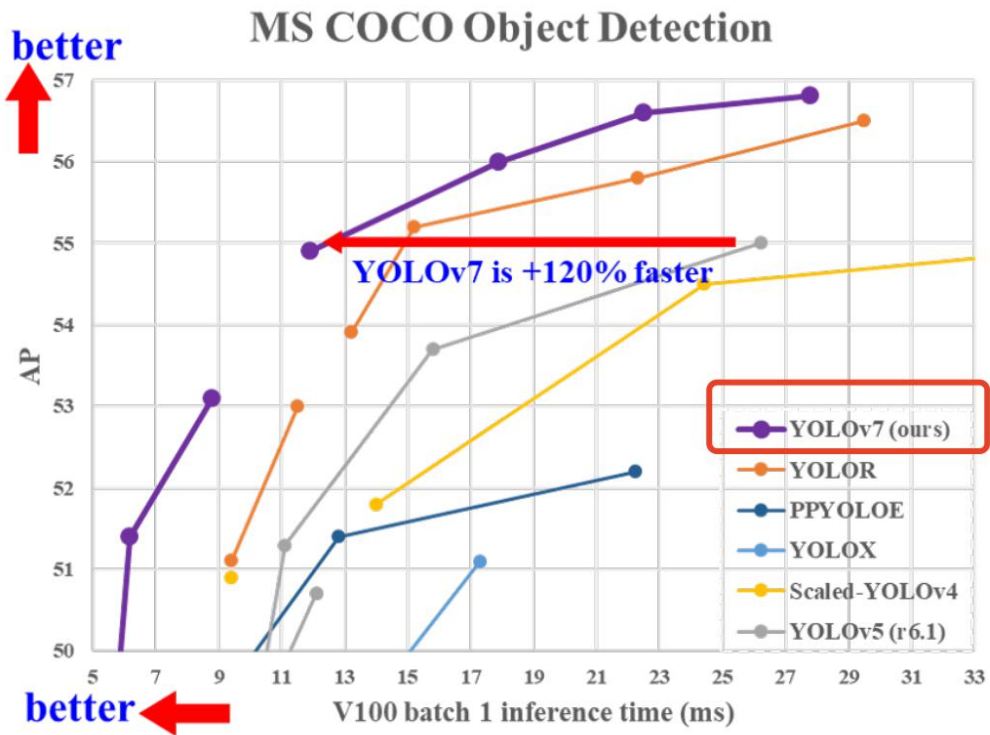


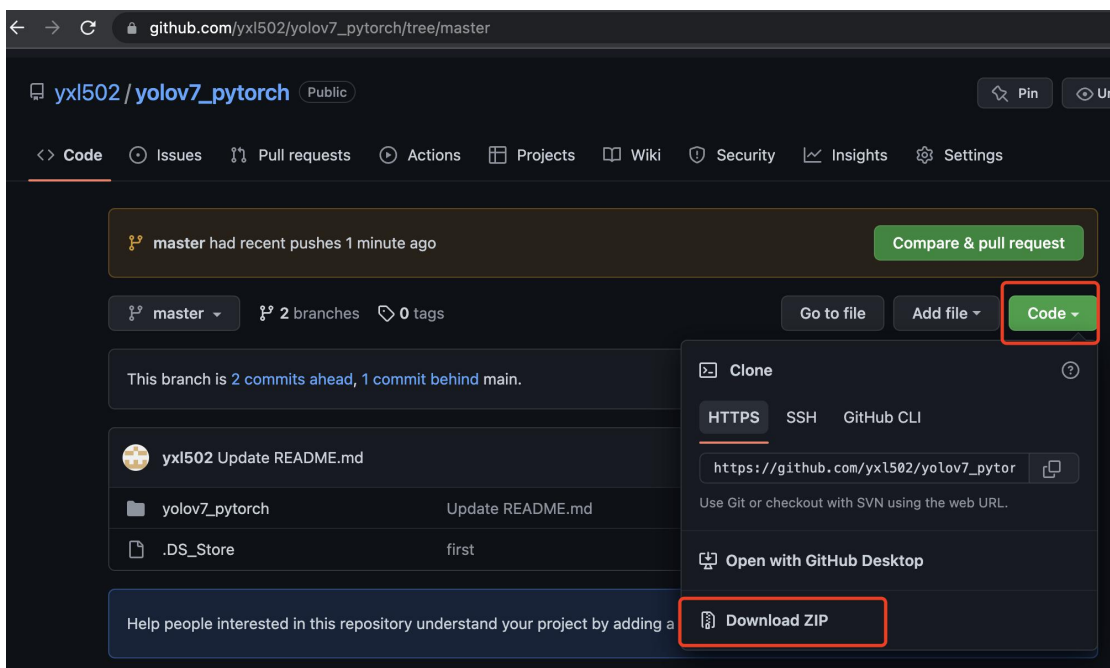
YOLO V7 小浣熊检测

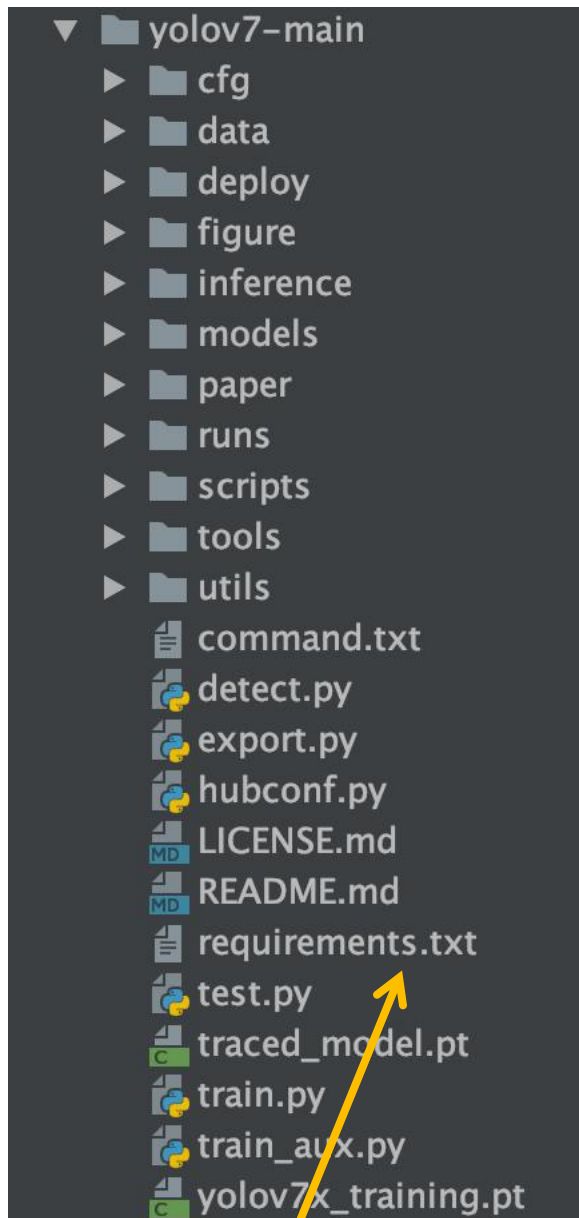


一、准备工作

1、代码克隆

Code: https://github.com/yxl502/yolov7_pytorch/tree/master





2、环境安装

```
cd yolov7-main  
pip install -r requirements.txt
```

3、权重下载

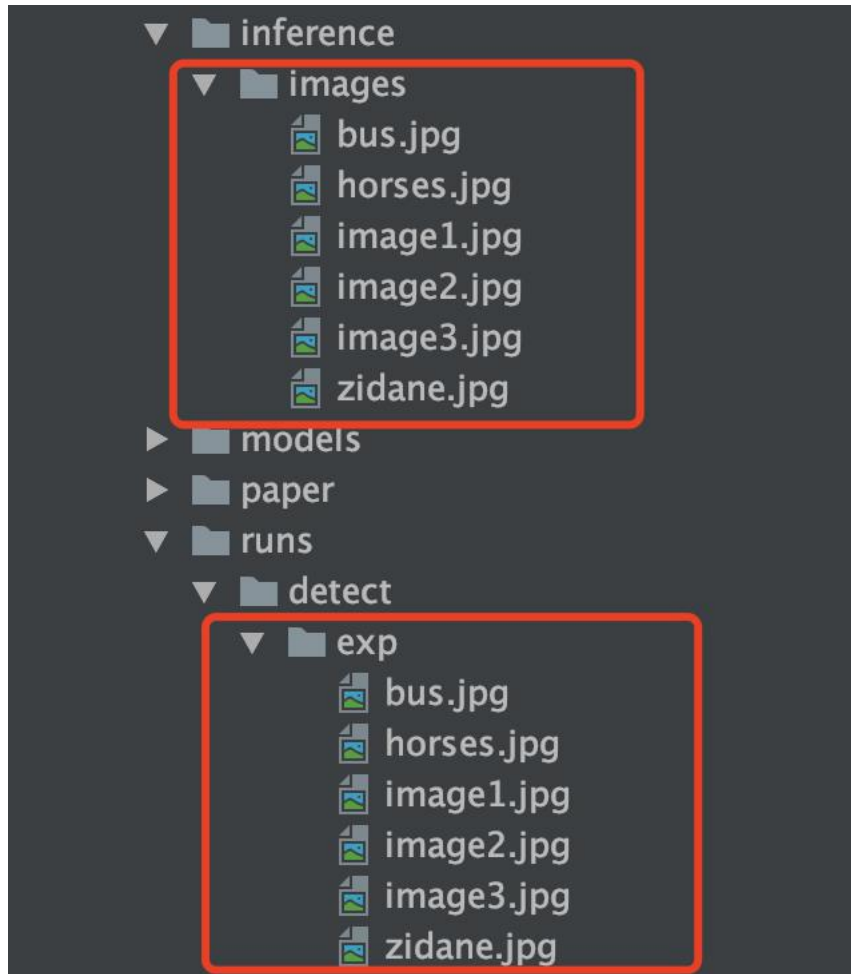
我们使用的预训练模型是 yolov7x_training.pt, 这个模型是在 MS COCO 数据集上进行训练的, 我们把这个模型下载到 yolov7-main/下

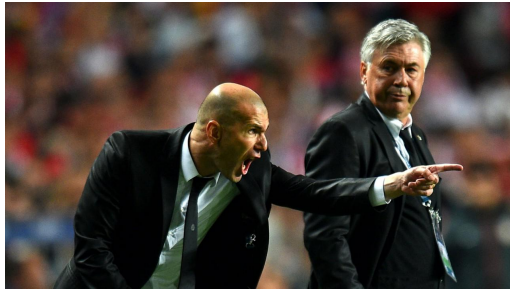
4、测试一下

```
python detect.py --weights yolov7x_training.pt --source inference/images
```

```
The image with the result is saved in: runs/detect/exp/bus.jpg
The image with the result is saved in: runs/detect/exp/horses.jpg
The image with the result is saved in: runs/detect/exp/image1.jpg
The image with the result is saved in: runs/detect/exp/image2.jpg
The image with the result is saved in: runs/detect/exp/image3.jpg
The image with the result is saved in: runs/detect/exp/zidane.jpg
Done. (3.963s)
```

运行成功，预测的图片被保存在了./runs/detect/exp/文件夹下

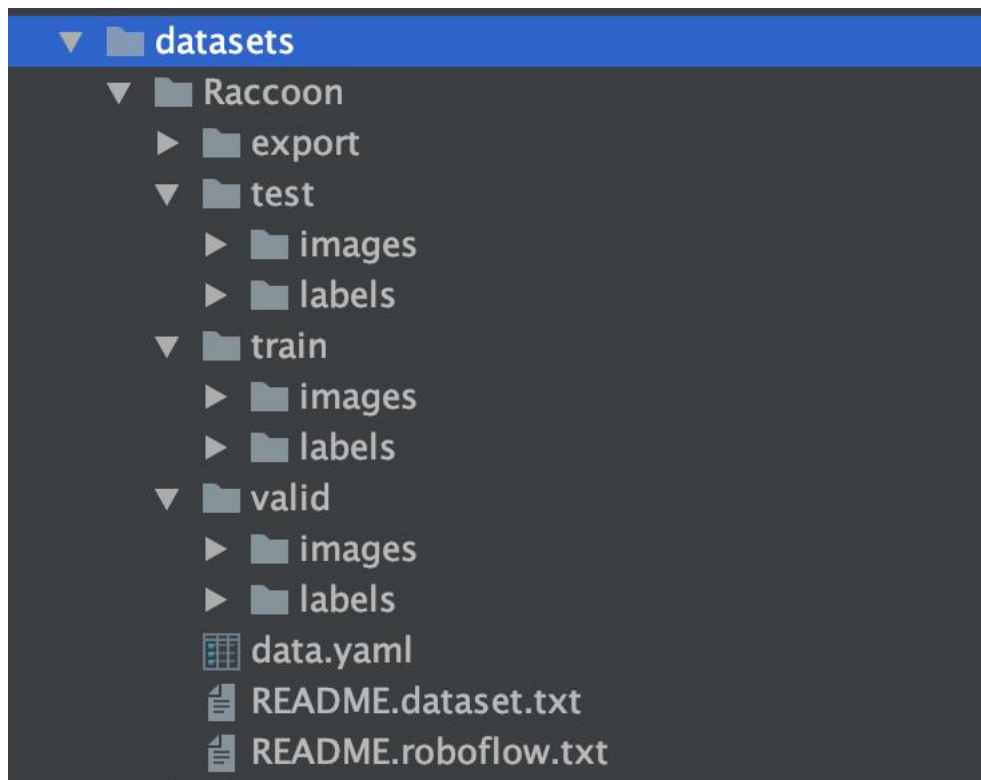




二、数据准备

1、准备 yolo 格式的数据

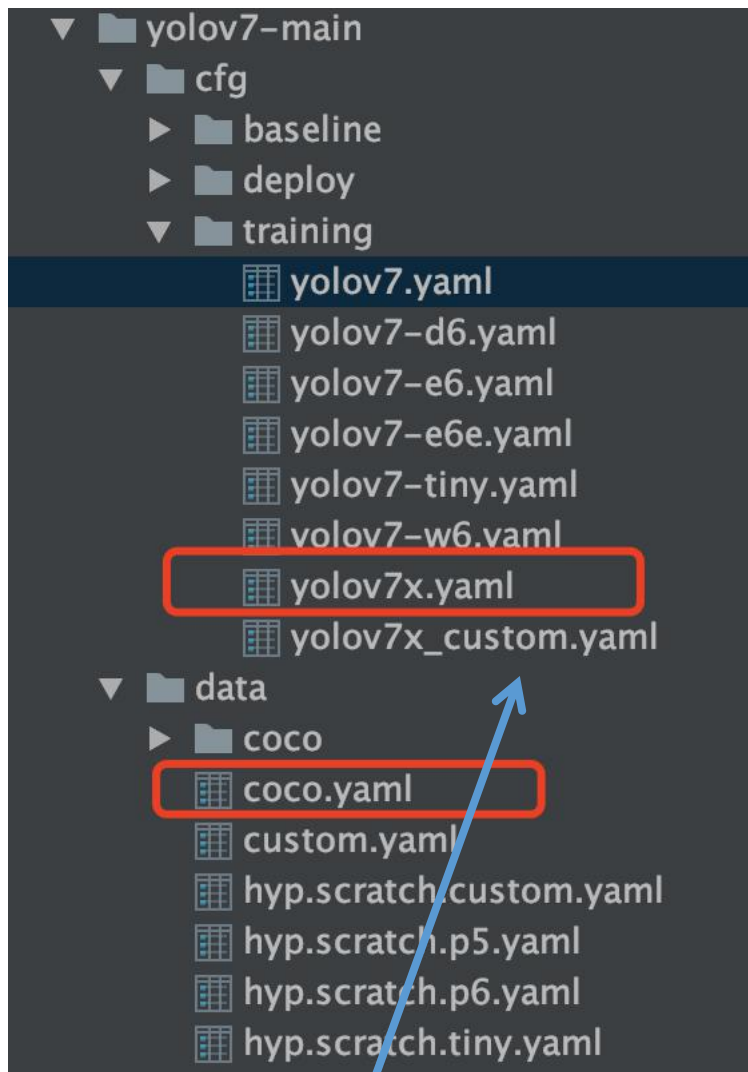
数据下载地址：链接：<https://pan.baidu.com/s/1W-MxEKyj0fPh4qf-UrvRQQ> 提取码：a6a1



三、配置训练的相关文件

1、配置文件说明

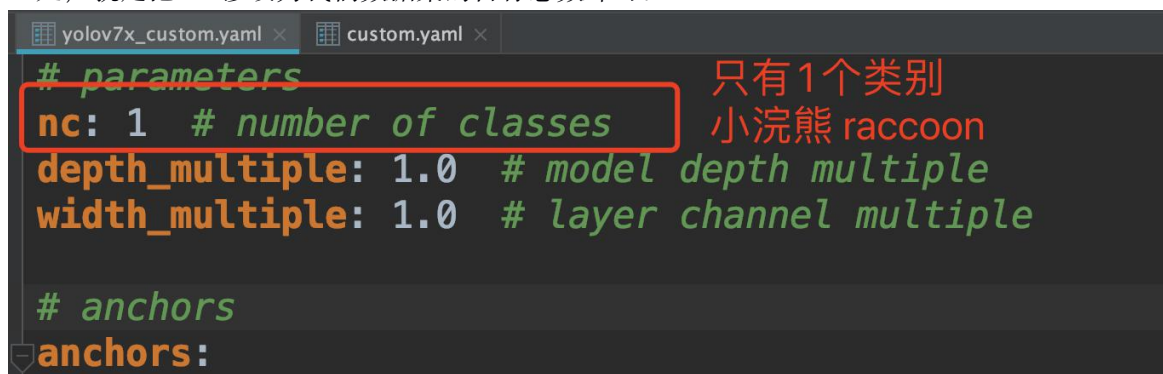
总共有两个文件需要配置，一个是 `yolov7-main/cfg/training/yolov7x.yaml`，这个文件是有关模型的配置文件；一个是 `yolov7-main/data/coco.yaml`，这个是数据集的配置文件。



1、配置模型文件

第一步，复制 yolov7x.yaml 文件到相同的路径下，然后重命名，我们重命名为 yolov7x_custom.yaml。

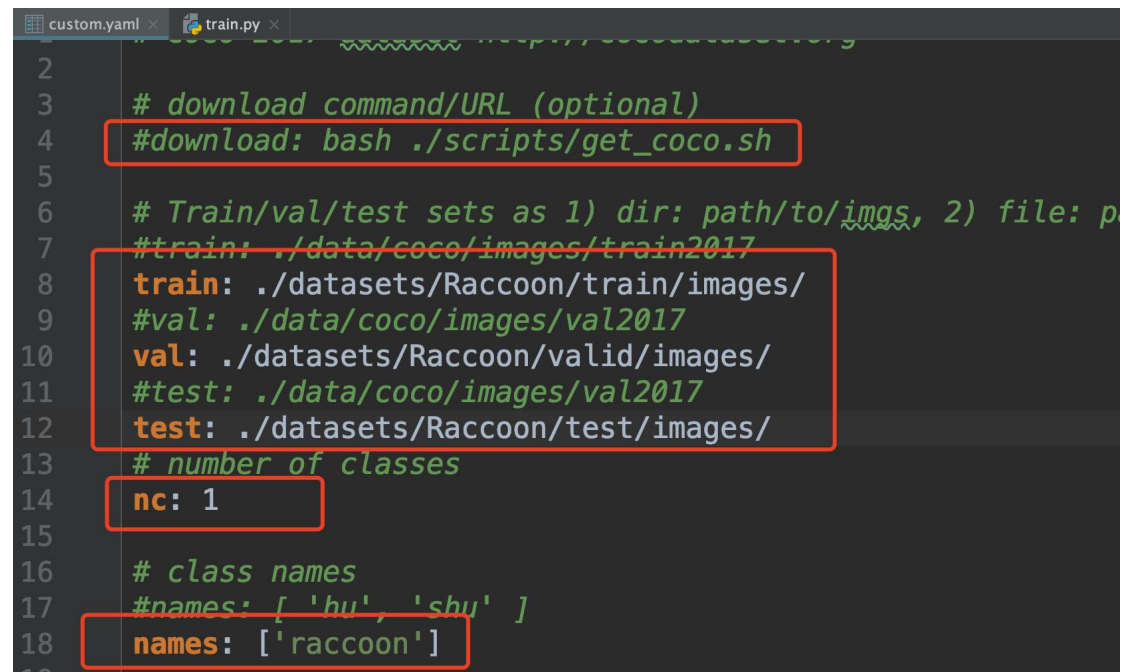
第二步，打开 yolov7x_custom.yaml 文件，进行如下图所示的修改，这里修改的地方只有一处，就是把 nc 修改为我们数据集的目标总数即可。



2、配置数据文件

第一步，复制 `coco.yaml` 文件到相同的路径下，然后重命名，我们命名为 `custom.yaml`。

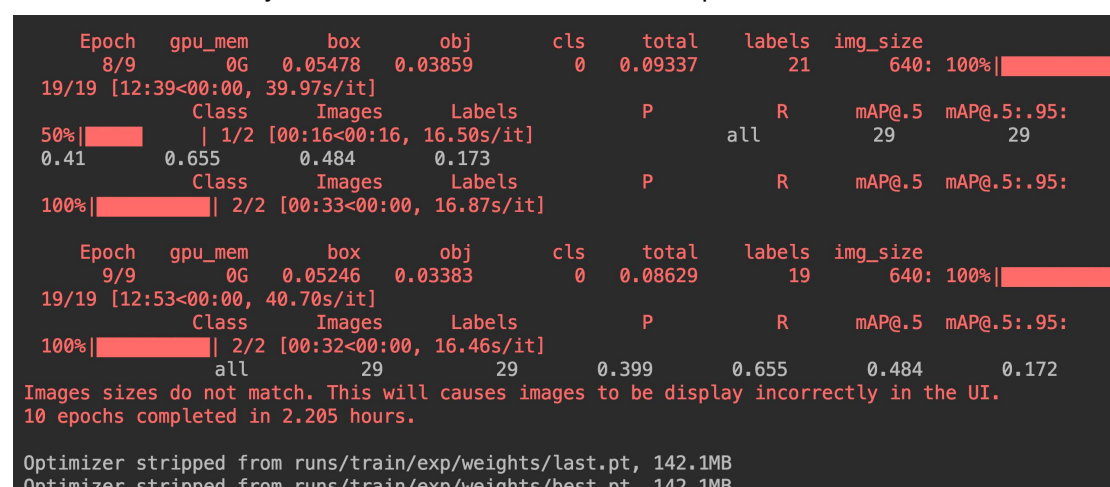
第二步，打开 `custom.yaml` 文件，进行如下所示的修改，需要修改的地方为 6 处。第一处：把代码自动下载 COCO 数据集的命令注释掉，以防代码自动下载数据集占用内存；第二处：修改 `train` 的路径；第三处：修改 `val` 的位置路径；第四处：修改 `test` 的位置路径；第五处：修改 `nc` 为数据集目标总数；第六处：修改 `names` 为数据集所有目标的名称。然后保存。



```
2
3 # download command/URL (optional)
4 #download: bash ./scripts/get_coco.sh
5
6 # Train/val/test sets as 1) dir: path/to/imgs, 2) file: p
7 #train: ./data/coco/images/train2017
8 train: ./datasets/Raccoon/train/images/
9 #val: ./data/coco/images/val2017
10 val: ./datasets/Raccoon/valid/images/
11 #test: ./data/coco/images/val2017
12 test: ./datasets/Raccoon/test/images/
13 # number of classes
14 nc: 1
15
16 # class names
17 #names: [ 'hu', 'shu' ]
18 names: ['raccoon']
19
```

四、开始训练

```
python train.py --weights yolov7x_training.pt --cfg cfg/training/yolov7x_custom.yaml
--data data/custom.yaml --device 0 --batch-size 64 --epoch 10
```



```
Epoch 8/9  gpu_mem 0G  box 0.05478  obj 0.03859  cls 0  total 0.09337  labels 21  img_size 640: 100%|
19/19 [12:39<00:00, 39.97s/it]
Class Images Labels P R mAP@.5 mAP@.5:.95:
50%| 1/2 [00:16<00:16, 16.50s/it] all 29 29
0.41 0.655 0.484 0.173
Class Images Labels P R mAP@.5 mAP@.5:.95:
100%| 2/2 [00:33<00:00, 16.87s/it]
Epoch 9/9  gpu_mem 0G  box 0.05246  obj 0.03383  cls 0  total 0.08629  labels 19  img_size 640: 100%|
19/19 [12:53<00:00, 40.70s/it]
Class Images Labels P R mAP@.5 mAP@.5:.95:
100%| 2/2 [00:32<00:00, 16.46s/it] all 29 29 0.399 0.655 0.484 0.172
Images sizes do not match. This will causes images to be display incorrectly in the UI.
10 epochs completed in 2.205 hours.

Optimizer stripped from runs/train/exp/weights/last.pt, 142.1MB
Optimizer stripped from runs/train/exp/weights/best.pt, 142.1MB
```

可能遇到的错误: (没错，请忽略)

1、调用 Tensorboard 时报错 `AttributeError: module 'setuptools._distutils' has no attribute 'version'`

<https://blog.csdn.net/fightforglory/article/details/123782837>

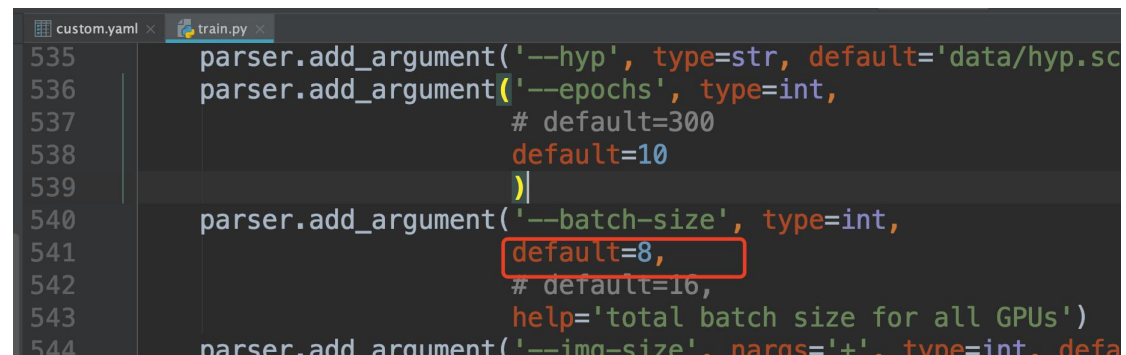
2、wandb.errors.UsageError: api_key not configured (no-tty). call
wandb.login(key=[your_api_key])

<https://blog.51cto.com/AIXhao/5247320>

3、Process finished with exit code 137 (interrupted by signal 9: SIGKILL)

程序并没有运行完，系统强制杀死进程，一般来说是因为内存不足或者 CPU 不够用。

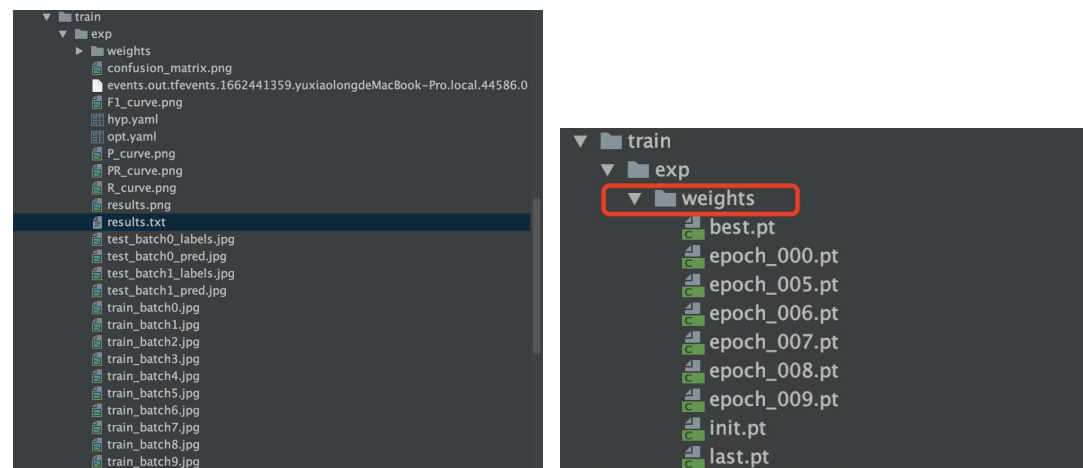
调整--batch-size



```
535 parser.add_argument('--hyp', type=str, default='data/hyp.sc
536 parser.add_argument('--epochs', type=int,
537                       # default=300
538                       default=10
539                       )
540 parser.add_argument('--batch-size', type=int,
541                     default=8,
542                     # default=16,
543                     help='total batch size for all GPUs')
544 parser.add_argument('--img-size', nargs='+', type=int, defa
```

五、训练结果与推理

在./runs/train/exp/下，保存了训练的结果文件，比如每个 epoch 的结果到保存到了 results.txt 文件里,在./runs/train/exp/weights/下保存了训练完成的模型文件，加载模型文件进行推理。



python detect.py --weights runs/train/exp/weights/best.pt --source
inference/raccoon_images