深度学习-物体检测及分类-YOLOv5实战-汽车检测 (一桥)

课程网址：

1. https://download.csdn.net/course/detail/37028

你将收获

1.全套物体检测分类实操经验
2.独立完成工作能力

适用人群

1.想学习人工智能的所有人员 2.想学习物体检测、分类的所有技术人员

课程介绍

物体检测应用场景：人脸检测，车辆检测，行人计数，自动驾驶，安全系统 ... 等众多场景
本课程将在90分钟内，让学员学会人工智能领域的【物体检测分类的实战技巧】，达到可以自己动手标注数据、训练模型、模型优化等方面技能，
拥有独立完成工作的能力。
课程主要内容：
1.数据标注
2.模型训练
3.预测
4.代码讲解
5.模型优化
6. 其他...

课程目录

- 1.课程介绍 免费
- 2.数据标注
- 3.环境搭建
- 3.2环境搭建_补充知识
- 4.训练集、测试集的制作
- 5模型训练
- 6.模型预测

总结：

1、课程介绍

目标检测实战：行人检测、汽车检测

目标-自动驾驶

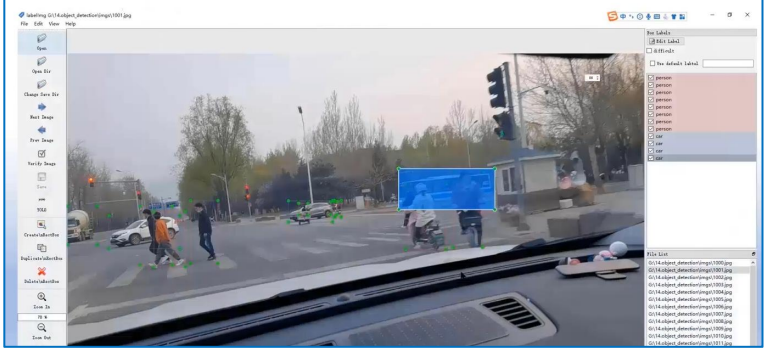
1. 课程介绍
2. 物体检测-任务说明
3. 环境搭建
4. 数据集标注
5. 训练集、测试集的制作
6. 代码讲解
7. 模型训练
8. 模型预测
9. 模型优化
10. 作业

物体检测-任务说明

讲解检测哪些物体

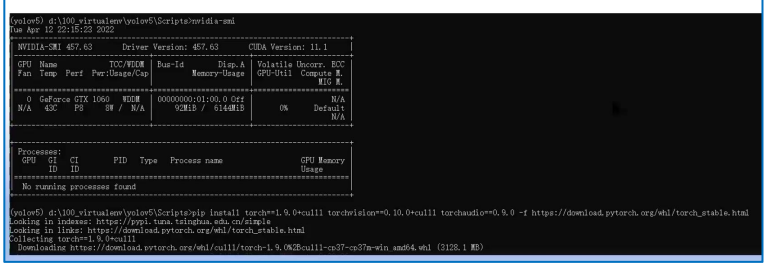
二、数据集标注

labelImg 下载



三、环境搭建

讲解如何搭建虚拟环境（cuda11.1+torch1.9.0）



四、训练集、验证集的制作

讲解模型训练所需的训练集、验证集文件 (train、val)

1.修改配置文件、创建相关文件夹
2.制作数据集
3:1

五、模型训练

1.预训练模型下载：

2.设置 vscode使用虚拟环境

3.设定参数，进行模型训练

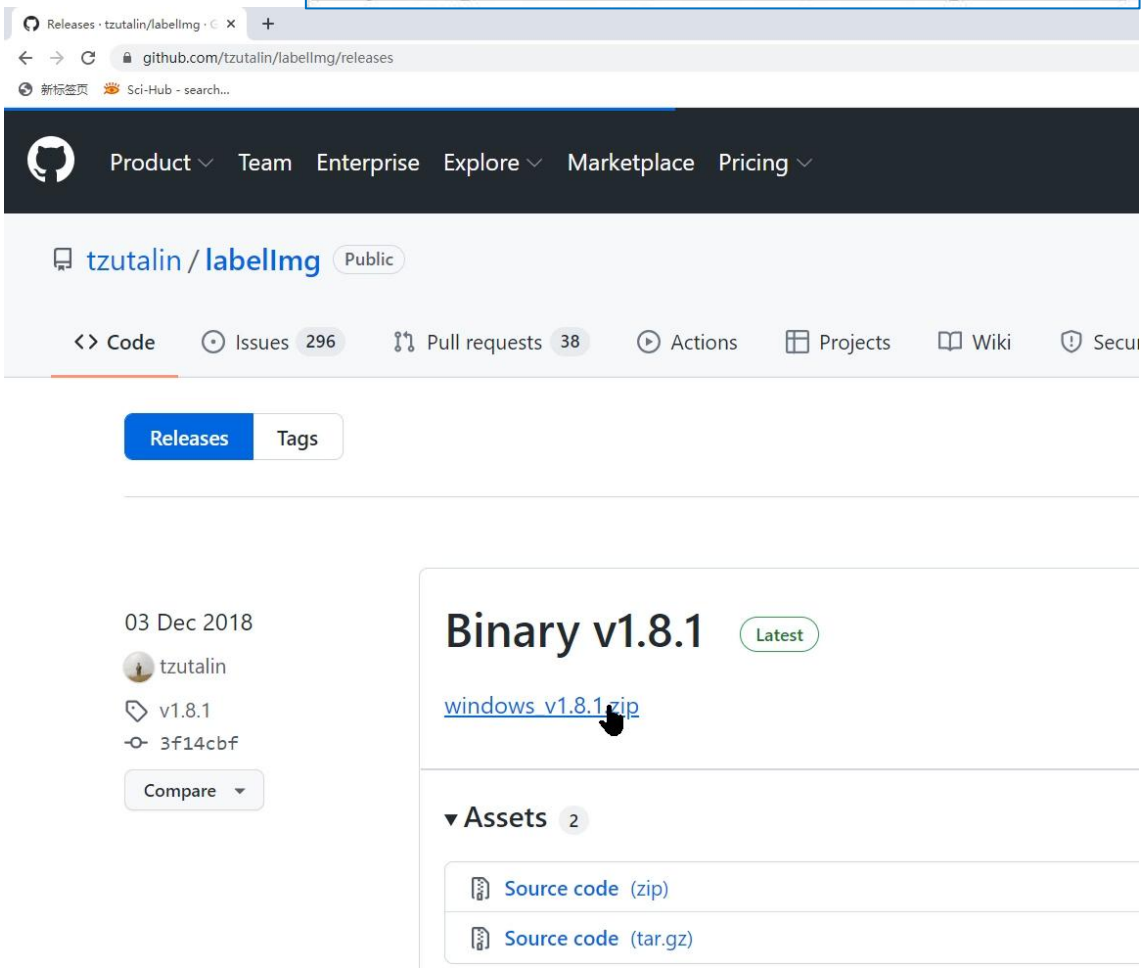
六、模型预测

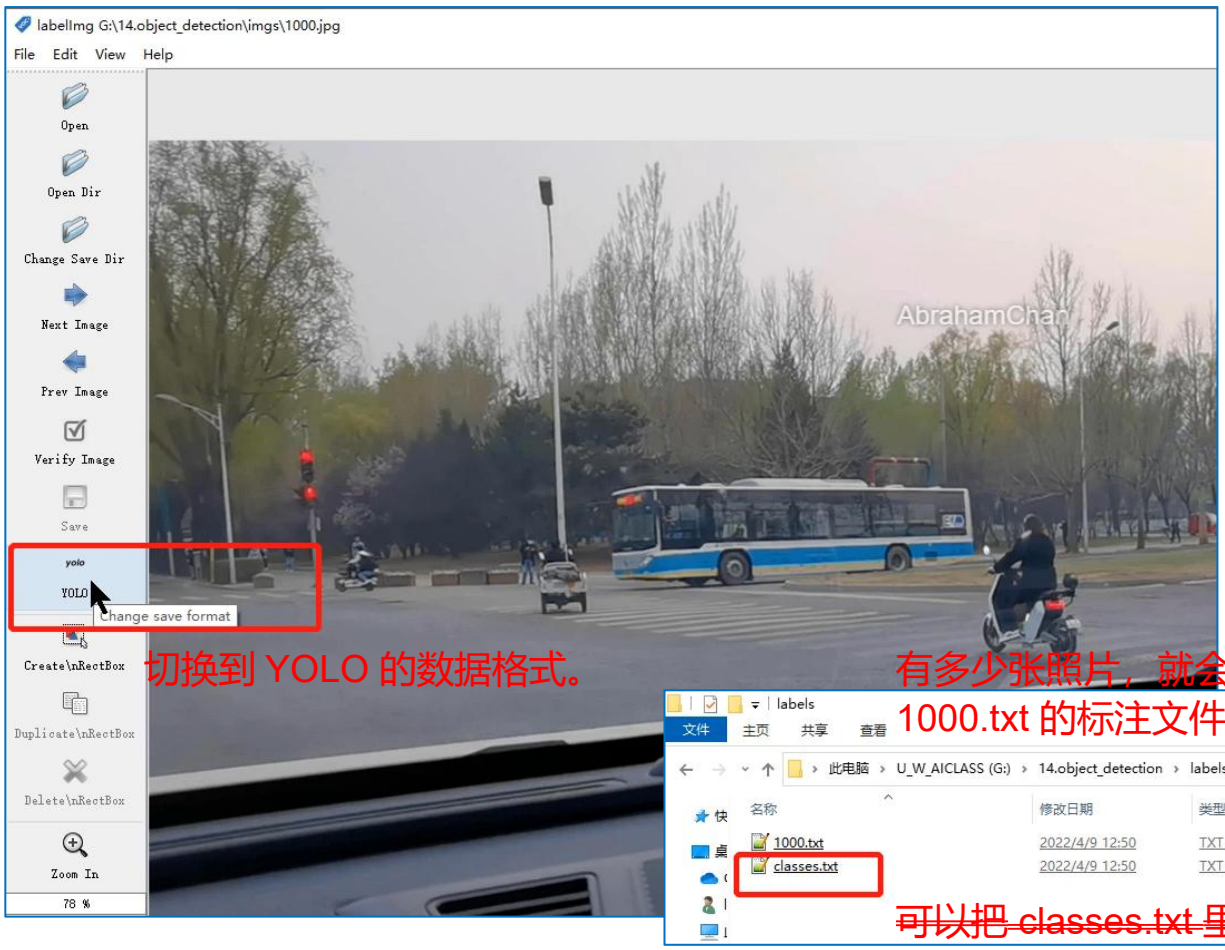
利用模型预测的方法：
1.参数设定
2.测试
3.效果预览

总结：

2、数据标注

https://github.com/tzutalin/labelImg/releases





切换到 YOLO 的数据格式。

有多少张照片，就会生成多少个类似 1000.txt 的标注文件。

~~可以把 classes.txt 里不用的类给删掉。~~

总结：

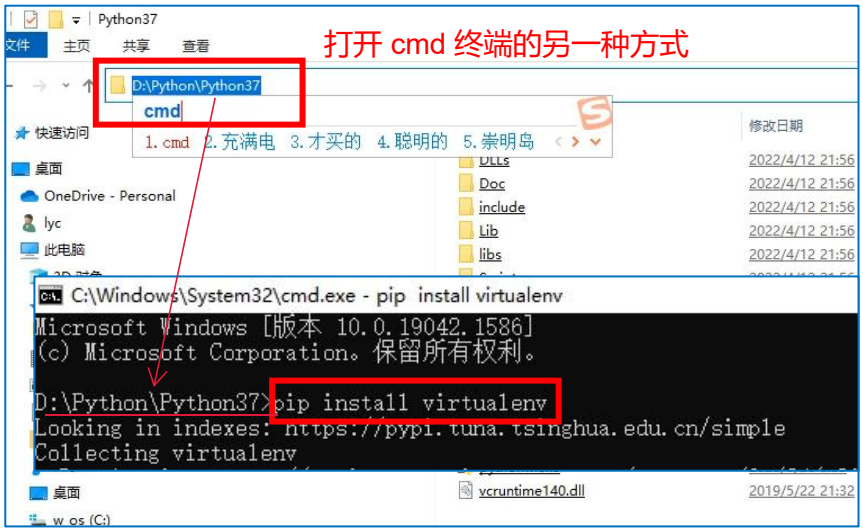
3、环境搭建（Windows）

多种方式：
1.Anaconda
https://www.anaconda.com/
a. 下载 b.安装 c. 创建虚拟环境 d.安装软件包

2.Virtualenv
a.Python : https://www.python.org/
b.安装virtualenv, 并创建一个虚拟环境（激活：activate）
c.在虚拟环境下安装软件包
注意：torch需要与我们的cuda版本相对应

3.docker

安装 virtualenv

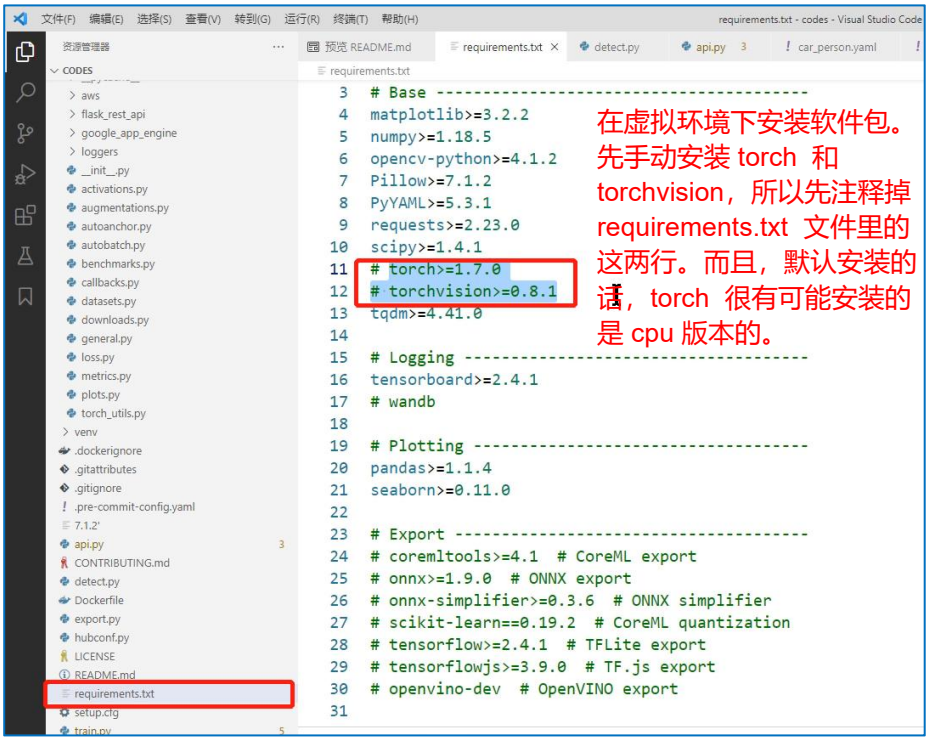


创建虚拟环境 yolov5

```
d:\100_virtualenv>virtualenv -p python3 yolov5
created virtual environment CPython3.7.8.final.0-64 in 398ms
  creator CPython3Windows(dest=D:\100_virtualenv\yolov5, clear=False, no_vcs_ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=C:\Users\lyc\AppData\Local\pypa\virtualenv)
    added seed packages: pip==22.0.4, setuptools==62.1.0, wheel==0.37.1
  activators BashActivator,BatchActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator

d:\100_virtualenv>dir
 驱动器 D 中的卷是 w_work
 卷的序列号是 DA66-C398

 d:\100_virtualenv 的目录

2022/04/12  22:05    <DIR>          .
2022/04/12  22:05    <DIR>          ..
2021/06/29  08:31    <DIR>          py3
2021/07/10  16:36                42 py3.6.5.txt
2021/12/20  16:59    <DIR>          py375_cv4.1.1.26
2022/04/12  22:05    <DIR>          yolov5
               1 个文件             42 字节
               5 个目录 53,593,276,416 可用字节

d:\100_virtualenv>cd yolov5
```

```
d:\100_virtualenv\yolov5>cd Scripts

d:\100_virtualenv\yolov5\Scripts>dir
 驱动器 D 中的卷是 w_work
 卷的序列号是 DA66-C398

 d:\100_virtualenv\yolov5\Scripts 的目录

2022/04/12  22:05    <DIR>          .
2022/04/12  22:05    <DIR>          ..
2022/04/12  22:05             2,147 activate
2022/04/12  22:05               987 activate.bat
2022/04/12  22:05             3,024 activate.fish
2022/04/12  22:05             2,567 activate.nu
2022/04/12  22:05             1,758 activate.ps1
2022/04/12  22:05             1,193 activate_this.py
2022/04/12  22:05               510 deactivate.bat
2022/04/12  22:05               682 deactivate.nu
2022/04/12  22:05           106,866 pip-3.7.exe
2022/04/12  22:05           106,866 pip.exe
2022/04/12  22:05           106,866 pip3.7.exe
2022/04/12  22:05           106,866 pip3.exe
2022/04/12  22:05                24 pydoc.bat
2022/04/12  22:05           522,768 python.exe
2022/04/12  22:05           522,256 pythonw.exe
2022/04/12  22:05           106,853 wheel-3.7.exe
2022/04/12  22:05           106,853 wheel.exe
2022/04/12  22:05           106,853 wheel3.7.exe
2022/04/12  22:05           106,853 wheel3.exe
              19 个文件      1,912,792 字节
               2 个目录 53,593,276,416 可用字节

d:\100_virtualenv\yolov5\Scripts>activate

(yolov5) d:\100_virtualenv\yolov5\Scripts>python
```

进入虚拟环境文件夹下的 Scripts 文件夹，激活该虚拟环境。



总结：

```
(yolov5) d:\100_virtualenv\yolov5\Scripts>pip install torch==1.9.0+cu111 torchvision==0.10.0+cu111 torchaudio==0.9.0 -f https://download.pytorch.org/whl/torch_stable.html
Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple
Looking in links: https://download.pytorch.org/whl/torch_stable.html
Collecting torch==1.9.0+cu111
```

```
(yolov5) d:\100_virtualenv\yolov5\Scripts>python
Python 3.7.8 (tags/v3.7.8:4b47a5b6ba, Jun 28 2020, 08:53:46) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import torch as tc
>>> tc.__version__
'1.9.0+cu111'
>>> tc.cuda.is_available()
True
```

```
(yolov5) G:\14.object_detection\codes>pip install -r requirements.txt
Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple
Collecting matplotlib>=3.2.2
```

环境搭建补充知识

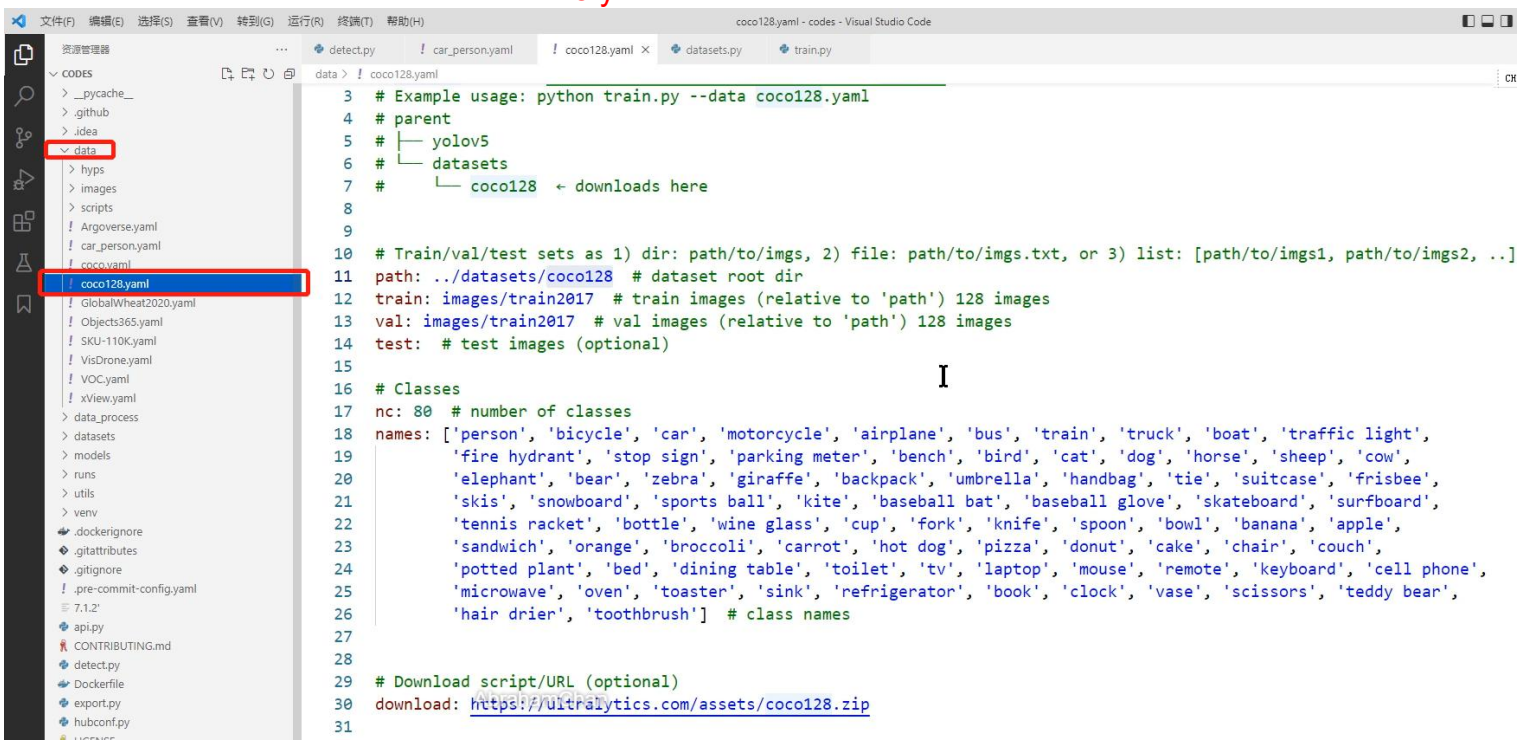
（略）

4、训练集、测试集的制作

讲解模型训练所需的训练集、验证集文件 (train、val)

1.修改配置文件、创建相关文件夹
2.制作数据集
3:1

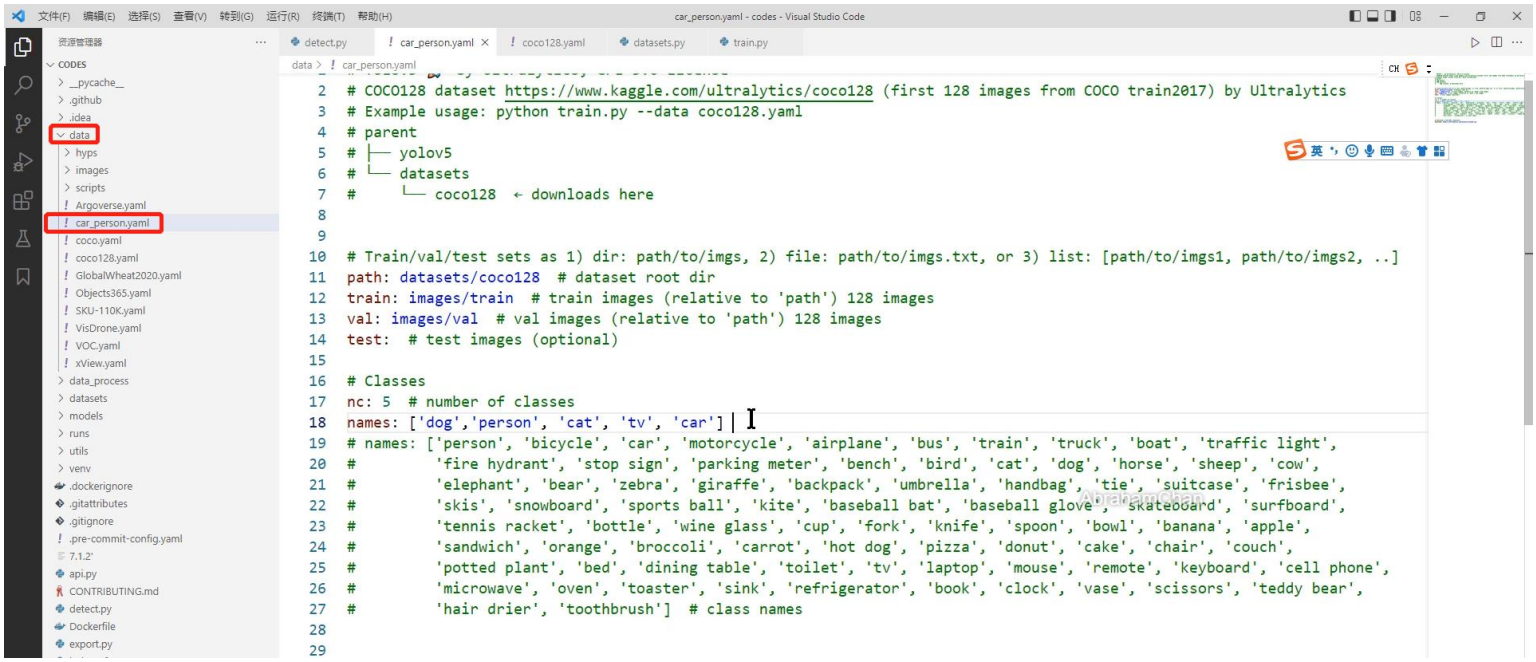
coco128.yaml



data 文件夹下 coco128.yaml 文件复制一份，改名字，如 car_person.yaml。然后修改里面内容。。

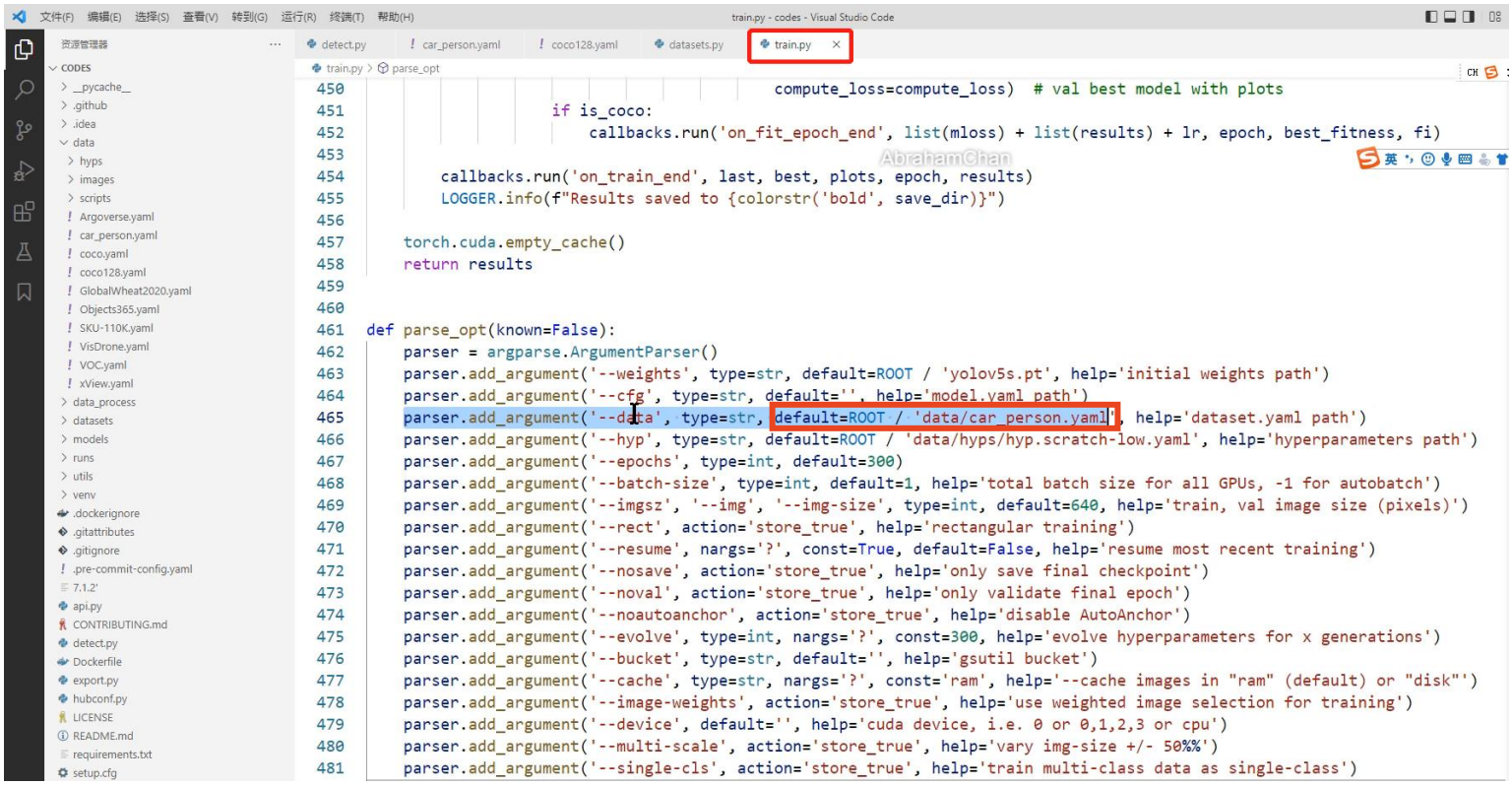
总结:

car_person.yaml （设置路径、类别数等配置参数。）



```yaml
# COCO128 dataset https://www.kaggle.com/ultralytics/coco128 (first 128 images from COCO train2017) by Ultralytics
# 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/train  # train images (relative to 'path') 128 images
val: images/val  # val images (relative to 'path') 128 images
test:  # test images (optional)

# Classes
nc: 5  # number of classes
names: ['dog','person', 'cat', 'tv', 'car']
# 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
```

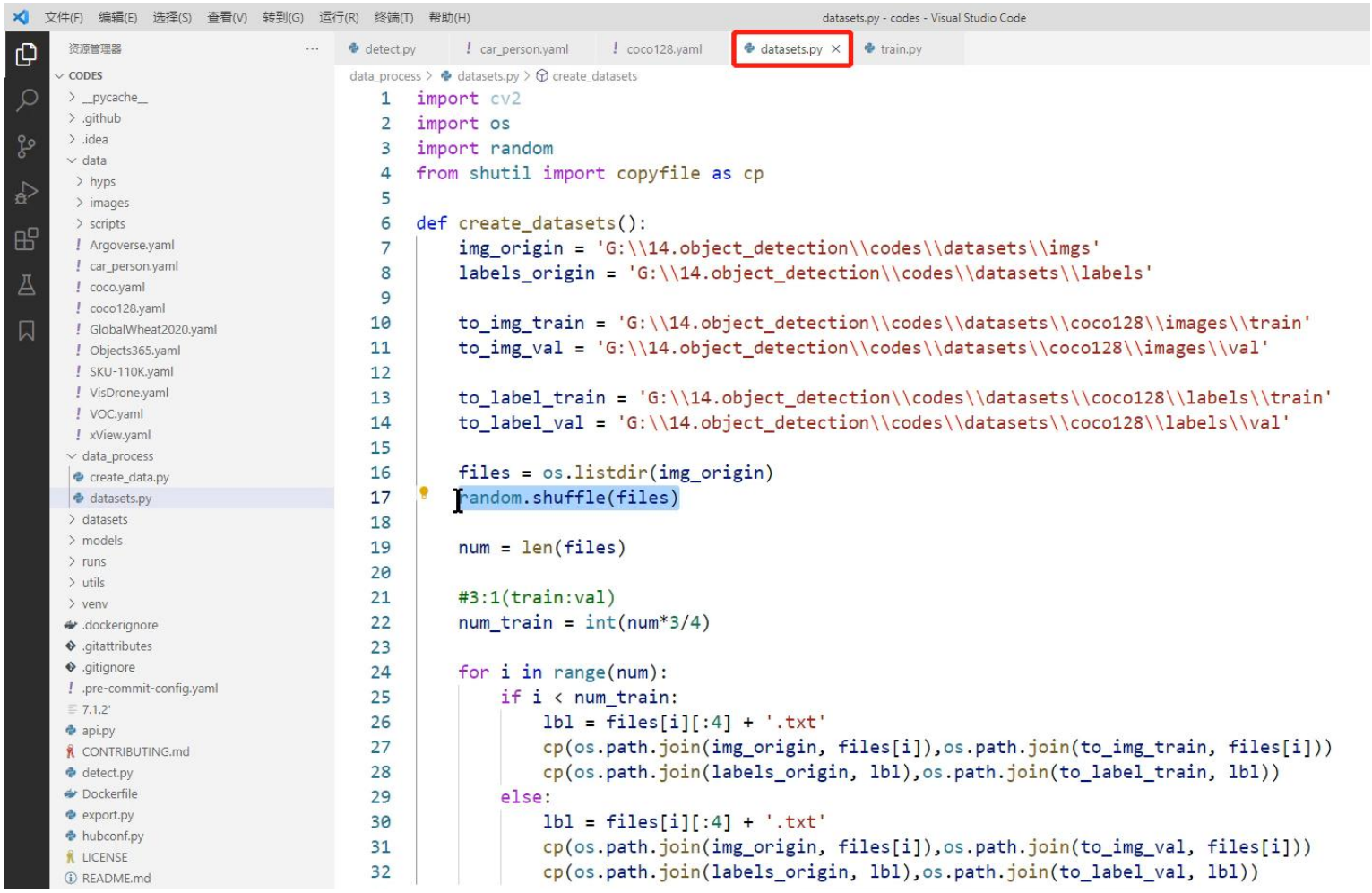
train.py



```python
                                    compute_loss=compute_loss)  # val best model with plots
                if is_coco:
                    callbacks.run('on_fit_epoch_end', list(mloss) + list(results) + lr, epoch, best_fitness, fi)

        callbacks.run('on_train_end', last, best, plots, epoch, results)
        LOGGER.info(f"Results saved to {colorstr('bold', save_dir)}")

    torch.cuda.empty_cache()
    return results


def parse_opt(known=False):
    parser = argparse.ArgumentParser()
    parser.add_argument('--weights', type=str, default=ROOT / 'yolov5s.pt', help='initial weights path')
    parser.add_argument('--cfg', type=str, default='', help='model.yaml path')
    parser.add_argument('--data', type=str, default=ROOT / 'data/car_person.yaml', help='dataset.yaml path')
    parser.add_argument('--hyp', type=str, default=ROOT / 'data/hyps/hyp.scratch-low.yaml', help='hyperparameters path')
    parser.add_argument('--epochs', type=int, default=300)
    parser.add_argument('--batch-size', type=int, default=1, help='total batch size for all GPUs, -1 for autobatch')
    parser.add_argument('--imgsz', '--img', '--img-size', type=int, default=640, help='train, val image size (pixels)')
    parser.add_argument('--rect', action='store_true', help='rectangular training')
    parser.add_argument('--resume', nargs='?', const=True, default=False, help='resume most recent training')
    parser.add_argument('--nosave', action='store_true', help='only save final checkpoint')
    parser.add_argument('--noval', action='store_true', help='only validate final epoch')
    parser.add_argument('--noautoanchor', action='store_true', help='disable AutoAnchor')
    parser.add_argument('--evolve', type=int, nargs='?', const=300, help='evolve hyperparameters for x generations')
    parser.add_argument('--bucket', type=str, default='', help='gsutil bucket')
    parser.add_argument('--cache', type=str, nargs='?', const='ram', help='--cache images in "ram" (default) or "disk"')
    parser.add_argument('--image-weights', action='store_true', help='use weighted image selection for training')
    parser.add_argument('--device', default='', help='cuda device, i.e. 0 or 0,1,2,3 or cpu')
    parser.add_argument('--multi-scale', action='store_true', help='vary img-size +/- 50%%')
    parser.add_argument('--single-cls', action='store_true', help='train multi-class data as single-class')
```

总结：

datasets.py



总结:

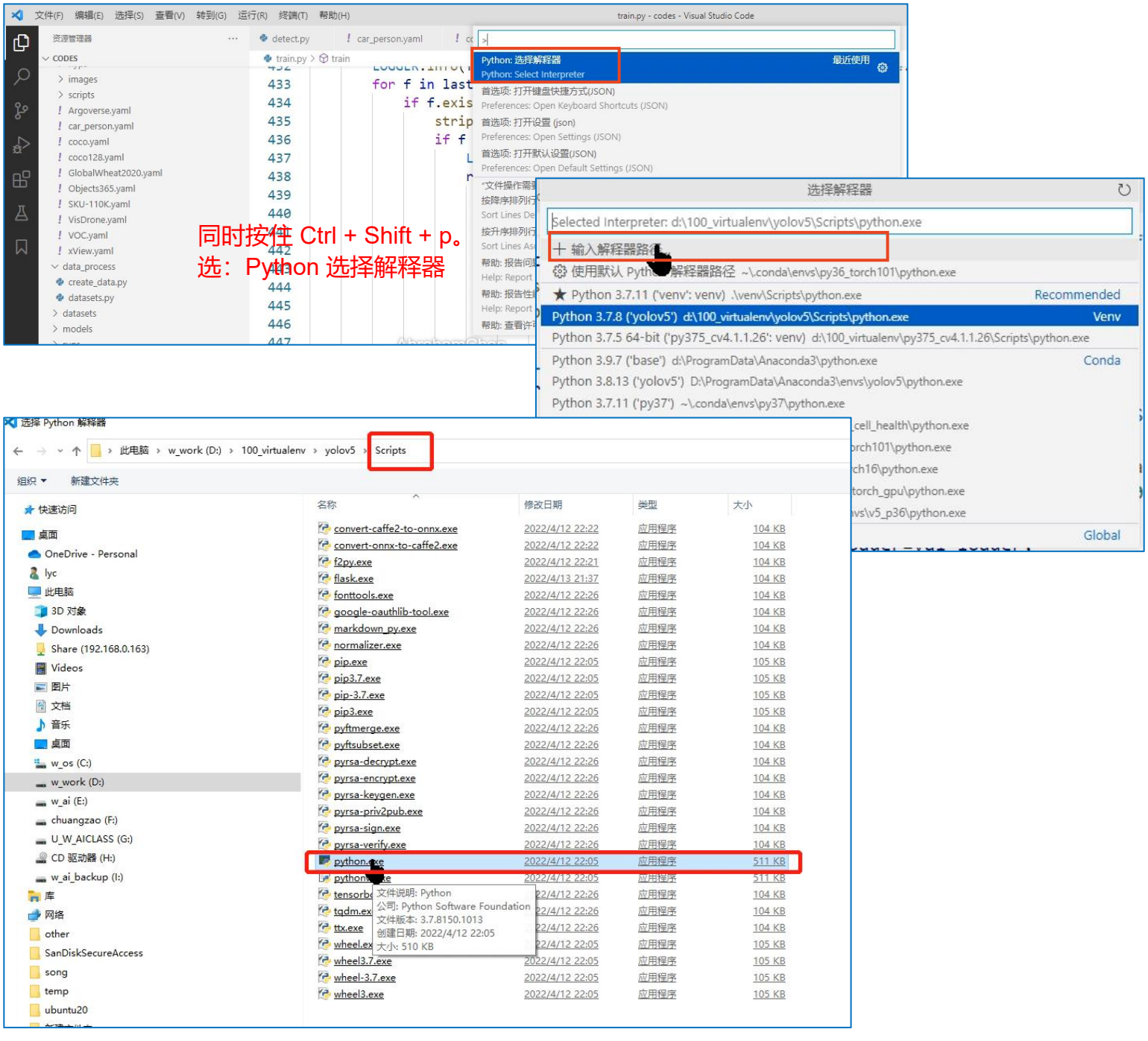
5、模型训练

1.预训练模型下载：
https://github.com/ultralytics/yolov5/releases/download/v6.1/yolov5s.pt

2.设置 vscode使用虚拟环境

3.设定参数，进行模型训练　train.py 文件里进行参数设定。

配置 vscode 使用虚拟环境。

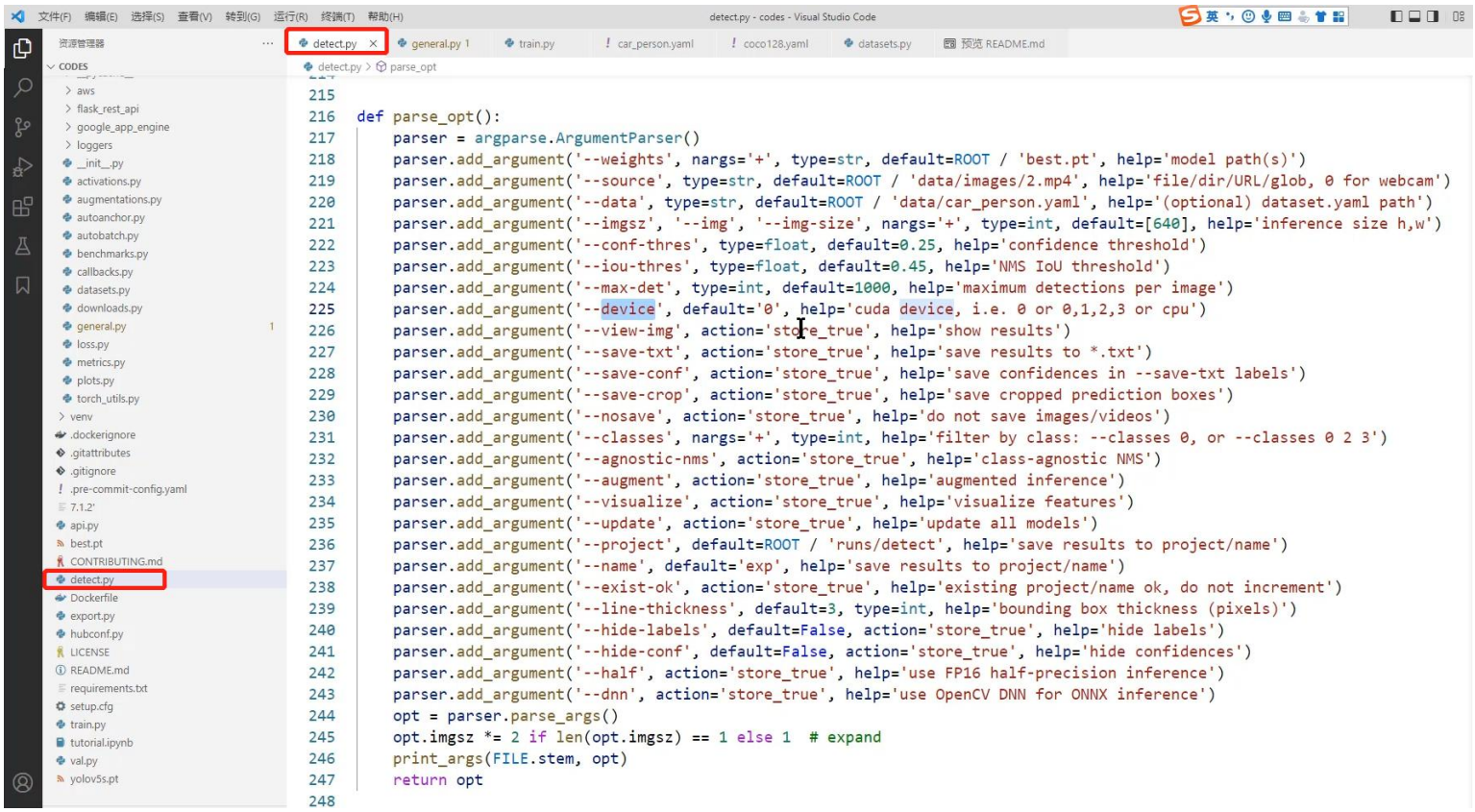


总结：

6、模型预测

利用模型预测的方法：
1.参数设定　（照片 or 视频等）
2.测试
3.效果预览

detect.py



总结：