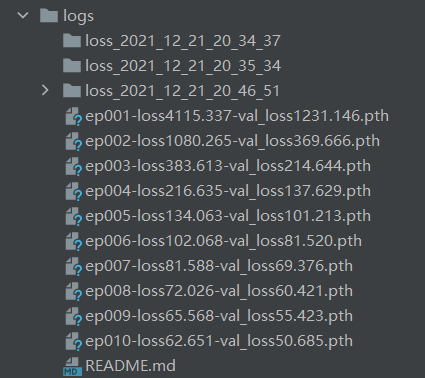
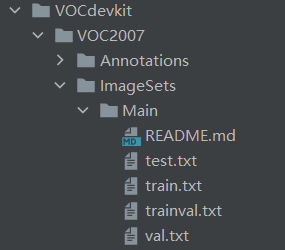
1. 首先删除一些东西：
2. logs文件夹下所有权重和日志文件



1. VOCdevkit/VOC2007/ImageSets/Main文件夹下的四个txt文件

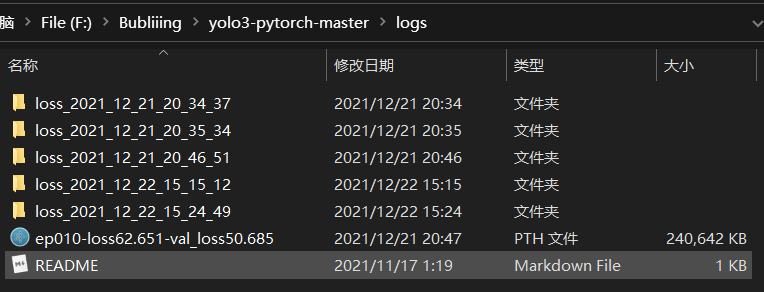


（3）2007\_train.txt和2007\_val.txt文件



（4）map\_out文件夹。因为保存的是上一次测试的结果，当然也可以不删，因为重新运行get\_map.py的话新的测试结果会覆盖掉原来的。

1. 用labelimg制作数据集，原图片需要是jpg格式。可以用python函数将png格式转成jpg格式，函数在F盘的DataSet\_VOC/convert\_img.py
2. 将制作的数据集中的图片（只能是jpg格式）和标签分别放在VOCdevkit\VOC2007文件夹下的JPEGImages文件夹和Anotations文件夹下。
3. 在model\_data文件夹下新建一个txt文件，里面是数据集里面的类别，要竖着写，比如dog、cat、person......。比如命名为my\_classes.txt。
4. 打开voc\_anotations.py文件：
5. 修改annotation\_mode = 0
6. 修改classes\_path = 'model\_data/my\_classes.txt'
7. 运行voc\_anotations.py生成所需txt文件
8. 打开train.py文件:
9. 修改classes\_path = 'model\_data/my\_classes.txt'
10. model\_path = 'model\_data/yolo\_weights.pth'是加载别人预训练的权重，一般不修改（要先把这个权重文件下载下来放在对应文件夹下面）
11. input\_shape= [416, 416]，可以根据自己数据集图片的大小修改，比如640，608，320等，但必须是32的倍数
12. 冻结阶段与解冻阶段的参数修改，如epoch、batch\_size、lr等
13. Freeze\_Train为True。利用了迁移学习。
14. num\_workers在Windows下最好设置为0，否则可能报错。在ubuntu下可以设置为2或者4等。
15. 运行train.py进行训练，同时会在logs文件夹下面保存损失函数信息，文件夹里面放的是损失函数（包括训练、测试与验证），以及模型权重。



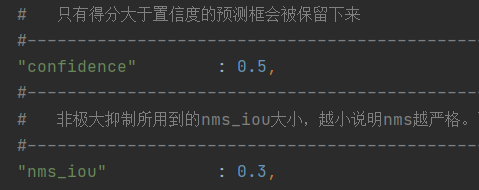
1. 打开predict.py文件，利用训练好的权重进行预测：
2. 在预测之前，进入YOLO()这个类里面调整一些参数：

首先是载入自己训练出的权重和指明数据集类别--->

"model\_path" : './logs/ep010-loss62.651-val\_loss50.685.pth',

"classes\_path" : 'model\_data/my\_classes.txt',

然后可以调整置信度阈值和nms\_iou阈值--->



1. 返回predict.py，设置model：predict、video、fps等。运行便能得到检测结果

如果没看到结果只看到原图，可能是confidence和nms\_iou设置不当，可以重新调整之后再进行检测。

1. 在训练完之后可以测试mAP：

打开get\_map.py文件。

（1）首先要修改classes\_path，指向自己的txt文件，

classes\_path = 'model\_data/my\_classes.txt'

（2）可以修改map\_mode为01234，但是若修改为4需要注意下注释

1. 选择修改：MINCVERLAP、map\_vis等
2. 运行get\_map.py可以看到结果
3. 程序运行之后会生成map\_out文件夹并保存结果，比如在result文件夹下面保存了AP、F1、Precision、Recall等。当然如果之前就存在这个文件夹的话，会被清空然后再重新保存结果，所以其实一开始可以将整个map\_out文件夹删掉。