

# *Imacessor*

——基于深度学习的图片处理网站

开发者：吴海隽 *Hoichun H. Ng*

**id: 2017013619**

## 一、项目描述

‘Imacessor’ is a website that provides image-processing services based on deep learning methods and models.

“易码世 Imacessor”是以 python/Django 为 Web 框架搭建的 Web 应用程序，使用经过 Deep Learning 方法训练的深度神经网络，对用户提供的图像进行实时处理与结果反馈。目前只发布了简单图像（整图）分类一种功能，当前版本为 1.0.0。

## 二、开发环境

操作系统：Windows 10

CPU：Intel® Core™ i7-7500U CPU @ 2.70GHz 2.90GHz

IDE：JetBrains PyCharm 2019.2.1 x64

编程语言：Python (Version 3.7.4)

内置数据库：SQLite3 (SQLite Studio 3.2.1 辅助开发)

Web 主框架：Django (Version 2.2.5)

## 三、功能介绍

### (1) 简单整图分类 Image Classification

- 公开深度模型
- 来源：<https://github.com/mrhelloboy/Image-classify-by-using-Django>
- 用户上传一张本地图片并提交，分类器判断当前图片描述对象的类别，最后结果返回到浏览器进行显示。
- 程序中内置若干图片，作为分类器的默认测试集。  
(位置相对路径：{BASE\_DIR 程序包存储根目录})

\imacessor\static\images\classifier\_test\_set)

- 用户对图片的分类操作会留下历史记录，历史图片记录的存储相对路径为：“{BASE\_DIR 程序包存储根目录} \imacessor\static\images\history”
- 该简单分类器只能对下述 12 种对象的图片进行分类：

The Great Wall	School	Restaurant	Mount Tai
Sea Beach	Street	ATM	Mangrove Tree
City Square	Airport	Station	Shopping Mall

## (2) 操作历史记录查看与删除

在历史记录查询界面，用户不仅可以查看过往载入图片进行分类的结果，还可以进行记录删除操作。普通用户只能查找和删除属于自己的数据，而管理员用户可以访问和操作在数据库当中的任意一条记录。

## 四、网页部署

### (1) 部署方式与步骤

# 以下仅介绍 win10 上的部署，且计算机必须预装 python 3.x。

- a. 解压“Imacessor.zip”压缩文件，无需安装；
- b. 打开 windows 命令行 cmd，进入 manage.py 所在的目录；
- c. 输入指令“python manage.py runserver”，等待其启动；
- d. 命令行提示的末尾应大致为

```
''' Starting development server at http://127.0.0.1:8000/
```

```
Quit the server with CTRL-BREAK. '''
```

在 Chrome 或者 Microsoft Edge 浏览器键入' <http://127.0.0.1:8000>'

并访问，即可启动该网页程序。

（若当前返回的地址与端口与上述情况不一致，则按照 cmd 提示访问即可）

### (2) 网页界面截图

图 1 - 网站主界面，可进行注册或者登录操作 ↓

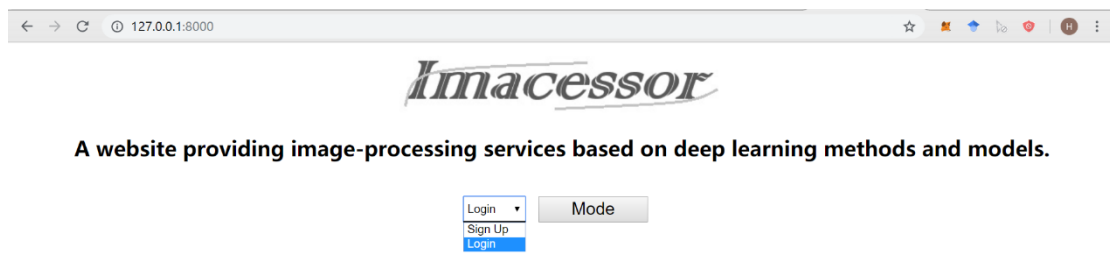


图 2 - 登录界面，使用已注册的账号登录 ↓

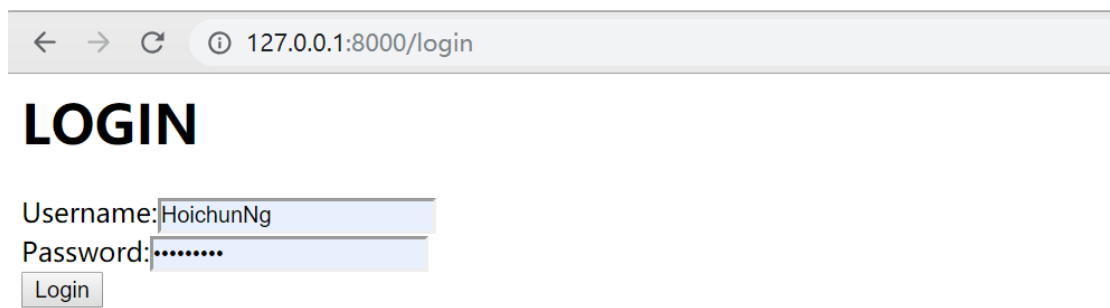
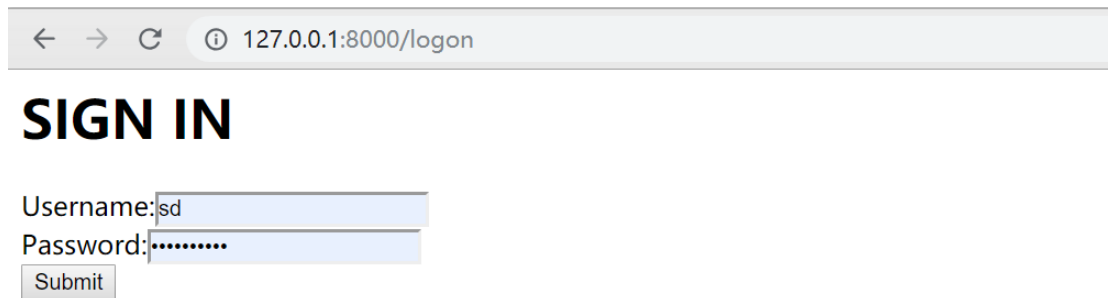


图 3 - 注册界面 ↓



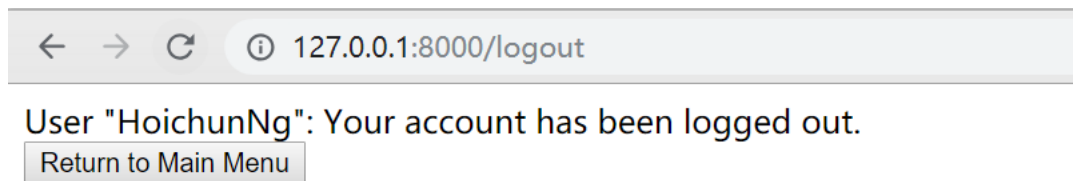
← → ↻ ⓘ 127.0.0.1:8000/logon

# SIGN IN

Username:

Password:

图 4 - 注销界面，退出后可返回主界面 ↓



← → ↻ ⓘ 127.0.0.1:8000/logout

User "HoichunNg": Your account has been logged out.

图 5 - 欢迎界面，上次登录未注销，再访问网站自动登录 ↓

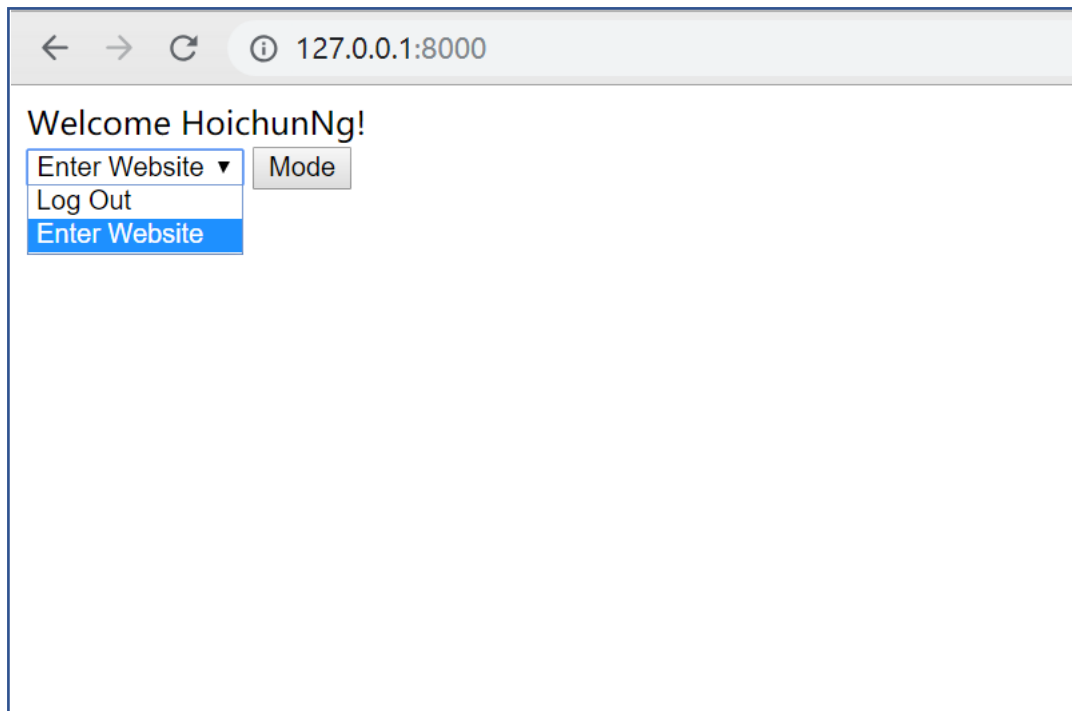


图 6 - 图片分类器界面，点击“选择文件”上传图片，点击 Submit 提交 ↓



图 7 - 图片分类器界面，成功获取分类结果 ↓

# Process Result

Your image: MountTai1.jpg



What's in the picture?

mount tai

[Try Another One](#)

[Check Operation History](#)

[Return to Main Page](#)

---

图 8 - 历史操作查询界面，可以删除一条到多条记录↓

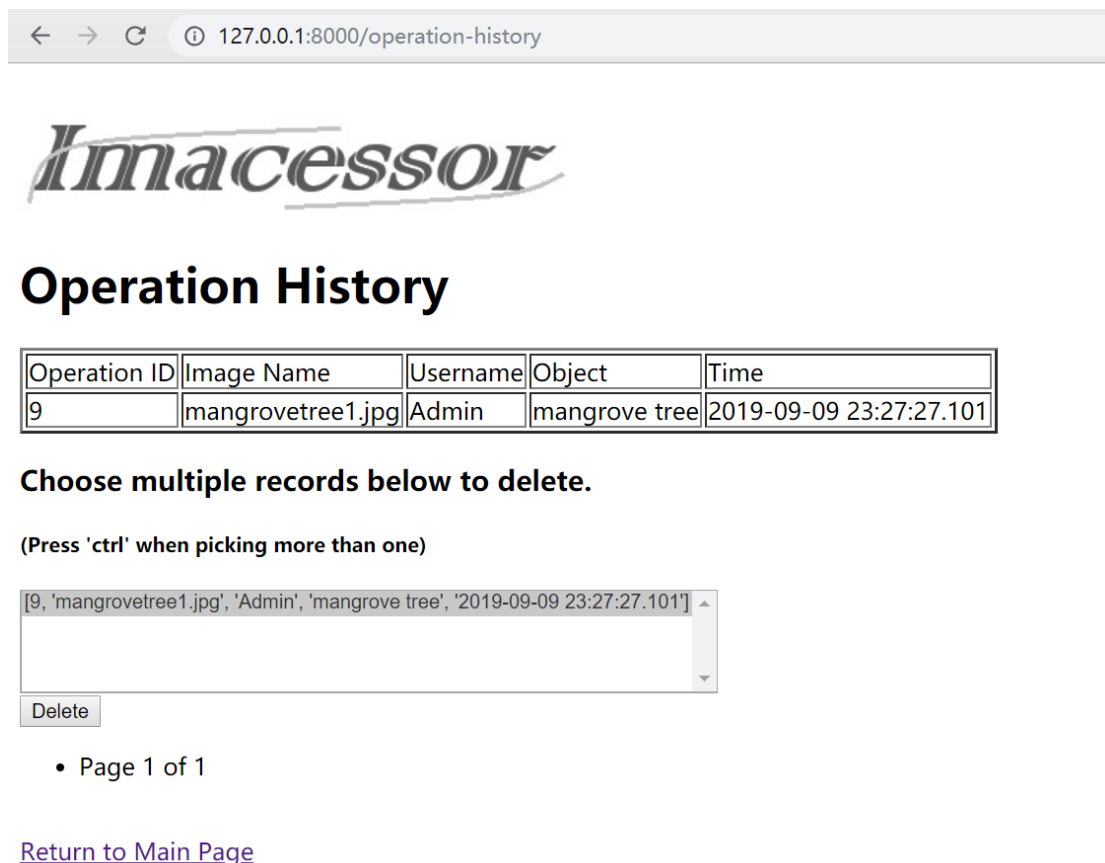
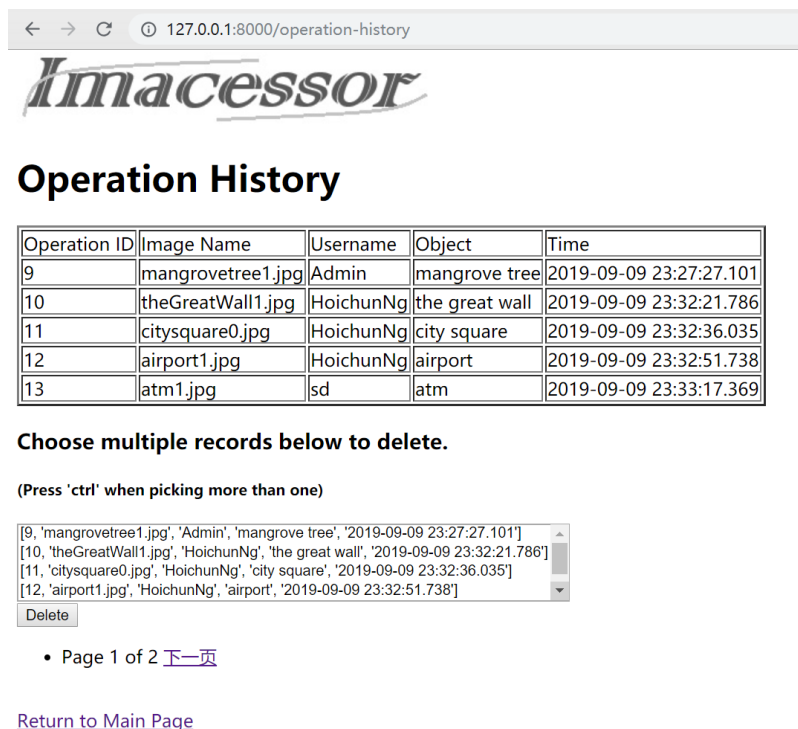


图 9 - 历史操作查询界面，实现分页显示与翻页功能 ↓



(3) Python 第三方库调用  
Django (Version:2.2.5)

Pillow (Version:6.1.0)  
Tensorflow (Version:1.14.0)

#### (4) 其他模块调用/参考

- 分页显示前端  
来源: <https://www.jianshu.com/p/332406309476>

## 五、 功能亮点说明

### 1. 实现管理员功能

管理员账户的账号和密码在内嵌数据库 onlineDB.db 中预存, 账号是 'Admin', 密码是 'iamadministrator'。管理员可以在历史记录显示中查看所有人的操作记录, 并且可以删除任意一条记录。

### 2. 支持用户批量删除历史记录

在查询历史记录的界面, 用户还可以进行记录删除操作。除了管理员外, 普通用户只能查找和删除属于自己的数据。界面下方有一个多选框, 支持批量删除操作记录, 图片历史文件夹中相应的图片也会同时被删除。

## 六、 实验体会

在完成本次综合实验之前, 我虽然先后学习了前端和后端的课程, 但是对二者是如何结合起来的, 以及二者如何将信息与表示分开诸如此类的问题一知半解。通过本次大作业的实践, 我对前端与后端的联系与配合有了更直观、更感性的认识。

同时, 在本次实验中, 我常需要将自己的代码与其他人的开源项目的代码拼接起来。一开始, 这个过程非常不顺利, 常常会报很多 bug 且难以查找病因。后来, 我发现如果一开始能够读一遍别人的代码, 大致知道其内部的逻辑与结构之后再去进行组装操作, 而不是完全将其当成黑盒去处理, 就能节省下不少临时去 debug 的时间和精力。