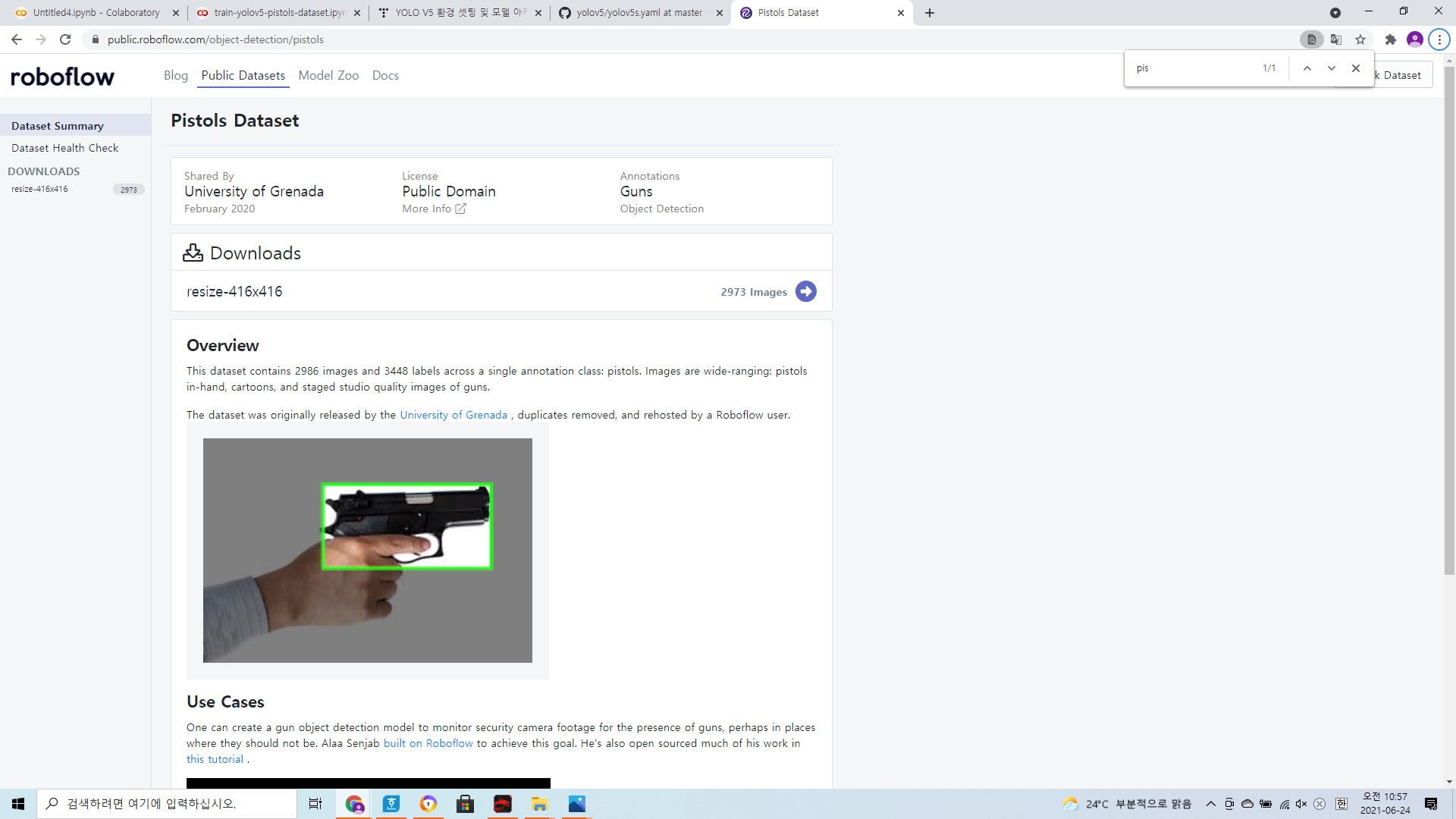
Yolov5 – Objectdetection(pistol)

데이터 수집



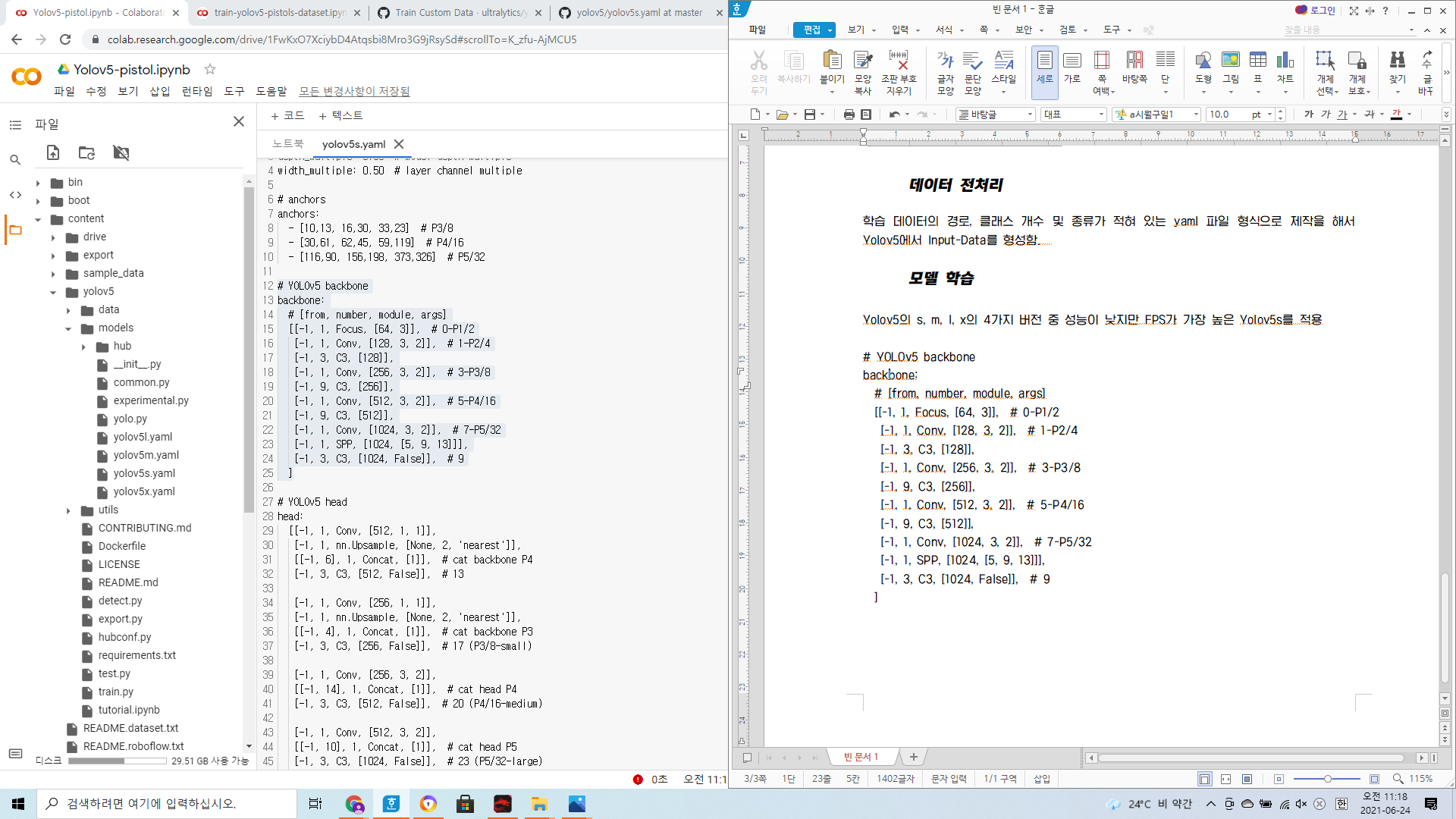
<https://public.roboflow.com/object-detection/pistols> 2971개의 pistol 이미지 파일과 bounding box의 좌표.txt 파일을 roboflow에서 얻음.

데이터 전처리

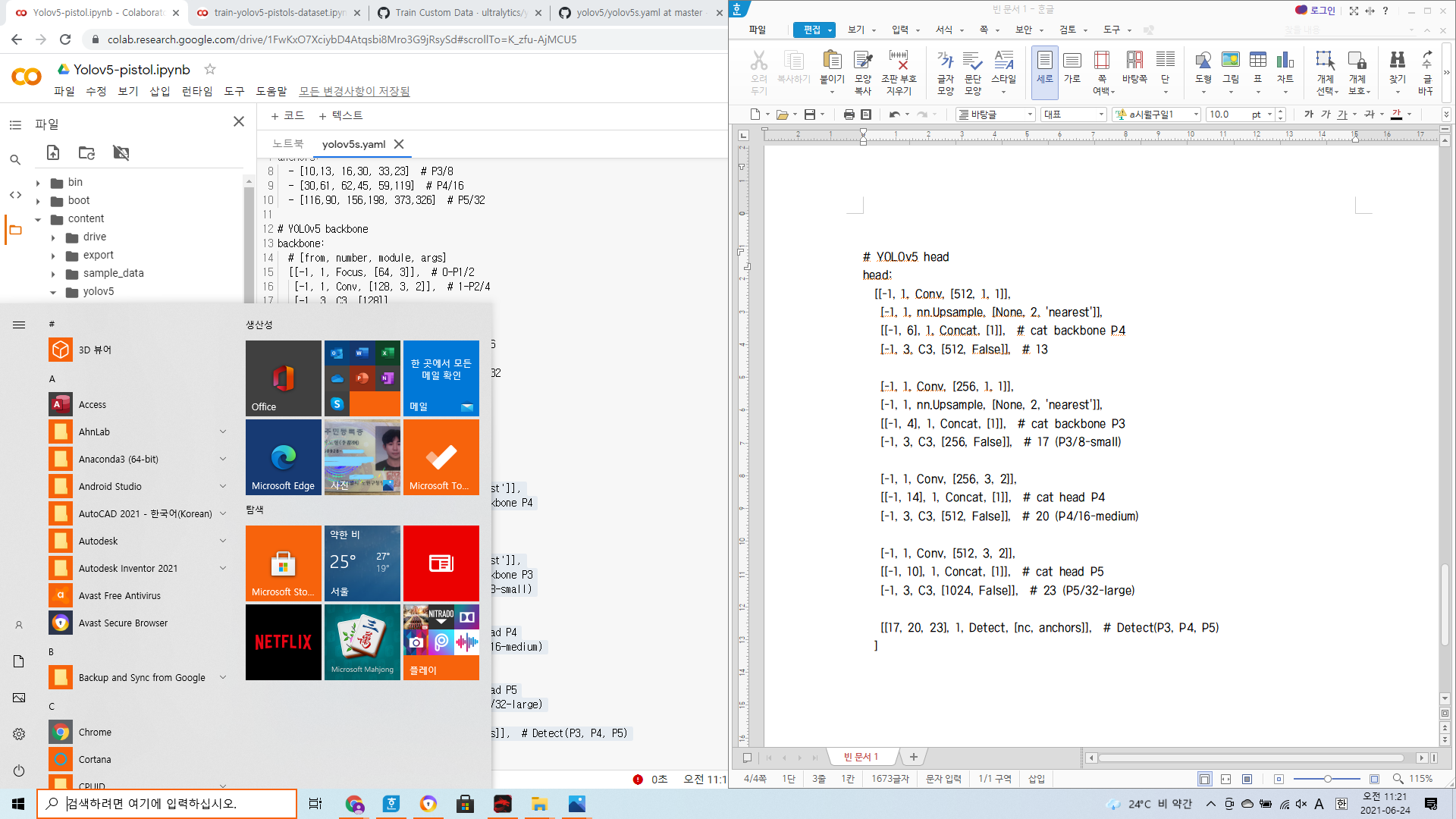
학습 데이터의 경로, 클래스 개수 및 종류가 적혀 있는 yaml 파일 형식으로 제작을 해서 Yolov5에서 Input-Data를 형성함.

모델 학습

Yolov5의 s, m, l, x의 4가지 버전 중 성능이 낮지만 FPS가 가장 높은 Yolov5s를 적용



Yolov5의 특징맵을 뽑는 backbone-net은

Focus에서 batch, channel, width, height를 바꿔주고 Conv는 일반적인 Conv+batch\_norm 레이어를 사용.

활성화 함수로는 Hard swish로 relu보다 좀 더 딥한 딥러닝에서 좋은 성능을 보인다고 함.

C3은 Bottleneck으로 ResNet과 동일하게 사용한 skip-connection이 있는 블록임.

SPP는 output을 27의 사이즈로 고정된 1차원 형태의 배열을 생성하고 Full-connet형식으로 입력으로 들어가게 해줌.

Yolov5의 위치를 뽑는 head에서는

Upsample로 피쳐맵의 각 배열을 늘려주고

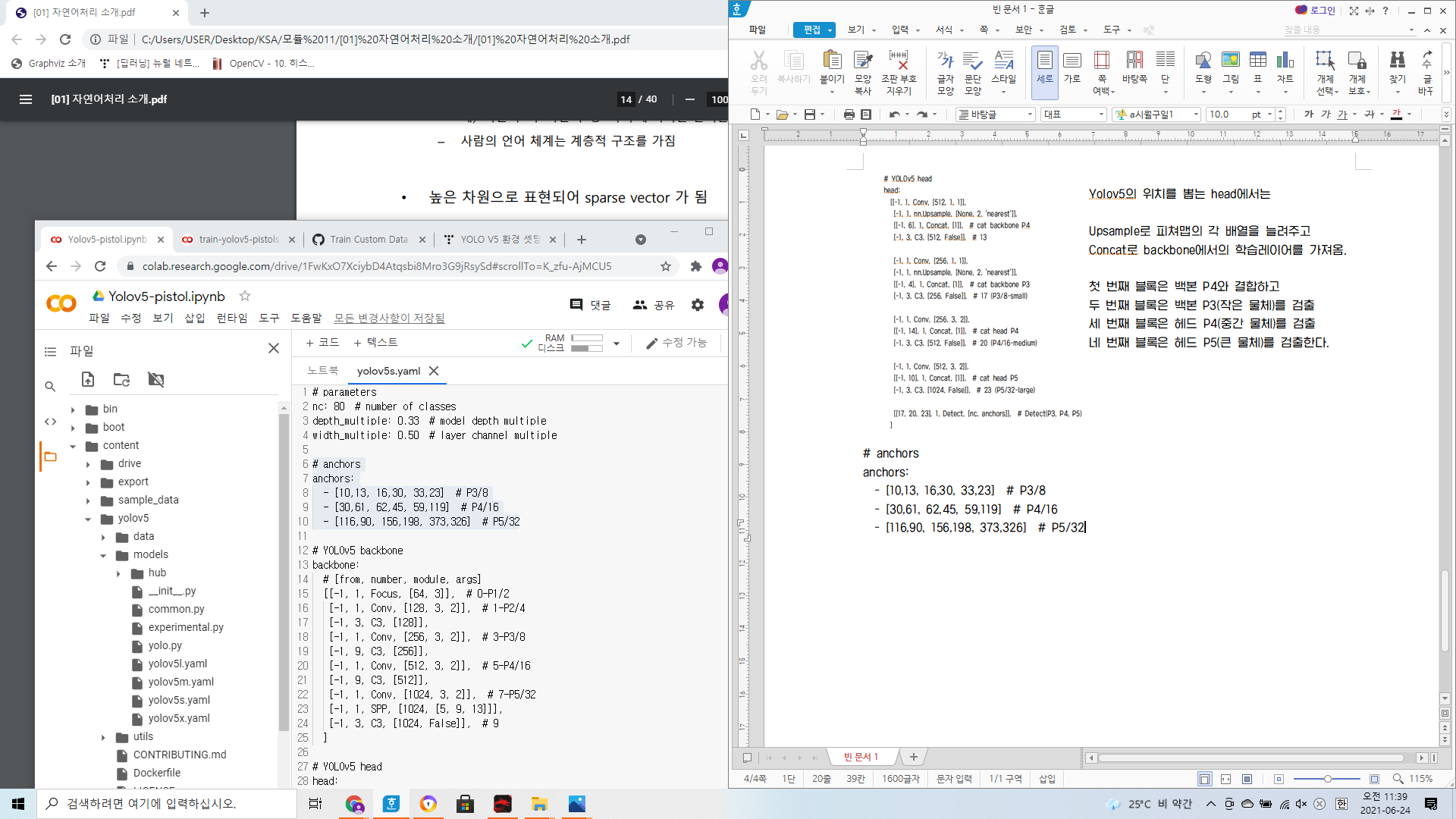
Concat로 backbone에서의 학습레이어를 가져옴.

첫 번째 블록은 백본 P4와 결합하고

두 번째 블록은 백본 P3(작은 물체)를 검출

세 번째 블록은 헤드 P4(중간 물체)를 검출

네 번째 블록은 헤드 P5(큰 물체)를 검출함.



세 개의 anchor box는 다음과 같이 구성함.

작은 객체

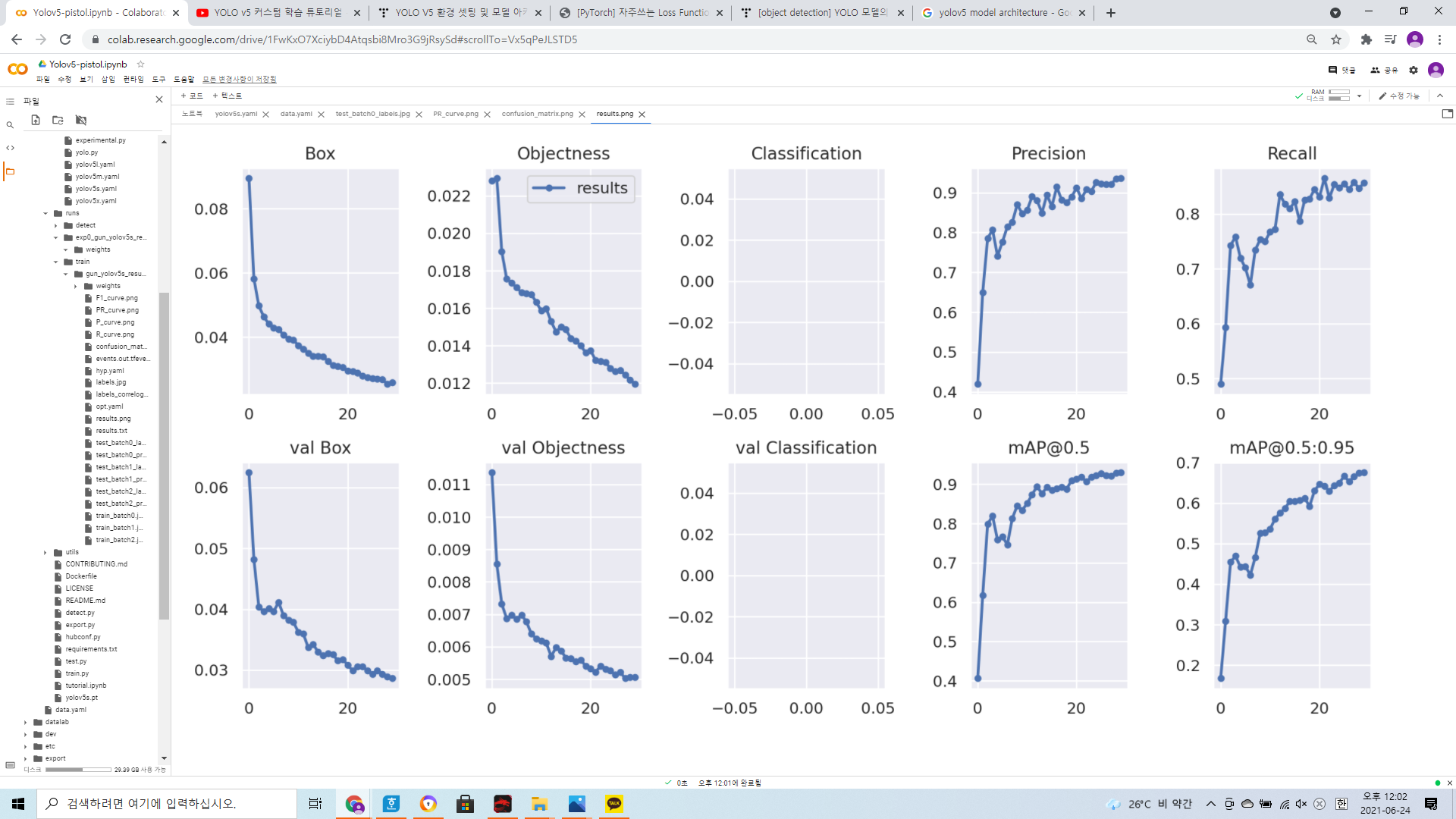
중간 객체

큰 객체

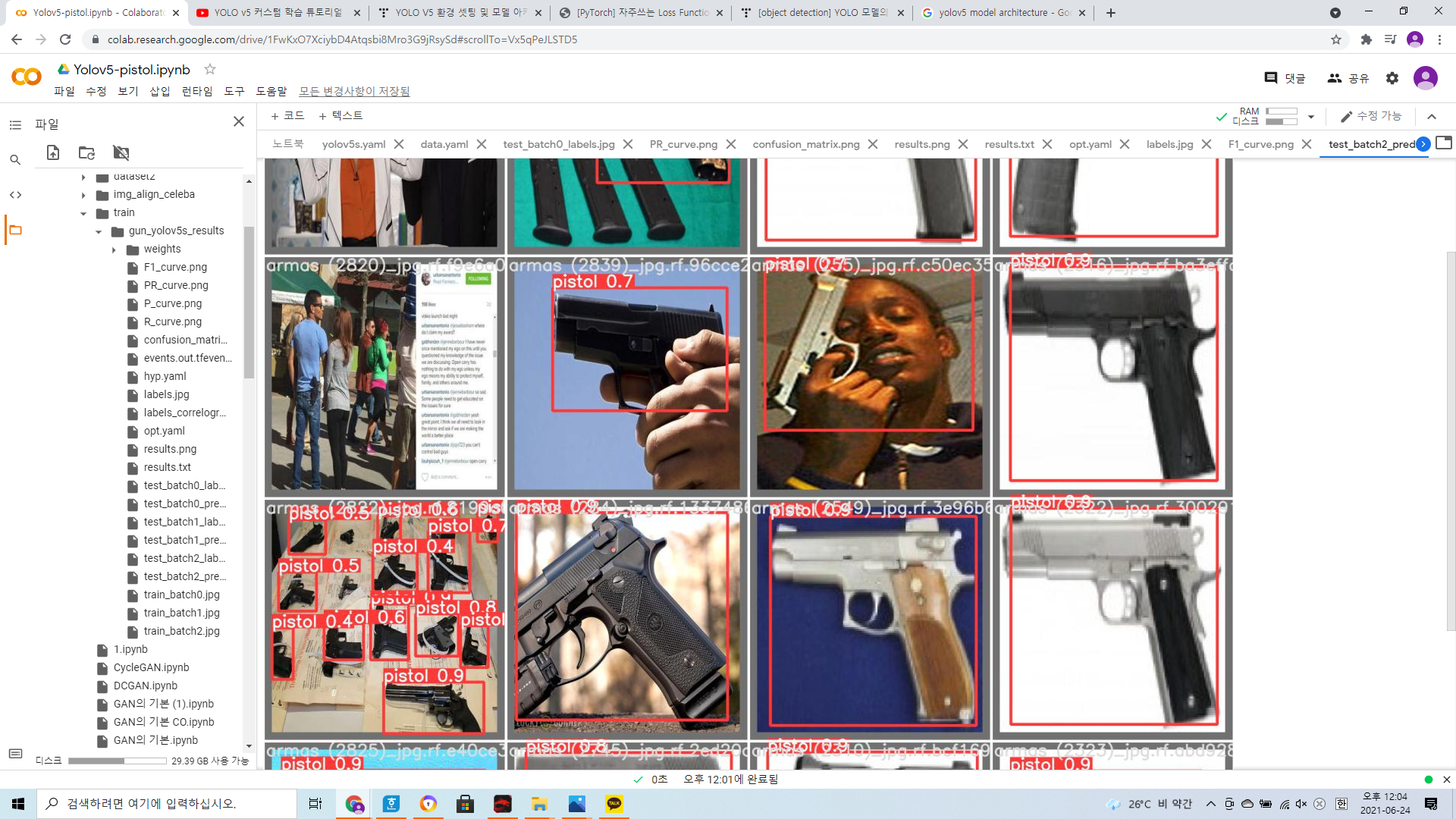
Loss

loss의 경우 1. giou\_loss, IOU를 활용한 Loss와 2. BCEwithLogitsLoss를 사용한다.

Result



단일 Class에 대해서 Object-Detection이 Box loss와 Object loss가 학습 진행에 따라 감소했고 정밀도와 재현율 또한 학습 진행에 따라 증가함.



Test Set의 학습 결과 이미지파일.