

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

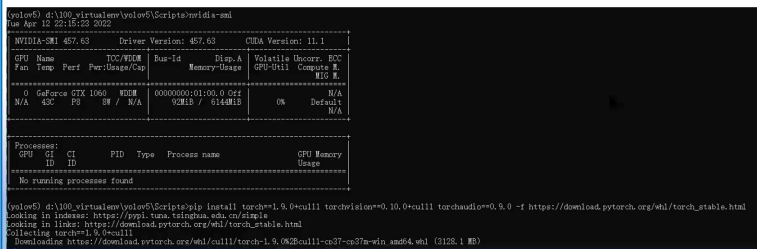
二、数据集标注

labellmg 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

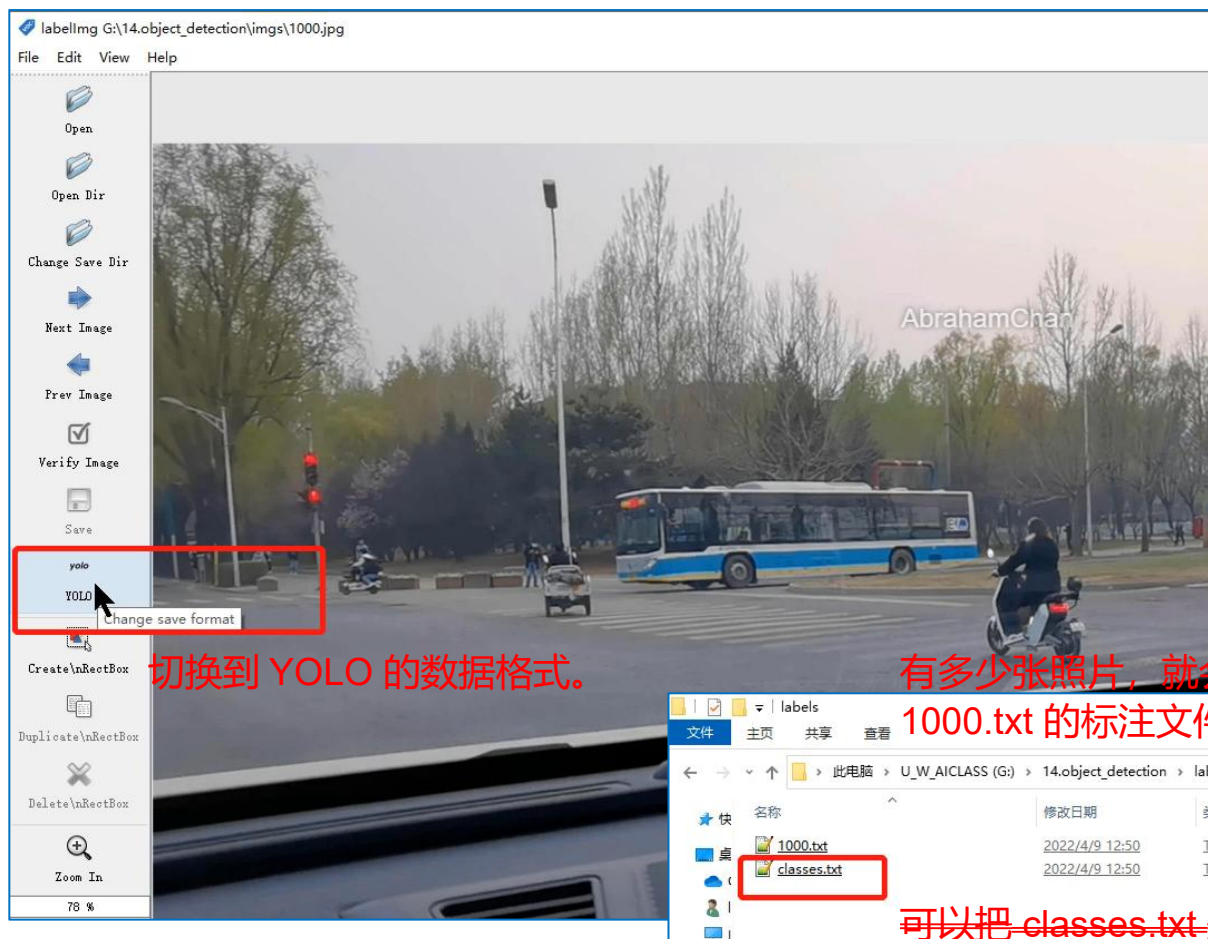
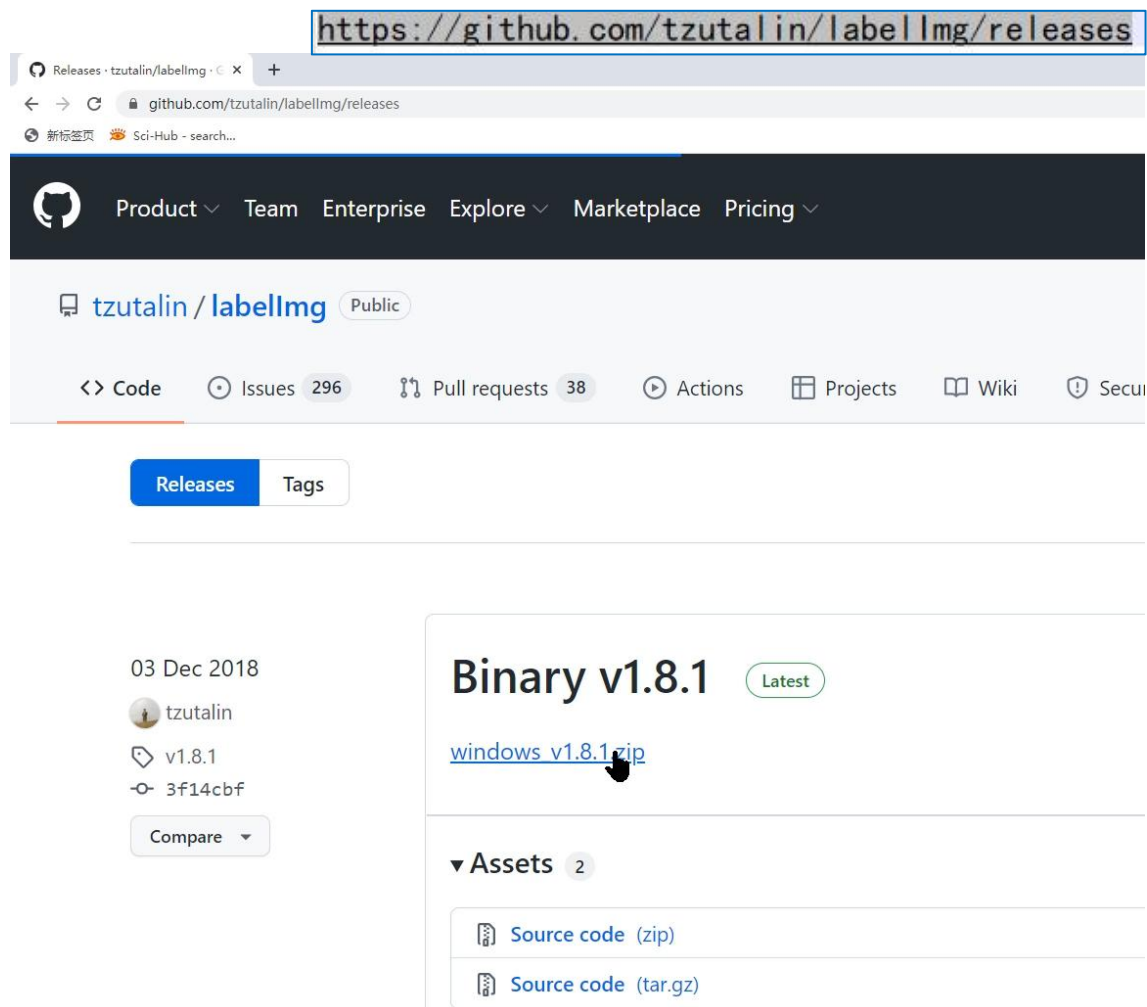
- 1.预训练模型下载：
- 2.设置 vscode使用虚拟环境
- 3.设定参数，进行模型训练

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

2、数据标注



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/>

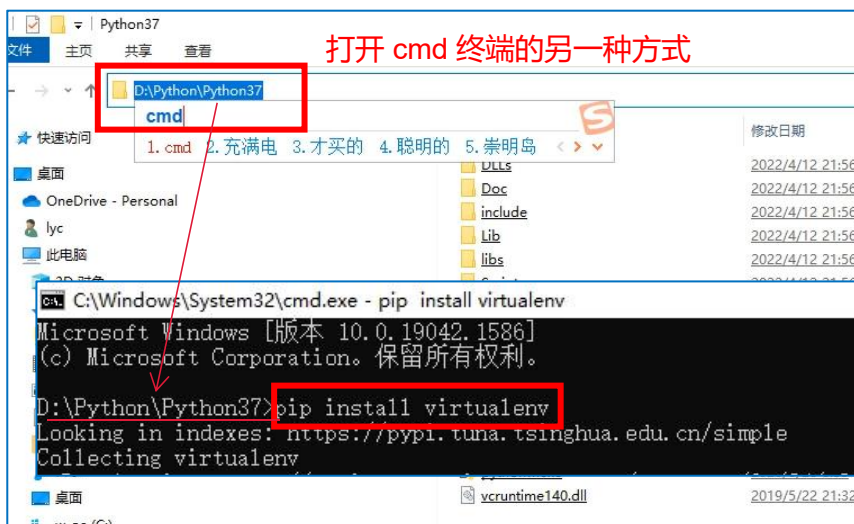
b. 安装 virtualenv, 并创建一个虚拟环境 (激活: activate)

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
```

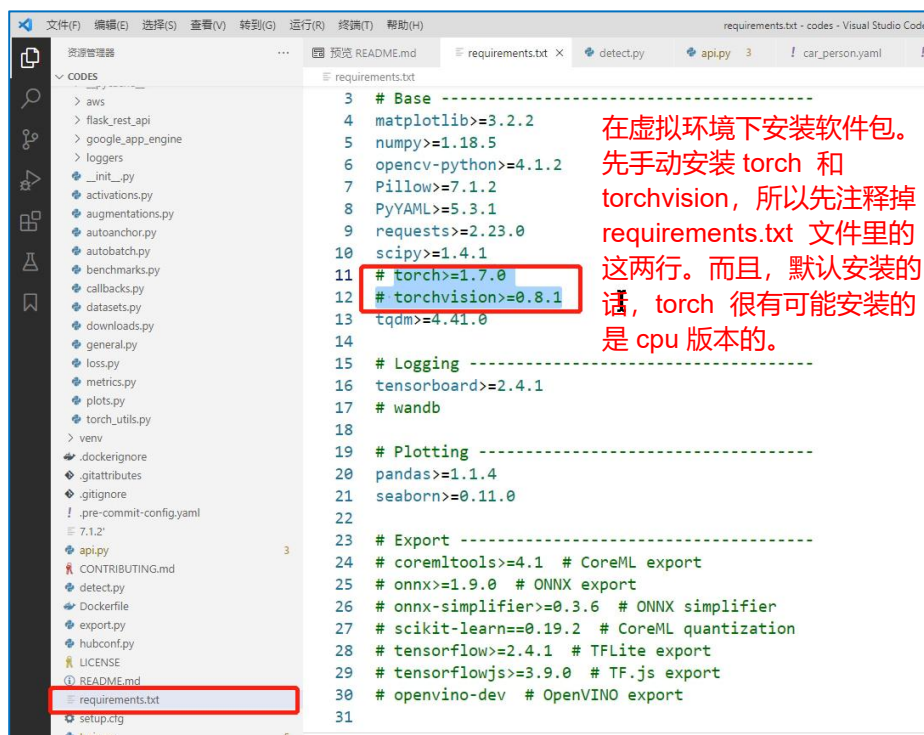
d:\100_virtualenv\yolov5\Scripts 的目录

```
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
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

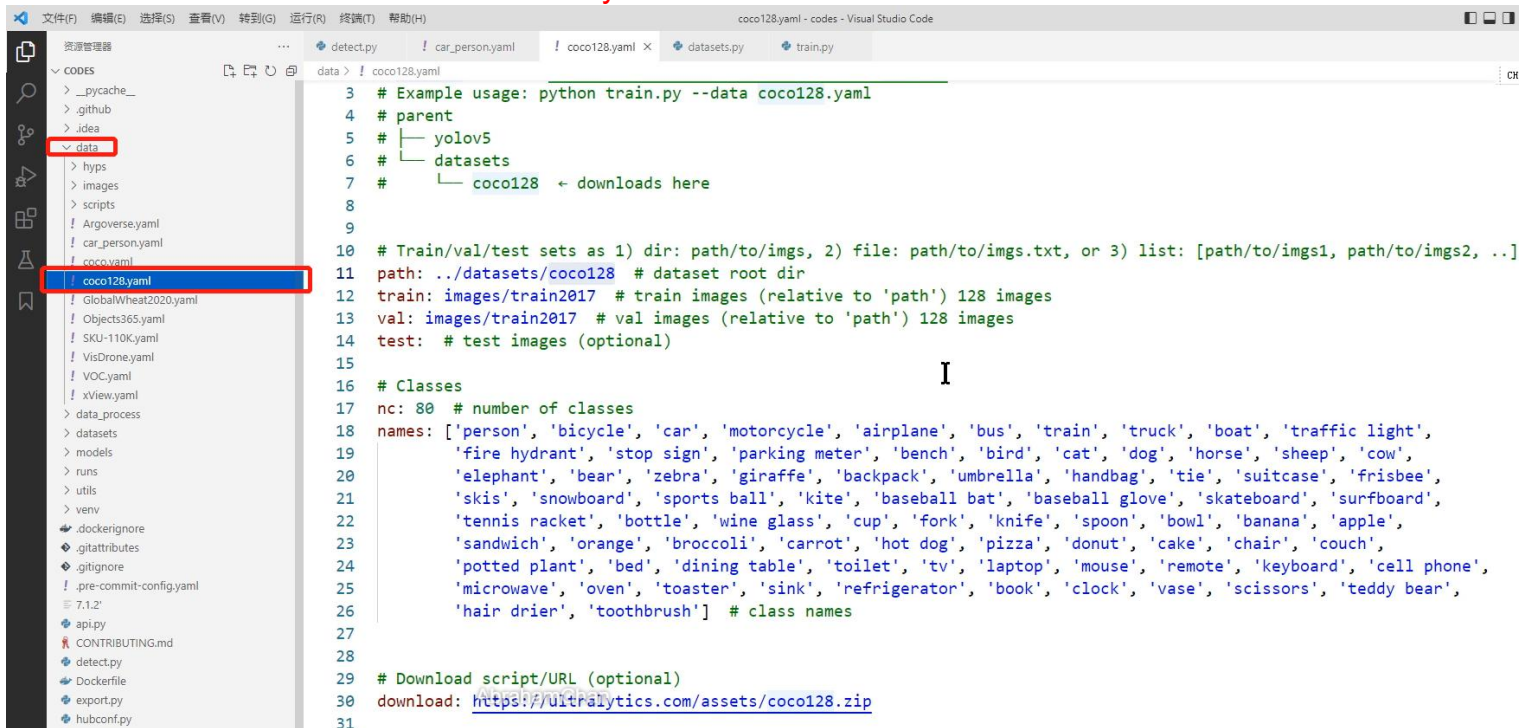
(略)

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

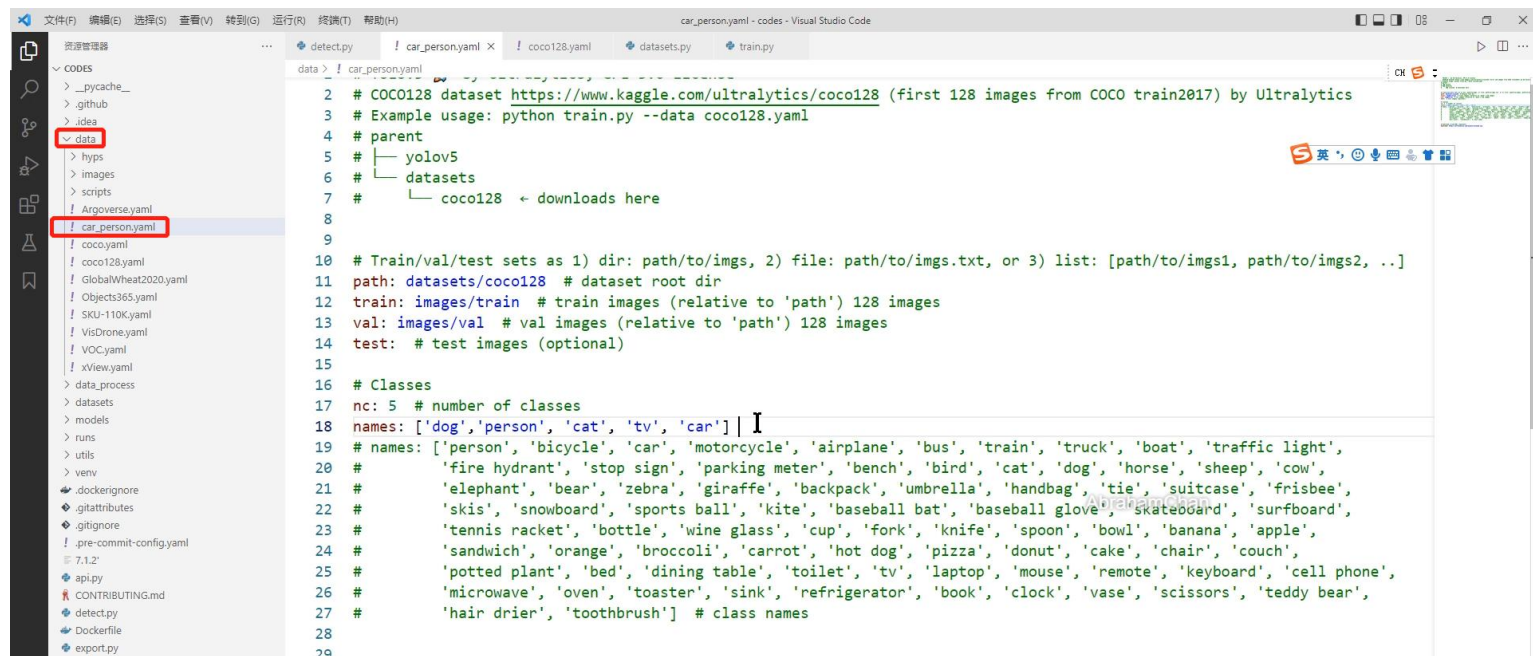
coco128.yaml



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。。

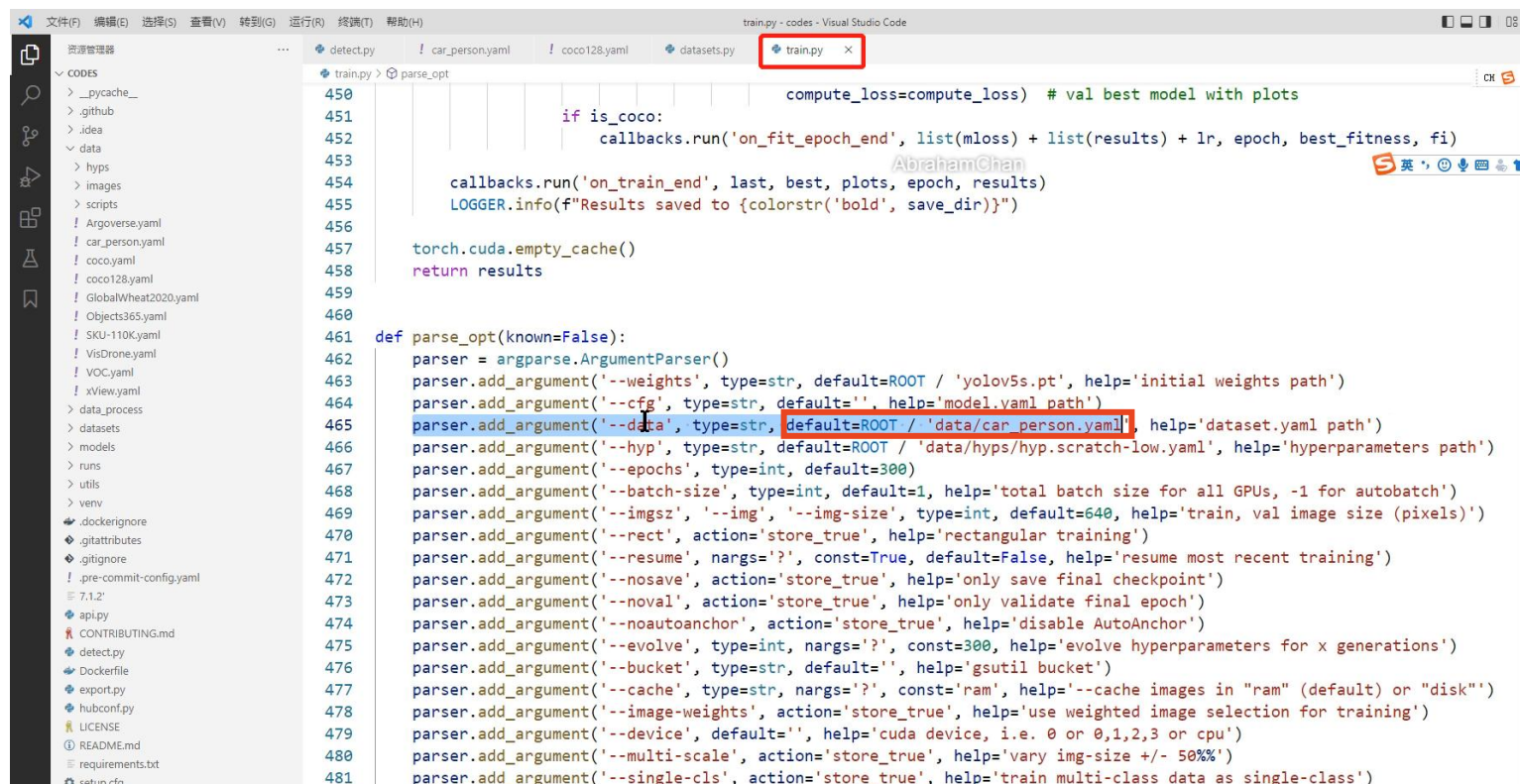
总结：

car_person.yaml (设置路径、类别数等配置参数。)

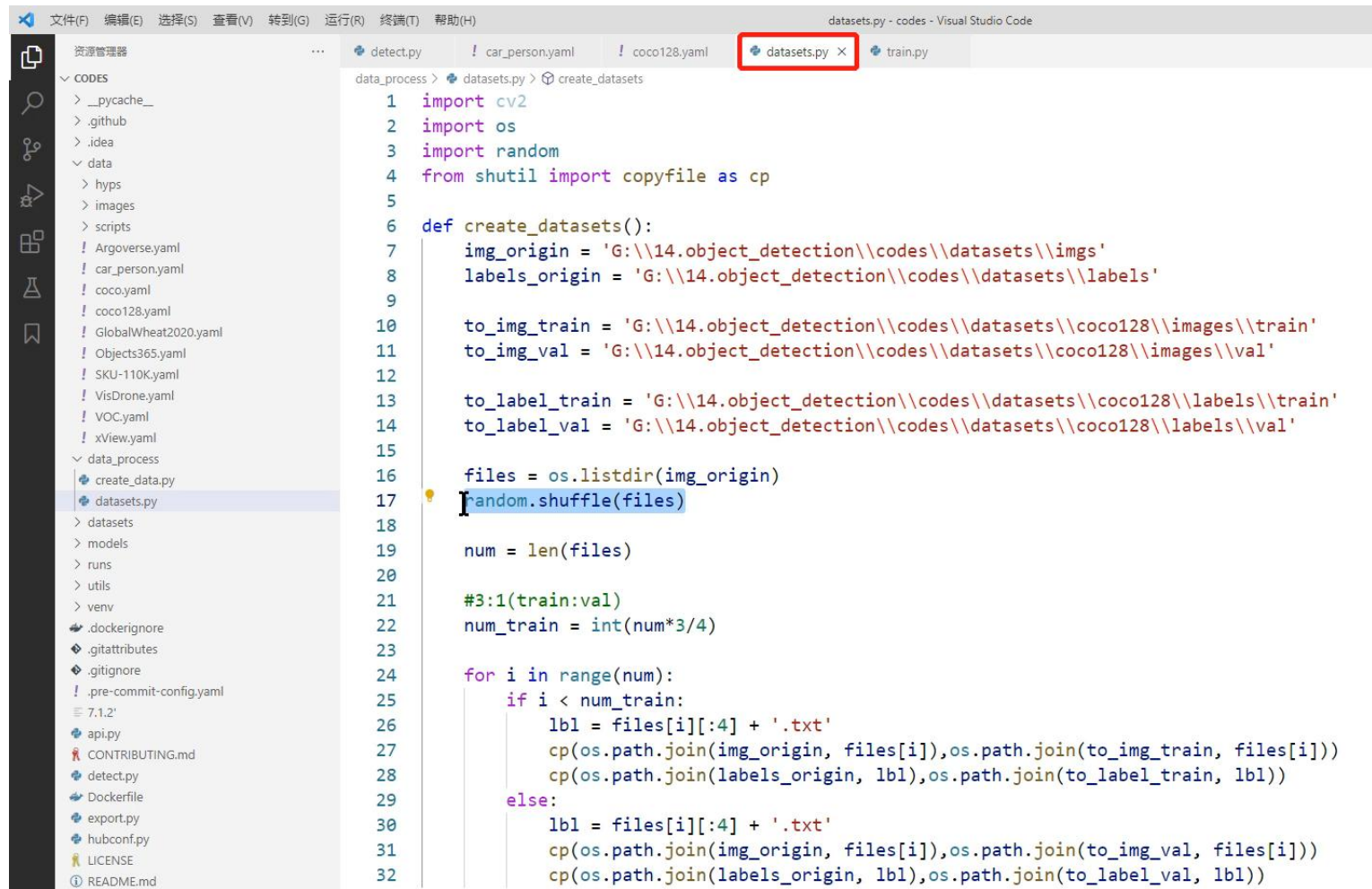


```
1 # COCO128 dataset https://www.kaggle.com/ultralytics/coco128 (first 128 images from COCO train2017) by Ultralytics
2 # Example usage: python train.py --data coco128.yaml
3 # parent
4 # |
5 # | yoloV5
6 # | datasets
7 # | coco128 ← downloads here
8
9
10 # Train/val/test sets as 1) dir: path/to/imgs, 2) file: path/to/imgs.txt, or 3) list: [path/to/imgs1, path/to/imgs2, ..]
11 path: datasets/coco128 # dataset root dir
12 train: images/train # train images (relative to 'path') 128 images
13 val: images/val # val images (relative to 'path') 128 images
14 test: # test images (optional)
15
16 # Classes
17 nc: 5 # number of classes
18 names: ['dog', 'person', 'cat', 'tv', 'car']
19 # names: ['person', 'bicycle', 'car', 'motorcycle', 'airplane', 'bus', 'train', 'truck', 'boat', 'traffic light',
20 # 'fire hydrant', 'stop sign', 'parking meter', 'bench', 'bird', 'cat', 'dog', 'horse', 'sheep', 'cow',
21 # 'elephant', 'bear', 'zebra', 'giraffe', 'backpack', 'umbrella', 'handbag', 'tie', 'suitcase', 'frisbee',
22 # 'skis', 'snowboard', 'sports ball', 'kite', 'baseball bat', 'baseball glove', 'skateboard', 'surfboard',
23 # 'tennis racket', 'bottle', 'wine glass', 'cup', 'fork', 'knife', 'spoon', 'bowl', 'banana', 'apple',
24 # 'sandwich', 'orange', 'broccoli', 'carrot', 'hot dog', 'pizza', 'donut', 'cake', 'chair', 'couch',
25 # 'potted plant', 'bed', 'dining table', 'toilet', 'tv', 'laptop', 'mouse', 'remote', 'keyboard', 'cell phone',
26 # 'microwave', 'oven', 'toaster', 'sink', 'refrigerator', 'book', 'clock', 'vase', 'scissors', 'teddy bear',
27 # 'hair drier', 'toothbrush'] # class names
28
29
```

train.py



```
450 compute_loss=compute_loss) # val best model with plots
451
452 if is_coco:
453     callbacks.run('on_fit_epoch_end', list(mloss) + list(results) + lr, epoch, best_fitness, fi)
454
455 callbacks.run('on_train_end', last, best, plots, epoch, results)
456 LOGGER.info(f'Results saved to {colorstr('bold', save_dir)}')
457
458 torch.cuda.empty_cache()
459 return results
460
461 def parse_opt(known=False):
462     parser = argparse.ArgumentParser()
463     parser.add_argument('--weights', type=str, default=ROOT / 'yoloV5s.pt', help='initial weights path')
464     parser.add_argument('--cfg', type=str, default='', help='model.yaml path')
465     parser.add_argument('--data', type=str, default=ROOT / 'data/car_person.yaml', help='dataset.yaml path')
466     parser.add_argument('--hyp', type=str, default=ROOT / 'data/hyps/hyp.scratch-low.yaml', help='hyperparameters path')
467     parser.add_argument('--epochs', type=int, default=300)
468     parser.add_argument('--batch-size', type=int, default=1, help='total batch size for all GPUs, -1 for autobatch')
469     parser.add_argument('--imgsz', '--img', '--img-size', type=int, default=640, help='train, val image size (pixels)')
470     parser.add_argument('--rect', action='store_true', help='rectangular training')
471     parser.add_argument('--resume', nargs='?', const=True, default=False, help='resume most recent training')
472     parser.add_argument('--nosave', action='store_true', help='only save final checkpoint')
473     parser.add_argument('--noval', action='store_true', help='only validate final epoch')
474     parser.add_argument('--noautoanchor', action='store_true', help='disable AutoAnchor')
475     parser.add_argument('--evolve', type=int, nargs='?', const=300, help='evolve hyperparameters for x generations')
476     parser.add_argument('--bucket', type=str, default='', help='gsutil bucket')
477     parser.add_argument('--cache', type=str, nargs='?', const='ram', help='--cache images in "ram" (default) or "disk"')
478     parser.add_argument('--image-weights', action='store_true', help='use weighted image selection for training')
479     parser.add_argument('--device', default='', help='cuda device, i.e. 0 or 0,1,2,3 or cpu')
480     parser.add_argument('--multi-scale', action='store_true', help='vary img-size +/- 50%')
481     parser.add_argument('--single-cls', action='store_true', help='train multi-class data as single-class')
```



```
1 import cv2
2 import os
3 import random
4 from shutil import copyfile as cp
5
6 def create_datasets():
7     img_origin = 'G:\\14.object_detection\\codes\\datasets\\imgs'
8     labels_origin = 'G:\\14.object_detection\\codes\\datasets\\labels'
9
10    to_img_train = 'G:\\14.object_detection\\codes\\datasets\\coco128\\images\\train'
11    to_img_val = 'G:\\14.object_detection\\codes\\datasets\\coco128\\images\\val'
12
13    to_label_train = 'G:\\14.object_detection\\codes\\datasets\\coco128\\labels\\train'
14    to_label_val = 'G:\\14.object_detection\\codes\\datasets\\coco128\\labels\\val'
15
16    files = os.listdir(img_origin)
17    random.shuffle(files)
18
19    num = len(files)
20
21    #3:1(train:val)
22    num_train = int(num*3/4)
23
24    for i in range(num):
25        if i < num_train:
26            lbl = files[i][:4] + '.txt'
27            cp(os.path.join(img_origin, files[i]), os.path.join(to_img_train, files[i]))
28            cp(os.path.join(labels_origin, lbl), os.path.join(to_label_train, lbl))
29        else:
30            lbl = files[i][:4] + '.txt'
31            cp(os.path.join(img_origin, files[i]), os.path.join(to_img_val, files[i]))
32            cp(os.path.join(labels_origin, lbl), os.path.join(to_label_val, lbl))
```

5、模型训练

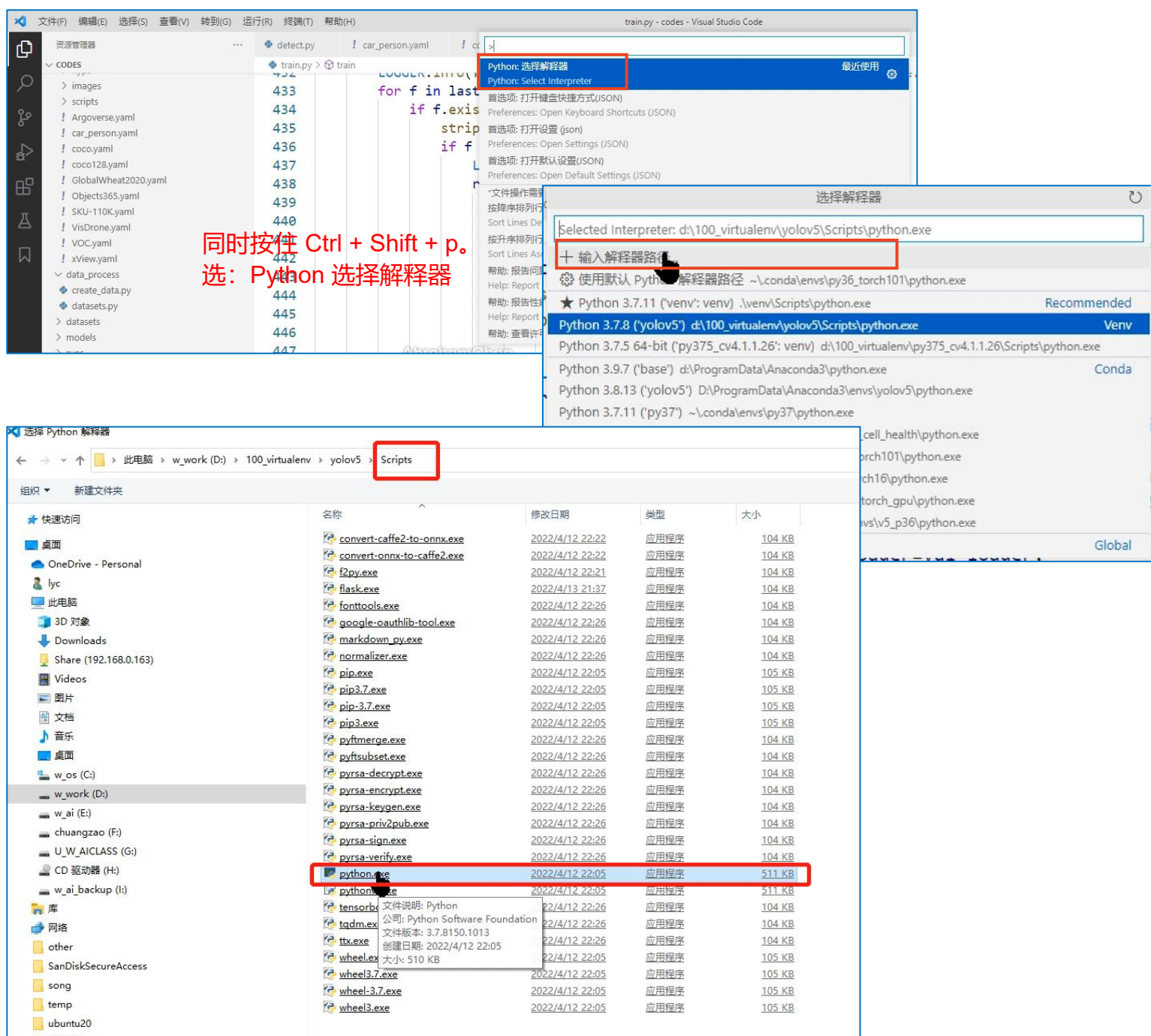
1.预训练模型下载:

<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。



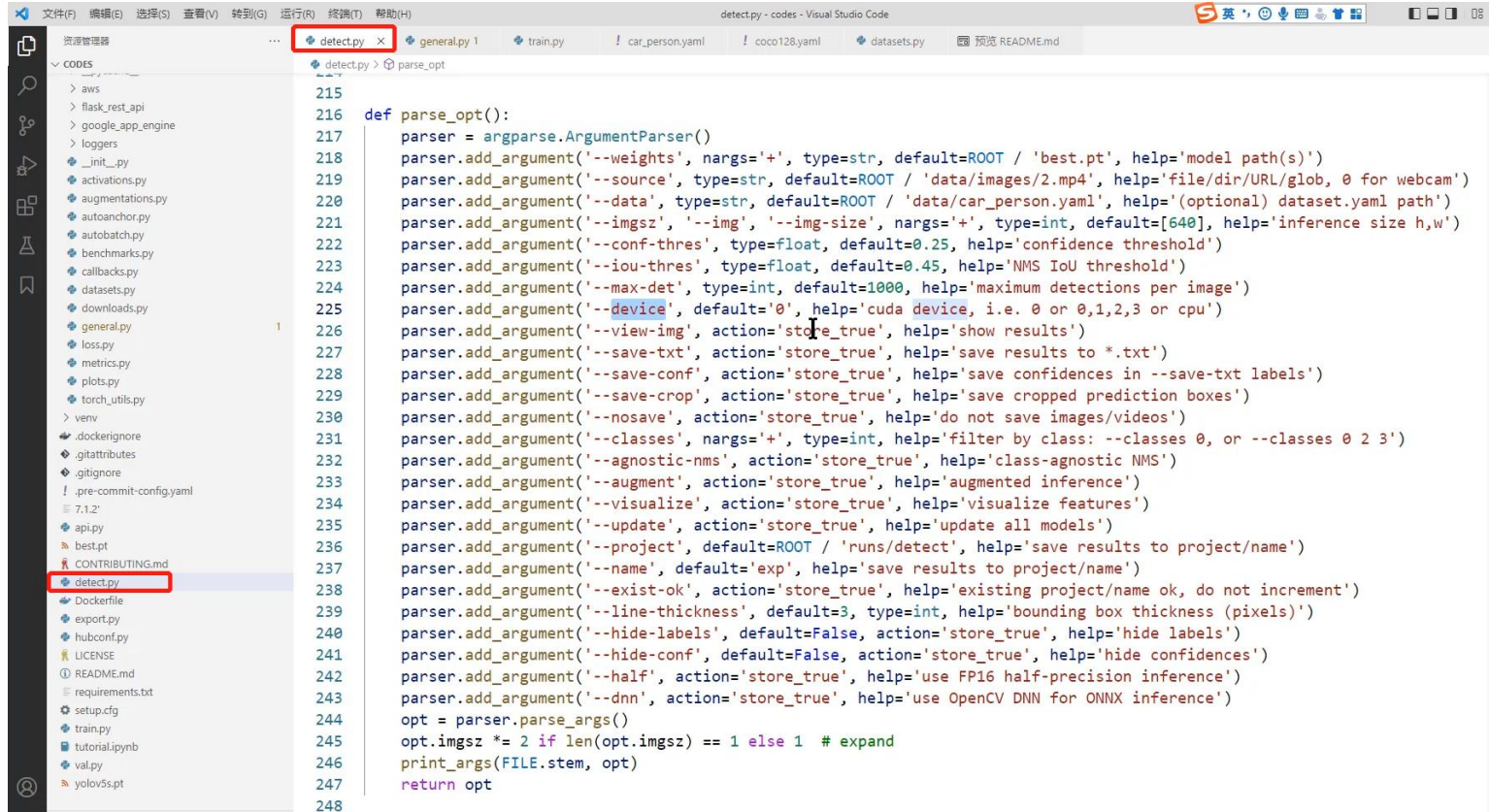
总结:

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



```
215
216 def parse_opt():
217     parser = argparse.ArgumentParser()
218     parser.add_argument('--weights', nargs='+', type=str, default=ROOT / 'best.pt', help='model path(s)')
219     parser.add_argument('--source', type=str, default=ROOT / 'data/images/2.mp4', help='file/dir/URL/glob, 0 for webcam')
220     parser.add_argument('--data', type=str, default=ROOT / 'data/car_person.yaml', help='(optional) dataset.yaml path')
221     parser.add_argument('--imgsz', '--img', '--img-size', nargs='+', type=int, default=[640], help='inference size h,w')
222     parser.add_argument('--conf-thres', type=float, default=0.25, help='confidence threshold')
223     parser.add_argument('--iou-thres', type=float, default=0.45, help='NMS IoU threshold')
224     parser.add_argument('--max-det', type=int, default=1000, help='maximum detections per image')
225     parser.add_argument('--device', default='0', help='cuda device, i.e. 0 or 0,1,2,3 or cpu')
226     parser.add_argument('--view-img', action='store_true', help='show results')
227     parser.add_argument('--save-txt', action='store_true', help='save results to *.txt')
228     parser.add_argument('--save-conf', action='store_true', help='save confidences in --save-txt labels')
229     parser.add_argument('--save-crop', action='store_true', help='save cropped prediction boxes')
230     parser.add_argument('--nosave', action='store_true', help='do not save images/videos')
231     parser.add_argument('--classes', nargs='+', type=int, help='filter by class: --classes 0, or --classes 0 2 3')
232     parser.add_argument('--agnostic-nms', action='store_true', help='class-agnostic NMS')
233     parser.add_argument('--augment', action='store_true', help='augmented inference')
234     parser.add_argument('--visualize', action='store_true', help='visualize features')
235     parser.add_argument('--update', action='store_true', help='update all models')
236     parser.add_argument('--project', default=ROOT / 'runs/detect', help='save results to project/name')
237     parser.add_argument('--name', default='exp', help='save results to project/name')
238     parser.add_argument('--exist-ok', action='store_true', help='existing project/name ok, do not increment')
239     parser.add_argument('--line-thickness', default=3, type=int, help='bounding box thickness (pixels)')
240     parser.add_argument('--hide-labels', default=False, action='store_true', help='hide labels')
241     parser.add_argument('--hide-conf', default=False, action='store_true', help='hide confidences')
242     parser.add_argument('--half', action='store_true', help='use FP16 half-precision inference')
243     parser.add_argument('--dnn', action='store_true', help='use OpenCV DNN for ONNX inference')
244     opt = parser.parse_args()
245     opt.imgsz *= 2 if len(opt.imgsz) == 1 else 1 # expand
246     print_args(FILE.stem, opt)
247     return opt
248
```

总结: