

모델 학습 프레임워크 데이터셋 지원

지원 데이터 셋 (기존 YOLOv7 공식 github로 배포된 프레임워크에서 정상적으로 수행 확인)

- MS COCO 2017 (학습 클래스 선택 가능)
- PASCAL VOC 2012 (학습 클래스 선택 가능)
- VISDRONE 2019 (학습 클래스 선택 가능)
- Cityscape (vehicle, person, traffic sign, traffic light 4개 클래스 학습 지원)

프레임워크 사용법

모델 학습 방법

1) `train.py` 파일 내부 `argparse`의 `--data` 속성에 학습에 사용할 데이터 셋 지정

지원하는 데이터셋 yaml 파일 목록

- `data/coco.yaml`
- `data/pascal.yaml`
- `data/visdrone.yaml`
- `data/cityscapes.yaml`

`argparse`에 학습할 데이터셋 yaml 파일을 지정한 뒤 다음 명령어로 학습 수행

```
python train.py
```

위 명령어를 실행하게 되면 커스텀 데이터 셋 구성을 위한 전처리 과정이 선행되고, 작업이 완료되면 구성된 커스텀 데이터 셋으로 모델 학습이 시작된다

프레임워크 주의사항

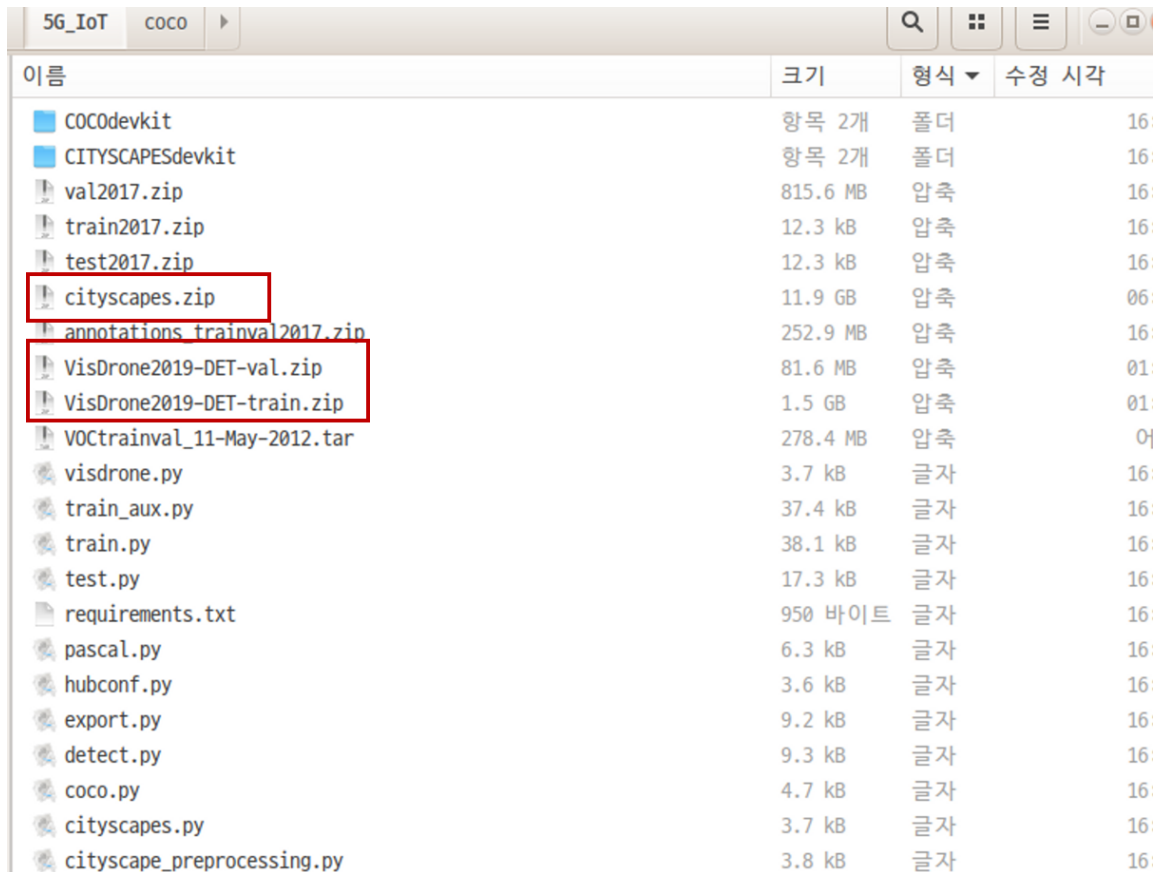
- Pascal voc 및 MS-COCO 데이터의 경우 데이터 다운로드 및 전처리를 모두 지원한다.

- visdrone 및 cityscapes 데이터는 미리 압축파일을 다운받아 프로젝트 폴더에 다음과 같이 위치시켜야한다.

visdrone의 경우 공식 홈페이지를 통해 다운받으면 되고,

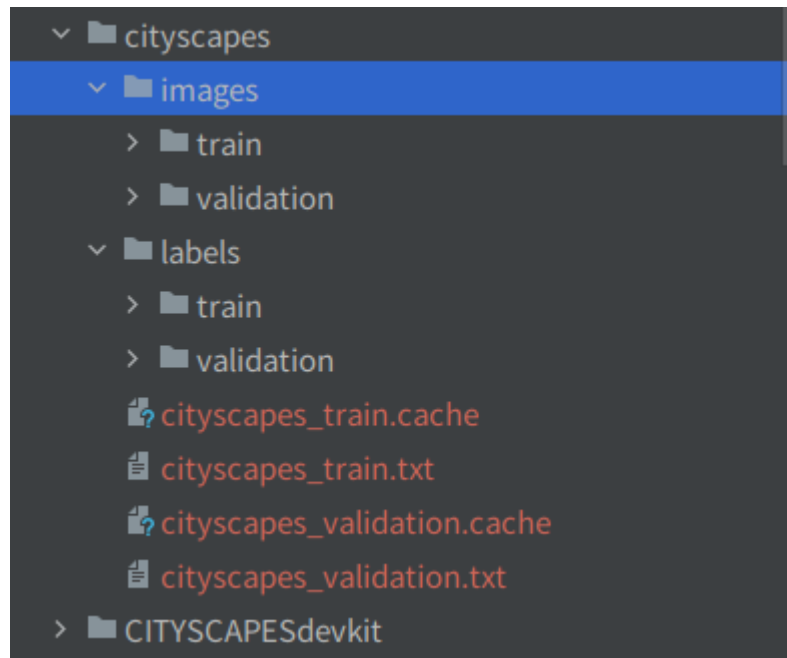
cityscapes는 아래 주소를 통해 데이터를 다운받는다

→ https://drive.google.com/uc?id=1GPueFgeVyHZBLc88fyjL6nSem_9lpAe9



이름	크기	형식	수정 시각
COCOdevkit	항목 2개	폴더	16
CITYSCAPESdevkit	항목 2개	폴더	16
val2017.zip	815.6 MB	압축	16
train2017.zip	12.3 kB	압축	16
test2017.zip	12.3 kB	압축	16
cityscapes.zip	11.9 GB	압축	06
annotations_trainval2017.zip	252.9 MB	압축	16
VisDrone2019-DET-val.zip	81.6 MB	압축	01
VisDrone2019-DET-train.zip	1.5 GB	압축	01
VOCtrainval_11-May-2012.tar	278.4 MB	압축	어
visdrone.py	3.7 kB	글자	16
train_aux.py	37.4 kB	글자	16
train.py	38.1 kB	글자	16
test.py	17.3 kB	글자	16
requirements.txt	950 바이트	글자	16
pascal.py	6.3 kB	글자	16
hubconf.py	3.6 kB	글자	16
export.py	9.2 kB	글자	16
detect.py	9.3 kB	글자	16
coco.py	4.7 kB	글자	16
cityscapes.py	3.7 kB	글자	16
cityscape_preprocessing.py	3.8 kB	글자	16

- Cityscapes 데이터 전처리 과정중 뜨는 “no Label ..” 문구는 전처리 과정중 생기는 문구로 모듈이 정상적으로 동작하는 것이니 걱정하지 않으셔도 됩니다.
- 모든 데이터 셋은 “(데이터 셋 명) Devkit” 이름의 폴더 내에서 압축이 풀리며, 데이터 셋 마다 지정된 폴더명으로 이미지와 라벨파일이 생성된다 (절대 수정 x)
- 생성된 각 데이터 셋 별 폴더 구조는 다음과 같이 구성된다
(아래 그림은 cityscapes 데이터셋 전처리 결과 예시)



커스텀 데이터 셋 구성 방법

데이터 셋 별 커스텀 클래스 선택은 아래와 같은 과정으로 설정한다. (cityscapes는 선택 불가)

1. yaml 파일 수정

아래 그림은 coco.yaml 파일 수정 예시를 나타낸다.

nc는 클래스 수를 의미하며, 아래 names에는 학습할 클래스의 이름을 지정하면된다

```
coco.yaml
# COCO 2017 dataset http://cocodataset.org

# download command/URL (optional)
download: bash ./scripts/get_coco.sh && python coco.py
train: ./coco/train2017.txt # 118287 images
val: ./coco/val2017.txt # 5000 images
test: ./coco/test-dev2017.txt
nc: 12
names: [ 'person', 'bicycle', 'car', 'motorcycle', 'airplane',
         'bus', 'train', 'truck', 'boat', 'traffic light',
         'fire hydrant', 'stop sign' ]
```

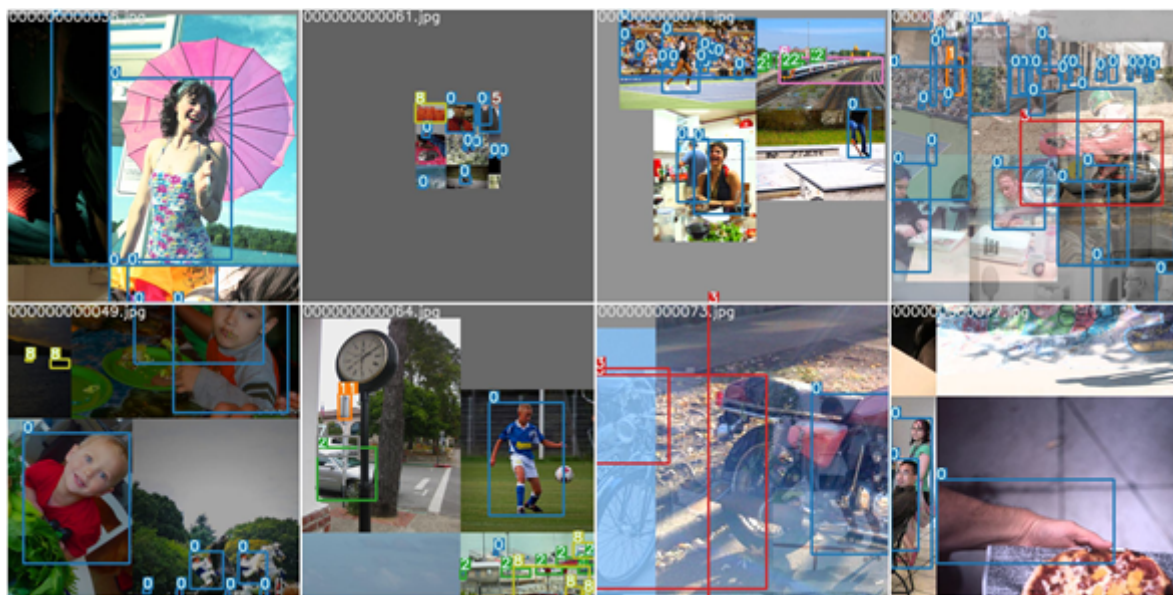
2. (데이터셋 명).py 수정

메인 디렉토리 안에 있는 coco.py, pascal.py, visdrone.py의 custom_cls 부분을 학습할 클래스 이름들로 지정해주어야 한다.

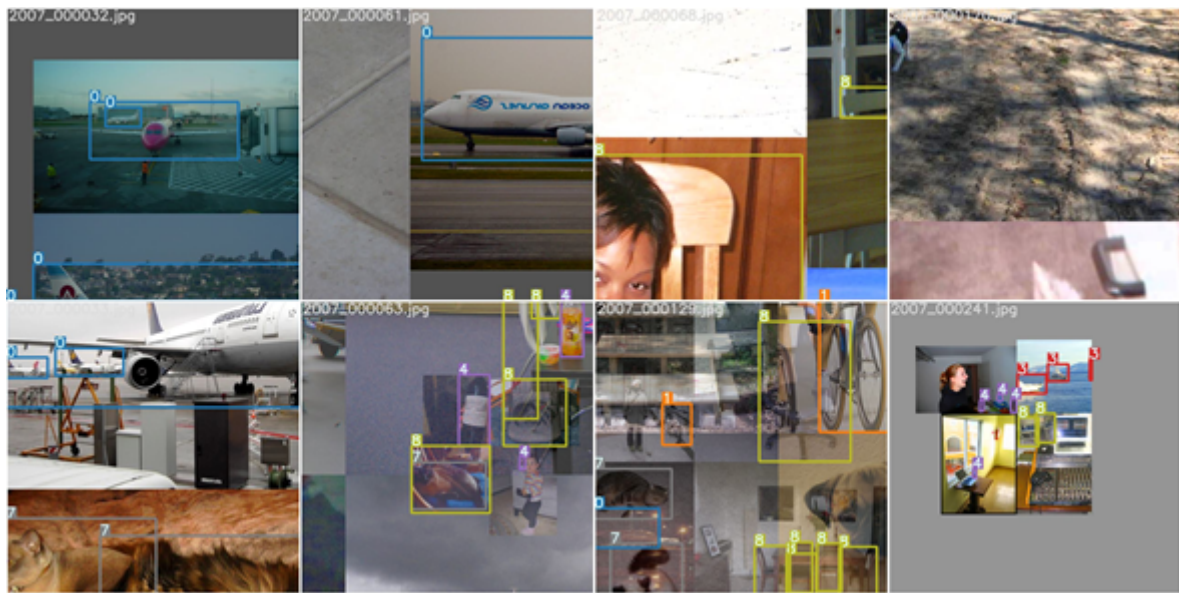
(yaml 파일에서 지정한 클래스 리스트와 동일하게 설정)

```
# Convert
#custom class
custom_cls = [ 'person', 'bicycle', 'car', 'motorcycle',
               'airplane', 'bus', 'train', 'truck', 'boat', 'traffic light',
               'fire hydrant', 'stop sign']
nc = len(custom_cls)
dir = './coco'
path = './COCOdevkit'
```

전처리 과정을 거친 후 생성된 데이터 셋 예시



<MS COCO 2017 예시>



<Pascal VOC 2012 예시>



<Visdrone 2019 예시>



<Cityscapes 데이터 예시>

모든 데이터가 지정된 클래스에 대해서만 라벨을 구성한 것을 확인할 수 있으며,
 마지막 Cityscapes 사진 예시에서도 데이터셋이 지정한 4개의 클래스에 대해서만 라벨을
 표시하는 것을 확인할 수 있다.