

1. 라이브러리 설치

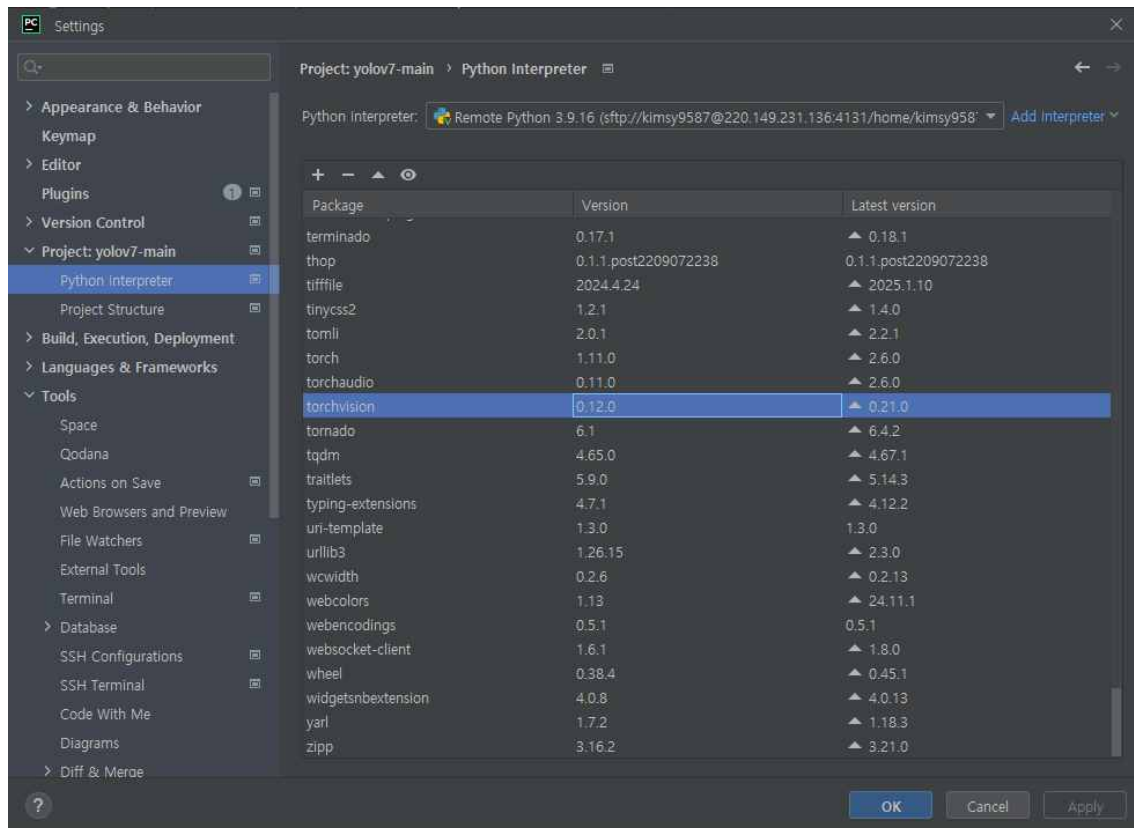
YOLOv7 모델을 돌리기 전에 파이썬을 설치해주고, 필요한 라이브러리들을 설치해주는 것이 중요해.

나는 Python 3.9.16 버전을 설치했고, Pytorch와 torchvision은 각각 1.11.0, 0.12.0을 설치했어. (굳이 이 버전을 써야만 한다 이런 것은 아님! 그렇지만 버전이 안 맞아서 오류가 뜨는 경우도 종종 있긴 해.)

<https://pytorch.org/get-started/previous-versions/>

이 사이트에서 Pytorch와 torchvision을 설치할 수 있는 코드가 제공되어 있어.

그리고 내가 보내준 폴더에 requirement.txt가 있을 텐데, 거기에 있는 라이브러리들은 모두 설치를 해주는 것이 좋아. (이건 꼭 기억하도록 해!)



2. 본격적으로 YOLOv7 튜토리얼

내가 보내준 수많은 파일들 중에서, YOLOv7 모델을 학습시킬 때는 train.py, 테스트시킬 때는 test.py, visualize할 때는 detect.py를 쓰면 돼. 주로 이 3개의 파일만 쓰게 될거야.

```

528     parser = argparse.ArgumentParser()
529     parser.add_argument('--weights', type=str, default='yolo7.pt', help='initial weights path')
530     parser.add_argument('--cfg', type=str, default='', help='model.yaml path')
531     parser.add_argument('--data', type=str, default='data/coco.yaml', help='data.yaml path')
532     parser.add_argument('--hyp', type=str, default='data/hyp.scratch.p5.yaml', help='hyperparameters path')
533     parser.add_argument('--epochs', type=int, default=300)
534     parser.add_argument('--batch-size', type=int, default=16, help='total batch size for all GPUs')
535     parser.add_argument('--img-size', nargs='+', type=int, default=[640, 640], help='[train, test] image sizes')
536     parser.add_argument('--rect', action='store_true', help='rectangular training')
537     parser.add_argument('--resume', nargs='?', const=True, default=False, help='resume most recent training')
538     parser.add_argument('--nosave', action='store_true', help='only save final checkpoint')
539     parser.add_argument('--notest', action='store_true', help='only test final epoch')
540     parser.add_argument('--noautoanchor', action='store_true', help='disable autoanchor check')
541     parser.add_argument('--evolve', action='store_true', help='evolve hyperparameters')
542     parser.add_argument('--bucket', type=str, default='', help='gsutil bucket')
543     parser.add_argument('--cache-images', action='store_true', help='cache images for faster training')
544     parser.add_argument('--image-weights', action='store_true', help='use weighted image selection for training')
545     parser.add_argument('--device', default='', help='cuda device, i.e. 0 or 0,1,2,3 or cpu')
546     parser.add_argument('--multi-scale', action='store_true', help='vary img-size +/- 50%')
547     parser.add_argument('--single-cls', action='store_true', help='train multi-class data as single-class')
548     parser.add_argument('--adam', action='store_true', help='use torch.optim.Adam() optimizer')
549     parser.add_argument('--sync-bn', action='store_true', help='use SyncBatchNorm, only available in DDP mode')
550     parser.add_argument('--local_rank', type=int, default=-1, help='DDP parameter, do not modify')
551     parser.add_argument('--workers', type=int, default=8, help='maximum number of dataloader workers')
552     parser.add_argument('--project', default='runs/train', help='save to project/name')
553     parser.add_argument('--entity', default=None, help='W&B entity')
554     parser.add_argument('--name', default='exp', help='save to project/name')
555     parser.add_argument('--exist-ok', action='store_true', help='existing project/name ok, do not increment')
556     parser.add_argument('--quad', action='store_true', help='quad dataloader')
557     parser.add_argument('--linear-lr', action='store_true', help='linear LR')
558     parser.add_argument('--label-smoothing', type=float, default=0.0, help='Label smoothing epsilon')
559     parser.add_argument('--upload_dataset', action='store_true', help='Upload dataset as W&B artifact table')
560     parser.add_argument('--bbox_interval', type=int, default=-1, help='Set bounding-box image logging interval for W&B')
561     parser.add_argument('--save_period', type=int, default=-1, help='Log model after every "save_period" epoch')
562     parser.add_argument('--artifact_alias', type=str, default="latest", help='version of dataset artifact to be used')
563     parser.add_argument('--freeze', nargs='+', type=int, default=[0], help='Freeze layers: backbone of yolov7=50, first3=0 1 2')
564     parser.add_argument('--v5-metric', action='store_true', help='assume maximum recall as 1.0 in AP calculation')
565     opt = parser.parse_args()

```

이 코드들은 YOLOv7 train.py의 argument 설정해주는 부분인데, weights, cfg, data, hyp 부분은 너가 직접 경로를 설정해주고, project, name 부분은 학습하고 나서 결과가 저장되는 부분이니 너가 직접 이름을 설정해주면 돼. epochs나 batch size의 경우는 너 취향껏 설정하면 돼. 디폴트 값 써도 되구.

3. YOLOv7 데이터 형태

YOLOv7에서는 주로 yaml 파일 형태를 사용해.

```
! custom_data.yaml X
C: > yolov7 > yolov7-custom > data > ! custom_data.yaml
1 |
2 train: ./data/train
3 val: ./data/val
4
5 # number of classes
6 nc: 1
7
8 # class names
9 names: [ 'Johnny Depp' ]
10
```

이 사진은 yaml 파일 형식이고, test까지 하고 싶다면 세 번째 줄 val 밑에 test 경로도 추가해야 해. class 개수와 class 이름도 직접 잘 세팅해주고.

train, valid, test data 모두 다 images와 labels가 있어야 해. images는 raw data고, labels는 객체 번호, x, y, w, h로 이루어진 txt 데이터일 거야.