Yolo5 사용자 데이터 학습 및 추론 방법

참조 싸이트 <https://dohyeon.tistory.com/38>

본 문서는 순수 python으로만 사용 적용되어 있으나 anaconda를 사용하는 유저도 대부분 내용이 동일하므로 쉽게 따라 할 수 있을 것이다.

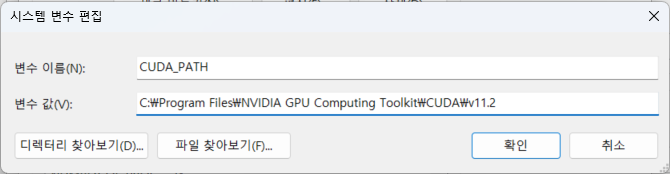
# 실행 환경

os : 윈도우11

Python: 3.7.9 (본인의 python은 3.7 이나 3.8이 권장됨)

Pytorch : torch.\_\_version\_\_'1.11.0+cu113'

환경 변수



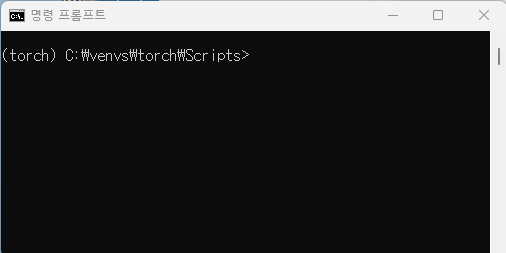
# 가상 환경

C:\venvs에 torch라는 가상환경을 만든다. 아래와 같이 명령을 준다.

Python –m venv torch

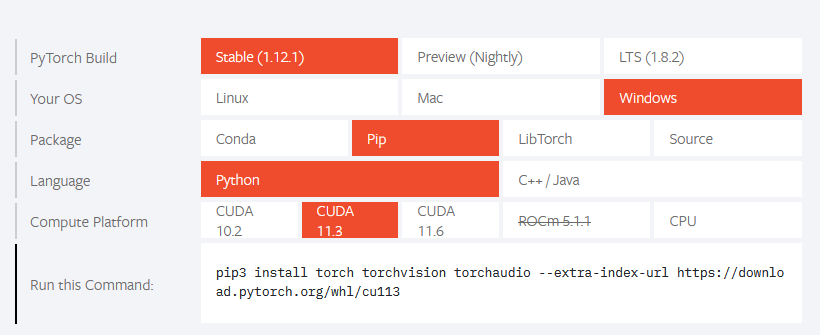
cd torch\scripts

activate



Torch 가상 환경 진입

<https://pytorch.org/> 싸이트에서



자신의 환경에 맞게 선택하면 pytorch 설치 명령이 생성 되고 이를 복사하여 command 창에서 실행하면 pytorch가 설치된다.

# Yolo5 설치

Github에 <https://github.com/ultralytics/yolov5>

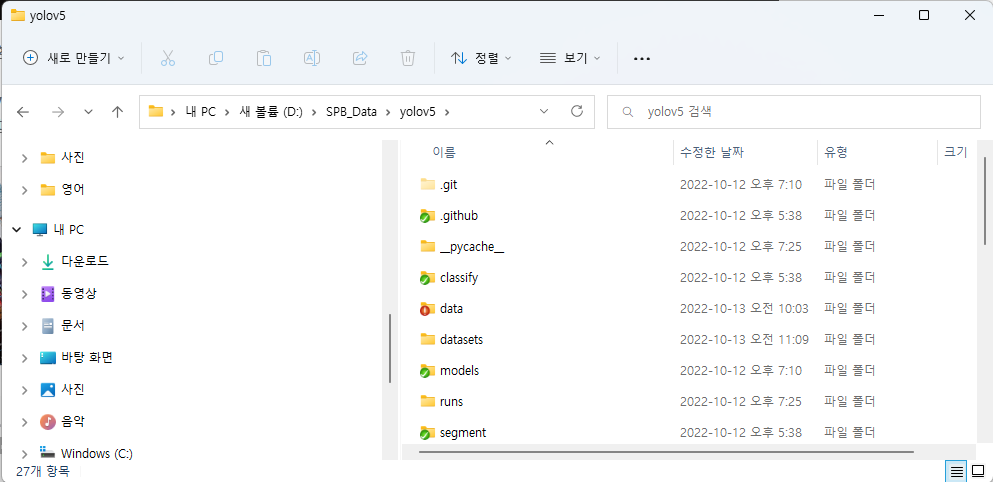
기본 소스가 올려져 있다.

자기가 다운받으려는 위치로 가서

git clone https://github.com/ultralytics/yolov5

을 실행한다.

설치 된 모습



pip instasll requirement.txt

환경이 다를 수 있기 때문에 위 명령을 실행 하지 말라고 하지만 난 그냥 실행 함.

# Yolo5 custom data 훈련, 추론

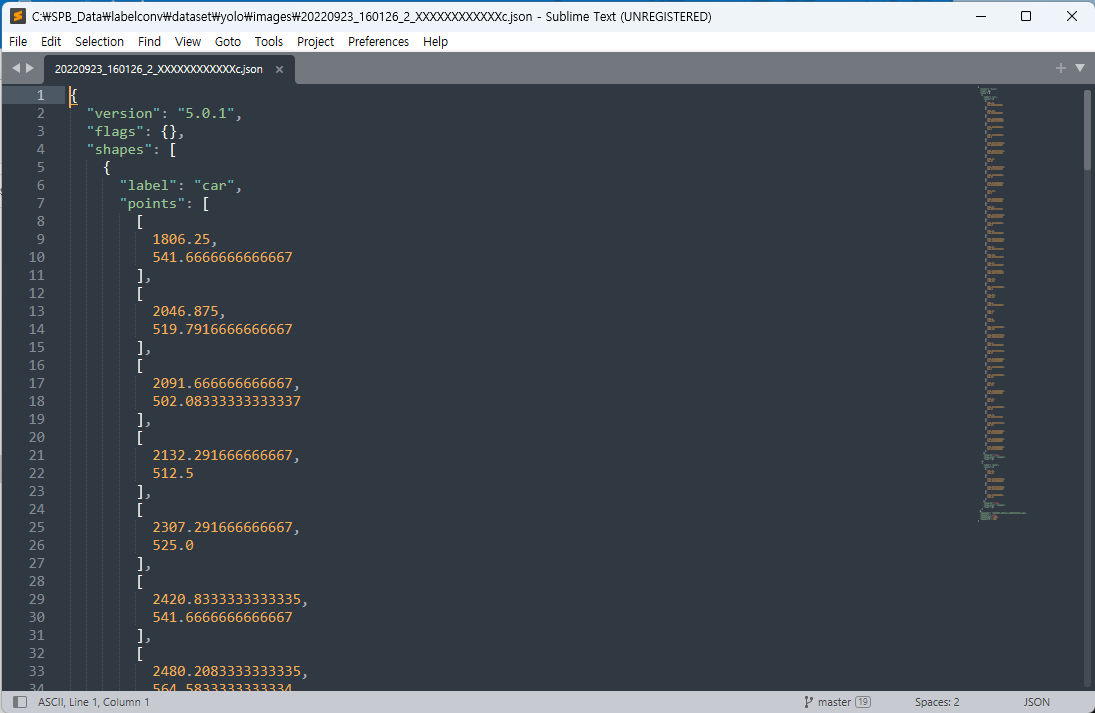
# labelme를 yolo버전 변환

labelme로 라벨링을 하면 labelme 고유 형식의 json으로 저장된다. 간단한 스크립트를 통하여 yolo 버전으로 변경 가능하다.

LPR\_Car-Plate\_Labels.txt 파일에 아래와 같이 레이블 명이 작성되어 있다.

1. car,승용
2. truck,트럭
3. bus,버스
4. bike,오토바이
5. bicycle,자전거
6. plate,번호판
7. human,사람

또한 json은 이에 따라 생성되어 있다.



이를

<http://175.123.189.22:32000/ksyoon/labelconv.git>

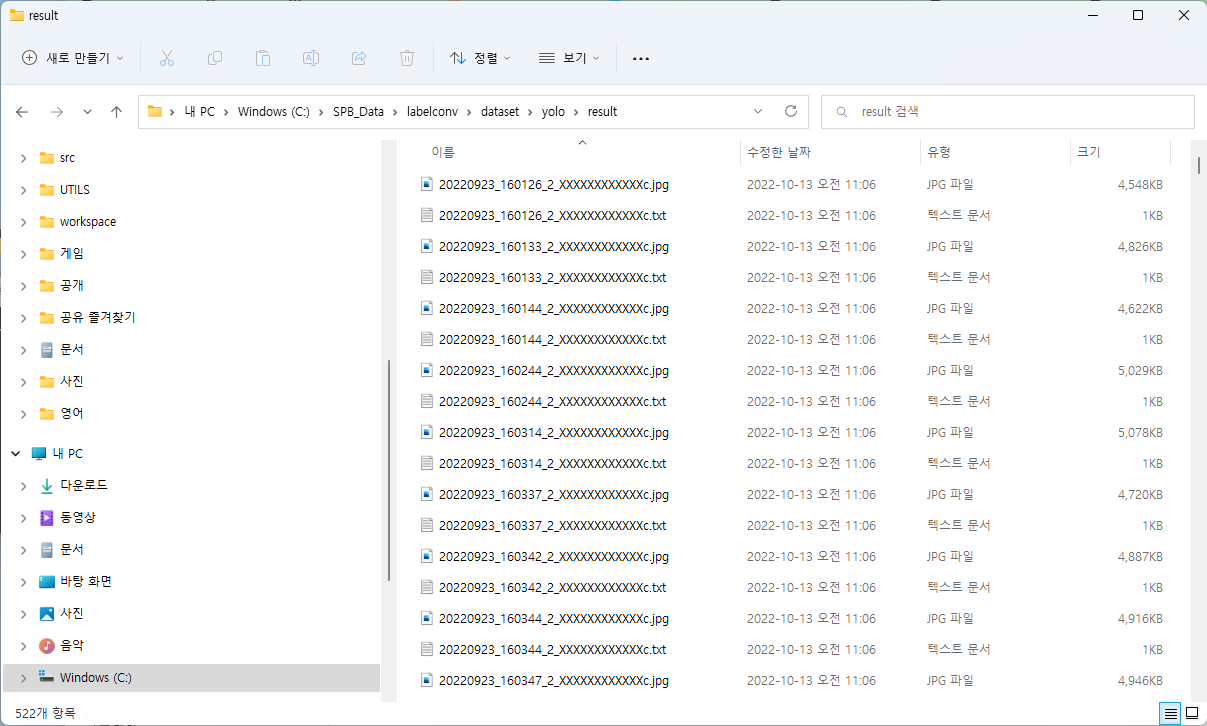
에 있는 json2yolo.py 를 실행하면

json을 yolo용 포멧으로 변경 해준다.

영상 및 json을 기본으로

C:\SPB\_Data\labelconv\dataset\yolo\images 에 넣어 두면

C:\SPB\_Data\labelconv\dataset\yolo\result 에 yolo 레이블이 생성 된다.



예)

1. 0 0.658042 0.456136 0.318579 0.455909
2. 5 0.646976 0.643636 0.090087 0.031818

Yolo 포멧에서 첫째 숫자는 class 번호이고 나머지 4개의 숫자는 박스 좌표이다.

박스는 영상의 넓이 높이를 1로 normalize 했을 때의 기준으로 한 크기이며 첫 두 좌표는 박스의 중심점 x,y 이고 그 다음 두 좌표를 중심점을 기준으로 한 width, height 이다.

# custom yaml 생성

yolov5\data 디렉토리를 보면

coco128.yaml이 이는데, 이를 수정하여 진행한다.

coco128.yaml을 jwisobjs.yaml이란 이름으로 바꾸고,

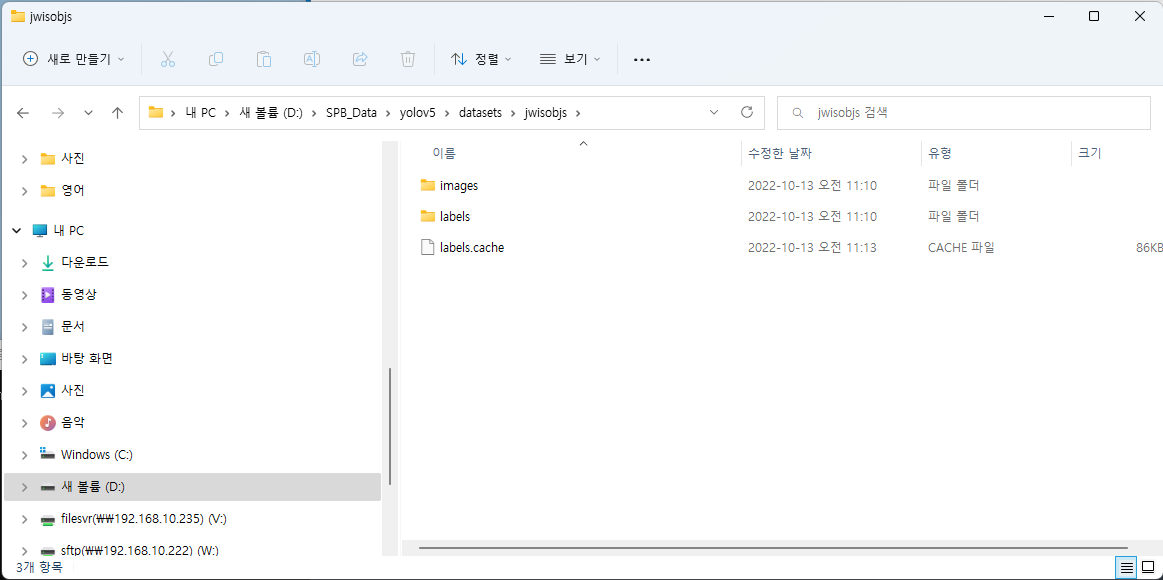
1. # YOLOv5 🚀 by Ultralytics, GPL-3.0 license
2. # COCO128 dataset https://www.kaggle.com/ultralytics/coco128 (first 128 images from COCO train2017) by Ultralytics
3. # Example usage: python train.py --data coco128.yaml
4. # parent
5. # ├── yolov5
6. # └── datasets
7. #     └── coco128  ← downloads here (7 MB)

10. # Train/val/test sets as 1) dir: path/to/imgs, 2) file: path/to/imgs.txt, or 3) list: [path/to/imgs1, path/to/imgs2, ..]
11. path: datasets  # dataset root dir
12. train: jwisobjs/images  # train images (relative to 'path') 128 images
13. val: jwisobjs/images  # val images (relative to 'path') 128 images
14. test:  # test images (optional)
16. # Classes
17. names:
18. 0: car
19. 1: truck
20. 2: bus
21. 3: bike
22. 4: bicycle
23. 5: plate
24. 6: human

class name을 자기가 원하는 대로 수정하고, path를 수정한다.

본인은 D:\SPB\_Data\yolov5\datasets\jwisobjs 을 만들고

images (영상저장)와 labels(레이블 저장)를 만들었다.



참고)

위와 같이 나머지 경로는 다 주석처리하고

train: 위 사진 참조

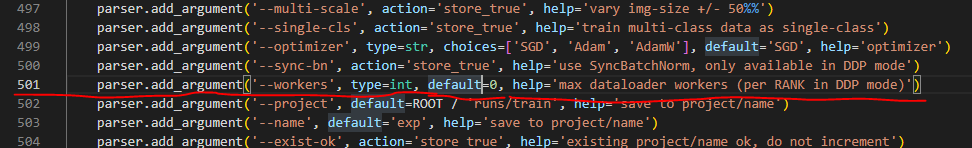
val: 위 사진 참조

test: 주석처리 할것

nc: 내가 학습시키고자 하는 class 개수

names: 내가 학습시키고자 하는 class 이름

# train.py 수정



yolov5 폴더 내 train.py 파일 들어가서 num\_worker default=0 으로 수정

위 line의 default=8로 되어있을 텐데 0으로 수정한다.

(위 과정을 진행하지 않으면 페이징 오류 발생!)

# training

yolo5가 있는 최상위 디렉토리에서 아래 명령을 수행한다.

python train.py --img 640 --batch 16 --epochs 100 --data jwisobjs.yaml --weights yolov5s.pt --device 0

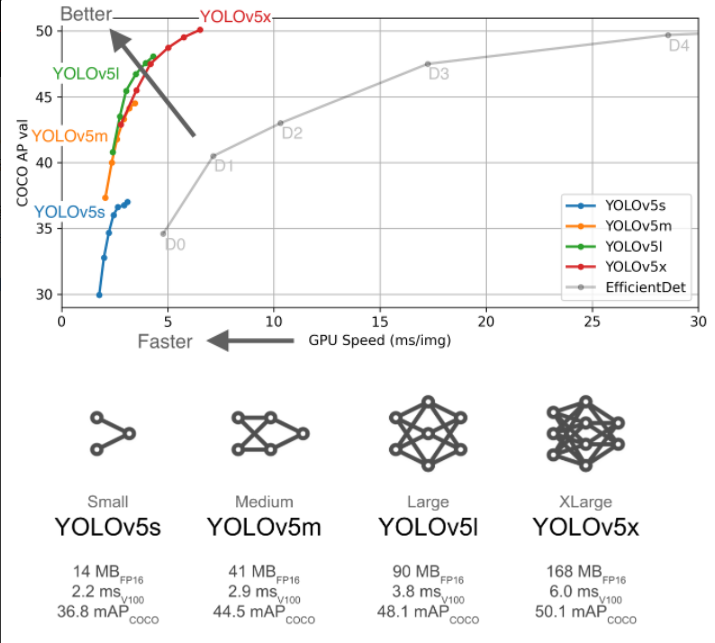
학습이 완료되면 yolov5 폴더 내에 runs 라는 폴더가 생성된다.

yolov5/runs/train/ 폴더에 들어가보면 exp2, exp3 등 트레이닝 횟수대로 폴더가 생성되어 있는데

최신 폴더에 들어가보면 weight 폴더에 best.pt 파일을 weight file로 쓰면 된다!

# 사용할 pre-trained 모델 확인

yolov5에는 n,s,m,l,x 5가지 버전이 있다. n부터 가장 가벼운 모델 x가 가장 무거운 모델이다. n는 모델이 가볍지만 mAP가 가장 낮고 x는 모델이 무겁지만 mAP가 높다. 이번에 yolov5s 모델을 사용 할 것이다.



# inference (추론)

python detect.py --source=d:/test/ --weight=D:\SPB\_Data\yolov5\runs\train\exp4\weights/best.pt --img 416 --conf 0.5 --save-txt

--img : 네트워크에 입력되는 이미지 크기

--batch : 배치 크기  
--epochs : epoch 횟수  
--data : 학습에 참조될 yaml file   
--weights : pre-trained model에 전이학습할 경우 (yolov5s.pt 등을 넣어주면 자동으로 다운로드 됨)  
--name : 학습된 모델의 저장이름

--source에서 각각의 파일 이름을 지정해도 되지만 여러 영상이 있으면 디렉토리 이름만 지정하면 영상들을 알아서 읽어 온다.

추론이 끝나면 yolo5\runs\detect 아래에 순서별로 폴더에 영상라 레이블이 저장된다.