

搞定系列：yolov5 训练自己数据集 windows 版 (云未归来)

课程链接：

1. <https://download.csdn.net/course/detail/36338>

你将收获

- 1、学会如何搭建yolov5环境
- 2、学会如何使用labellmg标注自己的数据集
- 3、学会如何转换自己的数据集到yolov5训练所需要格式
- 4、学会如何训练和测试自己模型

适用人群

想要入门yolov5的学者

课程介绍

本课程将带您进入yolov5框架搭建环境，标注，训练，测试等完整实现过程。主要内容如下

- 1、yolov5框架基本介绍
- 2、搭建windows上yolov5环境
- 3、用labellmg标注自己的数据集
- 4、转换数据集
- 5、配置参数
- 6、参数调节和训练
- 7、测试自己的模型

通过本课程您将学习到yolov5整个标注数据集和训练流程，本课程全部在windows10上实现，使用的GPU环境

课程目录

▶ yolov5框架基本介绍 免费

▶ 搭建windows上yolov5环境

▶ 用labellmg标注自己的数据集

▶ 转换数据集

▶ 配置参数

▶ 参数调节和训练

▶ 测试自己的模型

课程安排

- 1、^Iyolov5框架基本介绍
- 2、搭建windows上yolov5环境
- 3、用labellmg标注自己的数据集
- 4、转换数据集
- 5、配置参数
- 6、参数调节和训练
- 7、测试自己的模型

07:02

24:04

12:25

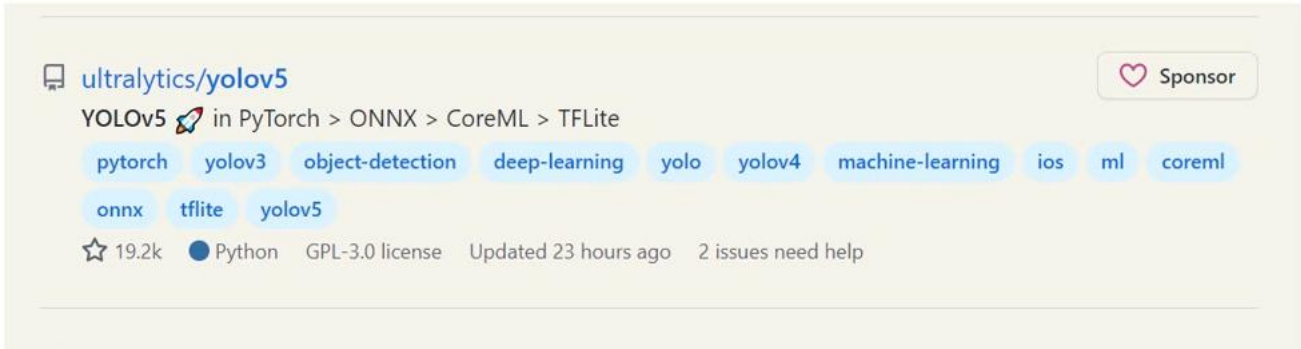
13:03

03:52

21:03

08:21

1、yolov5 框架基本介绍



下载地址：<https://github.com/ultralytics/yolov5>

本节内容				
yolov3,yolov4,yolov5区别				
		yolov3	yolov4	yolov5
输入端	数据增强		Mosaic数据增强	Mosaic数据增强
	锚框	anchor	anchor	自适应anchor
主干网络（backbone）	网络	Darknet53	CSPDarknet53	Focus+CSP
	归一化	BN	CMBN	BN
	激活函数	Leaky relu	Mish	Leaky relu
	正则化	无	Dropblock	
Neck		FPN+PAN	SPP+FPN+PAN结构	FPN+PAN
损失函数		IOU	CIoU	GIOU
非极大值抑制		nms	DIOU_nms	nms
FPN：空间金字塔（多尺度）		自顶向下的，将高层的特征信息通过上采样的方式进行传递融合，得到进行预测的特征图。 特征融合或特征相加		
PAN：路径聚合网络				

yolov5 总结参考一个文章：
https://blog.csdn.net/weixin_38842821/article/details/108544609

2、搭建 windows 上 yolov5 环境

yolov5训练自己数据集windows版

第二节：搭建windows上yolov5环境

常见pytorch安装方法:

1、conda安装方法

conda install pytorch torchvision cudatoolkit=10.2 -c pytorch

2、官方网站安装方法

pip install torch==1.7.0 torchvision==0.8.1 -f https://download.pytorch.org/whl/torch_stable.html

3、最简单方法就是最好方法，省时省力

离线安装方法

anaconda下载: **IDLE**

官方: <https://www.anaconda.com/>

国内源: <https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/>

anaconda常用命令:

创建环境: conda create -n yolov5 python=3.8

激活环境: conda activate yolov5

退出环境: conda deactivate

安装模块: conda install module_name

删除环境: 到文件夹envs\直接删除对应环境

恢复默认源: conda config --remove-key channels

删除所有的安装包及cache: conda clean --all

导出所有包到TXT: conda list -e > requirements.txt

从TXT导入包: conda install --yes --file requirements.txt

导出环境: conda env export > myenv.yml

从环境文件创建环境: conda env create -f myenv.yml

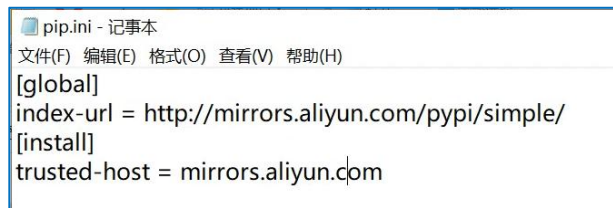
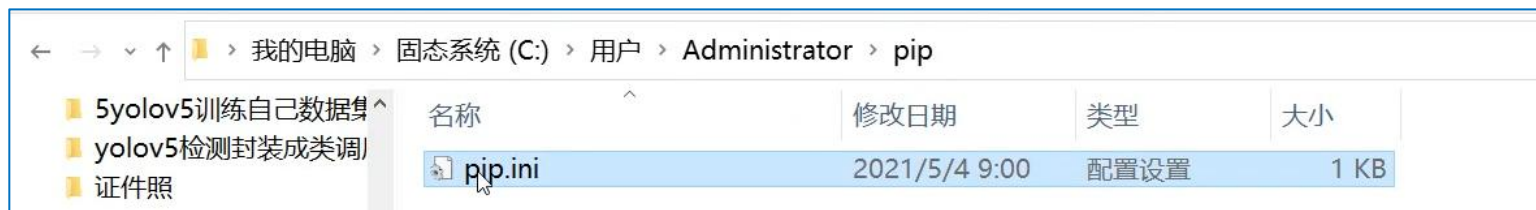
推荐安装到 C 盘之外的其它盘。推荐不要用含有**中文、空格或特殊字符**的路径。

在安装 anaconda 之前，推荐大家安装 VS2019，VS2015 和 VS2017 也是可以的。并且在安装时，其它插件都可以不勾选，但一定要勾选“使用C++的桌面开发”。否则后续安装不了 **pycocotools** (因为不能编译)。安装完 VS2019 后需要把 anaconda 终端重启，否则终端没反应。



总结:

接下来要安装 Cython，因为它是做编译用的。（在用户文件夹下的 pip.ini 文件里去修改和配置默认的镜像源。）在此基础上，才是安装 pycocotools。



```
(yolov5) C:\Users\Administrator>python
Python 3.8.12 (default, Oct 12 2021, 03:01:40) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import pycocotools
>>> exit()
```

离线安装方法:

下载cudatoolkit+cudnn

<https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/main/win-64/>

下载pytorch+torchvision

<https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/pytorch/win-64/>

如果您的显卡型号<=RTX2080

推荐pytorch==1.7.0 torchvision==0.8.1 cudatoolkit==10.1 cudnn==7.6.5

如果您的显卡型号>RTX2080

推荐pytorch==1.7.0 torchvision==0.8.1 cudatoolkit==11.0

注意：其实其他版本也是可以的，大家可以自行选择

pytorch==1.7.0 torchvision==0.8.1已经在windows/ubuntu上被我工作时候在yolov5-3.1,yolov5-4.0,yolov5-5.0,yolov5-6.0均可以使用。

```
(yolov5) C:\Users\Administrator>cd /d I:

(yolov5) I:\>conda install cudatoolkit-10.1.243-h74a9793_0.tar.bz2
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(yolov5) I:\>conda install cudnn-7.6.5-cuda10.1_0.tar.bz2
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(yolov5) I:\>conda install pytorch-1.7.0-py3.8_cuda101_cudnn7_0.tar.bz2
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(yolov5) I:\>conda install torchvision-0.8.1-py37_cu101.tar.bz2

Downloading and Extracting Packages
##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(yolov5) I:\>
```

总结:

3、用 labellmg 标注自己的数据集

1、学会在windows上安装labellmg

2、学会labellmg标注

安装 labellmg (可以安装在 base 环境下, 也建议这样子做)

```
pip install -i http://mirrors.aliyun.com/pypi/simple/ --trusted-host mirrors.aliyun.com labellmg
```

选择管理员: Anaconda Prompt (anaconda3)

```
(base) C:\Users\Administrator>pip install labellmg
Looking in indexes: http://mirrors.aliyun.com/pypi/simple/
Requirement already satisfied: labellmg in e:\anaconda3\lib\site-packages (1.8.6)
Requirement already satisfied: pyqt5 in e:\anaconda3\lib\site-packages (from labellmg) (5.15.4)
Requirement already satisfied: lxml in e:\anaconda3\lib\site-packages (from labellmg) (4.6.1)
Requirement already satisfied: PyQt5-sip<13,>=12.8 in e:\anaconda3\lib\site-packages (from pyqt5->labellmg) (12.9.0)
Requirement already satisfied: PyQt5-Qt5>=5.15 in e:\anaconda3\lib\site-packages (from pyqt5->labellmg) (5.15.2)
```

管理员: Anaconda Prompt (anaconda3) - labellmg

```
(base) C:\Users\Administrator>labellmg
```

打开 labellmg

1	Ctrl + u	加载目录中的所有图像, 鼠标点击Open dir同功能
2	Ctrl + r	更改默认注释目标目录(xml文件保存的地址)
3	Ctrl + s	保存
4	Ctrl + d	复制当前标签和矩形框
5	space	将当前图像标记为已验证
6	w	创建一个矩形框
7	d	下一张图片
8	a	上一张图片
9	del	删除选定的矩形框
10	Ctrl++	放大
11	Ctrl--	缩小
12	↑↓←→	键盘箭头移动选定的矩形框

用labellmg标注:
快捷键
w: 创建矩形框
D: 下一张图片
A: 上一张图片

labellmg

File Edit View Help

Open
Open Dir
Change Save Dir
Next Image
Prev Image
Verify Image
Save
PascalVOC
Create RectBox
Duplicate RectBox

labellmg - Save annotations to the directory

我的电脑 > Desktop >

搜索"Desktop"

快速访问

- Desktop
- 文档
- 图片
- official-m
- 360Down
- ai
- apex
- BaiduNet
- caffe-gpu
- CentOS 7

cuda算力查询

images labels yolov5-6.0 不常用

DLL合集 临时备份空间

文件夹: Desktop

建议图像文件与标签文件分文件夹开存放。当然放同一个文件夹也是可以的。

labellmg C:\Users\Administrator\Desktop\images\0000000006842.jpg [3 / 58]

File Edit View Help

☒ Auto Save mode
☐ Single Class Mode
☒ Display Labels
☒ Show/Hide Label Panel
☐ Advanced Mode

Ctrl+Shift+S
Ctrl+Shift+P
Ctrl+Shift+L
Ctrl+Shift+A

Hide RectBox Ctrl+H
Show RectBox Ctrl+A
Zoom In Ctrl++
Zoom Out Ctrl+--
Original size Ctrl+=
Fit Window Ctrl+F
Fit Width Ctrl+Shift+F

勾选这里

标注的要素:

- ① 不能错标。
- ② 不能漏标。
- ③ 标注框一定要尽可能紧框对象。不能太大, 也不能太小。

标注文件是 .xml 格式。建议至少标注100张以上的数据集。

总结:

4、转换数据集

学会将labellmg标注数据转换成yolov5训练的格式

转换流程:

- 1、将xml转成yolo格式的txt
- 2、将图片分割为训练集和测试集



假设图像的高和宽分别为 h, w , bbox 的左上角坐标为 (x_1, y_1) , 右下角坐标为 (x_2, y_2) , 则可求得 bbox 中心坐标 (x_c, y_c) 为:

$$x_c = x_1 + (x_2 - x_1) / 2 = (x_1 + x_2) / 2$$
$$y_c = y_1 + (y_2 - y_1) / 2 = (y_1 + y_2) / 2$$

假设 yolo 的5个数据分别为: label, x_c , y_c , w_c , h_c , 则有对应关系:

$$x_c = (x_1 + x_2) / 2w$$
$$y_c = (y_1 + y_2) / 2h$$
$$w_c = (x_2 - x_1) / w$$
$$h_c = (y_2 - y_1) / h$$

xml转换txt:

```
def convert(size, box):
    dw = 1. / (size[0])
    dh = 1. / (size[1])
    x = (box[0] + box[1]) / 2.0 - 1
    y = (box[2] + box[3]) / 2.0 - 1
    w = box[1] - box[0]
    h = box[3] - box[2]
    x = x * dw
    w = w * dw
    y = y * dh
    h = h * dh
    return (x, y, w, h)
```

总结:

xml转换txt:

```
def convert_annotation(xml_file,txt_file):
    in_file = open(xml_file)
    out_file = open(txt_file, 'w')
    tree=ET.parse(in_file)
    root = tree.getroot()          #这两行表示从xml文件中读取内容
    size = root.find('size')       #从读取到的xml内容中发现size
    w = int(size.find('width').text) #取宽度
    h = int(size.find('height').text) #取高度

    for obj in root.iter('object'):
        difficult = obj.find('difficult').text
        cls = obj.find('name').text
        cls_id = classes.index(cls)
        xmlbox = obj.find('bndbox')
        b = (float(xmlbox.find('xmin').text), float(xmlbox.find('xmax').text),
float(xmlbox.find('ymin').text), float(xmlbox.find('ymax').text))
        try:
            bb = convert((w,h), b)
        except:
            print(image_id)
        out_file.write(str(cls_id) + " " + " ".join([str(a) for a in bb]) + '\n')
    in_file.close()
    out_file.close()

if __name__ == '__main__':
    xml_dir=r'C:\Users\Administrator\Desktop\data\Annotations'

    save_dir=r'C:\Users\Administrator\Desktop\data\labels'
    if not os.path.exists(save_dir):
        os.makedirs(save_dir)
    for file in os.listdir(xml_dir):
        if file[-3:]!='xml':
            continue
        xml_file=os.path.join(xml_dir,file)
        convert_annotation(xml_file,os.path.join(save_dir,file[:-3]+'txt'))
```

分割数据集：

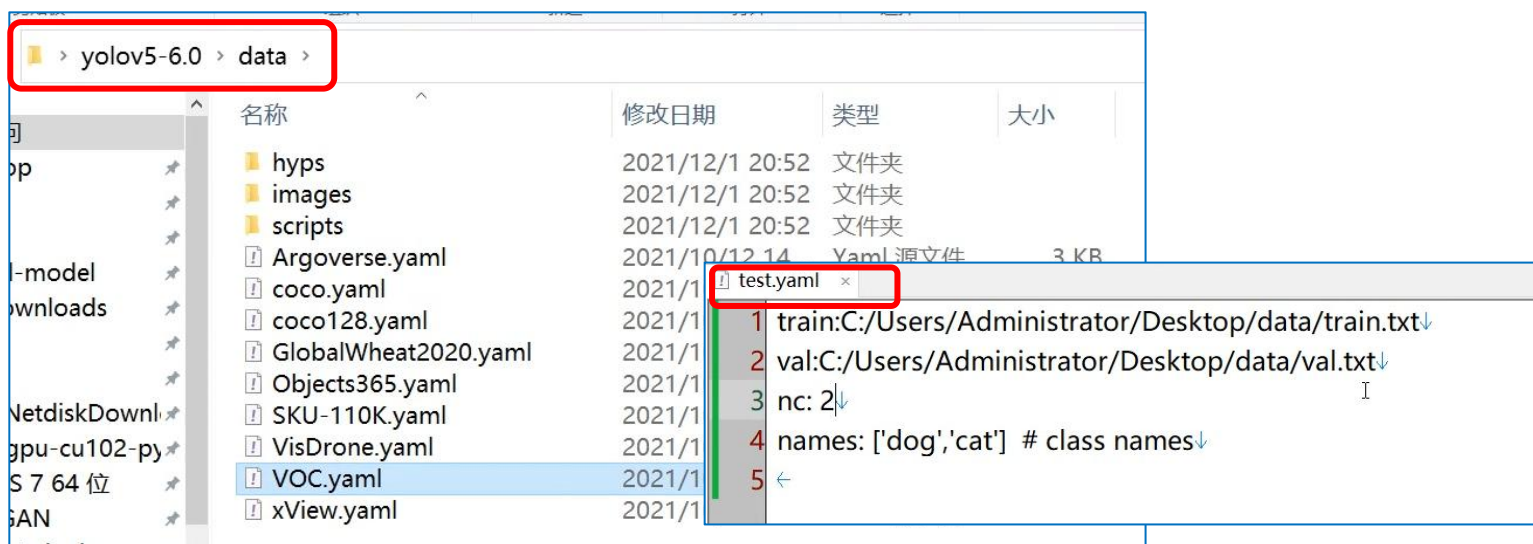
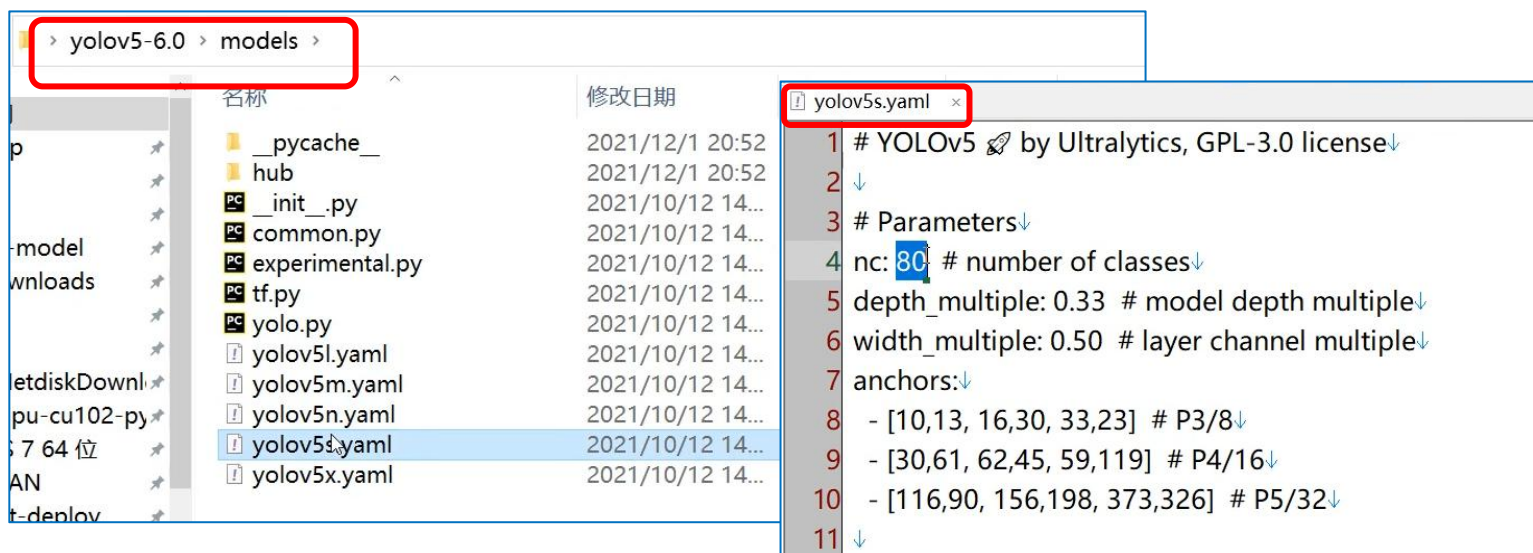
```
import os
import random
```

```
def convert(image_dir,train_ratio=0.9):
    file_list=[]
    for file in os.listdir(image_dir):
        image_file=os.path.join(image_dir,file)
        file_list.append(image_file)
    random.shuffle(file_list)
    train_count=int(len(file_list)*train_ratio)
    train_list=file_list[:train_count]
    val_list=file_list[train_count:]
    with open('train.txt','w') as f:
        f.write("\n".join(train_list))
    with open('val.txt','w') as f:
        f.write("\n".join(val_list))
image_dir=r'/home/wanghui/darknet-master/mydataset/VOC2012/JPEGImages'
convert(image_dir)
```

总结：

5、配置参数

- 1、修改网络配置文件
- 2、数据集配置文件



复制 VOC.yaml 文件，然后另文件名，如 test.yaml，然后修改这个文件里的内容。大部分内容都可以删掉，可以只删到只剩：train、val、nc、names，这几项。

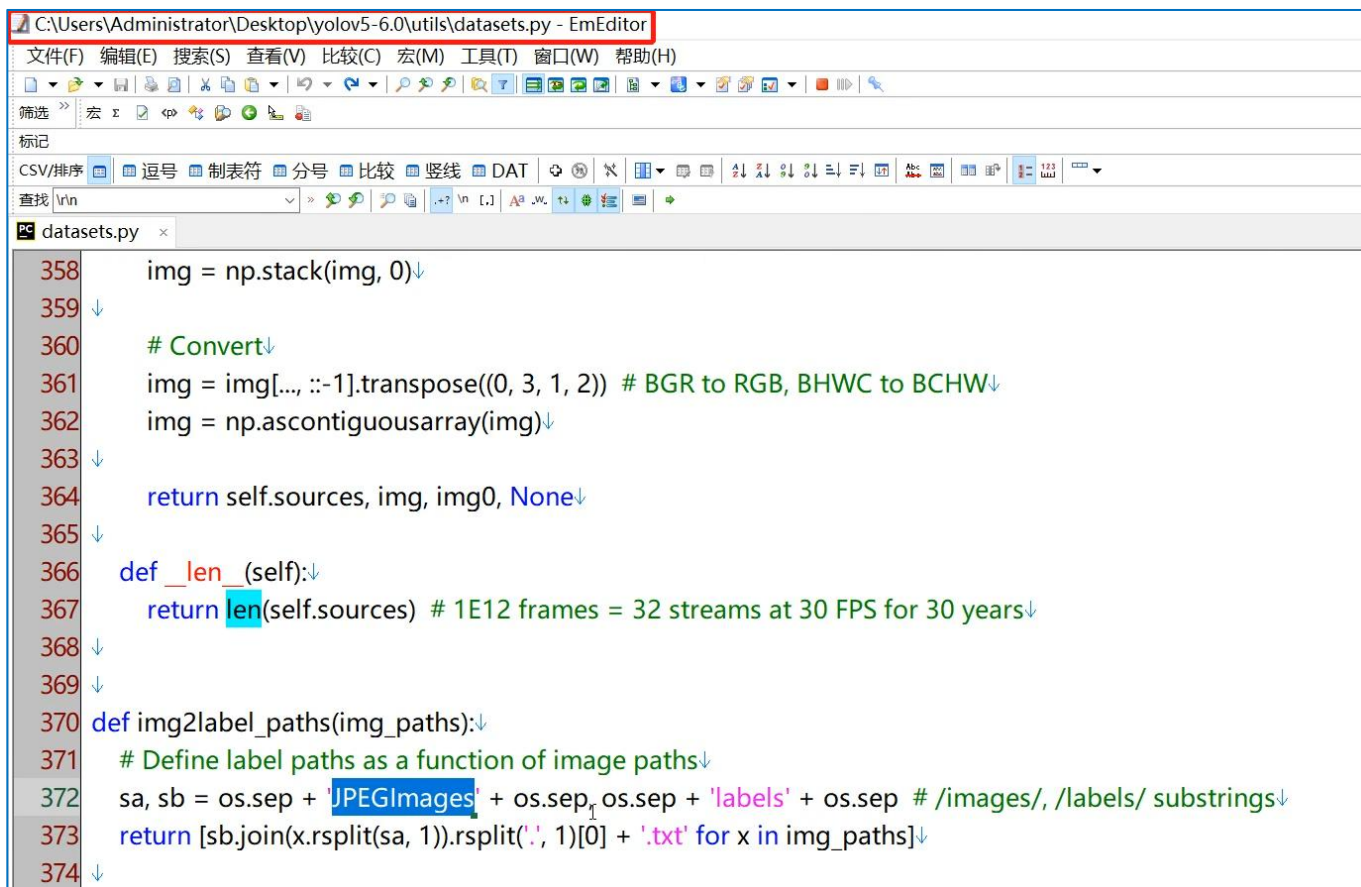


6、参数调节和训练

学会参数调节和训练

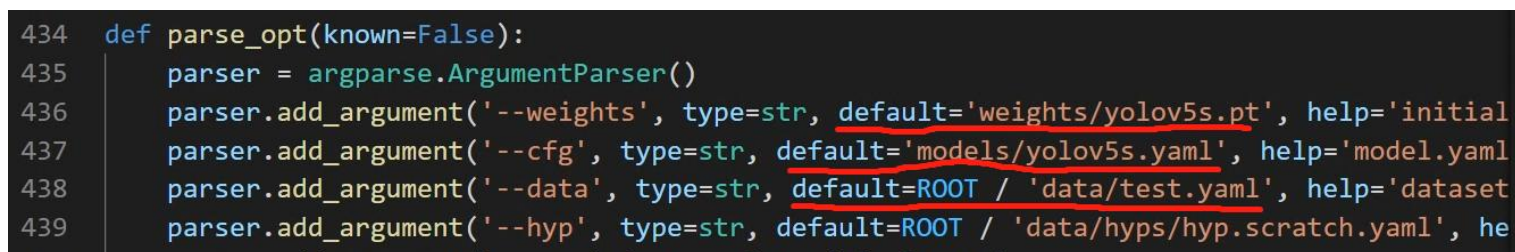
- 1、修改一些必要参数
- 2、训练技巧

- 修改 datasets.py 文件。



```
C:\Users\Administrator\Desktop\yolov5-6.0\utils\datasets.py - EmEditor
文件(F) 编辑(E) 搜索(S) 查看(V) 比较(C) 宏(M) 工具(T) 窗口(W) 帮助(H)
筛选 宏 逗号 制表符 分号 比较 竖线 DAT
查找 |r/n
datasets.py x
358 img = np.stack(img, 0)
359 ↓
360 # Convert
361 img = img[...,:-1].transpose((0, 3, 1, 2)) # BGR to RGB, BHWC to BCHW
362 img = np.ascontiguousarray(img)
363 ↓
364 return self.sources, img, img0, None
365 ↓
366 def __len__(self):
367     return len(self.sources) # 1E12 frames = 32 streams at 30 FPS for 30 years
368 ↓
369 ↓
370 def img2label_paths(img_paths):
371     # Define label paths as a function of image paths
372     sa, sb = os.sep + 'JPEGImages' + os.sep, os.sep + 'labels' + os.sep # /images/, /labels/ substrings
373     return [sb.join(x.rsplit(sa, 1)).rsplit('.', 1)[0] + '.txt' for x in img_paths]
374 ↓
```

- 修改 train.py 文件。



```
434 def parse_opt(known=False):
435     parser = argparse.ArgumentParser()
436     parser.add_argument('--weights', type=str, default='weights/yolov5s.pt', help='initial
437     parser.add_argument('--cfg', type=str, default='models/yolov5s.yaml', help='model.yaml
438     parser.add_argument('--data', type=str, default=ROOT / 'data/test.yaml', help='dataset
439     parser.add_argument('--hyp', type=str, default=ROOT / 'data/hyps/hyp.scratch.yaml', he
```

训练过程 weights 存放位置。

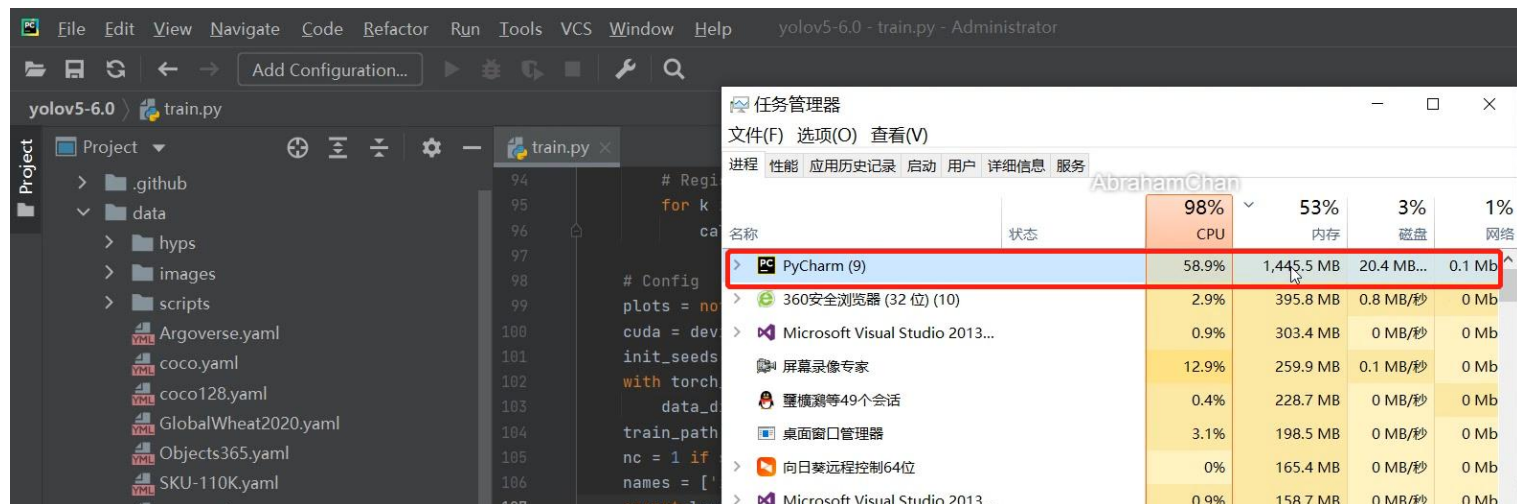
> yolov5-6.0 > runs > train > exp6 > weights

名称	修改日期	类型	大小
best.pt	2021/12/3 16:07	PT 文件	55,400...
last.pt	2021/12/3 16:07	PT 文件	55,400...

总结：

7、测试自己的模型

`python detect.py --weights best.pt --view-img --source XXX`



不建议在 pycharm 里进行训练，因为 pycharm 本身这个软件就要占用很大一部分内存。