程序设计实践 大作业文档

沈冠霖 2017013569

武笑石 2016011595

一． 环境说明、模型加载以及测试

1．依赖环境

本工程需要安装以下依赖：

库==版本号	开源协议	项目地址
python==3.6.7	GPL	https://www.python.org/
scipy==1.3.1	BSD 3-Clause	https://github.com/scipy/scipy
tensorflow==1.13.1	Apache License	https://github.com/tensorflow/tensorflow
urllib3==1.25.3	MIT License	https://github.com/urllib3/urllib3
sqlite==3.29.0	无限制	https://github.com/mackyle/sqlite
django==2.2.4	BSD	https://github.com/django/django
numpy==1.16.2	BSD 3-Clause	https://github.com/numpy/numpy
imageio==2.5.0	BSD 2-Clause	https://github.com/imageio/imageio
opencv==4.1.1	BSD 3-Clause	https://github.com/opencv/opencv
matplotlib==3.1.1	BSD 3-Clause	https://github.com/matplotlib/matplotlib
网络模型：		
MultiNet	MIT	https://github.com/MarvinTeichmann/MultiNet
GeoNet	MIT	https://github.com/yzcjtr/GeoNet

2．模型

除此之外，本项目需要手动下载模型参数并放置到正确的位置上。说明如下：

模型下载：

模型下载地址：https://cloud.tsinghua.edu.cn/d/efaabb8d8a3a4c1993ca/

有效期 30 天，如 30 天内未能及时下载，或发现链接失效，请联系武笑石。

微信：tgxs002

邮箱：13521428950@163.com

配置步骤：

请将 model_sn.meta, model_sn.data-00000-of-00001, model_sn.index, vgg16.npy 下载后放置到 Kitti/DATA 文件夹下。

请将 model.ckpt-15999 下载后放到 Kitti/RUNS/KittiSeg_pretrained 文件夹下

如实在无法成功配置，可访问我们的服务器：166.111.81.95:8000/logon 进行测试。我们的服务器将开放到 10 月 1 日。

2．测试

请使用“测试用图片”文件夹下的图片进行测试，为达到良好的测试效果，url 链接的图片也请选用与 KITTI 数据集接近的道路图片。

二．前端和前后端交互部分

前端主要用的是 html 自带的特性还有 django 的模板，在图片部分用了一点 ajax。前后端交互是 django 实现的。前后端交互的 url 读取在 hw4/url.py,前后端交互的处理函数在 hw4/view.py，前端代码在 templates 文件夹中，每个文件对应一个页面。

1.登录，注册，注销

登录，注册这两者实现差不多：都是前端用 html 自带的表单，使用 text/password 输入框进行输入，点击按钮发送对应的 post 请求到 django。django 调用后端接口进行登录，注册的处理。处理后在前端用 django 模板显示注册/登录的成功，失败信息。

注销在前端只用了一个按钮，点击后访问 logout，调用后端接口处理，注销后重定向到登录界面。

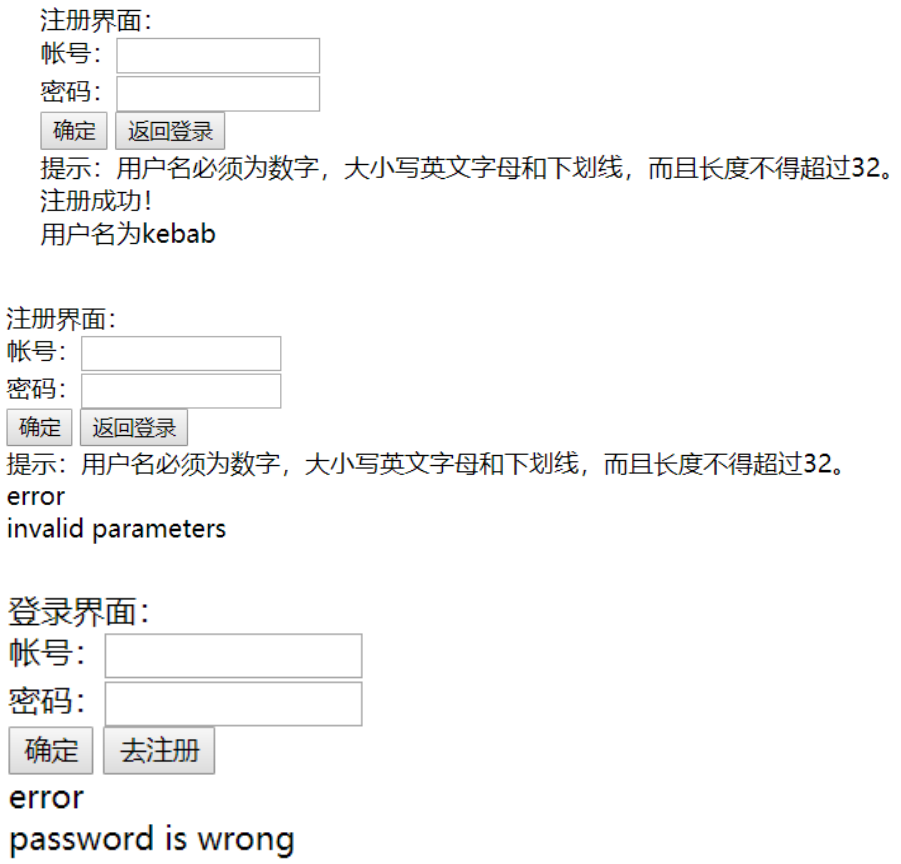

图 1. 注册和登录界面示意图

2.图片显示

图片的传输也是用 html 表单自带的文件选择框（image/jpeg）或者字符输入框（输入链接）实现的。不过请求的发送和接收我用了 ajax 技术。每输入一个图片，我会用 ajax 分别给后端发两个请求，对应两个不同网络的操作。因为 ajax 的性质--不需要刷新，请求成功后自动修改局部界面，我能做到某个网络完成后，立即返回对应结果。Ajax 代码在 index.html 中。

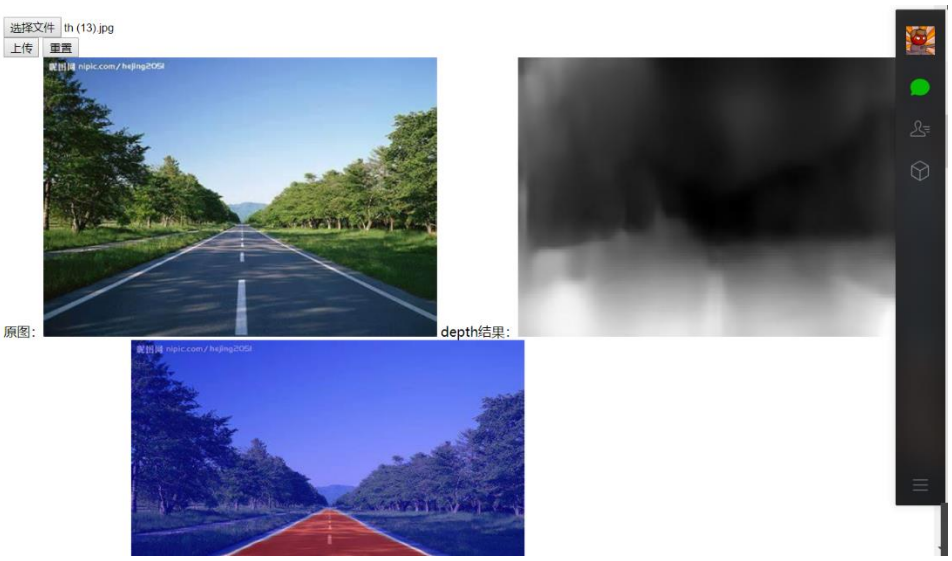

图 2. 选择图片或链接输入和显示结果图片。

3.历史记录查询

我通过 html 表单自带的日期选择框选择日期，手动将其转换为时间戳发送给后端，查询对应日期的历史记录。如果没选择日期，就查询所有历史记录。查询完毕后，我将记录按时间降序排序，之后用 django 的模板实现了分页显示,应用了 Paginator 库把每页分成了

10 个记录，可以显示当前页数和上一页，下一页。历史记录显示的代码在 history.html。

图 3.查询历史记录和显示历史记录

4.查询历史记录细节

我在之前的历史记录页面中，用 django 模板给每个历史记录的查询按钮按照用户名+id 绑定了显示细节的 url，点击按钮就会进入对应 url，发送用户名+id 给后端进行查询，返回对应的记录和图片文件。合法的历史操作（图片格式正确，可以下载）能正确显示各种细节。非法历史操作（图片格式不正确，无法下载等）会进入错误页面，显示这个历史操作非法。

用户名：sgl17
操作编号：10
操作类型：segmentation
文件来源：网络下载
操作时间：2019-09-09 21:00:26.575

原始图片：

图 4. 查看历史记录细节

5.删除历史记录

删除单个历史记录的步骤和之前类似，也是用 django 模板给每个历史记录的删除按钮绑定了 url，进入后发指令给后端删除记录。按时间批量删除则和查询类似，用 html 表单自带的日期选择框选择日期，然后转成时间戳，在后端查询所有对应的历史记录。然后将返回的记录挨个删除。

6.安全性

为了保证安全性，考虑到用户直接输入 url 的情况，我在每个直接和前端交互（url 中用到的函数，写在 view.py 里的）的函数前都加了登录判定，调用后端接口判断 session_id 是否已经登录。这样能保证只有登录的用户才能看主页，历史记录，进行注销和删除，只有未登录用户才能登录，注册。如果登录信息不正确，会自动跳转到错误界面。

在访问历史记录细节和删除历史记录的时候，用户也可能直接输入 url 访问不存在的，或者自己不可访问的（比如他人的数据），这里后端做了用户名验证，保证用户只能访问存在的，自己有权限访问的数据，否则就会报错。

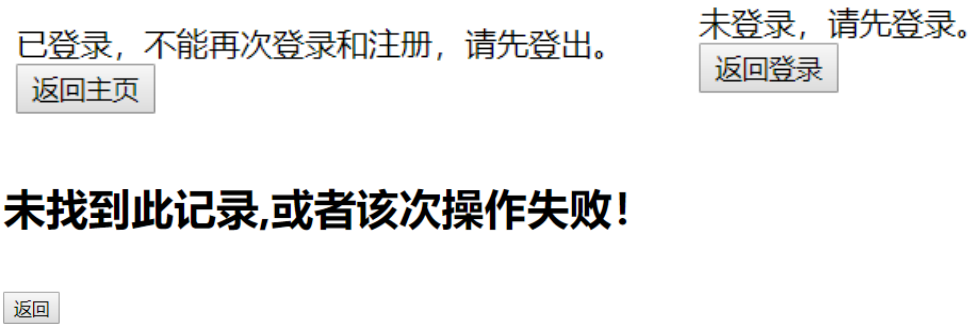

图 5. 错误提示页面。前两张是 error.html，第三张是 not_found.html 的。

三． 后端重点说明

本项目后端基于 homework4 的功能进行开发，在原有功能的基础上进行扩充。根据作业要求，本项目在 homework4 的功能中加入了图片存取以及神经网络的加载、推理功能。

1. 图片存取

本项目中图片以图片格式存放于数据库文件夹中，由操作系统的文件系统进行管理。当用户发出操作请求时，请求中的原图以及处理后的结果都会被保存。此后，当用户查询或请求删除具体某次操作记录时，后端将根据要求对文件夹中的对应图片进行处理。

2. 神经网络的加载与推理

神经网络是本次作业中的重点内容。我们组将 GitHub 上的开源深度学习项目集成到了本项目中，实现了 KITTI 数据集场景的道路语义分割以及深度图推理功能。

这两个任务是计算机视觉中的重要任务，在自动驾驶领域尤其重要。其中道路语义分割是指根据单张道路图片推断哪些像素属于当前的行车道，为自动驾驶提供依据。推断道路图片的深度图的意义更是不言而喻，因为场景的几何信息是自动驾驶得以实现的根基所在。

本项目在服务器启动时将模型加载入内存中，当用户请求到来时，即刻调用接口进行推理。

四．参考资料：

登录，注册：https://www.jianshu.com/p/033217dbfe25

ajax 请求和显示图片：https://code.ziqiangxuetang.com/django/django-ajax.html

https://github.com/yujuan123/file-upload/tree/master/public

分页：https://www.jianshu.com/p/332406309476

深度学习语义分割：

https://github.com/MarvinTeichmann/MultiNet

深度学习深度图推理：

https://github.com/yzcjtr/GeoNet