

Object detection Wrap-up Report

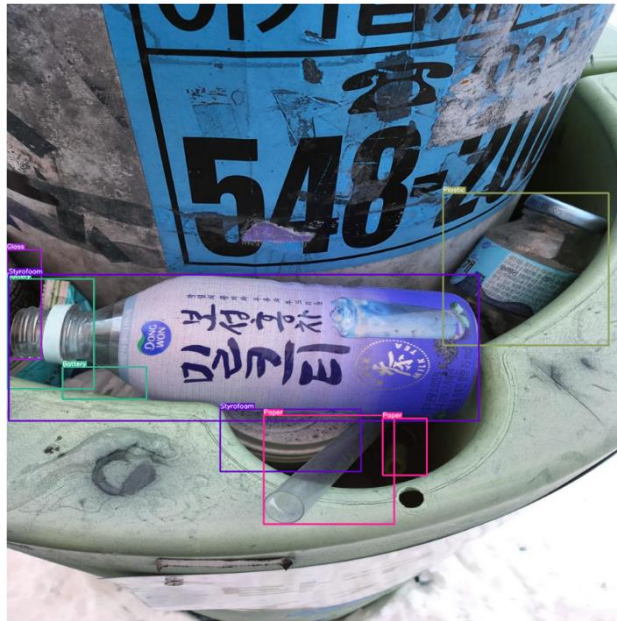
CV-9조 SCV

박상언 송지민 오왕택 이동호 이주헌 지현동

1. 프로젝트 개요

1-1. 프로젝트 주제

우리는 사진에서 쓰레기를 탐지하는 모델을 구축하여 정확한 분리수거를 도울 수 있는 프로젝트를 진행하였다. 데이터셋은 사진과 annotation file로 구성되어 있다. Annotation file에는 사진의 크기, 클래스, 경계 상자의 좌표, 경계 상자의 너비와 높이 등 사진의 정보들이 담겨 있다. 클래스는 쓰레기, 플라스틱, 종이, 유리 등 총 10개의 클래스를 분류하고, 해당 클래스가 존재하는 좌표 값을 구해야 했다.



2. 프로젝트 팀 구성 및 역할

이름	역할
박상언	Detectron2 baseline code 작성, 모자이크 코드 작성, 트랜스포머 모델 실험
송지민	EDA, MMDetection 구현, MMDetection 설명서 작성, 모델 실험(Backbone)

오왕택	EDA, YOLO, MMDetection baseline code 작성, 시각화 코드 작성, 증강 기법 실험,수도 라벨링, 모형 실험, 앙상블 실험
이동호	Torchvision baseline code 작성, Detectron2 실험, Anchor box 실험, WBF code 작성, 모형 실험
이주현	Detectron2 실험, YOLO 실험, Git 현황 관리
지현동	EDA, MMDetection 실험, 증강 기법 실험, Anchor box 실험, 모형 실험, Git 현황 관리

3. 프로젝트 수행 절차 및 방법

3-1. Time-line

기간	수행한 일
1월 3일 ~ 1월 5일	강의 수강, 제공 데이터 및 코드 확인
1월 6일 ~ 1월 10일	Baseline code 작성, Git branch 생성, EDA
1월 11일~ 1월 14일	MMDetection, YOLO로 라이브러리 결정, 시각화 코드 작성, 모형 실험, 증강 기법 실험, Validation set setting
1월 15일 ~ 1월 19일	Hyperparameter tuning, Anchor box 조정, 모형 실험, WBF, Pseudo-labeling

3-2. 프로젝트 수행 방법

1. EDA
2. Baseline code, 사용할 라이브러리 선택
3. Test set과 비슷한 지표를 가지는 validation set 찾기
4. 여러 가지 증강 기법 실험
5. 다양한 모형 실험
6. Anchor box adjusting
7. WBF
8. Pseudo labeling

4. 프로젝트 수행 결과

4-1. 데이터 살펴보기

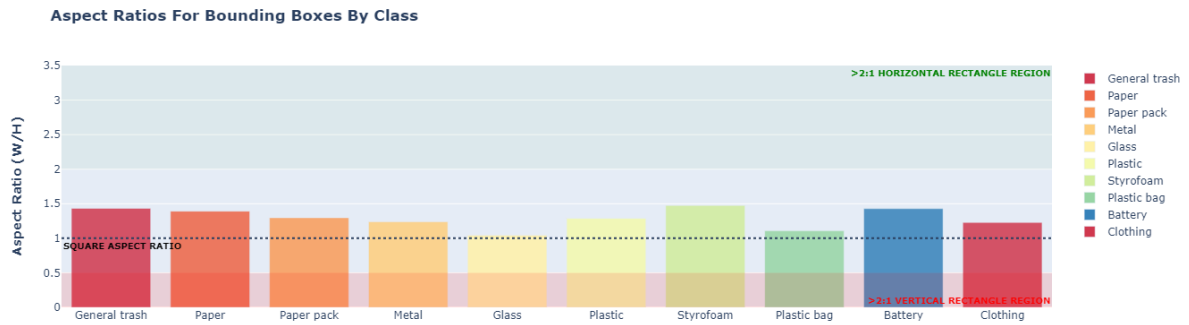
객체 탐지 작업을 수행하기 위해 데이터를 살펴보았다. 사진 속에서 하나의 객체가 존재하는 경우도 있고, 여러 개의 객체가 존재하는 경우도 있었다. 즉, 사진마다 속한 객체의 수가 매우 다양했다. 그리고 몇몇 경계 상자들은 조밀하게 있어 객체들을 분별하는데 쉽지 않았다.

또한 데이터 라벨링 부분에서도 모호한 부분들이 있었다. 쓰레기가 차 있는 일반 쓰레기 봉투의 경우 비닐 봉지로 라벨링이 되어있었다. 이외에도 종이 상자와 우유 갑 등을 분류하는 것이 쉽지 않아 보이는 등 객체가 여러 개가 모여있거나 객체의 크기가 다양하여 탐지하는 것이 쉽지 않아 보였다.



4-2. EDA

EDA를 해보았을 때, 기본적으로 클래스 불균형이 심하다는 것을 알 수 있었다. 특히, 배터리의 경우에는 비율이 매우 적어 해당 클래스를 맞추는 것이 어렵겠다고 생각을 했었다. 또한, 각 정답 경계 상자들의 크기와 비율 등이 매우 다양하여 이를 조정하는 것도 필요하다고 생각했다. 다만, 클래스별로 정답 경계 상자의 비율이 1:2를 넘어가는 경우는 없고 평균적으로 1:1.5 비율인 것을 확인하였다. 그리고 경계 상자의 크기는 의류의 경우 큰 반면에, 배터리의 경우에는 매우 작은 것을 확인할 수 있었다.



또한, 데이터 정규화를 위해서 사진들의 평균과 표준 편차를 계산하였다. 해당 사진의 경우에는 사진의 색깔 또한 정확도를 높이는데 의미가 있다고 생각을 하여서 학습 데이터셋에서 구한 평균과 표준 편차로 정규화를 수행하였다.

4-3. Baseline code 및 기본 모형 실험

수업에서 제공된 baseline code는 총 3 가지 종류였다. 우리는 해당 코드들을 다시 재작성하여서, torchvision, detectron, MMDetection을 모두 실험해보았다. Faster R-CNN ResNet50 FPN을 기본 모형으로 설정하고, 성능적인 차이와 앞으로 실험을 할 때 어떠한 라이브러리가 더 편할지를 생각하였다. 최종적으로는 MMDetection을 사용하기로 결정하였다. MMDetection은 다른 라이브러리에 비해 다양한 모형들을 실험할 수 있고, 증강 기법들도 적용하기 쉽다는 장점이 있었다.

추가적으로 Ultralytics의 YOLO 모형도 같이 실험을 하였다.

4-4. Validation set 찾기

K-fold validation을 통해서 test 점수와 가장 유사하게 나오는 검증 데이터셋을 확인하였다. 그리고 해당 데이터셋을 기준으로 여러 모형 실험의 결과들을 비교하였다.

4-5. 여러 가지 증강 기법 적용

레벨 1 이미지 분류 프로젝트와 마찬가지로 어떠한 증강 기법이 해당 데이터셋에 효과적인지를 실험하였다. 기본적으로는 R-CNN 계열의 모형에서 증강 기법을 실험한 후에, 최종적인 증강 기법을 다양한 모형에 적용하여 성능을 비교하였다. 우리는 mosaic, mixup, clahe, linear transform, color jittering, distortion 등 다양한 증강 기법을 적용하였다. Faster R-CNN로 실험을 하였을 때, 증강 기법이 뚜렷한 성능 향상을 가져오지는 않았다. 하지만 이를 여러 가지 조합을 통해 수행을 하였을 때는 성능의 향상이 존재하였다.

기법	Val mAP	기법	Val mAP
----	---------	----	---------

기본	0.4330	ShiftedScaleRotate	0.4448
Mosaic	0.4090	ColorJittering	0.4354
Clahe	0.4395	JpegCompression	0.4460
Mosaic, Clahe, Rotate, Colorjittering, JpegCompression	0.4530	Mosaic, Clahe, Rotate, Colorjittering, Cutout, JpegCompression	0.4670
정규화만 수행	0.4430	증강 기법 + 정규화	0.4690

그리고 성능이 가장 좋은 방법에서 NMS 대신에 soft NMS를 적용할 경우 성능이 0.4801로 더 뛰어난 성능을 갖게 됨을 알 수 있었다.

4-6. Anchor box adjusting

EDA에서 확인한 부분에 따라 앵커 박스를 조정하면서 실험을 하였다. Aspect ratio를 1:1.5로 조정, Aspect scale 조정, 클래스 불균형 해소를 위한 Focal loss 등 실험을 진행하였지만 기본적인 Faster R-CNN의 성능이 제일 좋았다. 특히, 우리의 모형이 큰 경계 상자는 잘 맞추는 반면에, 중간 크기와 작은 크기를 잘 못 맞춰서 이를 해결하기 위해 앵커 박스의 크기를 작게 만들었지만 큰 성능 개선은 없었다.

4-7. 모형 및 성능 평가

최대한 다양한 모형을 실험을 하여서 어떠한 모형이 가장 좋은 성능을 내는지를 확인하였다. 실험 결과, 모형보다 백본을 어떤 것을 사용하는지가 큰 영향을 끼친 것을 알 수 있었다. 특히, Swin transformer를 백본으로 사용을 하였을 때, 성능이 매우 좋음을 알 수 있었다. 해당 실험을 통해 객체 탐지에서 트랜스포머 기반의 거대 모형이 좋은 성능에 크게 기여한다는 것을 알 수 있었다.

또한, 거대 모형을 사용하니 우려와 다르게 배터리와 같은 클래스의 수가 적은 경우에도 잘 맞추는 것을 확인할 수 있었다.

모형(Backbone)	Val mAP	Test mAP
Faster R-CNN(Swin-L)	0.588	0.5756
Cascade R-CNN(ResNest)	0.497	0.4816
Cascade R-CNN(ConvNext)	0.558	0.5587
Cascade R-CNN(ConvNextv2-large)	0.676	0.676
Cascade R-CNN(HRNet)	0.495	0.5129
Cascade R-CNN(Swin-B)	0.512	0.4887

Cascade R-CNN(Swin-L)	0.596	0.5907
Dynamic Head(Swin-L)	0.62	0.613
DINO(Swin-L)	0.633	0.6103
Grounding DINO(Swin-B)	0.6912	0.6788

4-8. 하이퍼파라미터 튜닝 및 추가 방법

기본적으로는 옵티마이저로 SGD를 사용하였다. 이에 대해서 Adam, AdamW 등을 실험해보았지만, 오히려 SGD보다 성능이 하락하여 SGD를 사용하였다.

성능이 좋은 모형들의 앙상블을 위해서 WBF 기법을 사용하였다. DINO, Cascade R-CNN(Swin-L, ConvNext2-large)를 사용하였을 때, 최종적으로 성능이 나와 기존의 모형들보다 높은 성능을 기록하였다. 추가적으로 수도 라벨링도 시도를 하였지만, 이는 성능 개선에 미비하였다.

5. 자체 평가 의견

5-1. 잘했던 점

- MMDetection이라는 라이브러리를 최대한 활용을 하였다. 기본적인 학습 이외에 다양한 증강 기법 적용, Wandb 적용, 앵커 박스 조정 그리고 백본 교체 등을 실험해봄으로써 MMDetection의 활용 방법을 익힐 수 있었다.
- 이전 프로젝트에 비해 Git 관리는 보다 적극적으로 수행하였다.
- WBF, 수도 라벨링 등 이전에 시도해보지 않은 새로운 방법론들을 적용해볼 수 있었다.

5-2. 아쉬웠던 점

- 실험 결과들을 문서화하지 못한 점이 아쉽다. 다른 팀의 방법론들을 참고하여, 다음 프로젝트에는 바로 적용할 예정이다.
- 프로젝트 중간에는 데이터적인 관점보다는 모형을 돌리기에 급급했던 것 같다. 그리고 해당 모형들을 돌릴 때, 해당 모형의 작동 원리를 공부하지 않고 단지 사용만 한 점이 아쉬웠다.
- Detectron에 다른 모형을 랩핑해서 사용하는 것을 하지 못하였다.

5-3. 프로젝트를 통해 배운 점

- 객체 탐지 작업을 하는 방법을 배웠다는 것 자체만으로 의미가 있었다. 특히, MMDetection 사용법을 배운 것 자체가 의미가 깊었다
- 객체 탐지에서도 역시 증강 기법과 앙상블이 유효하다는 것을 알 수 있었다
- 데이터의 한계를 멀티 모달 또는 zero-shot 기반의 모형들이 해결을 하는 것을 보고, 좋은 모형을 쓰는 것도 중요하다는 점을 배울 수 있었다. 이는 앞으로 다양한 모형을 공부할 동기부여가 될 것이다.

개인 회고

박상언

Baseline code의 detectron2의 구조를 바꿔 프로젝트에 맞게 작성하였다. 작성된 코드를 활용하여 detectron2의 결과를 시각화하여 학습의 결과를 확인하며 과정을 진행하였다. 이후 모델을 학습하는 과정에서 mmdetection의 모델이 더 다양하고 실험가능한 요소가 많다는 사실을 확인하였고 mmdetection으로 라이브러리를 옮겨 나머지 작업을 진행하였다.

mmdetection에서는 모델과 증강기법을 비교하며 성능 향상을 시도하였고 팀원의 결과를 바탕으로 DINO계열을 실험하여 좋은 점수를 얻었다. 또한 hrnet 등 기존에 사용하지 않은 모델을 살펴보고 사용해보며 모델 간의 차이를 확인할 수 있었다.

학습의 미흡함

이번 프로젝트에서는 모델의 동작 원리 이해보다 단순 사용과 점수 향상에 더 중점을 두었다. 학습이라는 목표보다 점수를 더 중요시하면서 공부가 많이 부족했다. 또한 강의 정리에도 다소 부족함이 있어 프로젝트 진행 과정에서 부족함을 많이 느꼈다. 더욱 확실하게 학습을 진행하여 개념 기초를 강화할 것이다.

기여도 부족

시간이 많이 걸리는 모델을 중점으로 사용하여 더욱 적극적으로 파라미터를 튜닝하지 못하였다. 물론 다른 팀원이 적극적으로 실험을 진행하여 좋은 결과를 얻었지만 나 자신의 학습과 경험은 다소 미흡했다. 다음 프로젝트에서는 더욱 적극적으로 참여하여 기여도를 높여야겠다.

송지민

Baseline code로 mmdetection과 detectron2가 제공되었는데, 저는 mmdetection을 사용했습니다. mmdetection의 여러 모델을 config 파일을 이용해 학습하는 과정을 배웠습니다.

Cascade R-CNN과 성능이 좋아보이는 모델 DDQ, DINO 등을 사용했습니다. Cascade의 경우 약 4시간, DINO의 경우 12시간정도 학습시간이 걸렸는데, 프로젝트 초기에는 12시간의 학습이 오래걸린다고 생각하여 Cascade 12 epoch val_mAP50 0.4920 값을 기준으로 여러 모델을 실험했습니다.

DINO를 사용했을 때 점수가 가장 높아서 DINO의 backbone인 SwinTransformer를 다른 모델의 backbone으로 사용했습니다. Cascade R-CNN swin-L을 사용했을때 0.5907 점, Faster R-

CNN swin-L 을 사용했을때 0.5756 점으로 성능을 올릴 수 있었습니다. 또한 학습하는 과정에서 epoch 마다 val_mAP50 이 점진적으로 오르고 떨어지지 않아서 epoch 수를 늘리고 가장 점수가 높았던 모델을 선택했습니다.

최종적으로 Cascade R-CNN swin-L 에 AdamW 옵티마이저와 여러 Aumentation 을 적용하고 20 epoch 모델을 사용하여 mAP 0.6469 점을 얻었습니다.

아쉬운점 1: EDA 활용 부족

모델을 사용하기 전 EDA 를 통해 클래스 별 Annotation 의 개수와 크기 분포, 이미지 별 Annotation 의 개수와 unique 클래스의 개수를 확인했습니다. 막상 EDA 를 통해 얻은 결과를 실험 후에 어떤 클래스를 잘 맞추고, 어떤 클래스를 못 맞추는지, 어떤 크기의 박스를 잘 맞추는지 확인하지 않고 그냥 val50 결과만 확인하고 끝낸것이 아쉽습니다.

아쉬운점 2: Model 선택 실수

cascade R-CNN 에 backbone 네트워크에 swin-L 을 쓴 모델을 주로 썼는데, 1, 2 등 팀을 보니 dino 모델을 주로 사용했습니다. 저도 dino 모델을 사용해 좋은 결과를 얻긴 했었는데 너무 오래 걸려서 다른 모델로 방향을 돌렸던게 아쉽습니다.

아쉬운 점: Model 공부 부족

mmdetection 을 이용해서 object detection 을 하는 방법을 알았고 여러 모델을 써봤지만, 사용한 모델에 대해서 공부하지 않고 무작정 돌리고 결과를 보는데 치중한 것 같아서 반성해야 할 것 같습니다. mmdetection config 각 파라미터가 무엇을 의미하는지 모델의 구조를 알고 쓸 수 있도록 공부해야 할 것 같습니다.

오왕택

Data : Train 과 valid 데이터를 잘 나누어 대회 실험에 활용하고 EDA 를 통해 데이터 분포나 annotation 들을 확인까지 잘하고 문제점을 인지했지만, 데이터 클렌징이나 EDA 의 결과를 대회 실험에 적용시키는 과정이 부족했다고 생각합니다. 다른 팀들의 솔루션을 보며 데이터 클렌징만 하였어도 성능이 큰 폭으로 오를 수 있다는 생각이 들었습니다. 아쉬움을 계기로 남은 프로젝트 에서 데이터 클렌징을 진행하고 EDA 한 결과를 적극적으로 활용해, 더 높은 목표를 달성해보고 싶습니다.

Experiment : MMDetection 의 라이브러리가 잘 구성되어 있어 다양한 실험을 할 수 있었던 것 같습니다. 처음에 Faster R-CNN 을 바탕으로 성능 비교 실험과 다양한 모형들의 하이퍼파라미터 를 바꿔가며 성능을 최적으로 끌어올리기 위해 노력했고 최신 논문을 참고하여 기본으로 주여지지 않는 새로운 backbone 을 활용하여 높은 성능을 확인할 수 있었습니다. 딥러닝을 공부하면서 경험해보지 못한 Pseudo Labeling 과 Weighted Boxes Fusion 기법들을 사용하며 실험에 대한 식견을 넓힌 좋은 경험이었습니다. 모델을 더 완전히 이해해서 하이퍼파라미터를 수정했으면 더 좋

은 성능이 나오지 않았을까 라는 아쉬움도 있었고 시간의 한계에 부딪히기도 했지만, 3 주내에 많은 것들을 시도해본 모두 값진 경험이었습니다.

Github : 저번 프로젝트 보다 Github 의 기능을 적극적으로 사용했지만, 아직 배워야할 부분이 많다고 느꼈습니다. Github Commit 시, convention 을 정하고 사용한 것과 Issue, Pull Request 를 사용해봤지만, 많이 활용하지 못했고 Git Flow 전략 활용이나 Branch 만드는 과정에서 제대로 인지하지 못하고 사용했던 것이 아쉬웠습니다. 앞으로 Github 의 장점들을 살려 많은 협업 기능을 사용하여 익숙해지도록 남은 기간 노력하겠습니다.

마지막으로 모든 팀원들이 object detection에 대한 경험이 많지 않았고 짧은 시간이었지만, 역할 분담한 것들을 잘 수행해주었고 시간 투자 대비 좋은 결과가 나오게 되어 기쁩니다. 다음 프로젝트 때도 각자의 목표를 잘 달성해서 서로의 역량을 향상시키고 높은 수준의 성과를 얻기 위해 노력하겠습니다.

이동호

1. 잘한 점

Detectron2, MMDetection, torchvision code를 실험해보면서 객체 탐지 모형이 어떠한 방식으로 학습이 되는지를 알 수 있었다. 그리고 객체 탐지 library들을 사용해보면서 앞으로 객체 탐지 작업을 맡게 될 경우, 다양한 방식을 통해 실험을 할 수 있다는 점이 고무적이다. 그리고 객체 탐지에서 사용되는 기법들을 공부할 수 있었고, anchor box 조절, WBF 등 다양한 방식들도 공부할 수 있었다. 실제로 anchor box를 조절하면서 성능 변화를 파악해보는 것은 도움이 되었다. 양한 증강 기법들을 사용해보고, mosaic이라는 새로운 증강 기법도 공부할 수 있었다.

2. 아쉬운 점 및 개선 방안

몸 상태가 안 좋아서 실험을 많이 못해본 것이 아쉽다. 빠른 시일 내로 컨디션 관리를 해서 정상적인 몸 상태로 남은 프로젝트를 참여하고 싶다. 해당 부분에서 팀원들에게 미안한 부분이 컸다. 하지만 팀원 분들이 너무 잘해주셔서 고맙다고 이야기하고 싶다.

뒤늦게 데이터 중심의 관점에서 실험을 진행한게 아쉬웠다. 모형만으로도 어느 정도 성능이 나오다보니 근본적인 부분을 놓친 것 같다. 예를 들어, 큰 경계 상자는 잘 맞추지만, 작은 크기를 못 맞췄는데 해당 부분을 위해 anchor box 실험을 좀 더 다양하게 해봐야 했을 것 같다.

Data imbalance problem도 offline augmentation을 통해서 수가 적은 class에 대해서만 진행을 하고, dataset을 증강했으면 더 좋은 성능이 나오지 않았을까 싶다.

다음은 data-centric project인 만큼, data preprocessing에 집중해서 모형이 아닌 다른 관점에서 기계 학습을 바라보는 시각을 넓히고 싶다.

이주현

저는 detectron2에 대한 실험을 진행했습니다. 라이브러리에 대한 이해를 하려고 노력했고, 실제로 모델을 학습시켜보는 것 또한 진행했습니다. 그리고 팀 내에서 mmdetection을 사용하기로 결정된 후에 mmdetection의 여러 모델들에 대해서 팀원들과 의견을 나누고 팀 단위의 실험에 참여하였습니다. 맡은 모델을 실험하고 그에 대해서 기록하고 더 나아가 hyperparameter에 대한 실험에도 참여하였습니다. 또한 yolo v8에 대한 실험을 적극 진행하였습니다.

실험에 대한 성과나 팀 내에서의 성능 향상에 대한 확실한 기여를 했다고는 말하지 못하지만 팀원들에게 질문을 많이 하며 모델 실험에 참여한 것 만으로도 성장이 있었다고 생각합니다. 저번 프로젝트에서는 모델에 집중했지만 이번 프로젝트에서는 데이터에 대한 이해도 해보려고 노력하였습니다. 또한 git 을 통한 협업에 대해 생각해보며 프로젝트를 진행하였습니다.

팀 단위로는 다음 프로젝트에서 git을 더 적극적으로 활용하여 협업에 이용하며, 팀 단위의 스케줄이나 기록을 체계적으로 기록하여 자료를 남기고 싶습니다. 개인적으로는 데이터를 만지는 코드를 직접 만들어 보고 싶고 모델이나 라이브러리의 사용법 뿐만 아니라 좀 더 깊은 이해를 바탕으로 생각하며 프로젝트를 진행 해 보고 싶습니다.

지현동

* EDA 와 데이터 증강기법을 통한 모델 개선.

팀의 역할 분배에 따라 나는 데이터 증강기법과 EDA 를 통한 hyperparameter tuning 으로 모델을 개선하고자 했다. 효율적으로 실험을 진행하기 위해, 작은 모델에서 효과적인 데이터 증강기법은 큰 모델에서도 효과가 있을 것이라는 가정을 바탕으로, 기본 faster R-CNN 모델로 다양한 데이터 증강기법을 시도했다. Albumentation 라이브러리의 데이터 증강기법들과 cutout, mosaic 기법을 적용해서 0.03 향상된 성능 결과를 얻을 수 있었다.

이후 EDA 를 다시 진행하면서, aspect ratio 가 0.8 인 bbox 가 가장 많다는 것을 확인했다. 좀 더 정확한 크기의 anchor box 를 만들면 localization 을 더 잘 잡으므로써 mAP 가 오를 것이라 생각했고 실제 실험에서 미미하긴 했지만 성능이 향상된 것을 확인할 수 있었다. 그 밖에도 데이터셋 이미지의 RGB 평균, 표준편차 값을 구해서 정규화를 진행했고, 기존의 NMS 대신 Soft-NMS 로 대체하여 bbox 의 개수를 늘리는 효과를 통한 mAP 향상을 기대했고, 0.02 향상된 결과를 얻을 수 있었다.

* 실수에서 깨달은 딥러닝 기초 이론의 중요성.

기본 Faster R-CNN 모델을 학습시킬 때, 실수로 클래스의 개수를 실제 클래스의 개수보다 더 많은 상태로 학습시켰던 적이 있다. 하지만 문제가 발생하지 않고 학습이 잘 된 것을 확인할 수 있었다. 충분히 학습이 잘 된 모델은 정답 레이블보다 더 많은 output 노드를 가지고 있더라도, 학습을 통해 쓸모 없는 노드에 대한 가중치가 거의 0 에 수렴되기 때문이다. 실수를 통해 이론을 상기해볼 수 있는 좋은 경험이었다.

*** 문서 작업의 중요성을 느낀다.**

여러 실험이 동시 진행되면서, 어떤 흐름에서 나온 아이디어인지 한 번에 파악할 수 없어서 문서화의 중요성을 느꼈다. 다음 프로젝트부터는 문서화에 신경 쓸 것이다. 특히 가정-결과-분석으로 구성된 실험계획서를 작성할 것이다. 실험계획서를 작성하여 기록하는 것이 실험의 흐름을 이해하고 분석하는데 도움이 되고, 프로젝트 이후에 리뷰 시에도 큰 도움이 될 것 같다.

*** 아쉬운 점 1 : 잘못된 깃 전략**

깃을 활용하기 위해 팀내 컨벤션을 정하고, 이슈 템플릿과 풀리퀘스트 템플릿을 만들었으나 깃을 제대로 활용하지 못했다. 실험이 많은 대회 특성을 고려하지 못한, 깃 브랜치 전략을 사용했던 것이 원인이었던 것 같다. 다음 프로젝트 때는 이 부분을 보완해서 협업으로서의 깃을 제대로 활용할 수 있게끔 주도적으로 나서보려고 한다.

*** 아쉬운 점 2 : 데이터 클렌징 미수행**

EDA 를 통해 잘못된 어노테이션을 확인했지만, object detection 라이브러리를 익히는데 주안점을 두고 약간의 오차는 일반화 성능을 올려 줄 것이라는 기대와 함께 데이터 클렌징을 수행하지 않았다. 직접 실험을 통해 데이터 클렌징 전후의 성능을 비교해보지 못해서 아쉽다. 다음 프로젝트는 Data Centric 프로젝트인 만큼 annotation tool 을 적극 활용해 볼 것이다.