**基于Pytorch训练YOLO v4-Tiny目标检测模型**

**文件准备**

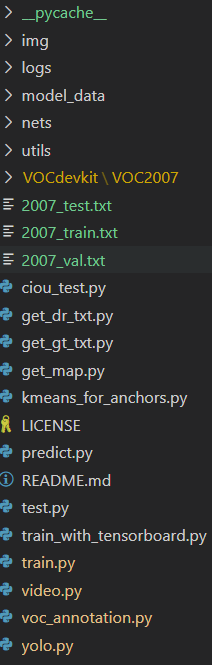
YOLO v4-Tiny模型可从GitHub上获得。首先在任意位置新建一个自己的项目文件夹用于存放代码文件（注意文件夹名字及路径中不能有**中文和空格**，否则之后会出现各种问题），打开命令行cd到此文件夹下，运行

git clone https://github.com/。。。

也可以直接从GitHub页面上下载压缩包，或是从百度网盘上下载，

链接 [https://pan.baidu.com/s/](https://pan.baidu.com/s/1xtVnJUwt97vqrzrLTsScOA" \t "_blank) 提取码: 。。。。

下载得到的YOLO v4-Tiny文件夹架构如下



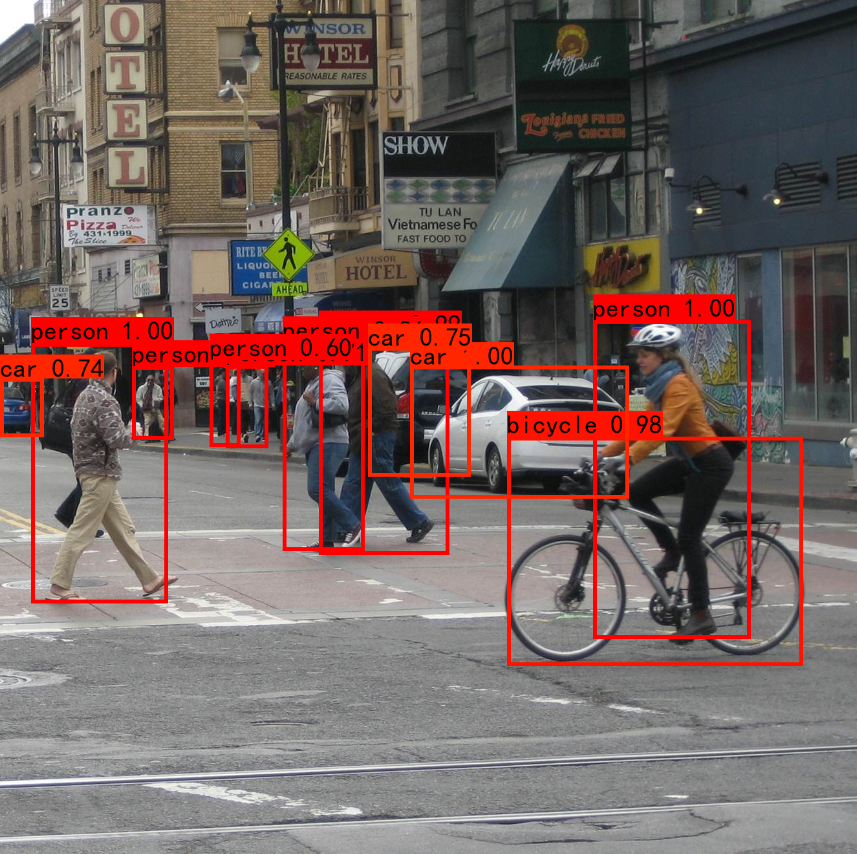
为了简化模型的训练过程，通常我们都会利用已预训练过的模型权重进行迁移学习，也就是将预训练好的模型权重参数作为自己模型的初始权重参数进行训练。因此，除了代码，还需要下载预训练好的模型权重文件。

可以从百度网盘上下载权重文件，有两个文件可供选择，分别是利用voc数据集预训练的yolov4\_tiny\_voc.pth和利用coco数据集预训练的yolov4\_tiny\_weights\_coco.pth，下载好后需要将权重文件放入model\_data文件夹下。

链接 [https://pan.baidu.com/s/](https://pan.baidu.com/s/1xtVnJUwt97vqrzrLTsScOA" \t "_blank) 提取码: 。。。。

**预训练模型测试**

1、Python运行predict.py示例程序，输入img/street.jpg，可以看到打开了img文件夹下名为street.jpg的图片，图中各种物体已经被框出并分类。



如出现FileNotFoundError错误，请学习python工作目录、相对路径等知识，有能力的同学可使用os.path模块改进代码，一劳永逸地解决相关问题。

2、Python运行video.py示例程序可以打开摄像头进行检测。

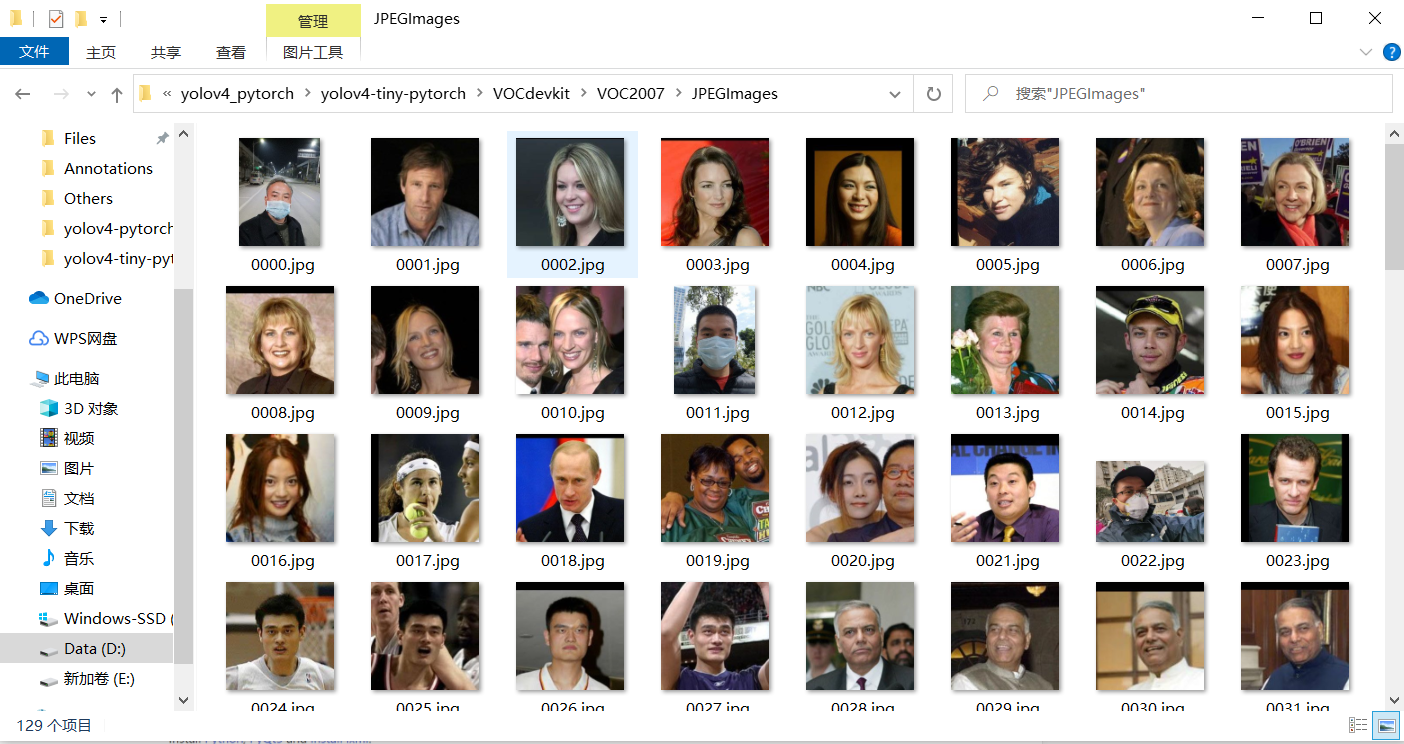
**模型训练**

**一、数据准备**

训练模型需要大量数据，有能力的同学可以自行在网上寻找高质量的数据集使用，这里我提供一些戴口罩与未带口罩的人脸图片，供同学们下载选用。

链接：[https://pan.baidu.com/s/](https://pan.baidu.com/s/1xtVnJUwt97vqrzrLTsScOA" \t "_blank) 提取码: 。。。。

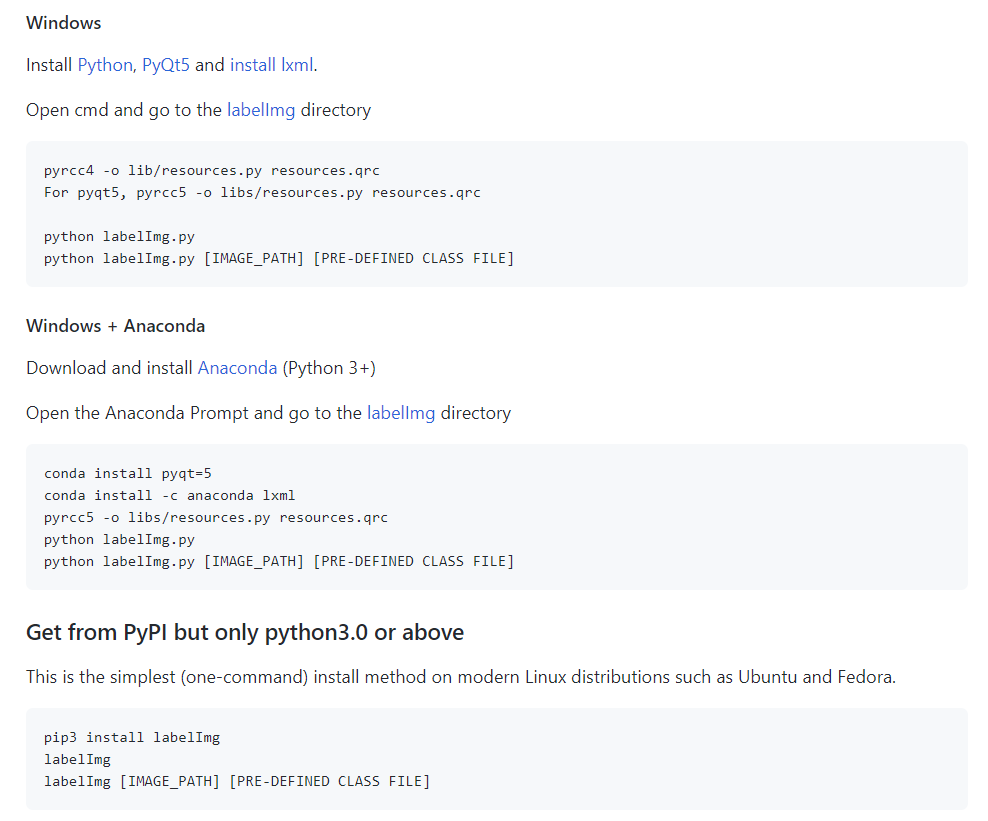
图片下载下来并挑选后，需要放到/VOCdevkit/VOC2007/JPEGImages文件夹下。注意，这些图片的文件名是混乱的，为了方便后续的标注和训练，应当将它们的文件名变成规律的，如“0000.jpg”、“0001.jpg”等。实现方法请同学们自行探索。



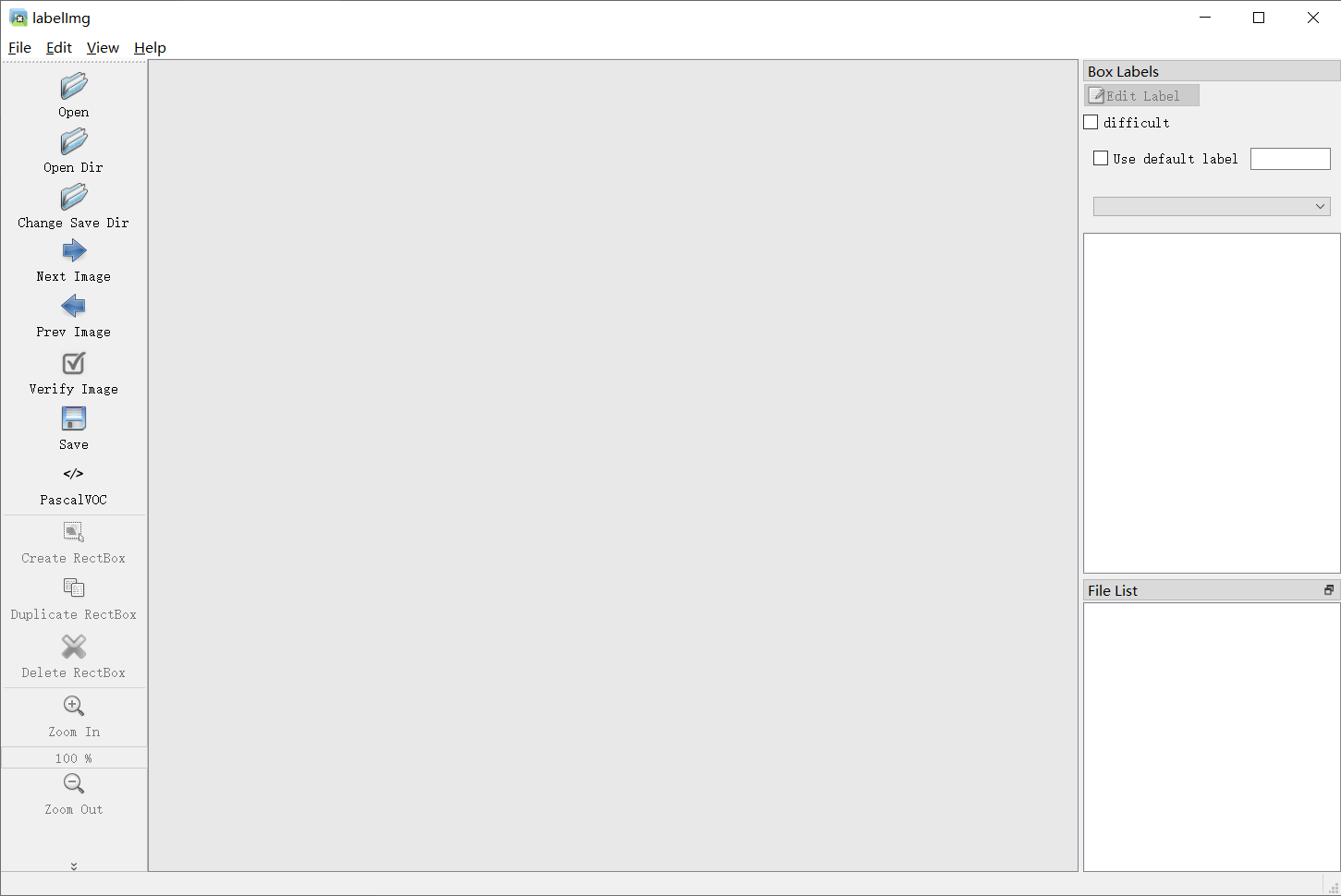
此外，有能力的同学可以利用数据增强（Data Augmentation）来扩充数据集，提高模型的鲁棒性。

**二、数据标注**

接下来进行数据标注，需要使用标注软件，这里用的是labelImg工具，可以从[此处](https://github.com/tzutalin/labelImg" \t "_blank)下载并学习安装方法。

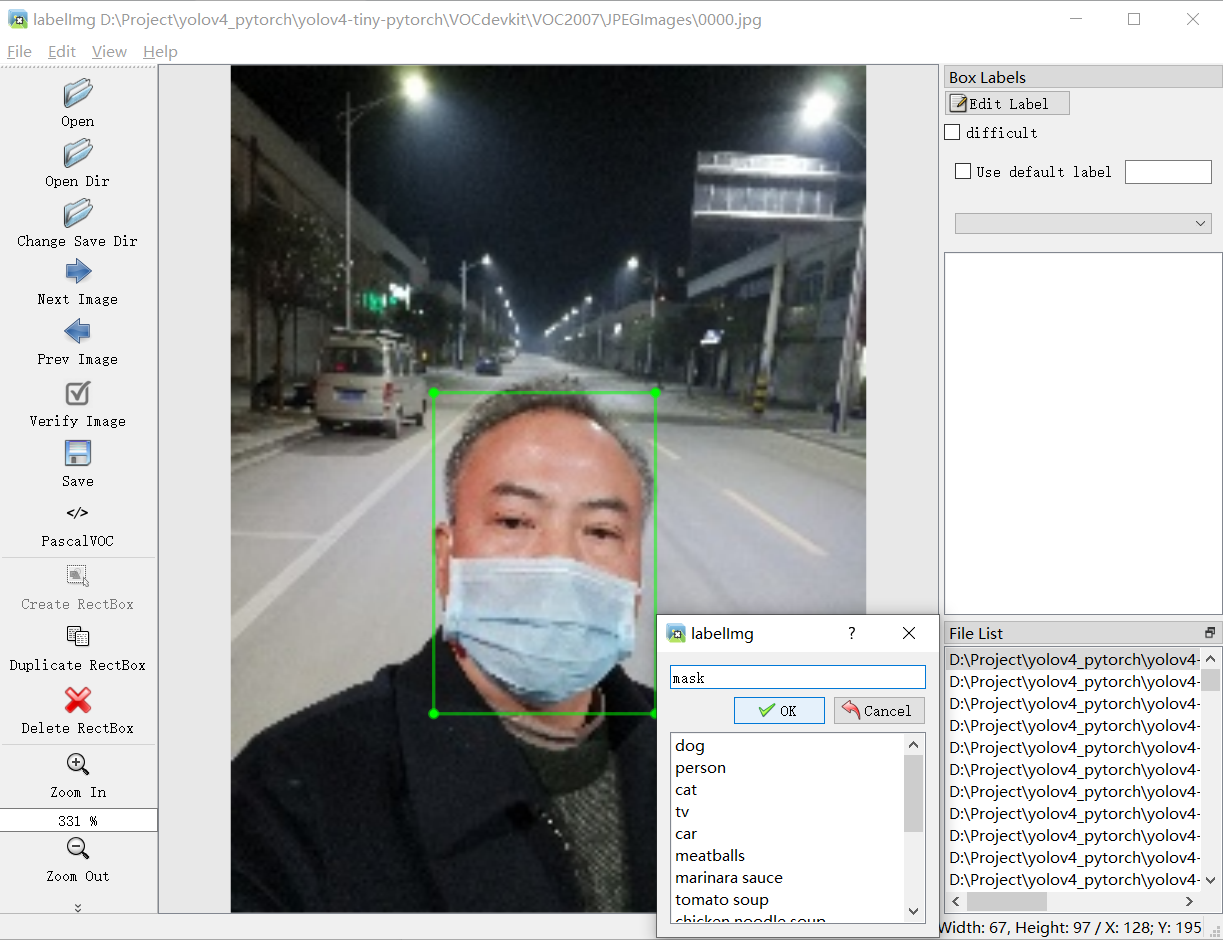


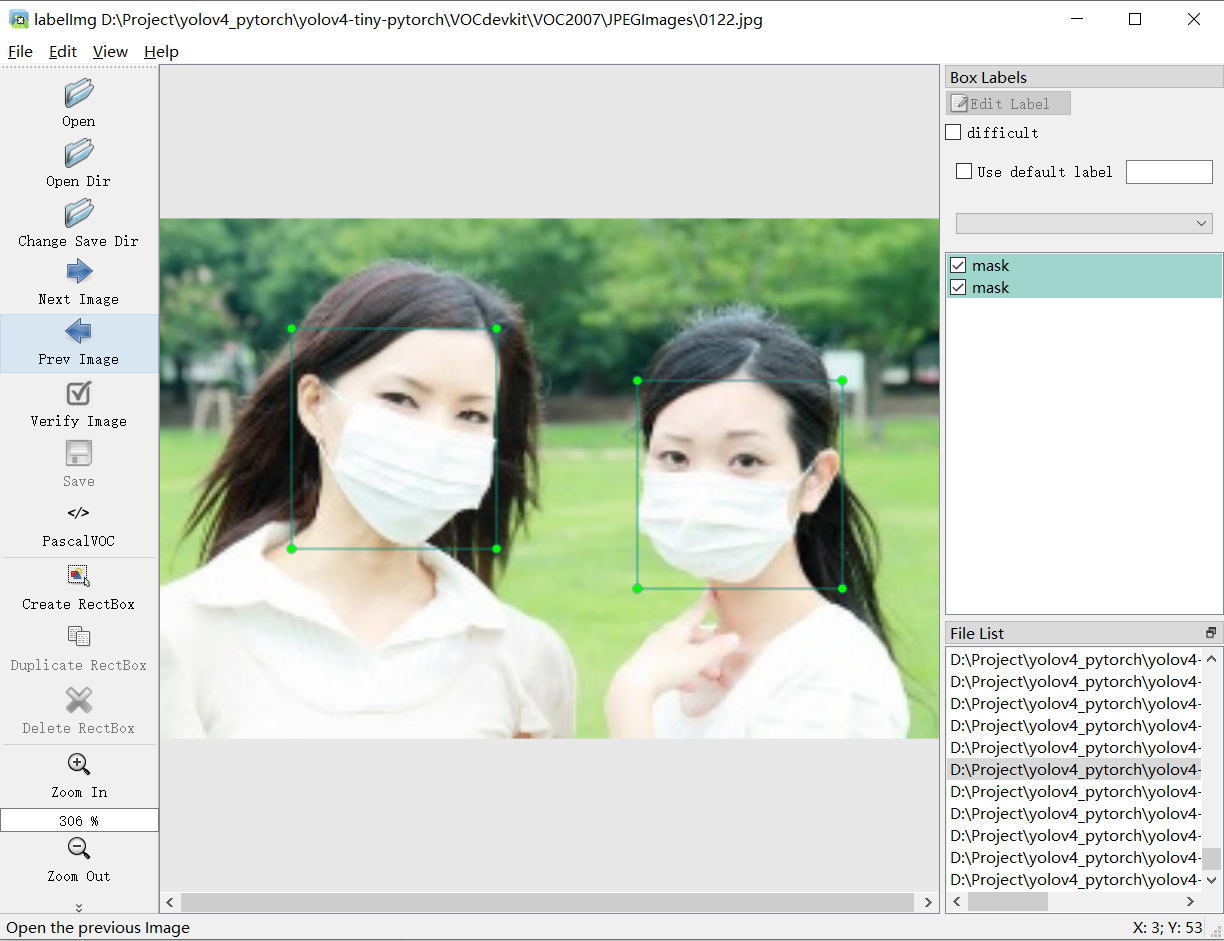
运行labelImg，打开界面



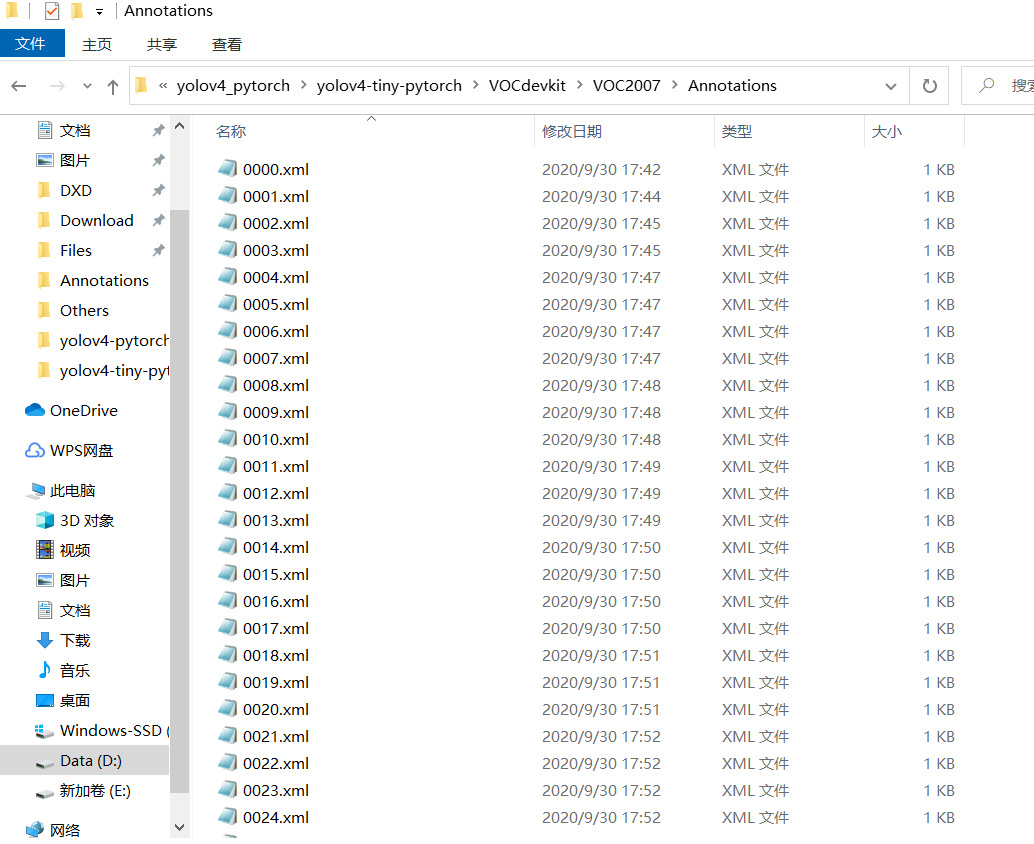
点击界面左上的Open Dir选择图片文件夹，即/VOCdevkit/VOC2007/JPEGImages，之后点击Change Save Dir选择标签存储文件夹为/VOCdevkit/VOC2007/Annotations，标注格式选择PascalVOC。

点击Create RectBox就可以进行框选标注，框选带口罩或不戴口罩的人脸，在弹出的标签窗口中输入mask或un\_mask即可进行标注，之后点Save保存，Annotations文件夹下就生成了xml格式的标签文件，之后点击Next Image继续进行标注（活用快捷键W、A、D、Ctrl+S）。

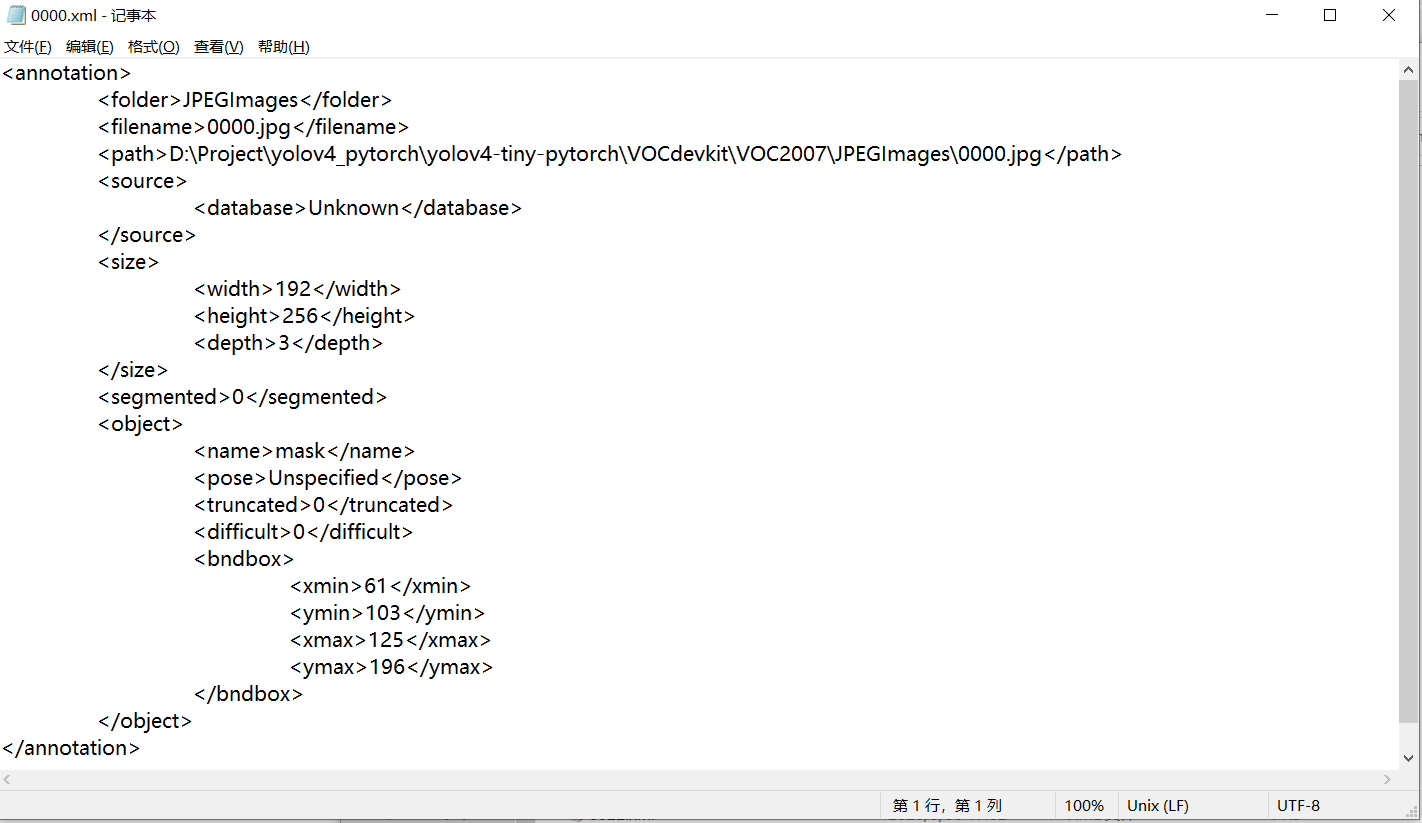




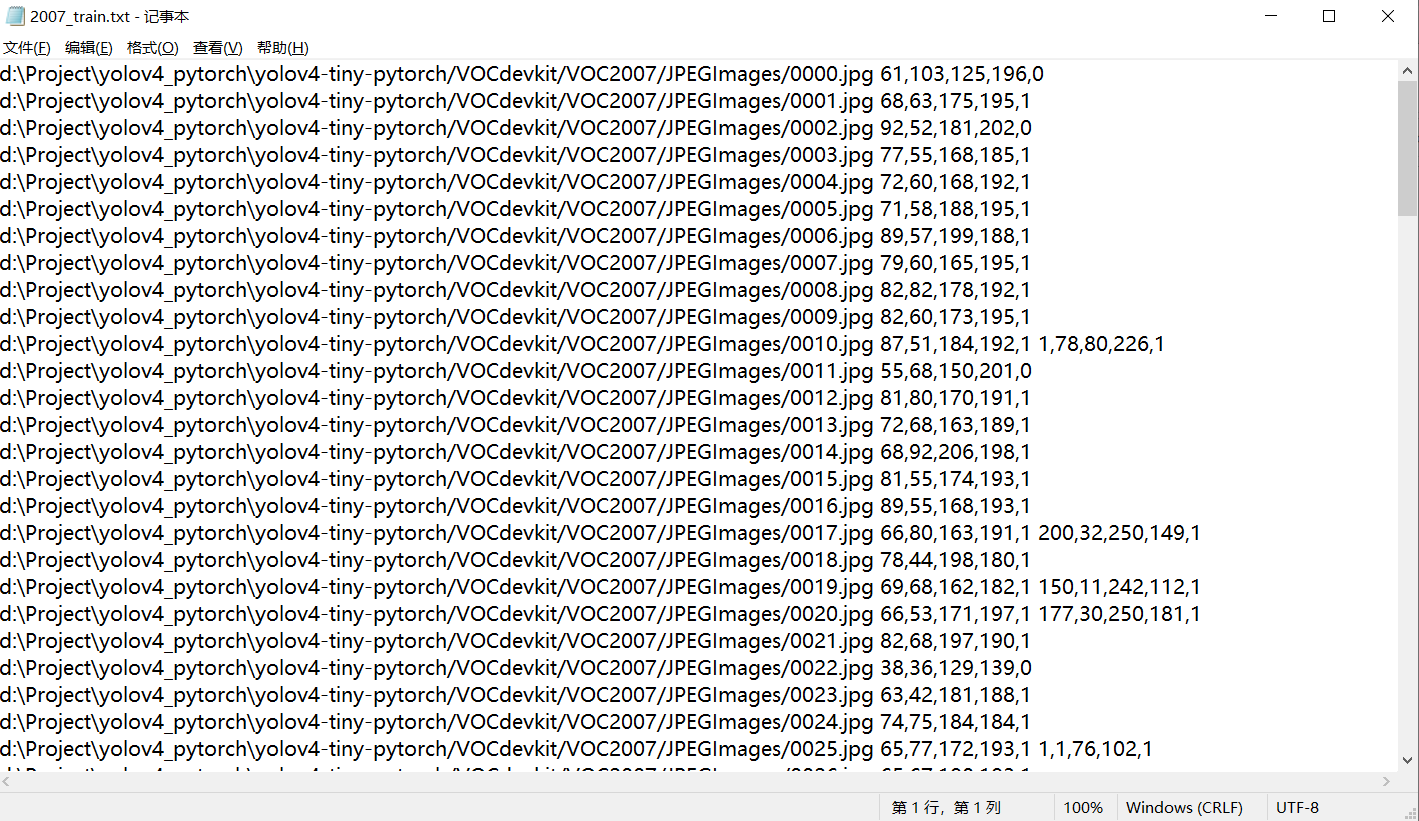
标注完成后，Annotations文件夹下会生成大量的xml文件



打开任意xml文件，格式如下

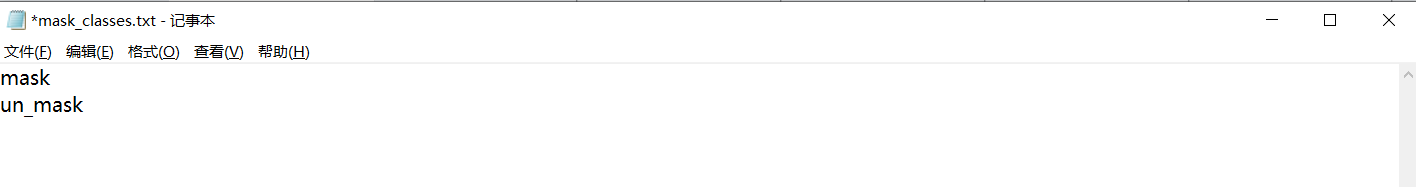


但此时标注文件尚不能被YOLO模型读取，要运行/VOCdevkit/VOC2007文件夹下的voc2yolo4.py提取文件索引，之后运行根目录下的voc\_annotation.py，运行前需要将classes改成自己需要的classes，即mask和un\_mask。运行后索引和标注信息被写入到了根目录下的2007\_train.txt文件中。



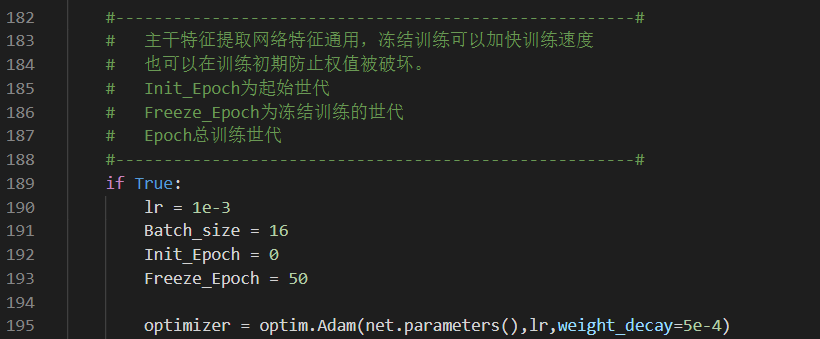
**三、训练模型**

接下来进入模型训练的环节，首先要在model\_data下新建一个txt文档，文档中输入需要分的类，同时在train.py中将classes\_path指向该文件。



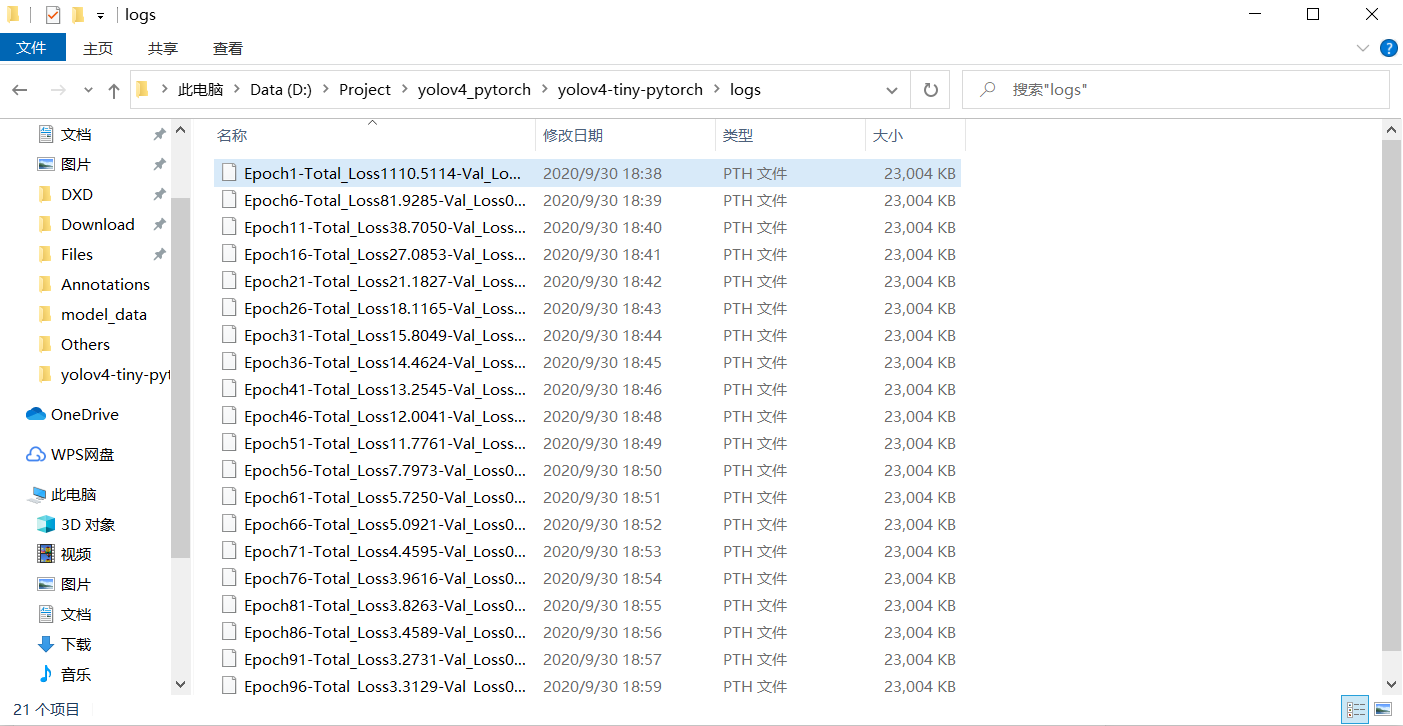


之后，在train.py中设置超参数，如学习率lr、批量大小Batch\_size、训练世代数等，可以先使用默认的值进行实验，之后再根据情况进行修改。



注意，代码中利用torch.save模块将训练出的模型权重文件保存在根目录的logs文件夹下，保存频率为每世代保存一次，同学们可以根据自己的意愿修改保存频率及路径。

一切就绪后，运行train.py开始训练，注意观察loss的下降。训练结束后logs文件夹下得到诸多权重文件。



找到最后世代的权重文件，将根目录下yolo.py中的model\_path指向该权重文件，并将classes\_path指向之前在model\_data下创建的txt文件。



接下来同测试预训练模型时一样，运行predict.py和video.py进行模型效果的测试。如测试结果不够理想，可以尝试修改超参数、扩大数据集等方法。