|  |  |  |  |
| --- | --- | --- | --- |
|  | Yolov8 | Yolov7 | Retina(mmdetect) |
| lr | 0.01 | 0.01 | 0.00125 |
| momentum | 0.937 | 0.937 | 0.9 |
| weight\_decay | 0.0005 | 0.0005 | 0.0001 |
| Type | SGD | SGD | SGD |
| Epoch | 50 | 100 | 12기준 0.6980 |
| 체감시간 | 40분 | 2시간 | 1시간 |
| 전이학습 | O | O | O |
| Batch size | 16 | 16 | 32 |
| mAP | 0.752 | 0.802 | 0.698 |

Object detection 연습

With yolov7,8 and mmdetection(model :retina)

사용 용이성 : Yolo 1stage 방식 사용 편리함

Mmdetection retina 2stage 방식 mmdetection 사용이 편리하진 않으나 범용성이 좋고 custom dataset, custom model 구성 가능하도록 지원

기본적으로 정해진 이미지와 annotation이 준비되었다고 했을 경우

Yolo의 경우 자동으로 annotation 내의 class 분류 및 자동으로 트레인셋을 구성해줌

반면에 mmdetection의 경우 지원하는 데이터셋에 맞는 데이터셋을 오버라이딩 하여 사용하는 방식

하지만 mmdetection의 경우 상속받는 모델을 기준으로 재정의 하기 때문에 파이프라인이나 모델의 백본, 넥 , 헤드 부분이 어떤 방식으로 구성되어 있는지 쉽게 파악할 수 있다. 이에 따라 다양한 모델을 경험하고 연습한다면 mmdetection의 각 model 을 훑어 보며 파악할 수 있다. 또한 mmdection은 이전의 모델이 지원하지 않았던 loss 값들을 개선하여 box loss 등 다양하게 제공 하고 있다는 점에서 좋은 학습용 툴이라고 할 수 있다.

Yolo의 경우에는 유저 친화적이며 dataset이 준비되어 있을 경우 커맨드라인으로 train.py 를 통해 바로 결과를 볼 수 있게 해준다. 하지만 모델 튜닝이나 optimizer 수정에서 간단한 수정만 가능하며 자체 smart optimizer 사용으로 깊이 들어가서 찾아야 한다 .

Yolo 8 predict 사진





2. 아래 질문에 답변 작성하기 (성능 개선 과정에 대해 자유롭게 작성 가능)  
      Q1) 어떤 옵티마이저, 로스 함수를 사용했는지?

SGD , box loss CLS loss   
      Q2) 처음 시도했던 Network Architecture는 어떤 종류인지?

Yolo,   
      Q3) 이후 시도해봤던 Network 들은 무엇인지?

여러가지  
      Q4) 과대적합을 피하기 위해 했던 작업들은 무엇인지?

과대적합을 피할만큼 많은 자료는 없었고 과소적합을 피하기 위해 에폭을

최대한 늘리며 데이터를 확인했다  
      Q5) 중요 하이퍼파라미터 어떻게 설정했는지? 이유?  
           위 표 참고

2-1. 추가로 그동안 학습했던 모델별 하이퍼파라미터와 성능 간략히 적어주시면 더 좋습니다!  
내용들을 슬랙방에 올려주시거나 깃허브 개인 폴더의 readme 파일에 작성하셔도 좋을 것 같습니다.  
(readme에 작성하는 게 좋을 것 같기도 합니다)작성해주신 것 토대로 내일 pdf 파일 만들어서 발표해볼까 합니다 ^\_^이렇게 개인 폴더 만들어주심 좋을 것 같아요!!