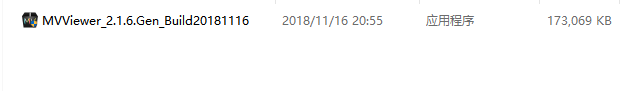
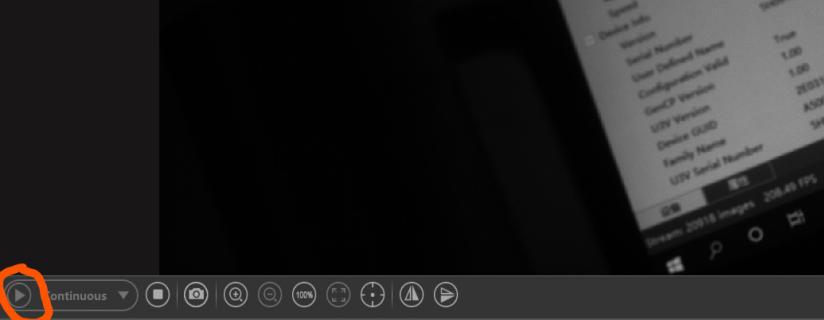
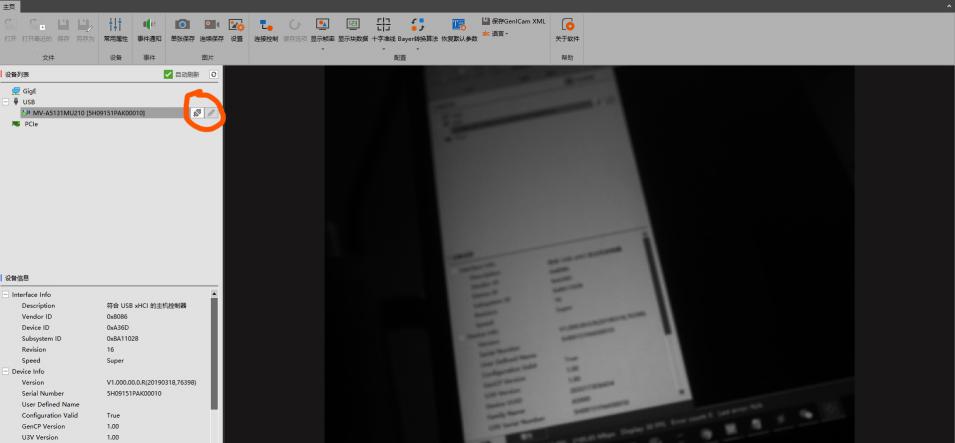
**测试工业摄像头**

（1）下载驱动



1. 测试摄像头



摄像头测试完成

1. 测试python调用程序

注：先将MVViewer的连接断开

进入对应目录：

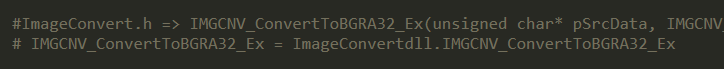


例程默认为32位，如需使用64位，需要修改加载的SDK库的处理。具体如下：

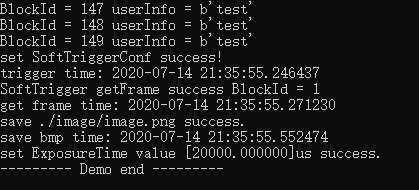
· MVSDK.py ： Line16注释掉，Line18去注释

· ImageConvert.py ： Line12注释掉，Line14去注释

然后注释ImageConvert.py的最后一行



python Demo.py”运行例程





Demo测试完成

**训练模型**

（1）标注数据

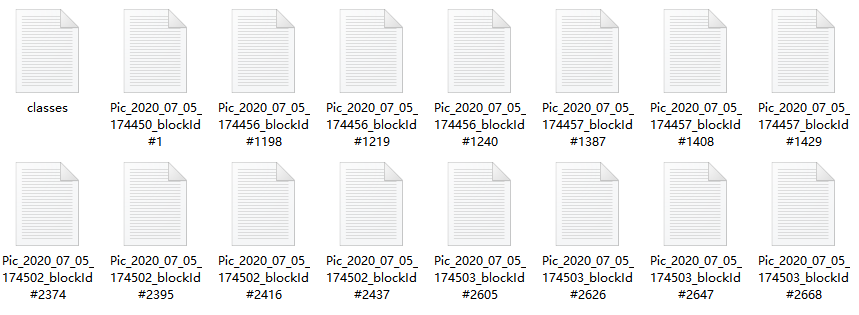
数据标注的软件使用的是LabelImg，进入它的目录，点击打开Anaconda Prompt窗口，然后输入python labelImg.py,

即可打开页面。

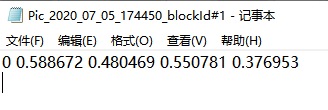


看到图片后，点击PascalVOC切换成YOLO模式，之后单击右键‘W’，即可框选我们需要的目标。然后框选之后输入标签，标签后，点击右边的“save”，保存txt文件到您想要存放的目录。这将作为我们模型训练的重要组成部分,txt文件的文件名和图片名一致。保存后，切换下一张，然后循环这个过程即可。

标签如下图所示：



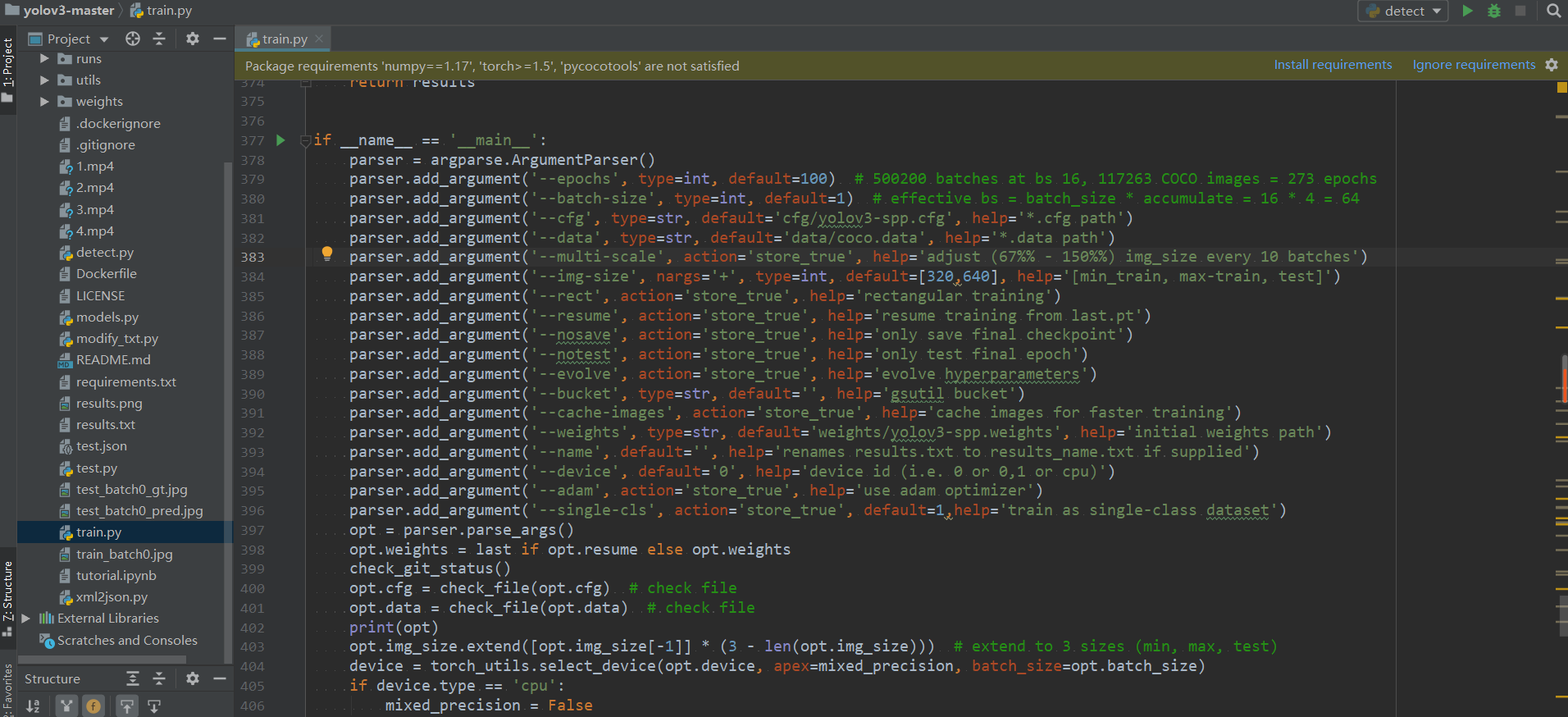
这个标签的内容是：标签序号+框选框的x,y,w,h坐标



（2）训练模型

模型的训练需要这几个步骤：准备数据集并放到相应的地方，设置好模型的参数，训练。

我们打开目标检测代码的train.py，记得要用pycharm创建新工程的方式打开yolo的文件夹。

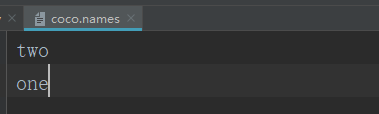
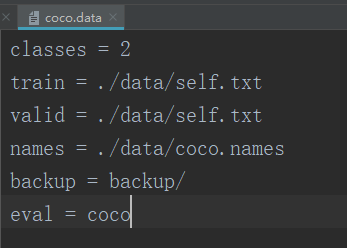


首先我们要确定我们要做的是几分类的目标检测,如二分类，三分类，假设是N分类，那么我们需要修改的文件有以下几个：

（1）./data/self下的coco.data和coco.names文件

coco.data文件修改的是第一行的classes，数值改成N

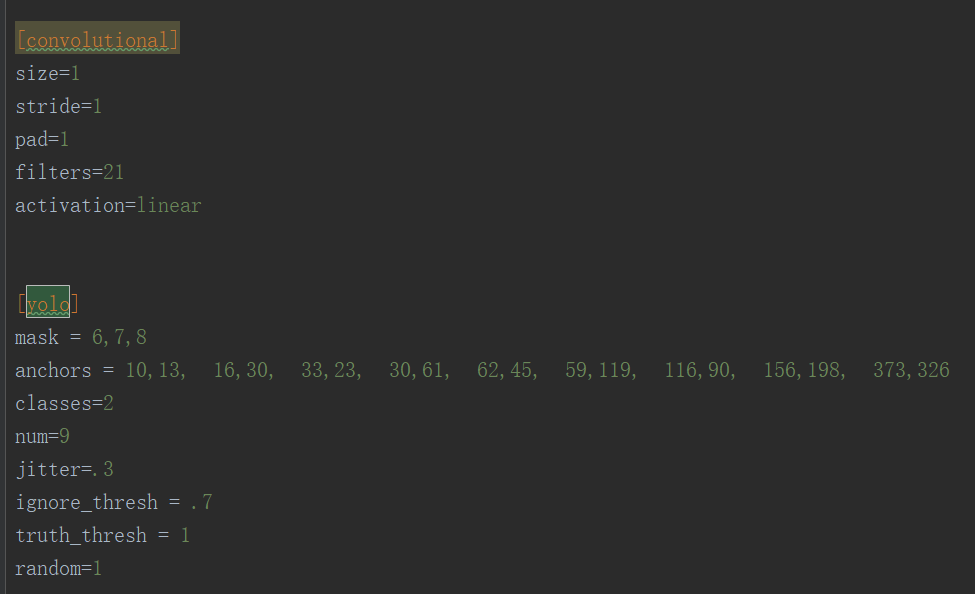
coco.names文件修改的是分类的标签



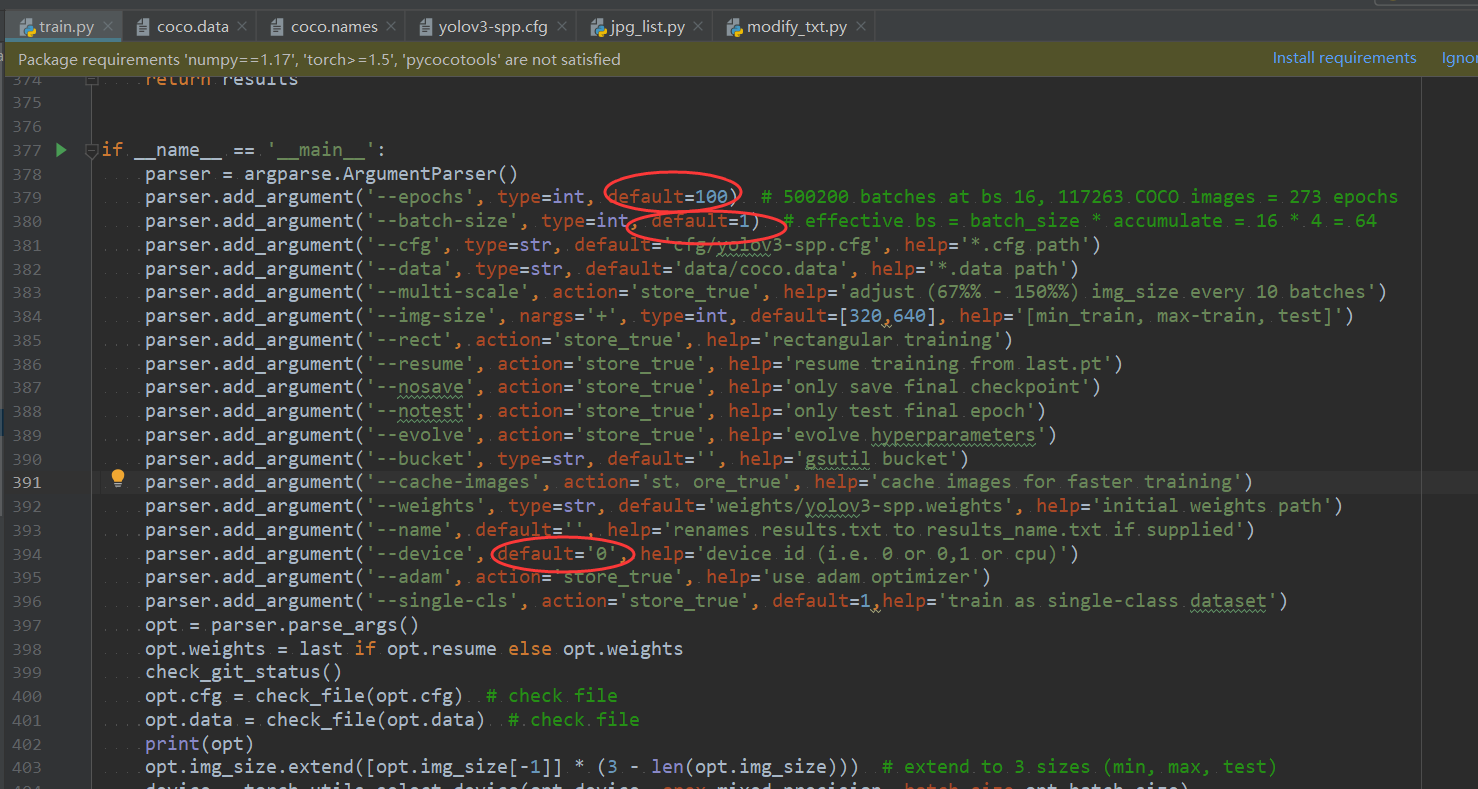
其中，标签0对应two，标签1对应one。

（2）cfg文件夹下的yolov3-spp.cfg文件：

在pycham下ctrl+F打开搜索框，搜索yolo，然后在每个yolo的那一块和它的前一个convolutional层修改一下参数，fliters = 3\*(5+2)=21,classes=2

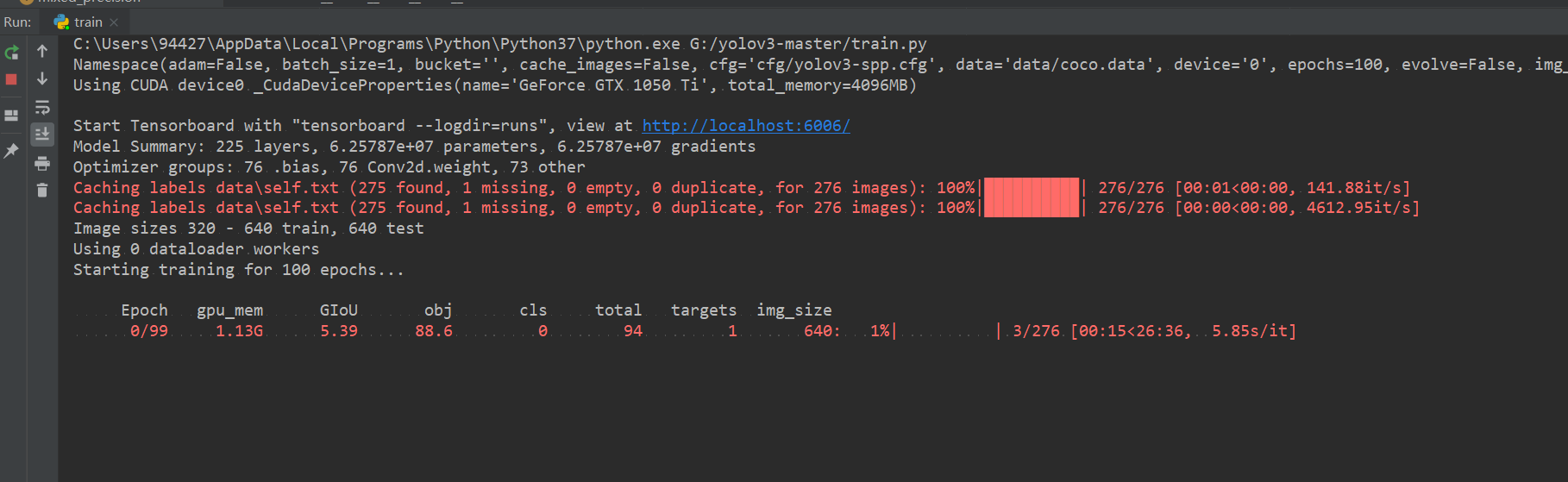


那么执行了一下这几步训练之前的基本准备工作就做完了。我们打开train.py



Device一般是说训练用的是GPU还是cpu，填’cpu’用的就是cpu，填‘0’就是用的第一块GPU。

这一切都设置好后，点开运行，



出现这样的输出时，便是开始训练了。

每轮训练保存的模型都在./weights/last.pt,这个就是训练模型文件，预测的时候用的话最好给它改个名。

至此训练过程介绍完毕，等待训练完成即可