1. 测试

1、测试对象：

主要针对后端的三个主要功能做测试：

(1)通过上传图片预测结果

(2)搜索对应类别的垃圾图片

(3)关键字搜索相应名称垃圾所属的种类

三个功能分别封装在三个类里面，在python的fastapi框架下可以很方便地使用。三个功能所属的类依次为：predictor、getpictures、Ksearch。所以只需针对这三个类做单元测试即可。

1. 测试框架

由于是使用的python语言，所以直接使用自带的unittest模块来进行测试。

3、测试方法

采用动态测试和静态测试结合的方法。既测试出代码的逻辑错误，又验证相应的函数能否在各个边界条件下正常运作。

1. 测试流程

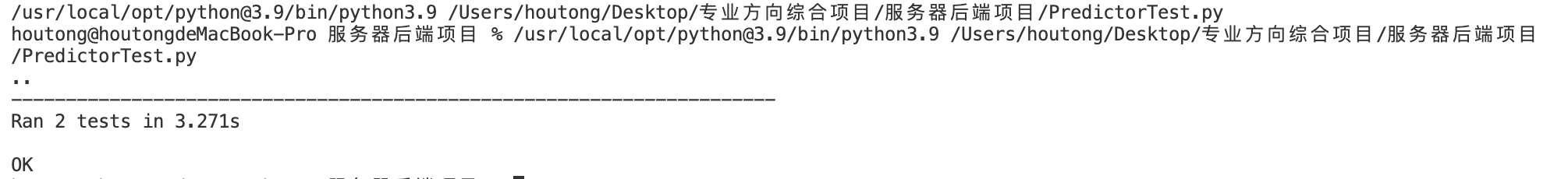
对于上述三个类分别编写测试类，再分别测试。(后端文件中，以test结尾的py文件都是测试类)

1. 测试环境

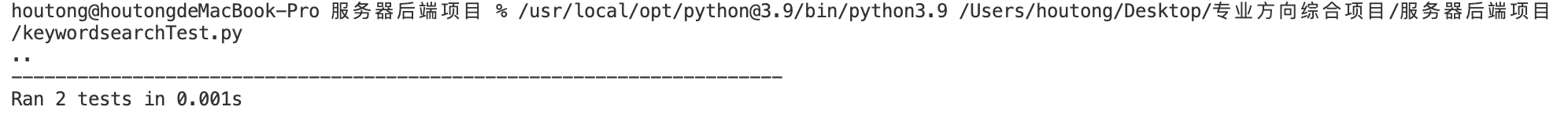
对于获取图片的getpictures类，需要在云服务器上才能测试(因为是访问服务器存储的图片)。其余两个类只要有模型就可以进行测试。

6、测试结果

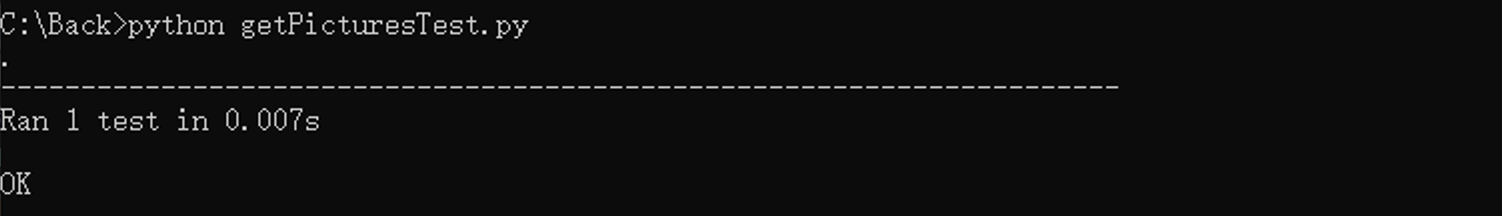
predictor\_test:



Ksearch\_test:



getpictures\_test(云服务器上运行):



1. 使用手册
2. 主页面



从上到下主要分为三个功能：

(1)通过上传图片预测结果，进入到2的界面

(2)搜索对应类别的垃圾图片，进入到3.1的界面

(3)关键字搜索相应名称垃圾所属的种类，进入到4的界面

1. 图像识别结果

上传图片之后，等待5~10秒钟，会进入如下界面，告知用户这张图片中的垃圾所属类别。



1. 获取图片

3.1 显示小类

进入页面后，会显示这种垃圾有哪些细分的种类，并且点击其中一种才会显示其对应的图片，并进入3.2的界面



3.2 显示图片

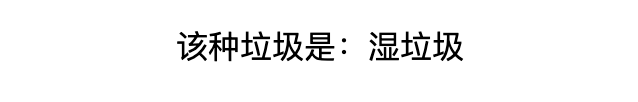
显示垃圾对应的图片。



1. 关键字搜索垃圾类别

如果搜索的结果在数据集中存在的话，显示4.1的界面；否则，显示4.2的界面

4**.**1 搜索成功



4.2 搜索失败

