YOLOv5

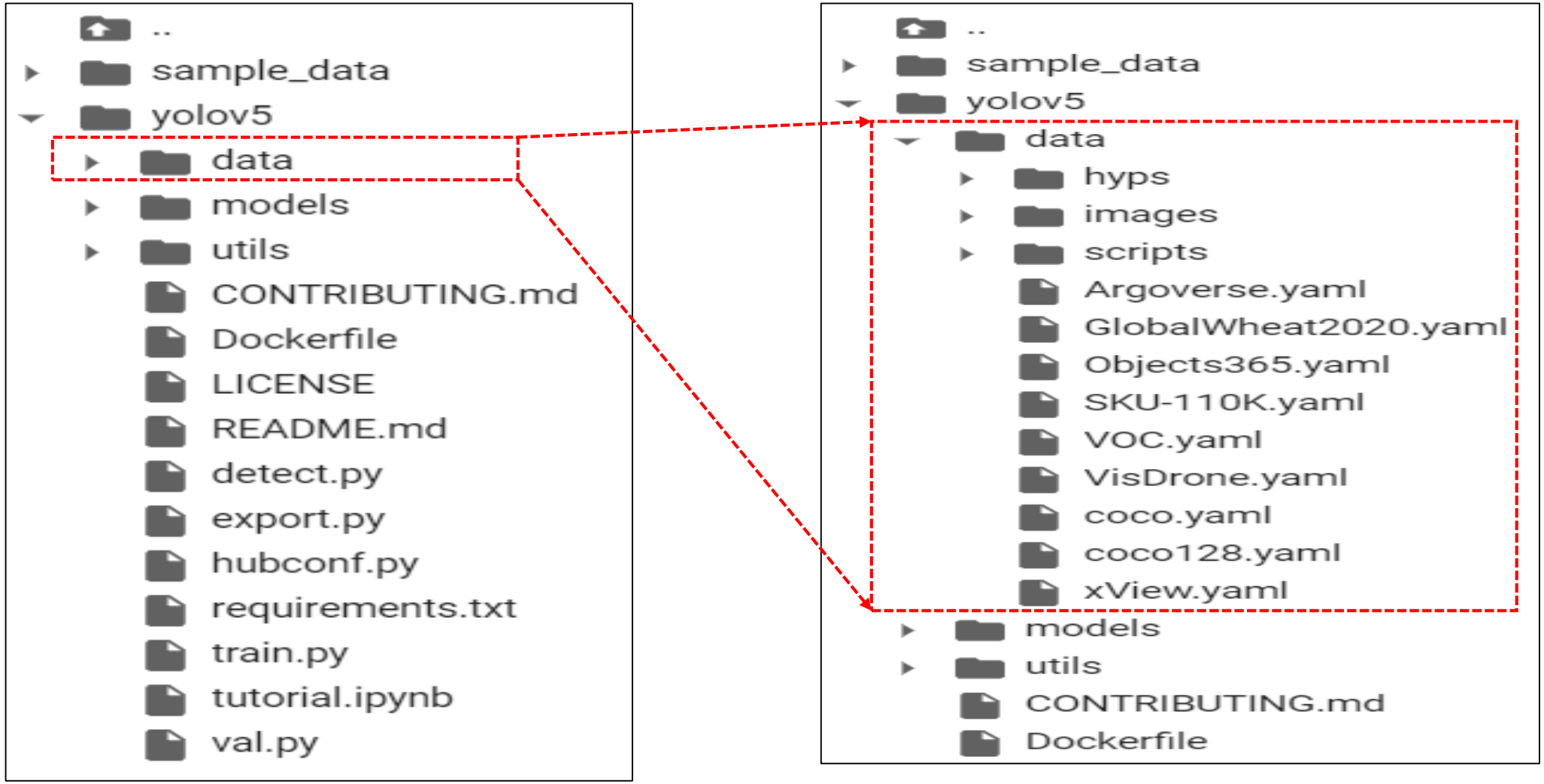
Basic Detection Process

(using pre-trained model)

박성호 (neowizard2018@gmail.com)

YOLOv5 git clone

```
%cd /content

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



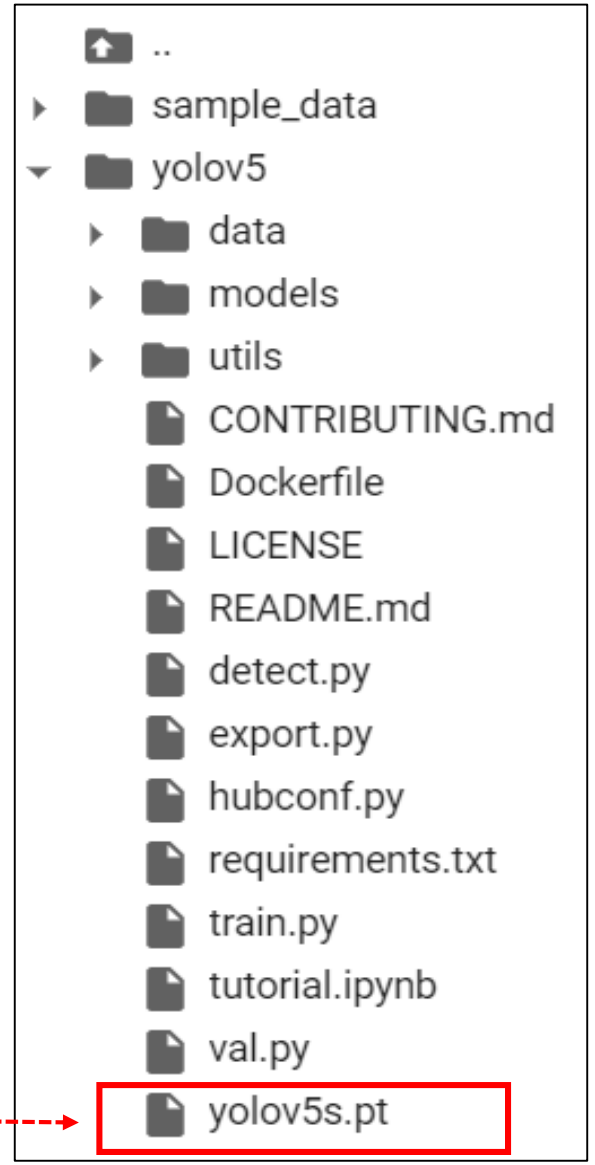
필수 라이브러리 설치

```
# 필수 라이브러리 설치

!pip install -r /content/yolov5/requirements.txt
```

```
Requirement already satisfied: matplotlib>=3.2.2 in /usr/local/lib/python3.7/dist-packages (from -r /conten
Requirement already satisfied: numpy>=1.18.5 in /usr/local/lib/python3.7/dist-packages (from -r /content/yo
Requirement already satisfied: opencv-python>=4.1.2 in /usr/local/lib/python3.7/dist-packages (from -r /con
Requirement already satisfied: Pillow>=7.1.2 in /usr/local/lib/python3.7/dist-packages (from -r /content/yo
Collecting PyYAML>=5.3.1
  Downloading PyYAML-6.0-cp37-cp37m-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_12_x86_64.manylinux2
     |████████████████████████████████| 596 kB 13.3 MB/s
Requirement already satisfied: requests>=2.23.0 in /usr/local/lib/python3.7/dist-packages (from -r /content
Requirement already satisfied: scipy>=1.4.1 in /usr/local/lib/python3.7/dist-packages (from -r /content/yol
Requirement already satisfied: torch>=1.7.0 in /usr/local/lib/python3.7/dist-packages (from -r /content/yol
Requirement already satisfied: torchvision>=0.8.1 in /usr/local/lib/python3.7/dist-packages (from -r /conte
Requirement already satisfied: tqdm>=4.41.0 in /usr/local/lib/python3.7/dist-packages (from -r /content/yol
Requirement already satisfied: tensorboard>=2.4.1 in /usr/local/lib/python3.7/dist-packages (from -r /conte
Requirement already satisfied: pandas>=1.1.4 in /usr/local/lib/python3.7/dist-packages (from -r /content/yo
Requirement already satisfied: seaborn>=0.11.0 in /usr/local/lib/python3.7/dist-packages (from -r /content/
Collecting thop
  Downloading thop-0.0.31.post2005241907-py3-none-any.whl (8.7 kB)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib
```

pre-trained model 다운로드 (option)

```
!wget -P /content/yolov5/  https://github.com/ultralytics/yolov5/releases/download/v6.0/yolov5s.pt

--2021-10-22 11:19:27--  https://github.com/ultralytics/yolov5/releases/download/v6.0/yolov5s.pt
Resolving github.com (github.com)... 140.82.113.4
Connecting to github.com (github.com)|140.82.113.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github-releases.githubusercontent.com/264818686/eab38592-7168-4731-bdff-ad5ede200
--2021-10-22 11:19:27--  https://github-releases.githubusercontent.com/264818686/eab38592-7168-4731-
Resolving github-releases.githubusercontent.com (github-releases.githubusercontent.com)... 185.199.
Connecting to github-releases.githubusercontent.com (github-releases.githubusercontent.com)|185.199
HTTP request sent, awaiting response... 200 OK
Length: 14698491 (14M) [application/octet-stream]
Saving to:  '/content/yolov5/yolov5s.pt'

yolov5s.pt          100%[===================>]  14.02M  --.-KB/s    in 0.1s

2021-10-22 11:19:27 (101 MB/s) -  '/content/yolov5/yolov5s.pt'  saved [14698491/14698491]
```



detect 실행 시에, 자동으로 다운로드 된 pre-trained model (default=yolo5s.pt). COCO Dataset으로 이미 학습되어 있음

학습데이터 정보 (nc, names, path 등) 가지고 있는 yaml 확인 및 수정 (현재는 확인)

```
# yaml 파일 확인 및 필요시 설정 (데이터셋 위치를 알려주는 config file)

%cat /content/yolov5/data/coco128.yaml
```

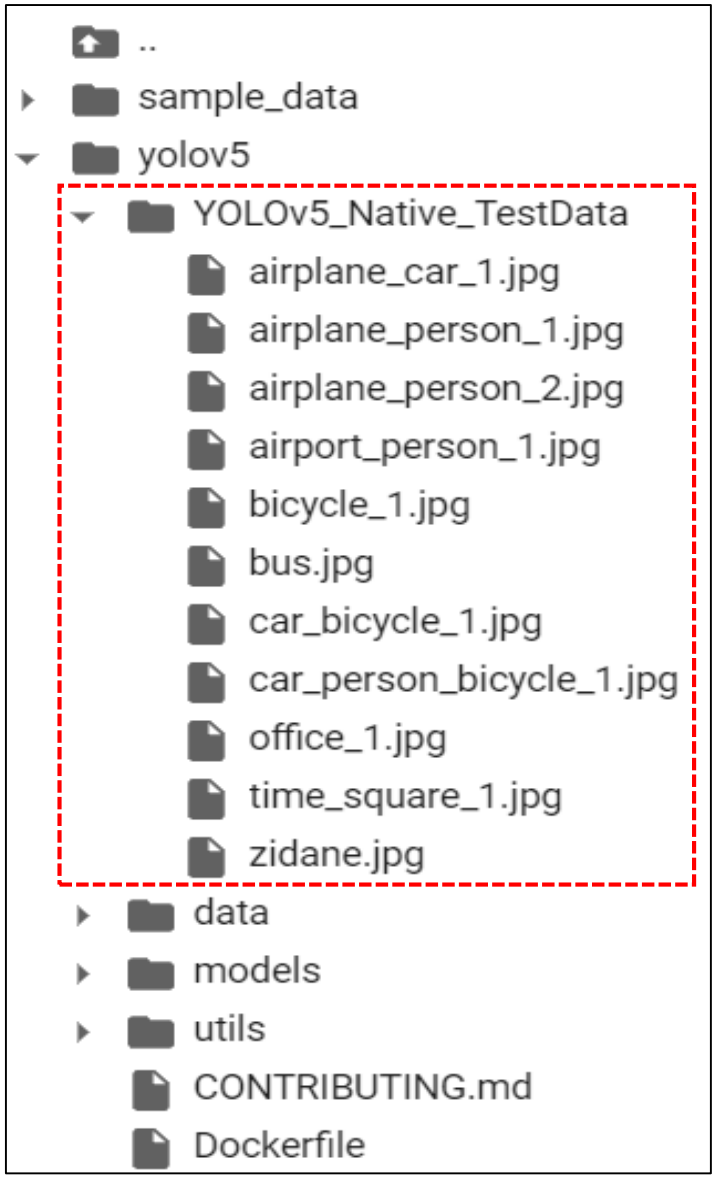
```
# YOLOv5 🚀 by Ultralytics, GPL-3.0 license
# COCO128 dataset https://www.kaggle.com/ultralytics/coco128 (first 128 images from COCO train2017)
# Example usage: python train.py --data coco128.yaml
# parent
# ├── yolov5
# └── datasets
#     └── coco128  ← downloads here


# 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, ..]
path: ../datasets/coco128  # dataset root dir
train: images/train2017  # train images (relative to 'path') 128 images
val: images/train2017  # val images (relative to 'path') 128 images
test:  # test images (optional)

# Classes
nc: 80  # number of classes
names: ['person', 'bicycle', 'car', 'motorcycle', 'airplane', 'bus', 'train', 'truck', 'boat', 'traffic light',
        'fire hydrant', 'stop sign', 'parking meter', 'bench', 'bird', 'cat', 'dog', 'horse', 'sheep', 'cow',
        'elephant', 'bear', 'zebra', 'giraffe', 'backpack', 'umbrella', 'handbag', 'tie', 'suitcase', 'frisbee',
        'skis', 'snowboard', 'sports ball', 'kite', 'baseball bat', 'baseball glove', 'skateboard', 'surfboard',
        'tennis racket', 'bottle', 'wine glass', 'cup', 'fork', 'knife', 'spoon', 'bowl', 'banana', 'apple',
        'sandwich', 'orange', 'broccoli', 'carrot', 'hot dog', 'pizza', 'donut', 'cake', 'chair', 'couch',
        'potted plant', 'bed', 'dining table', 'toilet', 'tv', 'laptop', 'mouse', 'remote', 'keyboard', 'cell phone',
        'microwave', 'oven', 'toaster', 'sink', 'refrigerator', 'book', 'clock', 'vase', 'scissors', 'teddy bear',
        'hair drier', 'toothbrush']  # class names


# Download script/URL (optional)
download: https://github.com/ultralytics/yolov5/releases/download/v1.0/coco128.zip
```

테스트 데이터 업로드 및 확인

```
import zipfile

with zipfile.ZipFile('/content/YOLOv5_Native_TestData.zip', 'r') as target_file:

    target_file.extractall('/content/yolov5/YOLOv5_Native_TestData/')
```

User-Defined Directory

```
# 테스트 이미지

import glob

test_image_list = glob.glob('/content/yolov5/YOLOv5_Native_TestData/*.jpg')

print(len(test_image_list))

test_image_list.sort()

for i in range(len(test_image_list)):

    print('i = ',i, test_image_list[i])
```



COCO Dataset으로 사전학습된 모델 이용하여 테스트 실행

```
# detect 실행

!python3 /content/yolov5/detect.py --weights /content/yolov5/yolov5s.pt  --source /content/yolov5/YOLOv5_Native_TestData/
```

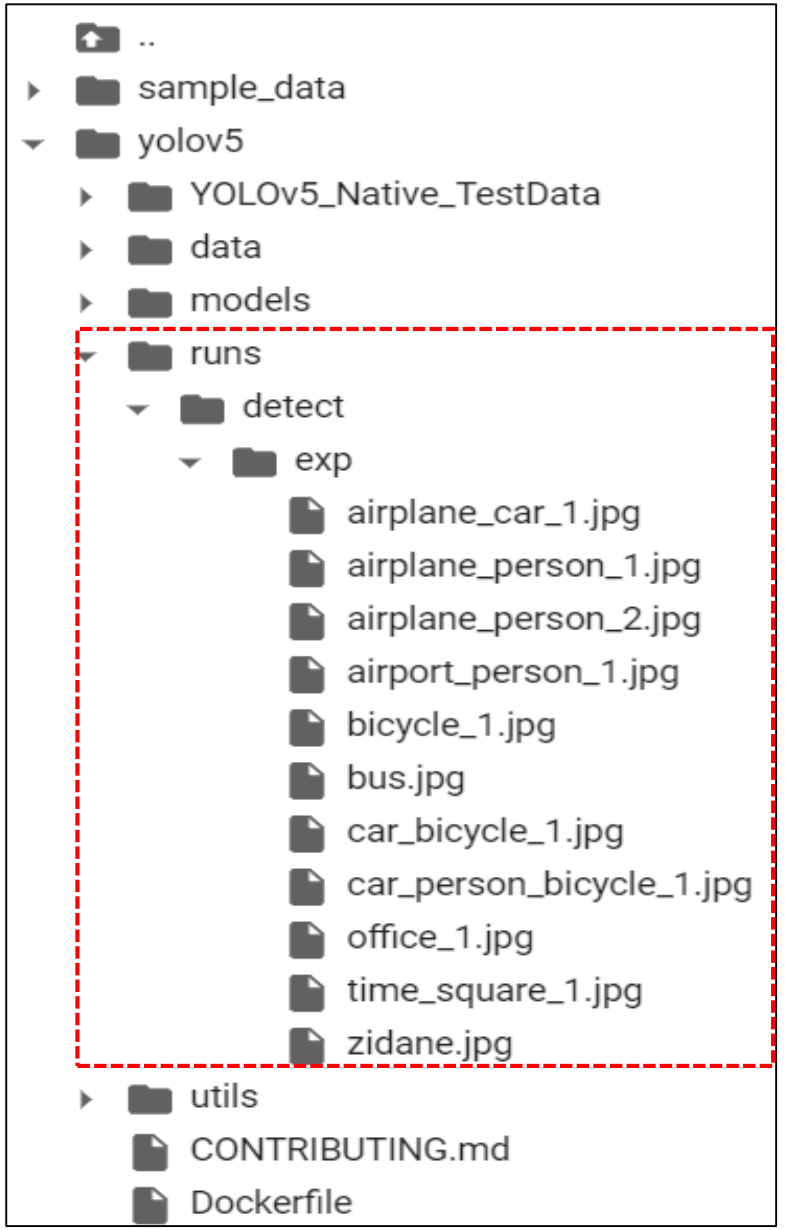
```
Downloading https://ultralytics.com/assets/Arial.ttf to /root/.config/Ultralytics/Arial.ttf...
detect: weights=['/content/yolov5/yolov5s.pt'], source=/content/yolov5/YOLOv5_Native_TestData/, imgsz=[640, 640], conf_thres=0.25, iou
YOLOv5 🚀 v6.0-25-g15e8c4c torch 1.9.0+cu111 CUDA:0 (Tesla K80, 11441.1875MB)

Fusing layers...
Model Summary: 213 layers, 7225885 parameters, 0 gradients
image 1/11 /content/yolov5/YOLOv5_Native_TestData/airplane_car_1.jpg: 512x640 2 persons, 1 car, 3 airplanes, 7 trucks, Done. (0.035s)
image 2/11 /content/yolov5/YOLOv5_Native_TestData/airplane_person_1.jpg: 448x640 1 person, 1 airplane, Done. (0.031s)
image 3/11 /content/yolov5/YOLOv5_Native_TestData/airplane_person_2.jpg: 448x640 1 person, 6 airplanes, Done. (0.029s)
image 4/11 /content/yolov5/YOLOv5_Native_TestData/airport_person_1.jpg: 384x640 8 persons, 1 handbag, 1 suitcase, 1 cell phone, Done. (
image 5/11 /content/yolov5/YOLOv5_Native_TestData/bicycle_1.jpg: 416x640 9 persons, 2 bicycles, 1 car, Done. (0.031s)
image 6/11 /content/yolov5/YOLOv5_Native_TestData/bus.jpg: 640x480 4 persons, 1 bus, Done. (0.033s)
image 7/11 /content/yolov5/YOLOv5_Native_TestData/car_bicycle_1.jpg: 480x640 6 persons, 3 cars, 6 motorcycles, 2 umbrellas, Done. (0.03
image 8/11 /content/yolov5/YOLOv5_Native_TestData/car_person_bicycle_1.jpg: 352x640 2 persons, 1 bicycle, 7 cars, 1 bus, 1 traffic ligh
image 9/11 /content/yolov5/YOLOv5_Native_TestData/office_1.jpg: 320x640 1 person, 1 cup, 1 potted plant, 1 laptop, 1 vase, Done. (0.027
image 10/11 /content/yolov5/YOLOv5_Native_TestData/time_square_1.jpg: 448x640 18 persons, 3 cars, 1 truck, 2 traffic lights, Done. (0.0
image 11/11 /content/yolov5/YOLOv5_Native_TestData/zidane.jpg: 384x640 2 persons, 1 tie, Done. (0.030s)
Speed: 0.5ms pre-process, 30.7ms inference, 5.7ms NMS per image at shape (1, 3, 640, 640)
Results saved to yolov5/runs/detect/exp
```

result directory

결과 확인

```
import glob

detetced_image_list = glob.glob(('/content/yolov5/runs/detect/exp/*.jpg'))

detected_image_nums = len(detetced_image_list)

print(detected_image_nums)

print(detetced_image_list)
```

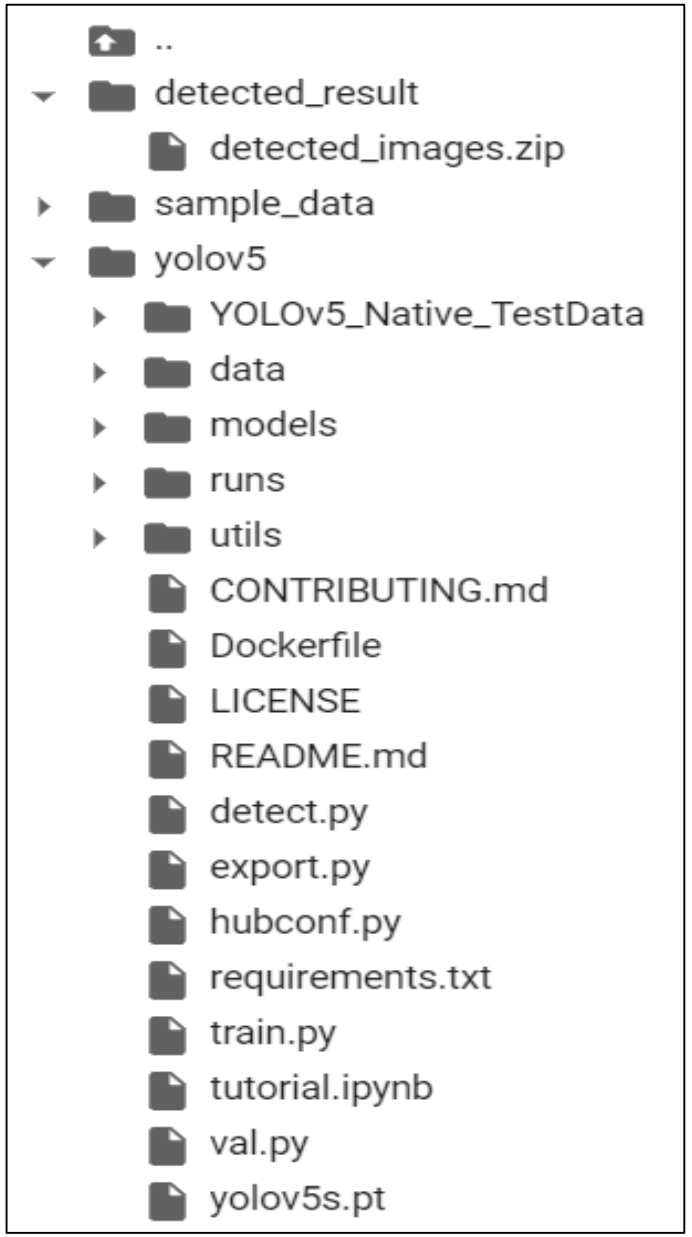
```
11
['/content/yolov5/runs/detect/exp/car_bicycle_1.jpg', '/content/yolov5/runs/detect/exp/bicycle_1.jpg', '/c
```



결과 다운로드

```
# 다운로드를 위한 inference image 압축

import zipfile
import os

if not os.path.exists('/content/detected_result/'):
    os.mkdir('/content/detected_result/')
    print('detected_result dir is created !!!')


with zipfile.ZipFile('/content/detected_result/detected_images.zip', 'w') as detected_images:
    for idx in range(detected_image_nums):
        detected_images.write(detetced_image_list[idx])
```

```
# 압축파일 다운로드

from google.colab import files

files.download('/content/detected_result/detected_images.zip')
```



[참고] detect..py 실행 옵션 (--source, --weights, --conf,...)

```
python detect.py --source 0   # webcam
                         file.jpg   # image
                         file.mp4   # video
                         path/   # directory
                         path/*.jpg   # glob
                         'https://youtu.be/NUsoVlDFqZg'   # YouTube video
                         'rtsp://example.com/media.mp4'   # RTSP, RTMP, HTTP stream
```

```
python detect.py --weights /content/yolo5/weights/best.pt  --conf 0.5  --source ....
```