 계속해서 변경하며 box 를 잘 잡는 것에 가중치를 두지, class 를 잘 구별하는 것에 가중치를 두지 고민하며 이 parameter 를 계속해서 변경하며 실험을 했습니다.

2. 모델 크기 변경

hyper parameter 변경으로는 큰 변화가 나타나지 않아 모델의 크기를 늘리는 방법을 사용했고, 가장 작은 모델인 YOLOv8n 모델에서 가장 큰 모델인 YOLOv8x 모델로 변경한 결과 validation 과 제출 시 모두에서 mAP 가 크게 상승했습니다.

마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- 하이퍼 파라미터를 계속 변경해도 성능이 오르지 않았던 문제로 2일정도를 고생했는데, 모델을 가장 작은 YOLOv8n 모델을 사용하고 있던 것을 깨닫지 못했던 점이 아쉬웠습니다. 가장 중요한 모델의 크기를 늘리는 점을 간과하고 있었던 문제로 시간을 많이 날렸으며, 성능을 개선하기 위해 하이퍼 파라미터를 변경하는 것에만 몰두하고 있어 다른 점을 변경하는 것을 놓친 것이 아쉬웠습니다.

- 깃을 더 많이 사용했던 점은 좋았지만, 제가 직접 깃에 PR 을 올리거나 했던 일이 적어 아쉬웠습니다. 팀원들과 깃을 통해 협업하며 서로 변경한 점이나 유용한 기능들을 만들어서 깃에 배포를 했는데, 이를 더 많이 하지 못한 것이 아쉬웠습니다.

- 팀원들과 다른 내용을 혼자서 고집했던 것이 아쉬웠습니다. 같이 MMDetection 을 사용하며 체계적으로 실험하다 어느 순간부터 YOLO 를 사용해야겠다고 마음을 먹은 후로 계속 혼자서 YOLO 라이브러리를 사용하며 팀원들이 어떤 환경으로 실험을 하고 있는 지 놓치게 된 것 같습니다.

한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇일까?

- 다음 프로젝트에서는 팀원들과 함께 실험을 계획하고 더 소통하며 진행을 하려고 합니다. 혼자 독단적으로 다른 내용을 진행하며 프로젝트 도중에 동떨어진 느낌을 받았으며, 팀원들에게 제가 진행중인 내용 공유를 더 많이 했다면 팀원들의 도움을 받았을 수도 있다고 생각합니다. 따라서 제가 하고 있는 내용을 다음 프로젝트에서는 더 많이 공유하려고 합니다.

- 제 실험을 체계적으로 진행하려고 합니다. 따로 제 실험을 진행하더라도, 제 실험에 대해 여러 변인들을 조절할 때 변인 통제를 잘 하지 않고 무작위로 변경해가며 실험을 진행해서 성능에 어떤 변인이 영향을 미치는 지 잘 몰랐던 것 같습니다. 따라서 다음엔 변인을 통제하며 더 체계적인 실험을 하려고 합니다.

개인 회고: 천지은

나는 내 학습목표를 달성하기 위해 무엇을 어떻게 했는가?

저의 학습목표는 팀원들과 지속적인 소통과 계획적인 실험으로 협업을 경험해보는 것이었습니다. 팀원들과 협업을 위해 대회를 진행하기 앞서 정한 규칙들을 잘 지킬 수 있도록 하고, 맡은 역할에 대한 구현 코드 및 실험 결과를 바로바로 공유할 수 있도록 노력하였습니다. 또한, git branch 전략으로 git flow를 사용하였는데 처음에는 두려움이 있었고 팀원에게 의지를 했었는데, 나중에는 github를 잘 다룰 수 있게 되어 적극적으로 활용하고 빠른 공유를 할 수 있었던 것 같습니다.

모델 개선을 위해 어떤 방식을 시도해봤는가?

1. offline Augmentation

먼저 EDA를 통해 클래스별 불균형이 있는 것을 파악하였습니다. 그래서 가장 부족했던 Battery 클래스에 대해 Up-Sampling을 진행하여 성능을 개선시켰습니다. 그리고 mmdetection 내에 online Augmentation으로 Mosaic이 존재하였지만, 데이터셋을 증강시키는 효과는 없으므로 아직도 부족한 클래스에 대해 Mosaic을 offline Augmentation으로 구현하여 성능을 개선시킬 수 있었습니다.

2. Pseudo Labeling

그리고 학습시킨 모델을 통해 Test Dataset의 결과를 Pseudo Labeling하여 다시 학습시키는 방법을 시도해봤습니다. 이를 통해 성능 향상을 기대해보았지만, 임계값을 어느 정도로 설정해서 학습 데이터로 사용할 지 몰랐어서 시도해본 저조차도 확신이 없던 모델이었습니다.

3. Ensemble

팀원들과 의논하여 성능 향상이 기대되는 모델들과 Ensemble 기법을 정하여 Ensemble을 시도해본 결과 최종적으로 Ensemble을 적용하기 전보다 성능이 매우 향상된 mAP를 기록했습니다.

마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

EDA를 통해 매우 적은 Battery 클래스에 초점을 맞췄었는데, 추후 Confusion Matrix 결과를 통해 Battery 클래스의 문제가 아니었음을 뒤늦게 파악했습니다. 또한, 팀원들과 EDA를 열심히 진행하였다고 생각했지만 문제점을 제대로 파악하지 못했고 이외에도 놓치고 있던 부분이 많이 있었던 것 같아 아쉬움이 남습니다.

그리고, Pseudo Labeling을 이번에 처음 배우고 시도해보았어서, 이를 사용할 때 레이블의 신뢰도를 파악하지 못한 채 어림짐작으로 임계값을 변경하며 시도한 것이 아쉬웠습니다.

한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇일까?

다음 프로젝트에서는 EDA를 세밀하고 신중하게 진행하여, 정확한 문제를 정의하고 분석하여 해결 방법까지 생각해보는 시간을 가져보려고 합니다. 그리고 github PR을 진행할 때, 꼼꼼한 코드 리뷰도 진행하여 놓치는 부분이 없도록 할 것 입니다.