

개인회고_이현진

학습 목표 및 과정 + 모델 개선 방안

학습 목표

1. object detection task에서 필요한 오픈소스 라이브러리인 `mmdetection` 및 `detectron2` 에 대해서 알아볼 수 있다.
2. 이 외의 `yolo family` 의 모델들에 대해서 알아볼 수 있으며 custom dataset에 대해서 training 해볼 수 있다.
3. object detection task에서의 `ensemble 기법` 에 대해서 알아볼 수 있다.
4. 앞으로의 다양한 `object detection task` 에 대해서 적합한 기법 및 model을 선정할 수 있다.

학습 과정

1. mmdetection을 사용해보기 이전에 faster rcnn model의 jupyter notebook 파일을 `python idle 형식` 으로 프로젝트를 구성해보았다. 이를 통해 `faster rcnn` 의 pipeline을 좀 더 깊이 확인해볼 수 있었다.
2. model을 사용하는 데에 있어서 필요한 부분들은 관련 논문 또는 블로그들을 찾아보아
3. 정리하며 학습할 수 있었다.

시도한 점

1. 학습 과정에서의 class별 mAP를 확인한 결과, `general trash` 의 mAP가 현저히 낮은 것을 확인하고 general trash가 50%, 70% 이상인 image에 대해서 `oversampling` 해주었다.
2. image dataset에 bbox가 없는 object들이 많았다. 이는 학습에 있어서 방해가 될 것이라 생각하여 가능한 제거해주고자 하였다. 따라서, image 속 모든 bbox를 포함하는 가장 작은 사각형의 boundary를 찾아 그 외는 black 처리해주었다.
3. 데이터 전수 조사

☐ 불필요 object 제거

image object를 지워주는 오픈 소스인 `cleanup.pictures` 를 사용하여 bbox가 없는 object 중 학습에 방해가 될 것 같은 요소들을 지워주었다.

☐ bbox 정보 추가 및 수정

정확한 class matching이 가능한 object에 대해서는 `roboflow` 를 사용해 bbox를 추가해 주었다. 다음번에는 `COCO annotator` 도 사용해보고 싶다.

4. Augmentation을 사용해보았다.

☐ `RandomRotate90`

쓰레기의 경우 일정한 형태를 지니고 있지 않기 때문에 rotate를 통해 형태가 다양해져 성능이 향상된 것으로 예측 가능하다.

☐ `CLAHE`

CLAHE는 계산 전에 히스토그램 높이에 제한을 뒤서 특정 높이 이상에 있는 pixel 값들을 재분배하는 방법이다. 따라서 이미지 속 밝기의 불균형에 대해서 해결해주어 어두운 부분의 object를 더 잘 잡아내어 성능이 향상된 것으로 예측 가능하다.

☐ `Mosaic`

Mosaic은 dataset에 있는 random한 4장의 이미지를 1장의 고정된 크기의 이미지로 변환하는 augmentation 기법이다. 이는 batch size가 증가하게 되는 효과를 볼 수 있으며 4장의 이미지가 작아지기 때문에 small object에 대해서 성능을 향상시킬 수 있다.

5. Pseudo labeling을 사용해보았다.

- ☐ 리더보드 상 Public mAP 값이 가장 큰 submission을 기준으로 pseudo labeling 진행
- ☐ 처음 모델을 학습시킨 후 inference 결과를 기준으로 pseudo labeling 진행
- ☐ fine tuning weight vs. pretrained weight

fine tuning으로 진행해본 결과 overfitting으로 성능이 떨어졌다.

따라서 pretrained weight로 실험 진행

배운 점

1. object detection을 수행함에 있어서 필요한 `config` 와 `pipeline` 을 작성하는 법을 알 수 있었다.
2. pseudo labeling 시에는 확실한 label만 달아주기 위해서 confidence threshold를 0.8로 정하였다. 하지만, 오히려 학습 성과는 떨어진 것을 확인할 수 있었다. 그 후 pseudo labeling을 시각화 하여 확인해본 결과 detection되지 않은 object들이 많았다. 이런 결과에 따라 confidence threshold를 0.35로 낮춰 pseudo labeling을 확인해본 결과 detection되지 않은 object의 비율이 현저하게 줄어들었고 정확도도 높아졌다. 이를 통해서 모든 실험들을 하기 전에 dataset을 시각화를 통해 정확하게 확인해보아야한다는 것을 알 수 있었다.
3. object detection에서의 ensemble 기법들에 대해서 알아볼 수 있었다.

아쉬운 점 + 앞으로의 방향

1. project 보고서 및 학습 정리를 하는데 있어 수행은 하였지만, 아직 체계적인 형식을 정하지 못한 것 같았다. 다음번에는 일별 project 보고서의 형식을 미리 정해두고 작성하는 습관을 가지도록 할 것이다.
2. 모델을 관리하는 데에 있어서 `wandb` 를 사용함으로써 수월하게 진행이 되었다. 하지만, 다음번에는 현업에서 더 많이 사용되어지는 `mlflow` 를 사용을 목표로 해볼 것이다.
3. 협업하는데 있어서 `git` 의 `branch` 기능을 사용해볼 수 있었지만, 원활한 `baseline` 공유 및 `issue` 사용 이 진행 되지 못한 점이 아쉽다. 우린 Git-flow를 사용하고 있어요 를 참고하여 규칙을 정하고 원활한 협업활동을 진행 해 볼 것이다.
4. cross validation을 진행 후, 각 fold에 대해서 검증을 하지 못하였다. 이 부분으로 인해서 test dataset가 가장 적합한 fold를 찾지 못한 점이 점수를 최대치로 올리지 못한 이유가 될 수 있을 것 같다. 다음번에는 모든 fold를 실험을 통해 앞으로의 실험 fold를 정해볼 것이다.
5. object detection에서의 평가 방법 중 하나인 `mAP` 에 대해서 알 수 있었다. 하지만, classification에서의 평가 방법인 focal에 적합한 focal loss처럼 mAP에 적합한 loss를 찾아보지 못해던 점이 아쉽다.