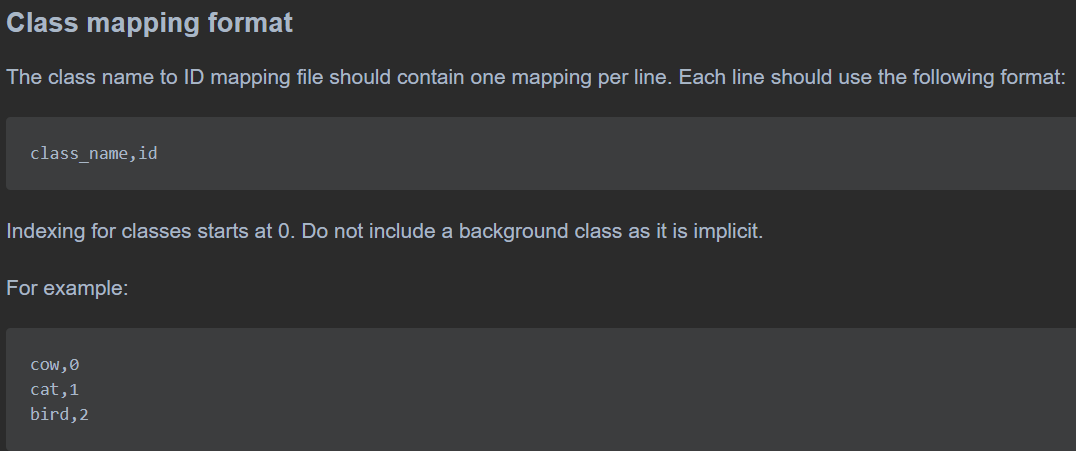
1 目前还需要做的一些工作：  
**(1.调研yolov5，Faster-RCNN，Retinanet，SSD，Cascade-RCNN模型的区别、特点、分别适用于什么样的数据集**  
**(2.调研相应的数据集**  
**(3.用这些数据集在网络上进行训练和测试，最终输出  [图片名称,IOU值]  这样的形式**

-------------------------------------------------------------------------------  
**注：目前的这个Retinanet项目已经可以实现最终输出 [图片名称,IOU值] 的工作。在pytorch-retinanet\output文件夹中的output.txt文件内有更详细的说明。**-------------------------------------------------------------------------------  
2 目前所构建的Retinanet项目使用流程。  
2.1数据集需要转换成的格式说明

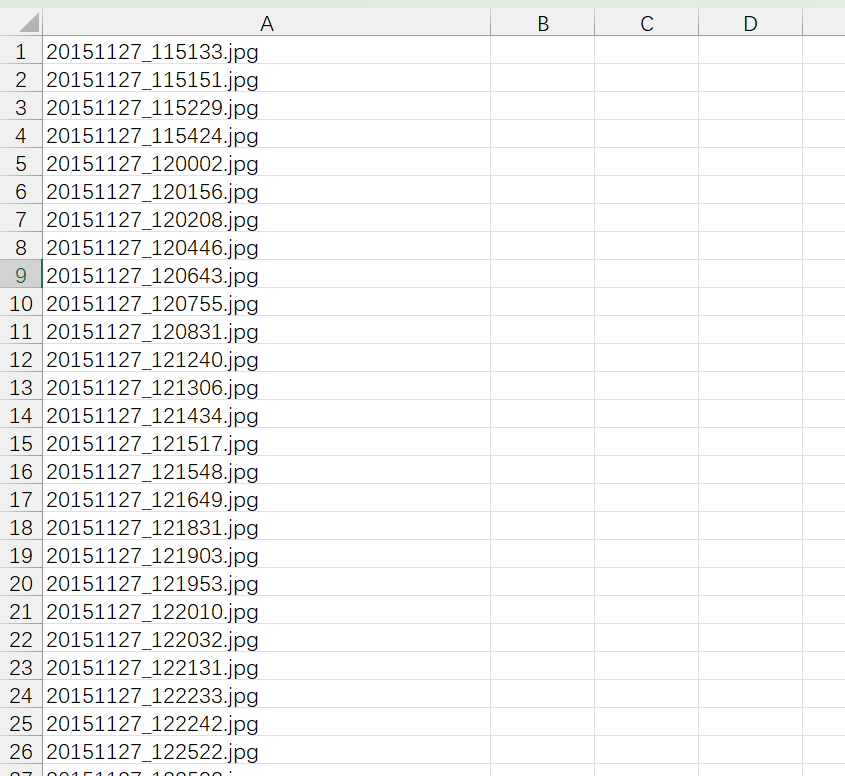
(1)数据集中所有的jpg格式图片放在pytorch-retinanet\data\images文件夹内

(2)创建class\_list.csv文件，文件中所写的内容格式如下：



可参考目前项目中的class\_list.csv文件进行修改

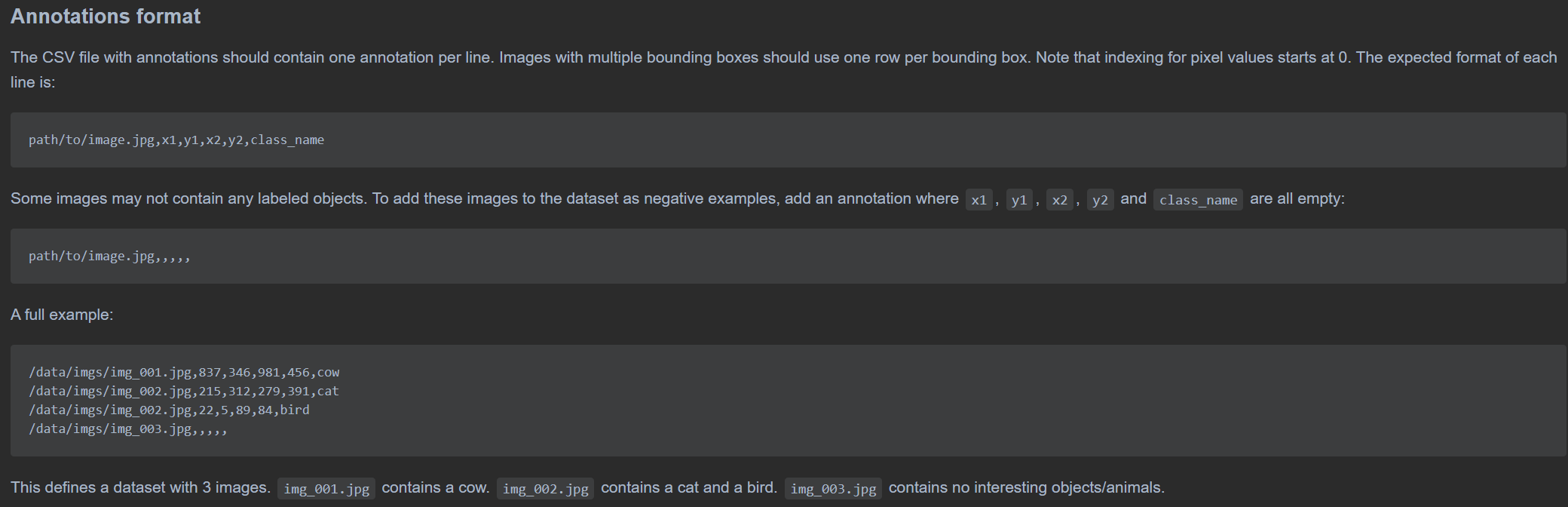
(3)创建train.csv文件，文件中所写的内容格式如下：



包含训练集所有图片的名称

(4)创建val.csv文件，文件中所写的内容格式与上述2.1的(3)相同，不同点是该文件中包含测试集所有图片的名称

(5)创建train\_annots.csv文件，文件中所写的内容格式如下：



第一列：训练集每张图片存放路径

第二列~第五列：图片上相应类别的标注框的左上角和右下角坐标(坐标原点为图片左上角，坐标单位为像素)

第六列：类别名称

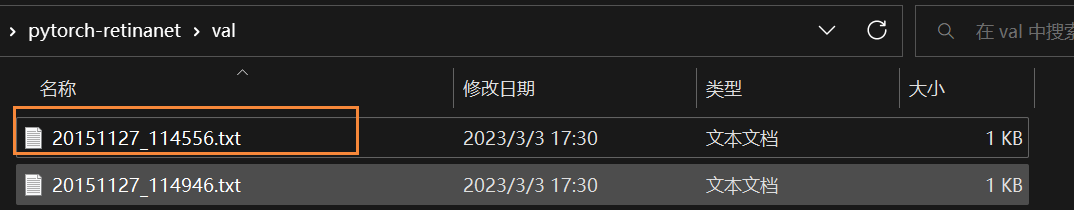
可参考目前项目中的train\_annots.csv文件进行修改

(6)创建val\_annots.csv文件，文件中所写的内容格式和上述2.1的(5)相同，不同点是该文件中描述的是测试集，可参考目前项目中的val\_annots.csv文件进行修改

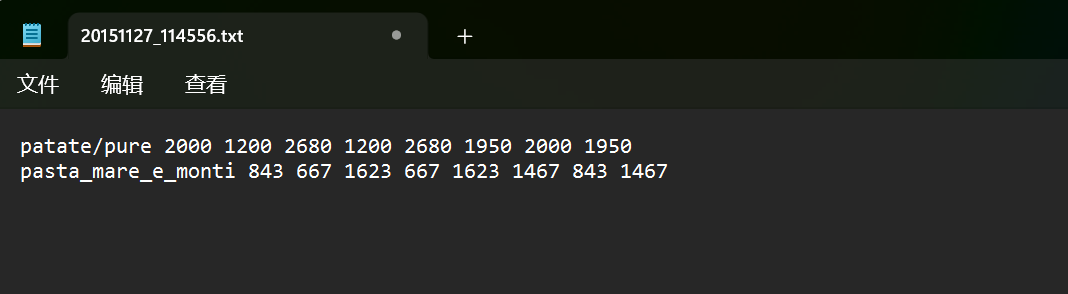
(7)在val文件夹下存放所有txt文件，每个txt文件其实对应一张图片。代表要将哪些图片最后输出为[图片名称,IOU值]的形式。每个txt文件中包含有对应图片上的所有类别名称以及类别对应的标注框信息(左上，右上，右下，左下角坐标)。

**注：在实际构建的时候这些txt文件可以只写对应图片上的所有类别名称即可，类别对应的标注框信息有无均可(仅限2.1的(7)部分可以这样操作)**

例如：



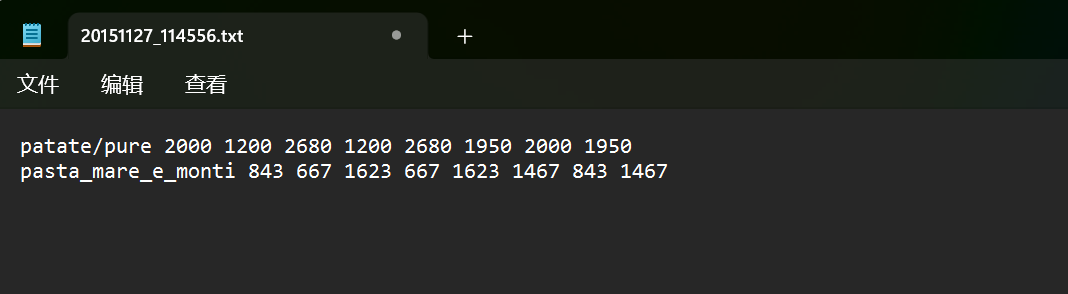
比如说20151127\_114556.txt文件，其实对应了数据集中的20151127\_114556.jpg图片。打开20151127\_114556.txt文件可以看到



这代表了20151127\_114556.jpg图片上有两个类别，以及这两个类别对应的标注框信息。

在实际构建的过程中，这里的txt文件中的内容格式如下两种均可：

第一种：



第二种：



2.2训练

2.1完成后，可直接运行train.py文件进行训练，目前训练所用的超参数设置如下(未调参)：

batchsize=1（bz=1时16G内存会占用大约98%）

epoch=5，

学习率1e-5，

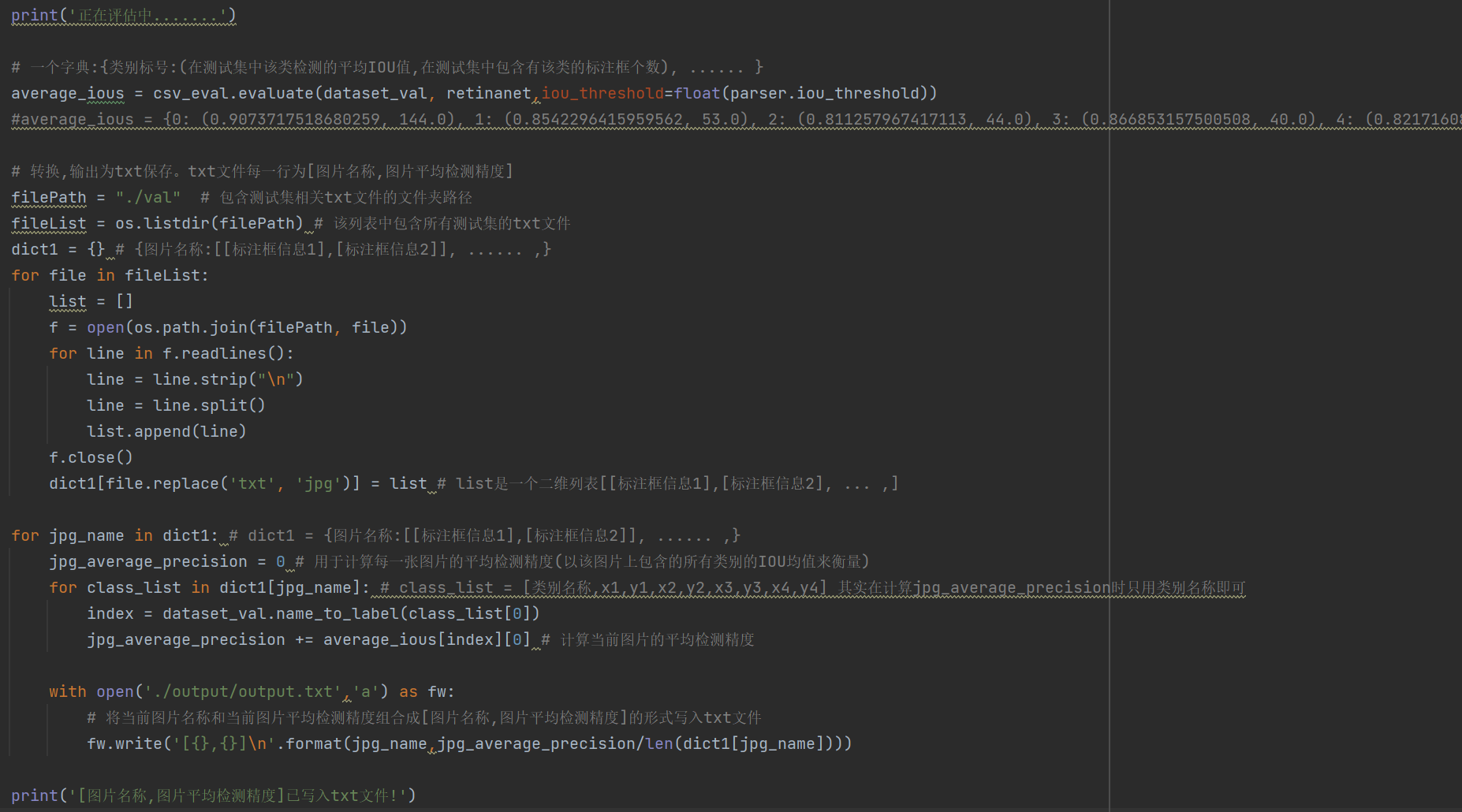
优化器为Adam

depth=50(表示此时特征提取网络采用resnet50，可选值有18, 34, 50, 101, 152)

训练结束后会保存一个训练好的模型文件model\_final.pt

2.3测试

2.2完成后，可直接运行csv\_validation.py文件，模型默认加载model\_final.pt，IOU阈值默认为0.5。该部分最终输出 [图片名称,IOU值] 的具体过程代码在csv\_validation.py文件中有详细的注释说明：



最后会在pytorch-retinanet\output文件夹中生成一个以当前时间命名的txt文件，文件内即保存了所有 [图片名称,IOU值] 信息。