**为方便大家测试代码是否能正常运行，我们给出了Image\_test和Anno\_test，请将**img\_pat和anno\_path**分别设为这两个文件夹对应的路径进行代码的运行测试。**

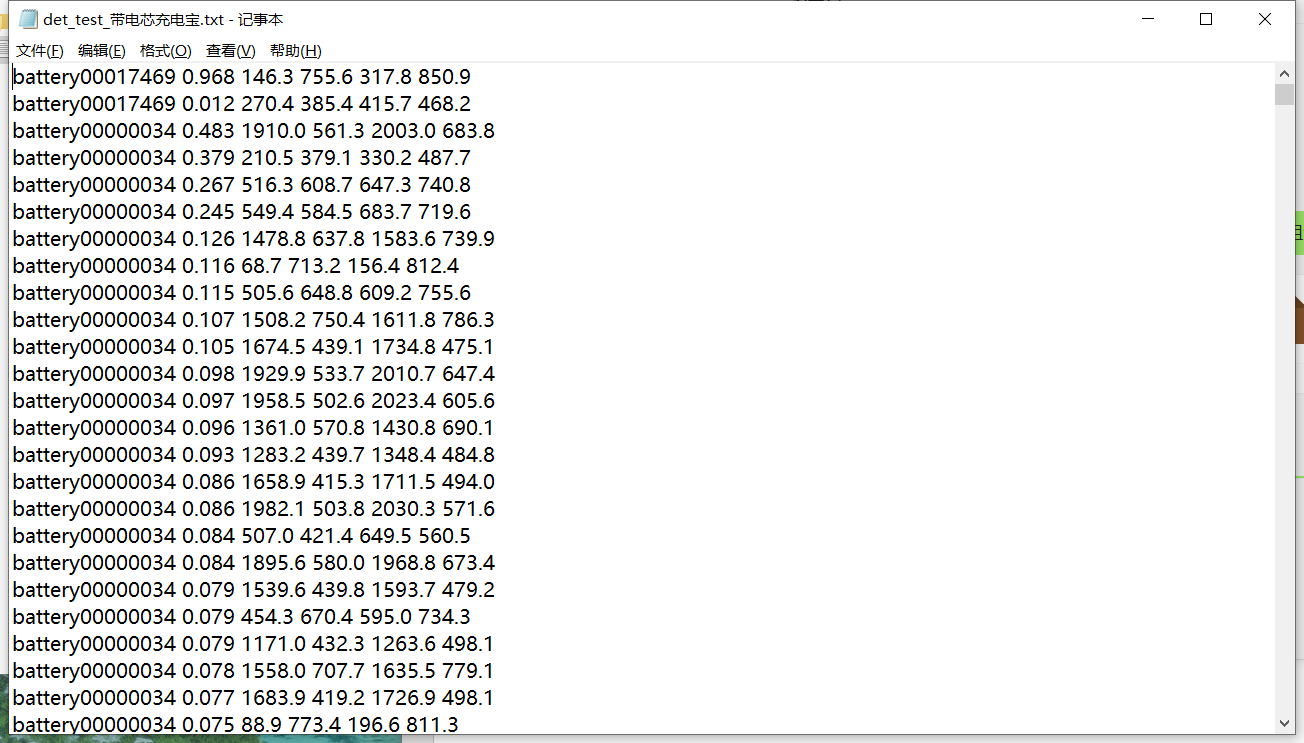
**输入要求：**

每个组的项目中要有test.py文件，其中包含test(img\_path，anno\_path)。

我们会写脚本调用test.py文件中的test()函数，并将img\_path和anno\_path修改为测试集图片的路径和对应图片标注的路径。

**输出要求：**

test函数需要输出两个txt文件（分别是带电芯充电宝和不带电芯充电宝的txt文件），文件的结构如下：

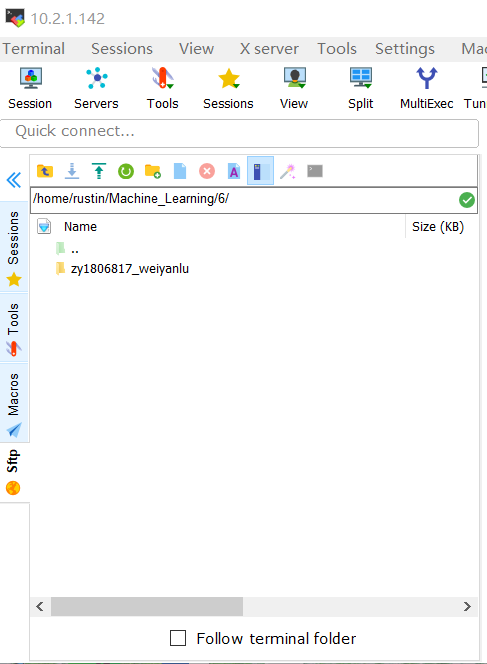


每行保存的是图片名、检测框的置信度、危险品的位置（左上角坐标和右下角坐标）。

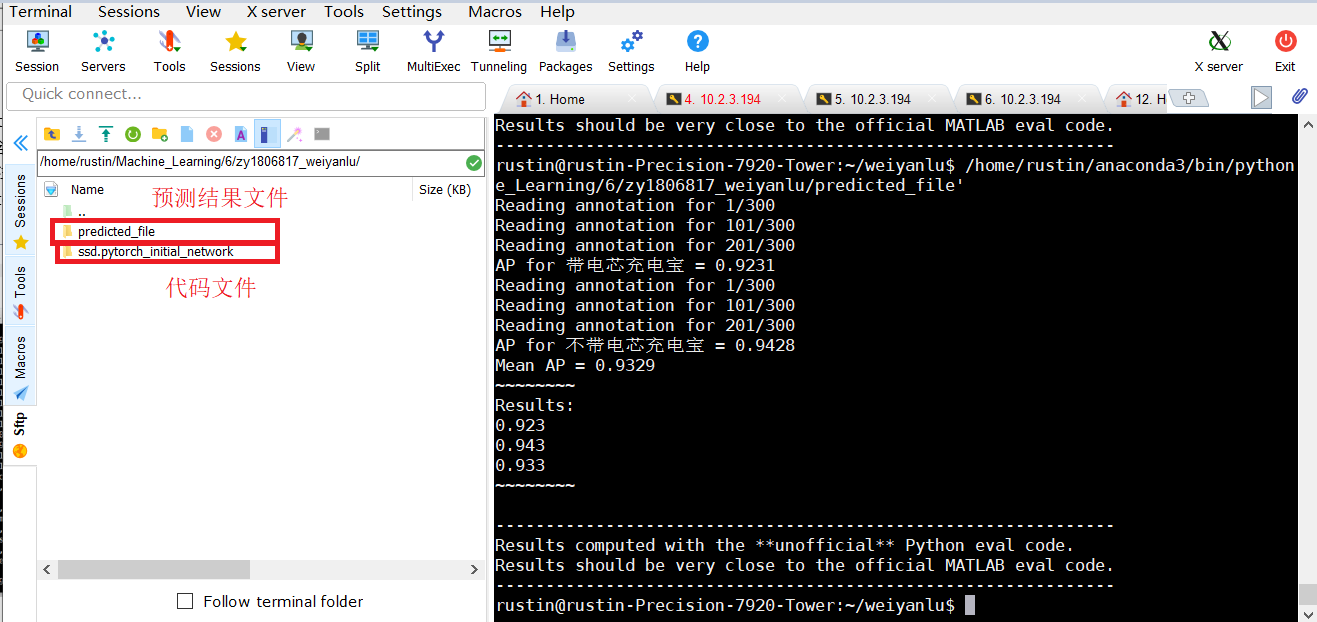
对于每张图片，需要保存置信度在0.01以上的所有预测框，这些预测框会用于map的计算。

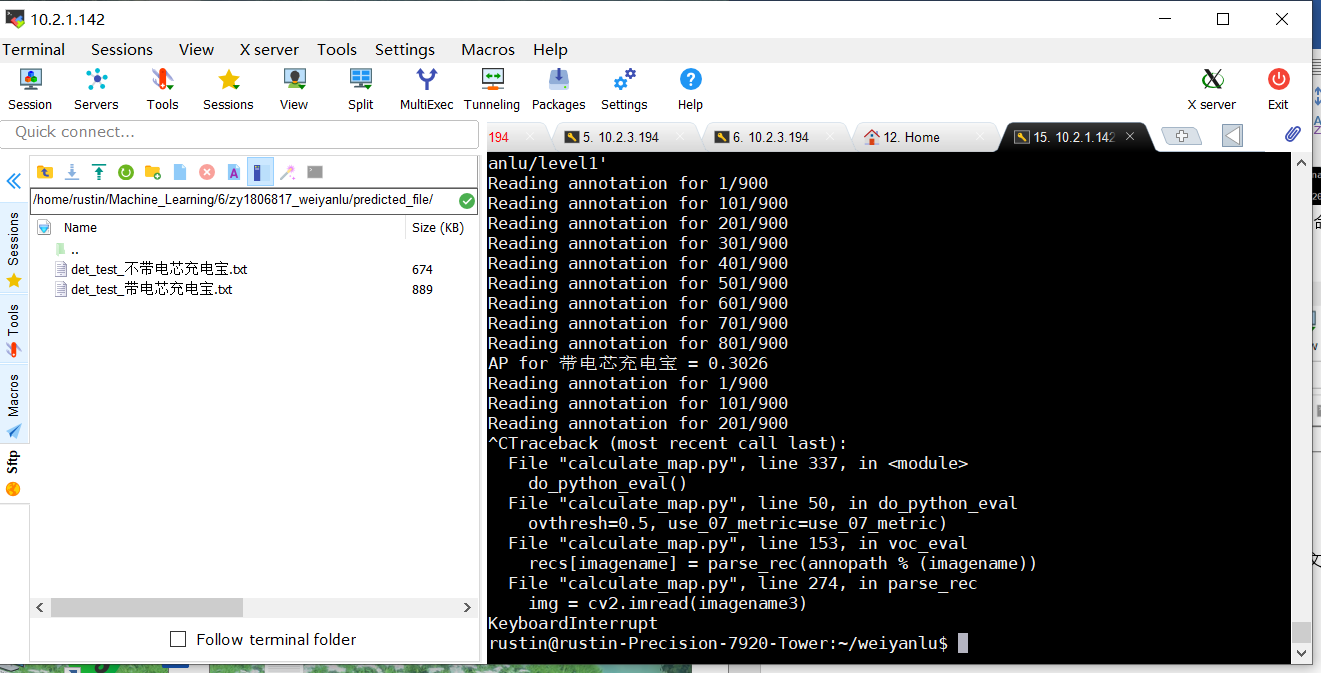
**示例：**

创建自己组的文件夹，命名格式为学号+组长姓名。（最后上交的代码就是这个打包的文件）

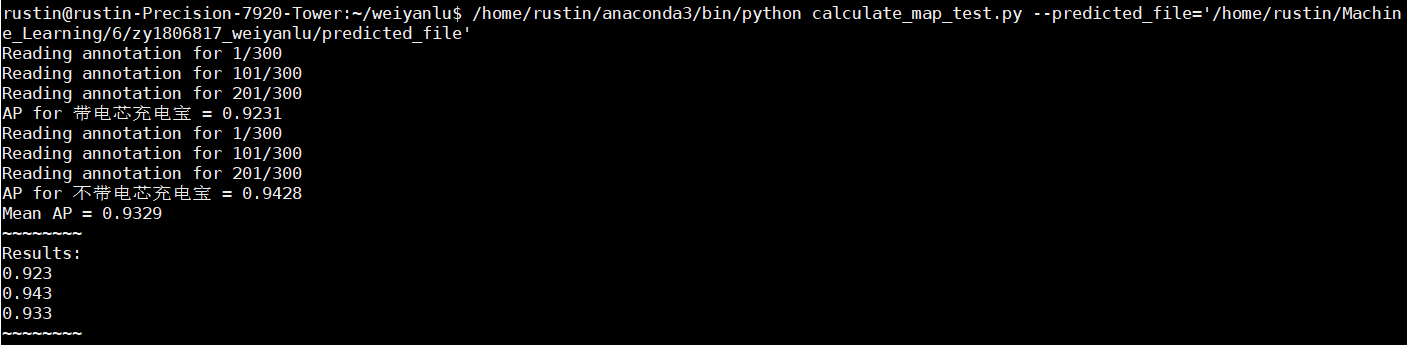


在该文件夹里放入自己的代码，并将输出结果的txt文件，文件命名要求和下图一致，放到一个predicted\_file文件夹中，如下两图，里面包含不带电芯和带电芯两种充电宝的预测结果（即**输出要求**中要求输出的预测文件）[注：若是遮挡问题可以建立predicted\_file\_level1、predicted\_file\_level2、predicted\_file\_level3三个文件夹]。





在得到上面的预测文件之后，执行给出的map计算脚本：



即可看到输出的结果。

评分标准：

（1）遮挡问题：

我们最后将修改测试文件中的数据集路径，将其改为测试集所在的路径（遮挡问题有3个不同遮挡等级的测试集）。最后分别得出3个遮挡等级各自的map。根据3个map与对应遮挡等级的权重（等级1权重0.2，等级2权重0.3，等级3权重0.5）相乘后得到的分数相加得出最终的分数。

（2）样本不平衡问题：

我们最后将修改测试文件中的数据集路径，将其改为测试集所在的路径。之后得出模型在测试集上的map作为最终的分数。