

# 2018-2019 夏季学期 程序设计实践

## 大作业要求

修订日期：2019 年 9 月 1 日

### 整体要求

使用 Python 3 语言开发一个简单的网站，向使用者提供基于深度学习的图片处理服务。

主要考察服务器端接口的开发，对前端网页除实现基本操作和图片展示之外无其它功能或性能要求。

作业中可以使用第三方库来负责 HTTP 协议解析、数据库处理、深度学习图像处理等服务，但主要网络接口的业务逻辑需要是自己写的。

### 项目细节

### 项目环境

1. 项目应当支持在 Python 3.5、3.6 或者 3.7 上正确稳定运行。
2. 项目应当支持在 AMD64 CPU 平台的 Windows 10 或者 Ubuntu 16/18（64 位）系统上正常运行
  - a) CPU 使用量可控，在没有用户请求、没有排队中的识别任务的时候占用很低的 CPU
  - b) 深度学习相关代码支持在无 CUDA 或 OpenCL 的环境下运行（评分时接受运算很慢的情况，不扣分）
  - c) 深度学习模块的内存消耗稳定可控
  - d) （不考虑第三方库的稳定性的前提下）内存占用量有限可控
3. 项目应当支持 MySQL 5.7+ 和 SQLite 3 数据库中的至少一种
4. 提供部署方式的说明文档，且文档中包括部署后的网页界面截图
  - a) 需要列出项目用到的 Python 第三方库，包括项目地址（比如 GitHub 网址或 PyPi 上包的网址）、授权协议（指 MIT、Apache、BSD 或者 GPL 等协议）
5. 项目应尽量使用 Python 官网提供的 CPython 和 PyPi 包就可以执行，如果确实需要 Conda 等第三方发行的环境的，需要在说明文档中解释原因

### 基于深度学习的图像处理

常见的支持深度神经网络的库有 PyTorch、MXNet、Caffe、Tensorflow 等，它们在大部分系统都有官方提供的 Python 包，Windows 平台也可以从 <http://www.lfd.uci.edu/~gohlke/pythonlibs/> 下载。

使用深度学习技术处理图片，常见的对象识别类任务有：整图分类 Classification、对象检测 Detection、图像分割 Segmentation 等；和语义相关的任务有打标签（Image Labeling）和起标题等；甚至还能实现风格转换、抹除水印等复杂需求。比如一个很小的 MTCNN 网络就可以在图像中检测出绝大多数人脸。

**本次作业中要求使用至少一个深度神经网络处理图片，推荐使用两个不同类型的神经网络来得到不同类型的结果。**项目可以使用没有版权问题的公开深度模型和框架。本次作业的考察重点在正确加载模型文件并调用模型推理（Inference）接口上。

运行网络时的峰值内存占用应该小于 12GB，每个深度网络的计算耗时应当在 20 秒以内（测试 CPU 主频不低于 3GHz，型号可能是 i7-8xxxU）。如果所用模型过于庞大，可以适当缩小输入网络的图片尺寸，也可以替换成精度可能低一些的简化模型。

## 用户模块

项目需要提供用户**注册、登录和注销**等基本功能。这三个接口可以使用 **Django** 等框架提供的功能来完成。项目需要支持同一用户在多处同时登录，可以使用 **Cookies** 来维护登录状态。

用户使用图片处理功能时，应当记录每次操作的输入图片和处理结果，图片应当存储到服务器程序所在机器的某个专用文件夹下（可以是子文件夹），处理结果和操作时间戳等信息可以存放到数据库。

用户应当可以通过网页查看自己曾经执行的图片处理操作，相关接口应当自己实现，而不是使用调用 **Django Admin** 模块等直接暴露数据表等方式。

用户应当可以删除某一条或者某些操作历史记录。

**推荐**在各个接口都检查用户身份（注册和登录除外），身份不一致则拒绝操作。

**管理员功能是可选完成的。**如果实现此功能，则管理员应当可以登录、查看所有人或者指定用户的操作记录，而且有权限删除任意操作记录。管理员账户的账号密码可以是预先创建在数据库中的，或者是项目配置文件中的，**不应该是直接写到代码里的。**

## 网页基础操作

用户需要先登录，方可使用图片处理功能。

图片处理功能至少**需要支持用户从网页上传一张 JPEG 格式的图片给后台识别**，推荐也支持用户上传图片的 HTTP 网址的方式，来由后台自动下载并处理。

处理完成时，前端网页应当将输入图片本身和处理结果在同一页面（的不同位置）都展示出来。

如果后台使用了多个深度网络处理图片，**推荐使用 AJAX、StreamHttpResponse** 等技术把每个网络的处理结果立即返回给前端网页。

显示历史操作列表时，**要求支持分页显示**：显示某个操作的具体信息时，应当能够看到操作输入的图片内容，且图片需要是从服务器后台发送过来的，而非指向用户操作时提供的 URL。

**推荐**提供按时间段查询历史记录的功能，还可以实现批量删除等复杂操作。

## 评分标准

如果完成了以上章节要求必须完成的那些功能，没有实现任何推荐的功能，且代码功能基本正确，则得分在 80 分左右。文档完整、代码风格良好、无安全漏洞的最多可以得到 85 分。代码混乱、文档描述不清，但无 bug 的，得分最少为 75 分。发现少数 bug 的最少给 70 分，bug 很多的得分最少给 60 分。

如果实现了调用两个不同类型的深度网络并正确返回处理结果，则评分基准为 84 分，如果实现了更多标记了“推荐”的功能，评分基准逐级上浮。所选深度网络的使用过程较为复杂的也可能加分。

实现了所有推荐功能、网页功能完整、代码无逻辑 bug 或安全漏洞的评分基准为 90 分，代码风格良好加分，文档完整加分，代码风格过于混乱则扣少量分，发现 bug 或者安全漏洞视严重程度扣分。

高分段（95 分）以上的，根据不同项目的总体质量的高低排序并给分。

没有实现调用深度网络的，基准分为 65 分。根据 bug 数量、其它接口的实现数量增减。