系统设计报告

***（“识食物者”---美食识别系统）***

东软教育

**更改履历**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 版本 | 更改时间 | 更改人 | 更改章节 | 状态 | 更改描述 |
| 1 | v0.1 | 0801-13:20 | 郭龙强 | 3 | 新建 | 后端设计 |
| 2 | v0.1 | 0801-13:10 | 武泳桦 | 2 | 新建 | 系统技术框架 |
| 3 | v0.1 | 0801-13:30 | 李纪元 | 4 | 新建 | 深度学习模型训练 |
| 4 | v0.1 | 0801-13:20 | 王蕊 | 2.2 | 新建 | 小组成员分工 |
| 5 | v0.1 | 0801-13:20 | 崔爽 | 3 | 新建 | 系统公共代码、数据爬取 |
| 6 | v0.1 | 0801-13:00 | 王蕊 | 3.4 | 新建 | 界面原型图 |
| 7 | v0.2 | 0801-16:00 | 武泳桦 | 2.1.1 | 修改 | 修改场景视图 |
| 8 | v0.2 | 0801-16:05 | 崔爽 | 3.1 | 修改 | 公共代码部分流程图及功能说明 |
| 9 | v0.2 | 0801-16:05 | 郭龙强 | 3.3 | 修改 | 修改流程图及功能说明 |
| 10 | v0.3 | 0801-16:05 | 崔爽 | 3.2 | 修改 | 修改流程图及功能说明 |
| 11 | v0.4 | 0801-16:00 | 王蕊 | 3.4 | 新建 | 前端框架说明 |
| 12 | v0.4 | 0801-16:30 | 王蕊 | 3.4 | 修改 | 界面原型图 |
| 13 | v0.5 | 0801-17:10 | 王蕊 | 4.1 | 新建 | 保护环境与可持续发展说明 |
| 14 | v0.5 | 0801-17:44 | 李纪元 | 1.2 | 修改 | 添加了术语表词条 |
| 15 | v0.6 | 0801-17:53 | 武泳桦 | 2.1.2 | 修改 | 修改逻辑视图 |
| 16 | v0.6 | 0801-17:56 | 武泳桦 | 2.1.3 | 修改 | 修改物理视图 |
| 17 | v0.6 | 0801-18:01 | 武泳桦 | 2.1.4 | 修改 | 修改处理视图 |
| 18 | v0.6 | 0801-18:10 | 武泳桦 | 2.1.5 | 修改 | 修改开发视图 |
| 19 | v0.7 | 0801-18:11 | 武泳桦 | 1.4 | 新建 | 需求说明 |
| 20 | v0.7 | 0801-18:20 | 武泳桦 | 2.3 | 新建 | 系统开发环境 |
| 21 | v0.8 | 0801-18:03 | 李纪元 | 1.4.1 | 增加 | 添加功能性需求 |
| 22 | v0.8 | 0801-18:10 | 武泳桦 | 1.4.2 | 增加 | 添加非功能性需求 |
| 23 | v0.8 | 0801-18:22 | 李纪元 | 2.3 | 增加 | 添加了开发环境条目 |
| 24 | v0.9 | 0801-18:47 | 武泳桦 | 2.3 | 增加 | 添加了数据增强公共代码设计 |
| 25 | v0.9 | 0801-18:47 | 武泳桦 | 2.3 | 增加 | 添加了模型部署公共代码设计 |
| 26 | v0.9 | 0801-18:54 | 李纪元 | 3.1 | 增加 | 添加了公共代码设计条目 |
| 27 | v0.9 | 0801-19:02 | 李纪元 | 3.5.2 | 修改 | 修正了流程图的错误 |
| 28 | v0.9 | 0801-19:03 | 李纪元 | 3.5.4 | 修改 | 修改了相关数据表的条目 |
| 29 | v0.9 | 0801-19:07 | 王蕊 | 2.3 | 增加 | 添加了数据规范化公共代码设计 |
| 30 | V1.0 | 0801-19:12 | 李纪元 | 3.5.5 | 新建 | 新建了关于模型训练设计的说明 |
| 31 | V1.0 | 0801-19:30 | 武泳桦 | 4.1 | 修改 | 修改了环境与可持续发展 |
| 32 | v1.0 | 0802-12:10 | 王蕊 | 4.1 | 增加 | 保护环境与可持续发展说明 |
| 33 | v1.0 | 0802-13:00 | 王蕊 | 3.4.3 | 修改 | 时序图 |
| 34 | v1.0 | 0802-15:00 | 王蕊 | 3.4.4 | 新建 | 流程图 |
| 35 | v1.0 | 0802-15:17 | 李纪元 | 3.5.2 | 增加 | 图片说明文字 |
| 36 | v1.0 | 0802-15:18 | 李纪元 | 3.5.3 | 增加 | 图片说明文字 |
| 37 | v1.0 | 0802-15:18 | 李纪元 | 3.5.2 | 修改 | 替换了图片 |
| 38 | v1.0 | 0802-15:19 | 李纪元 | 3.5.3 | 修改 | 替换了图片 |
| 39 | v1,0 | 0802-19:00 | 武泳桦 | 2.1 | 增加 | 图片文字说明 |
| 40 | v1,0 | 0802-19:00 | 武泳桦 | 2.1.4 | 修改 | 修改处理视图 |
| 41 | v1.0 | 0802-19:30 | 崔爽 | 3.1.2 | 增加 | 增加流程图说明 |

状态：新建、增加、修改、删除。

目录

[1 引言 1](#_Toc19390)

[1.1 编制目的（负责人：李纪元） 1](#_Toc23666)

[1.2 术语表（负责人：全体成员） 1](#_Toc3872)

[1.3 参考资料（负责人：郭龙强） 2](#_Toc24869)

[1.4 需求说明 2](#_Toc9115)

[1.4.1 功能性需求（负责人：李纪元） 2](#_Toc3736)

[1.4.2 非功能性需求（负责人：武泳桦） 3](#_Toc25273)

[2 系统开发环境 3](#_Toc25212)

[2.1 系统技术构架（负责人：武泳桦） 3](#_Toc31853)

[2.1.1 场景视图 3](#_Toc31199)

[2.1.2 逻辑视图 4](#_Toc6890)

[2.1.3 物理视图 5](#_Toc14869)

[2.1.4 处理视图 6](#_Toc10844)

[2.1.5 开发视图 7](#_Toc12832)

[2.2 人员分工（负责人：武泳桦） 7](#_Toc2873)

[2.3 系统开发环境（负责人：李纪元） 8](#_Toc46)

[3 系统设计 9](#_Toc27714)

[3.1 系统公共代码设计（负责人：全体成员） 9](#_Toc9246)

[3.1.1 功能说明 9](#_Toc12657)

[3.1.2 流程图 9](#_Toc12086)

[3.2 数据爬取及数据处理（负责人：崔爽） 13](#_Toc14175)

[3.2.1 功能说明： 13](#_Toc615)

[3.2.2 流程图： 13](#_Toc32500)

[3.3 Python web Flask后端（负责人：郭龙强、王蕊） 14](#_Toc6100)

[3.3.1 注册 14](#_Toc27471)

[3.3.2 登录 15](#_Toc31099)

[3.3.3 图片识别 16](#_Toc19537)

[3.3.4 用户反馈 17](#_Toc22059)

[3.3.5 后台管理 18](#_Toc568)

[3.4 Flask-Bootstrap前端（负责人：王蕊） 18](#_Toc28464)

[3.4.1功能说明 18](#_Toc11033)

[3.4.2原型图设计（图片为绘制） 19](#_Toc11357)

[3.4.3流程图 20](#_Toc4424)

[3.4.4时序图 22](#_Toc20821)

[3.4.5相关数据表 22](#_Toc2543)

[3.5 深度学习训练模块（负责人：李纪元、武泳桦） 23](#_Toc26189)

[3.5.1 功能说明 23](#_Toc11615)

[3.5.2 流程图 23](#_Toc14230)

[3.5.3 时序图 24](#_Toc24287)

[3.5.4 相关数据表 25](#_Toc10922)

[3.5.5 设计说明 26](#_Toc2056)

[4 环境与可持续发展 （负责人：王蕊、武泳桦） 26](#_Toc23711)

# 引言

## 编制目的（负责人：李纪元）

为开发人员提供明确的需求方案、流程设计、数据库设计、包及类结构时间，以明确开发思想。

通过本文档的整体与详细设计，开发团队可以定义结构化、阶段化的开发流程，以对软件开发各阶段进行更清晰的评审，降低软件的开发风险与消耗，使软件具有良好的可拓展性。

本文档通过对软件架构与接口的详细定义，来提高软件的可维护性，帮助软件维护人员快速建立起对于软件整体的认识，并进一步在细节上得到更深入的理解，以求在之后的维护中进行改善。

## 术语表（负责人：全体成员）

|  |  |  |
| --- | --- | --- |
| 词汇名称 | 词汇含义 | 备注 |
| CNN | 卷积神经网络 |  |
| 数据生成器 | 用于导入数据并进行增强 |  |
| 卷积基 | CNN模型中一系列的池化层与卷积层 |  |
| 分类器 | 由若干Dense层在卷积基之上相叠而成的神经网络层次 |  |
| 模型微调 | 通过“解冻”卷积基的若干层次，使得预训练模型的卷积基可以更好地适应新的数据 |  |
| 优化器 | 优化器决定神经网络反向传播修改权重的策略 |  |
| 损失函数 | 神经网络衡量误差的标准 |  |
| 评估函数 | 神经网络评估模型正确率的标准 |  |
| 数据增强 | 通过对图片进行位移、翻转、错切等加强数据，能够有效防止过拟合 |  |
| 过拟合 | 模型在训练集上的正确率显著高于验证集正确率的现象 |  |
| 训练集 | 用于训练的数据集 |  |
| 验证集 | 用于验证的数据集，每轮训练进行一次验证 |  |
| 测试集 | 用于模型训练完毕后评估正确率的数据集 |  |
| 爬虫 | 按照一定的规则，自动地抓取万维网信息的程序或者脚本 |  |
| Flask | Flask是使用 Python 编写的 Web 微框架。 |  |
| Web 框架 | Web 框架可以让我们不用关心底层的请求响应处理，更方便高效地编写 Web 程序。 |  |
| WSGI | Web服务器网关接口（Python Web Server Gateway Interface，缩写为WSGI）是为[Python](https://baike.baidu.com/item/Python" \t "https://baike.baidu.com/item/WSGI/_blank)语言定义的[Web服务器](https://baike.baidu.com/item/Web%E6%9C%8D%E5%8A%A1%E5%99%A8" \t "https://baike.baidu.com/item/WSGI/_blank)和Web应用程序或[框架](https://baike.baidu.com/item/%E6%A1%86%E6%9E%B6" \t "https://baike.baidu.com/item/WSGI/_blank)之间的一种简单而通用的[接口](https://baike.baidu.com/item/%E6%8E%A5%E5%8F%A3" \t "https://baike.baidu.com/item/WSGI/_blank)。 |  |
| Jinja2 | Jinja2 是一个现代的，设计者友好的，仿照 Django 模板的 Python 模板语言。 它速度快，被广泛使用，并且提供了可选的沙箱模板执行环境保证安全 |  |
| Bootstrap | 用于开发响应迅速、移动优先网站的受欢迎的集HTML、CSS和JS于一体的框架。 |  |
| 场景视图 | 展示系统的主要应用场景 |  |
| 逻辑视图 | 展示系统的功能型需求，即系统给用户提供哪些服务 |  |
| 物理视图 | 展示系统的部署环境 |  |
| 处理视图 | 用于描述系统软件组件之间的通信时序 |  |
| 开发视图 | 开发视图关注软件开发环境下实际模块的组织，反映系统开发实施过程 |  |

## 参考资料（负责人：郭龙强）

[1] Flask Web 开发实战 ：入门、进阶与原理解析／李辉著 ．—北京：机械工业出版社，2018.8 (Web 开发技术丛书）

[2] Flask Web开发 : 基于Python的Web应用开发实战 / (美) 格林布戈 (Grinberg,M.) 著 ; 安道译. -- 北京人民邮电出版社, 2015.1 （图灵程序设计丛书）

[3] Python深度学习/(美)弗朗索瓦·肖莱 著 张亮(hysic) 译 人民邮电出版社出版时间：2018-08（图灵程序设计丛书）

## 需求说明

### 功能性需求（负责人：李纪元）

|  |  |
| --- | --- |
| 需求 | 级别 |
| 该程序允许用户上传图片，并识别出图片为汉堡、牛排、甜甜圈、蛋挞、冰激凌、披萨中的一种，然后返回结果给用户 | 紧急 |
| 该程序可以在网页端运行，允许用户在电脑、手机端使用 | 一般 |
| 该程序允许用户注册并登录，通过加密手段保护用 | 一般 |
| 该程序允许用户进行反馈来进一步调整正确率 | 一般 |

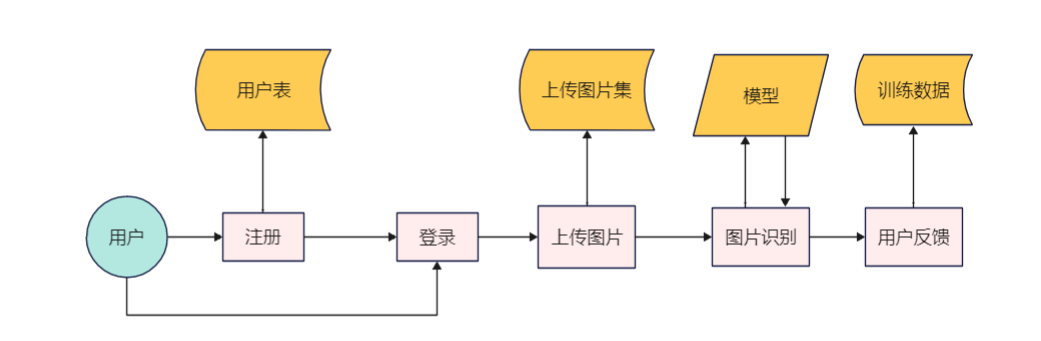
### 非功能性需求（负责人：武泳桦）

|  |  |
| --- | --- |
| 支持大图片的识别，如2048×1024像素 | 紧急 |
| 实时性：识别速度10-20s | 紧急 |
| 网站美观有食欲，不同端总是图片居中、满屏显示，  可以多平台使用，比如手机、平版、PC，要和电脑端做对比看是否是响应式的。识别完显示出来时怎么识别 更好看。 | 紧急 |
| 把程序挂到网上：华为云或腾讯云 | 一般 |

# 系统开发环境

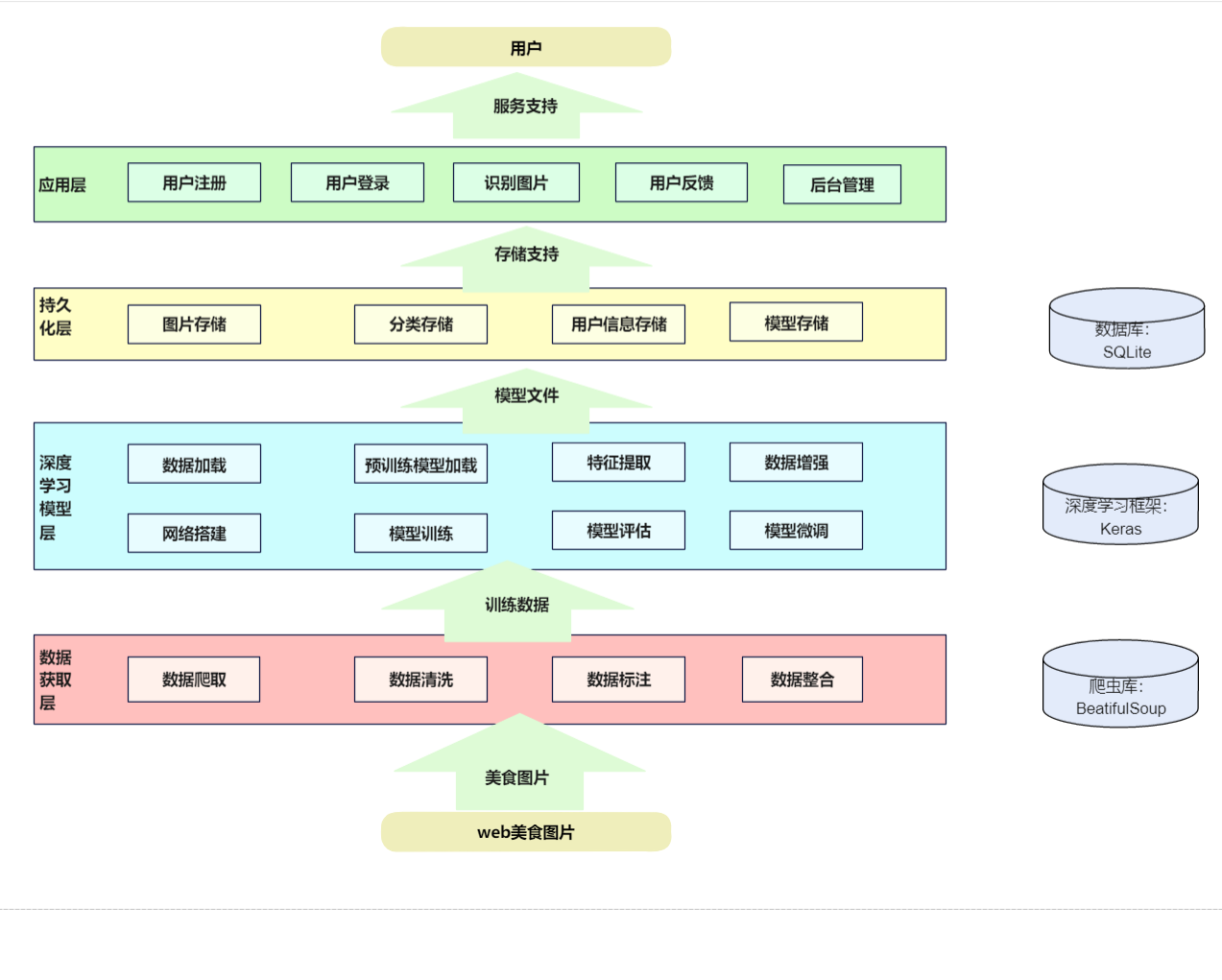
## 系统技术构架（负责人：武泳桦）

### 场景视图

* 图2.1.1 场景视图*

未注册的用户需要先注册才能使用系统的功能，注册成功后将用户信息写入数据库。已经注册了的用户直接登陆后，先上传想识别的图片，然后进行图片识别。图片上传到服务器中进行保存，然后调用模型进行图片识别，最后将识别结果返回给用户。用户可选择对本次识别结果是否满意做出反馈。对于满意的反馈，系统将记录本次用户上传的图片和标签作为一条训练数据。

### 逻辑视图

**

*图2.1.2 逻辑视图*

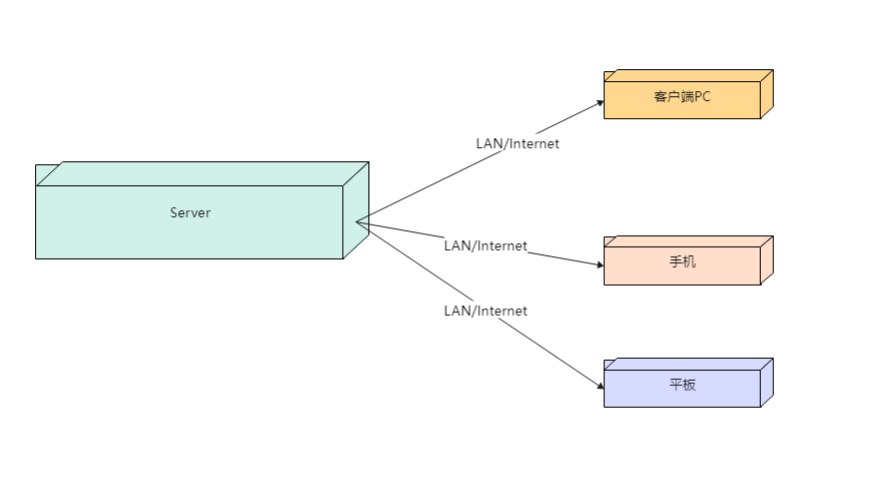
数据获取层主要负责从海量的web美食图片中爬取我们需要的六类美食数据并进行清洗、标注和整合，最后得到训练数据。这里爬虫技术主要是用的是爬虫库BeautifulSoup。

深度学习模型层主要负责模型的训练。本层展示的主要是模型训练使用到的子模块。主要采用的keras深度学习框架，利用数据获取层提供的训练数据进行训练，将生成的最好模型保存，以便应用层使用。

持久化层主要负责数据存储。主要存储图片数据、用户信息数据、模型，并为图片提供分类存储功能。数据库采用SQLite。

应用层面向用户，为用户提供服务。主要服务有用户注册、用户登录、识别图片、用户反馈和后台管理。

### 物理视图

图2.1.3 物理视图

软件系统主要运行在Server上，客户端PC、手机或平板可以通过LAN或Internet去访问Server的服务。

### 处理视图

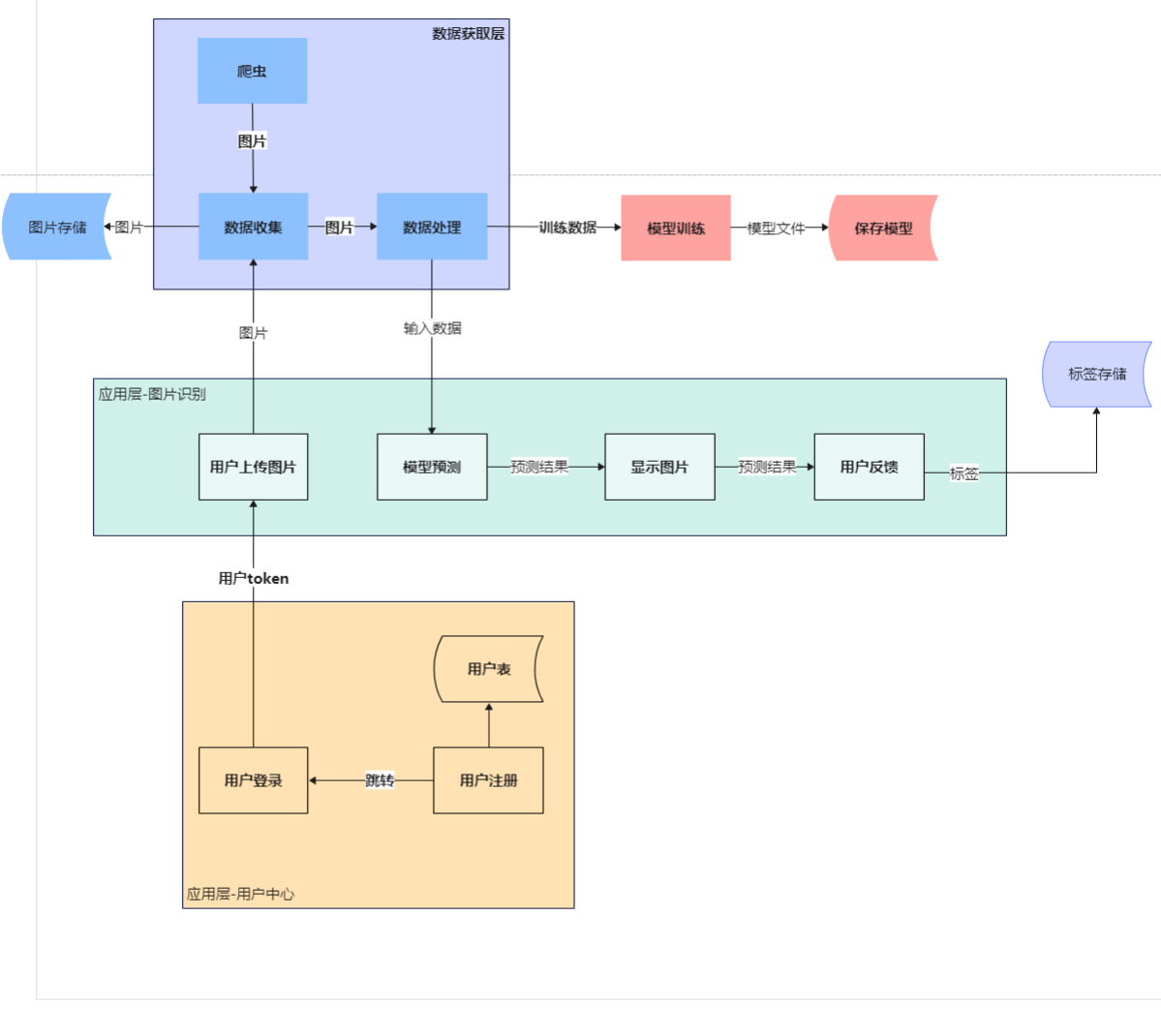


图2.1.4 处理视图

处理视图，又称过程视图、运行视图。用于描述系统软件组件之间的通信时序，数据的输入输出。用户注册后用户信息存储到数据库中，然后跳转到用户登录界面。用户登录后将用户token发送到系统，之后用户有有权限使用系统功能。用户上传图片后，系统先将图片保存，然后将图片处理成模型输入要求的数据，接着进行模型预测。然后显示图片和预测结果，用户可对本次预测结果进行反馈。用户的反馈会将本次识别结果作为训练数据保存，如果是失望的反馈，会请求用户输入正确的识别结果。

### 开发视图

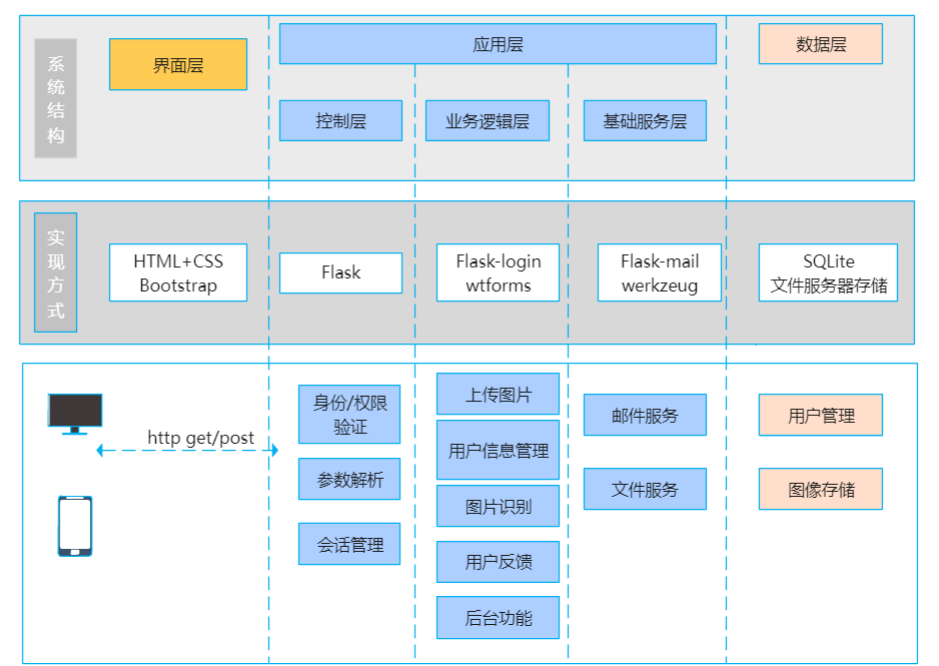


图2.1.5 开发视图

开发视图采用经典的三层架构。在界面层使用HTML+CSS+Bootstrap技术进行前端展示，通过http协议向控制层发送get/post请求。应用层中又划分为三层，分别为控制层、业务逻辑层、基础服务层。控制层中主要是用Flask框架中的功能对身份/权限验证、实现参数解析、会话管理。业务逻辑层使用Flask-login和wtforms插件，实现上传图片、用户信息管理、图片识别、用户反馈、后台功能。基础服务层使用Flask-mial,werkzeug库为系统提供邮件服务、文件服务。数据层使用SQLite和服务器本地存储提供用户管理信息存储和图像存储。

## 人员分工（负责人：武泳桦）

|  |  |
| --- | --- |
| 小组成员 | 分工 |
| 武泳桦 | 搭建神经网络并训练模型、系统功能扩展、系统整合和调试 |
| 李纪元 | 搭建神经网络并训练模型 |
| 郭龙强 | 通过flask框架搭建项目网站，主要负责上传图片、模型部署、用户反馈功能。 |
| 王蕊 | 通过flask框架搭建项目网站，主要负责界面以及登录、注册功能。 |
| 崔爽 | 网络爬虫以及数据处理 |

## 系统开发环境（负责人：李纪元）

|  |  |
| --- | --- |
| 软件/包名 | 版本 |
| Python | 3.8 |
| Keras | 2.9.0 |
| Tensorflow-gpu | 2.9.1 |
| Flask | 2.1.3 |
| Graphviz | 0.20.1 |
| Jinja2 | 3.1.2 |
| Jupyter Notebook | 6.4.12 |
| Matplotlib | 3.5.2 |
| Numpy | 1.23.1 |
| Opencv-python | 4.6.0.66 |
| Pandas | 1.4.3 |
| Pip | 22.2 |
| Requests | 2.28.1 |
| BeautifulSoup4 | 4.11.1 |
| Seaborn | 0.11.2 |
| Sklearn | 0.0 |
| Pycharm | 2022.1.4 |
| Flask-Bootstrap | 3.3.7.1 |
| Flask-SQLAlchemy | 2.5.1 |
| Flask-WTF | 1.0.1 |

# 系统设计

## 系统公共代码设计（负责人：全体成员）

### 功能说明

|  |  |
| --- | --- |
| 类名 | 功能 |
| ImgRename | 将图片统一重新命名为相同格式 |
| ImgResize | 将图片统一重新修改为相同大小 |
| DataPartition | 将数据按比例切分成训练集、验证集、测试集 |
| ImgDuplicateRemoval | 图片去重 |
| ImgSave | 存储图片 |
| ImgRead | 读取图片 |
| DataNormalize | 数据规范化 |
| PredictPic | 对传入的单张图片进行处理并预测后返回结果 |
| ModelFit | 令模型在传入的数据集上训练指定周期，用于用户反馈后的模型修正 |
| PlotAccAndLoss | 绘制模型训练的正确率与损失图像 |
| dataEnhance | 数据增强 |
| ModelBulid | 模型部署 |
|  |  |
|  |  |
|  |  |

### 流程图

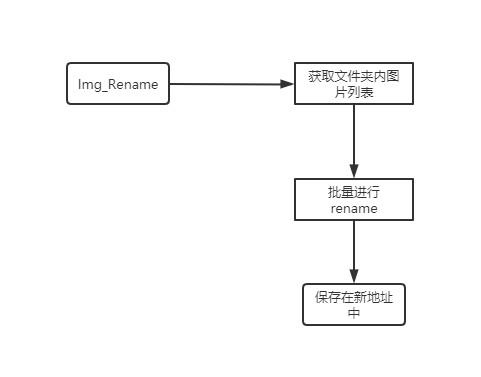
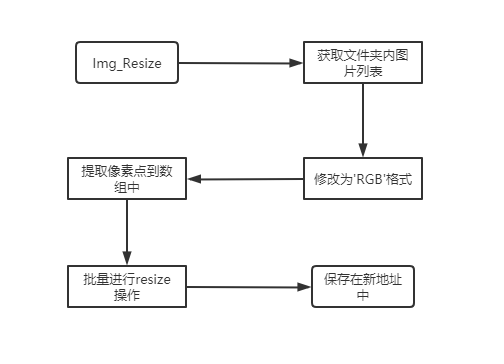


图3.1-1：图像Rename流程概要 图3.1-2：图像Resize流程概要

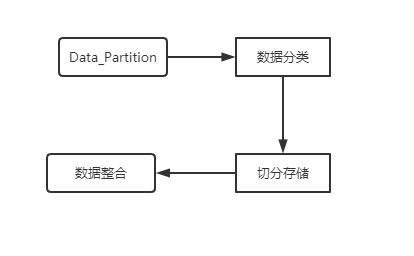
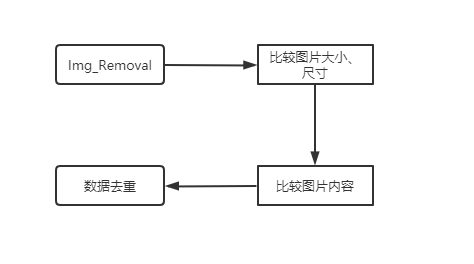
 

图3.1-3：图像数据切分流程概要 图3.1-4：图像数据去重流程概要

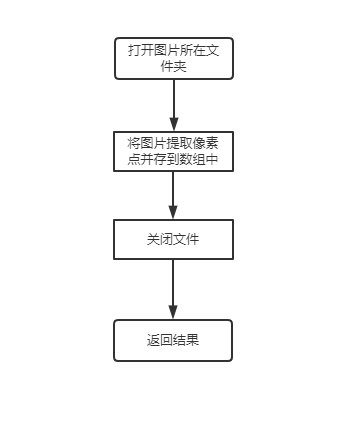
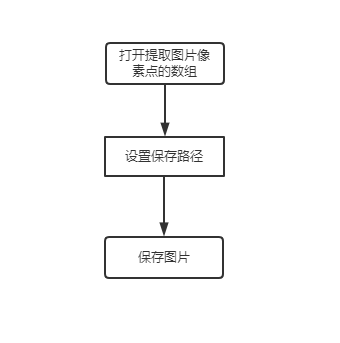


图3.1-5： 图像存储流程概要 图3.1-6：图像读取流程概要

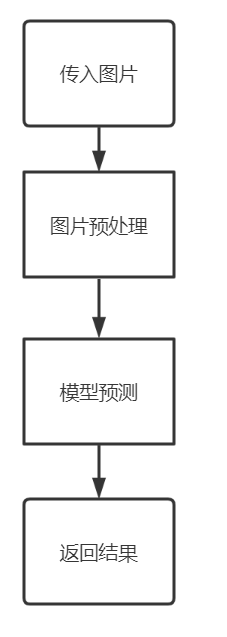
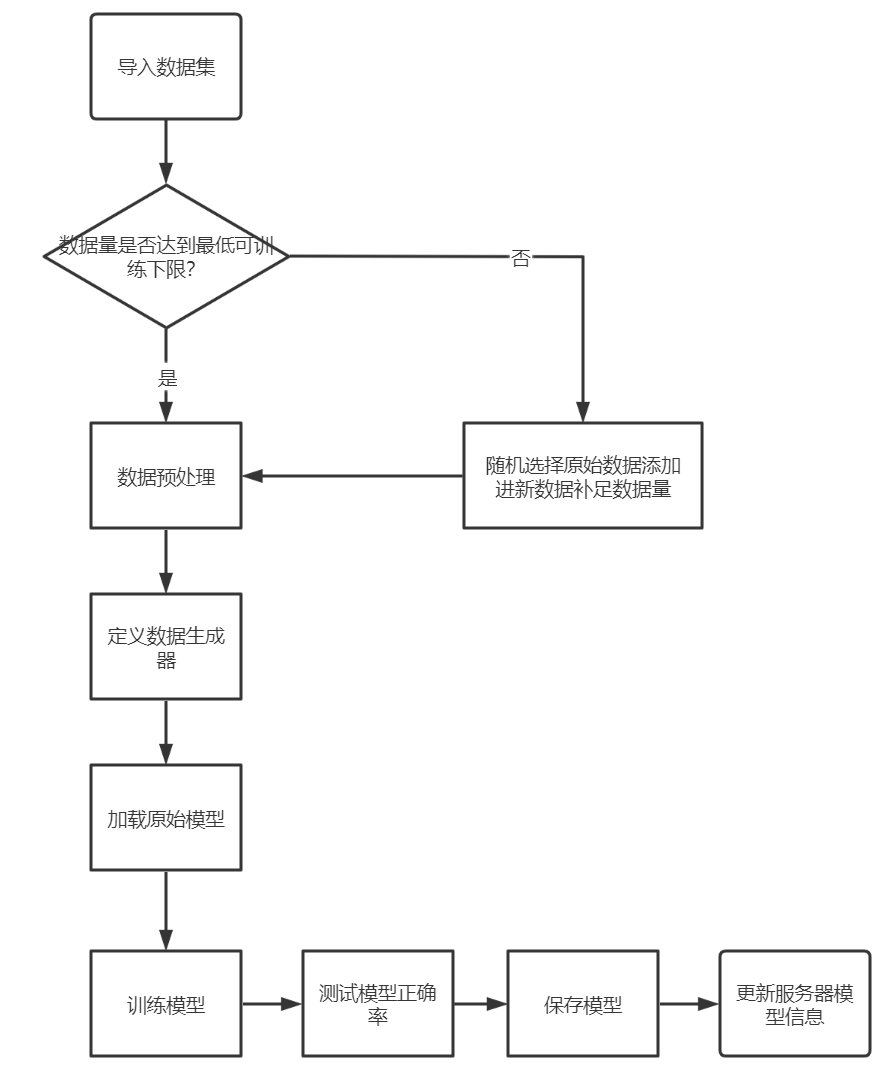
 

图3.1-7：单图像预测流程概要 图3.1-8：模型训练流程概要

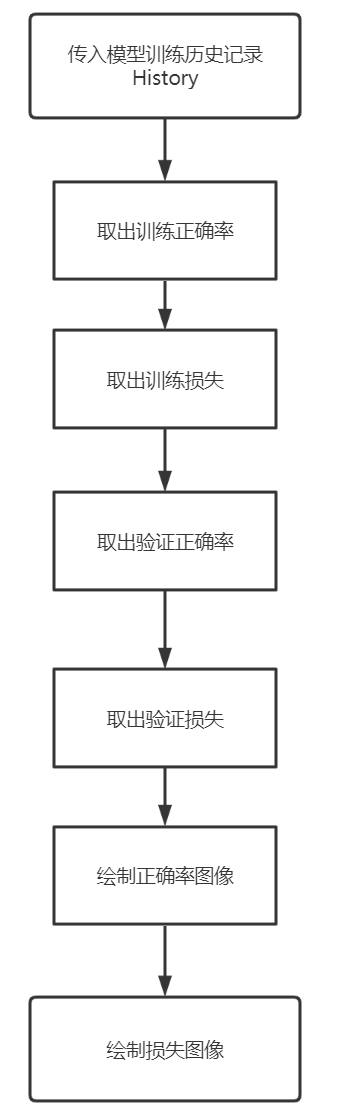
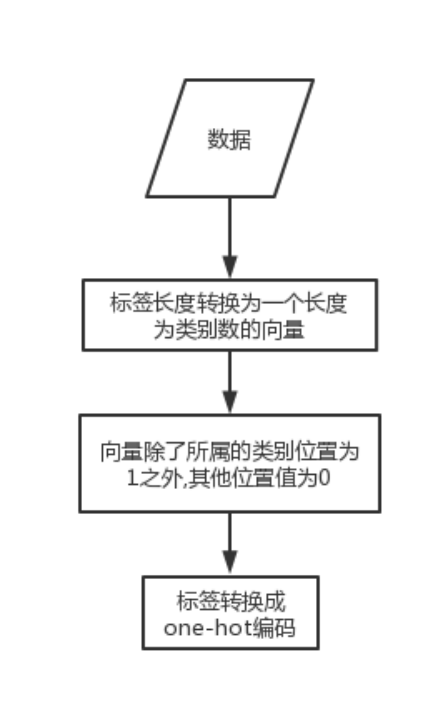
 

图3.1-9：模型正确率与损失图像流程概要 图3.1-10数据规范化将标签设定为one-hot格式

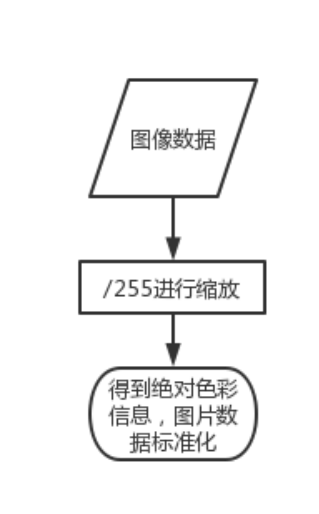


图3.1-11图像数据规范化

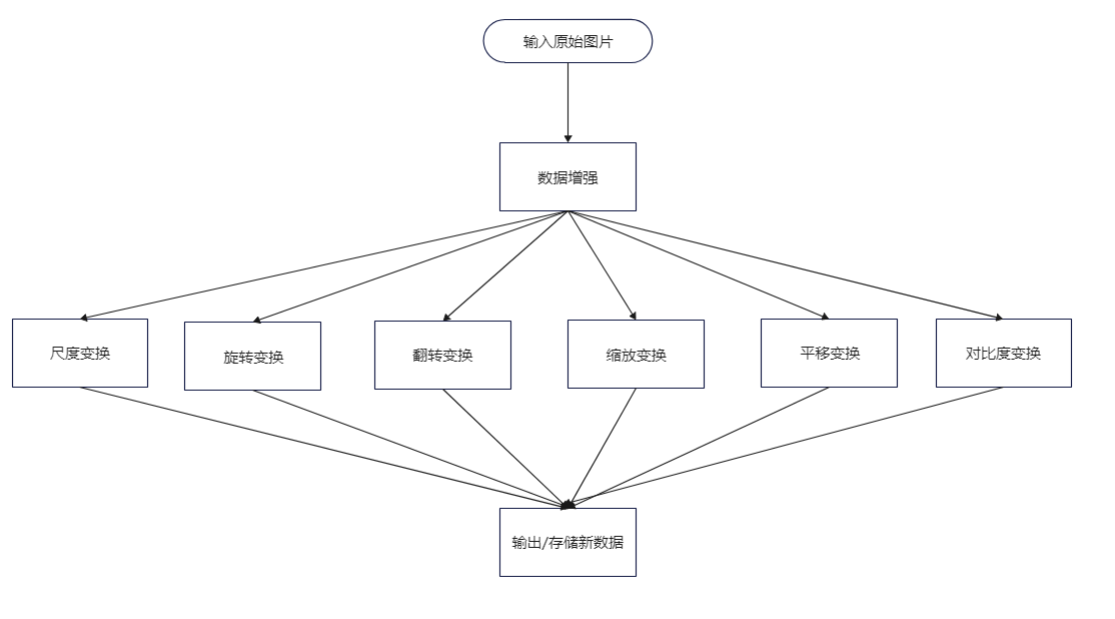


图3.1-12数据增强

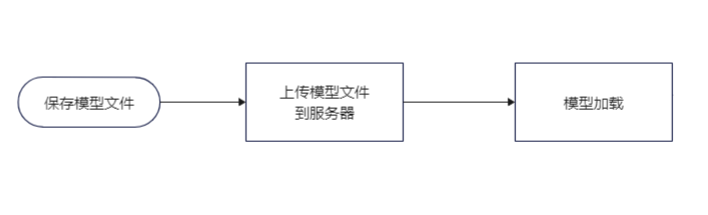


图3.1-13模型部署

## 数据爬取及数据处理（负责人：崔爽）

### 功能说明：

输入需要爬取的食物的关键字，在指定网站上进行图片爬取，以提供模型训练所需的原始数据。我们选用beautifulsoup库作为主要的爬取方法，因为这种方法用起来比较简便，同时又能很好的节省编程时间，最重要的是它可以很好的处理不规范标记并生成剖析树。定义三个函数并抽象出三个方法，GetImage负责从原图的url中将原图保存到本地；findImgUrlFromHtml负责从缩略图列表页中找到原图的url，并返回这一页的图片数量；getStartHtml负责获取缩略图列表页。数据处理即对图片进行resize、rename、removal等操作以使最终的数据可以被准确的提交到负责模型训练的同学手中，这部分已经写在公共代码中。

### 流程图：

首先使用伪头部代理模拟浏览器请求访问网站的url，获取到网站的html页面之后利用方法寻找到高清大图所在的标签，在标签中找到原图的url，然后再将图片保存到本地。爬取完所有种类图片之后进行数据清洗工作，对图片进行重命名、格式化、去重等操作

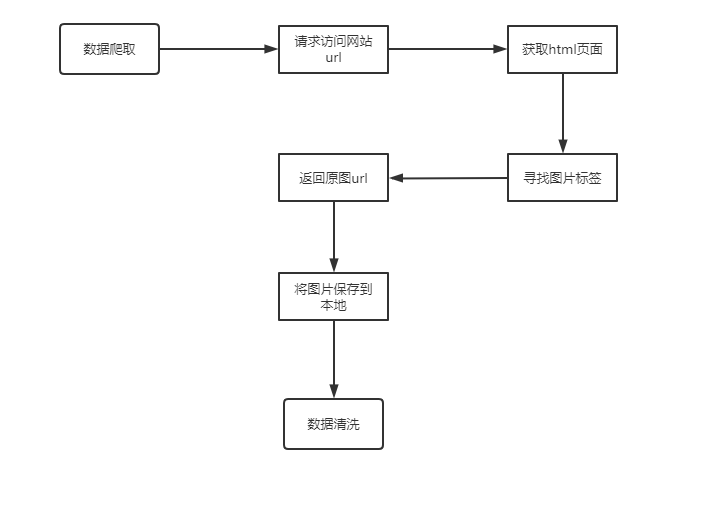


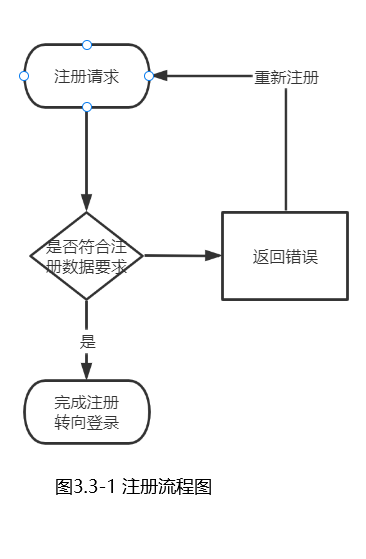
图3.2-1：数据爬取流程概要

## Python web Flask后端（负责人：郭龙强、王蕊）

使用Flask框架进行小型项目的部署开发，Flask是使用 Python 编写的 Web 微框架。Web 框架可以让我们不用关心底层的请求响应处理，更方便高效地编写 Web 程序。因为 Flask 核心简单且易于扩展，所以被称作微框架 (micro framework) 。Flask有两个主要依赖，一个是 WSGI (Web Server Gateway Interface, Web 服务器网关接口）工具集 Werkzeug (http: //werkzeug.pocoo.org/），另一个是 Jinja2 模板引擎（http: //jinja pocoo.org） 。Flask只保留了 Web 开发的核心功能，其他的功能都由外部扩展来实现，比如数据库、表单认证、文件上传等 如果没有合适的扩展，你甚至可以自己动手开发。Flask 不会替你做决定，也不会限制你的选择。总之，Flask 可以变成任何你想要的东西，一切都由你做主。

### 注册

1. 功能说明：为用户提供注册功能，对前端传回的数据进行判断处理返回结果。使用Flask\_wtf表单扩展为注册提供支持，通过表单验证数据是否符合要求进行相应处理。通过用户名注册登录之后使用系统功能，防止恶意访问使用导致系统负载过高，以免浪费服务器性能。
2. 流程图：

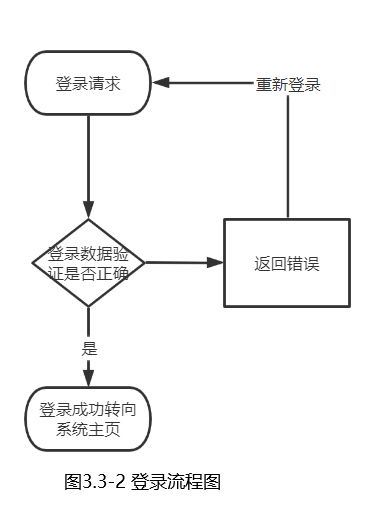
**

1. 相关数据表

|  |  |  |  |
| --- | --- | --- | --- |
| *字段名* | *字段类型* | *字段限制* | *字段说明* |
| *name* | *str* | *lenth（8-16）* | *用户名* |
| *password* | str | ^\w{6,20}$ | *用户密码* |
| confirm | str |  | *确认密码* |
| phonenum | str | ^1[35789]\d{9}$ | *手机号* |
| email | str | e-mail | *邮箱* |
| submit | str |  |  |

### 登录

1. 功能说明：为用户提供登录功能，对前端传回数据进行判断比对并返回结果。登录之后进入系统主界面可以使用图像识别功能。
2. 流程图：

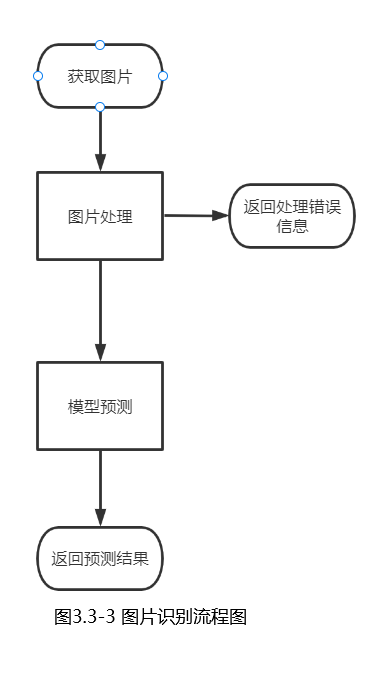
**

1. 相关数据表

|  |  |  |  |
| --- | --- | --- | --- |
| *字段名* | *字段类型* | *字段限制* | *字段说明* |
| *name* | *str* | *lenth（8-16）* | *用户名* |
| *password* | str | ^\w{6,20}$ | *用户密码* |
| submit | str |  | *提交* |

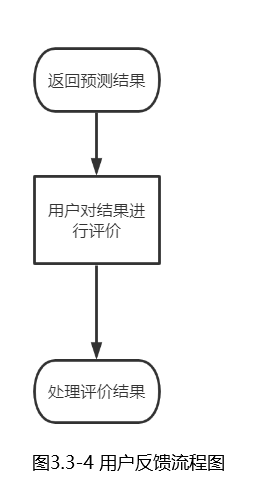
### 图片识别

1. 功能说明：为用户提供美食图片识别功能，对前端上传文件进行判断，预防攻击手段，将图片进行处理后交给模型进行识别，将识别结果返回。
2. 流程图：

**

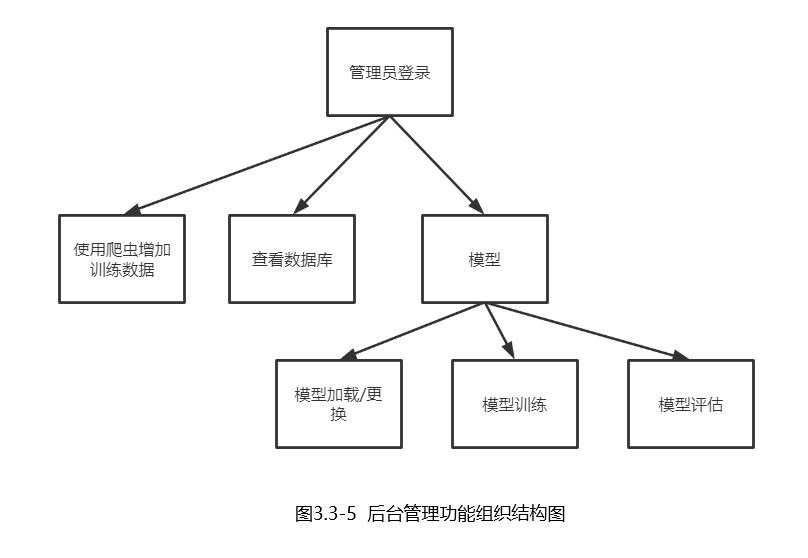
### 用户反馈

1. 功能说明：用户对系统识别结果进行评价，从用户反馈中获得模型改进数据。
2. 流程图：

**

### 后台管理

1. 功能说明：系统管理员对模型部署，数据库等进行维护。
2. 组织结构图：

**

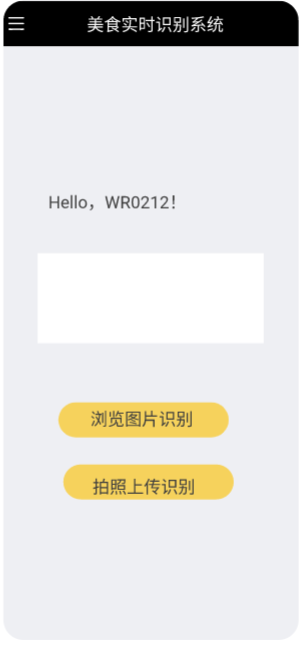
## Flask-Bootstrap前端（负责人：王蕊）

### 3.4.1功能说明

后端采用Flask框架进行开发，因为要创建主要在手机端运行的下拉式响应网页，所以界面方面采用Flask-Bootstrap。Bootstrap 是用于开发响应迅速、移动优先网站的受欢迎的集HTML、CSS和JS于一体的框架，适用于我们的美食实时识别系统。要想在程序中集成Bootstrap，需要对模板做所有必要的改动，我们使用简单的方法，使用一个名为Flask-Bootstrap的Flask扩展，简化了集成的过程。设计开始页、登录页面、注册页面、识别页面，为用户提供交互。

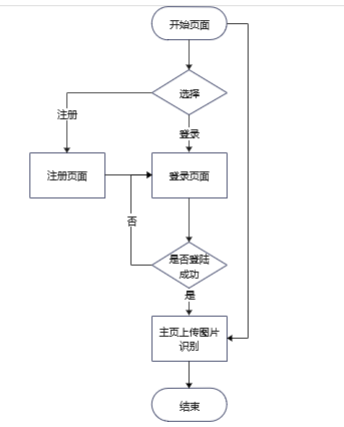
### 3.4.2原型图设计（图片为绘制）



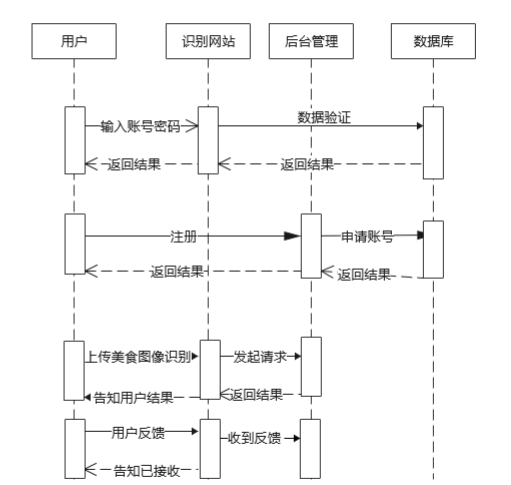


### 3.4.3流程图

用户首先进入开始页，已有账号可以选择登录，如果用户密码准确，进入识别页面，上传图片进行识别，识别后可以进行用户反馈，如果没有账号，可以进入注册页面注册，如果用户名或密码不符合规范会出现提醒，或者直接以游客身份进入美食识别页，上传图片进行识别。



### 3.4.4时序图



### 3.4.5相关数据表

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 字段说明 |
| name | String | 用户名 |
| password | String | 密码 |
| confirm | String | 确认密码 |
| phonenum | String | 手机号 |
| email | String | 邮箱 |

## 深度学习训练模块（负责人：李纪元、武泳桦）

### 功能说明

该模块对爬取到的图片进行标准化等预处理，使用数据生成器容纳并标记训练集、验证集、测试集，使用预训练模型和CNN神经网络模型对爬取到的食物图片数据集进行训练、验证、测试，最终得到一个可以对食物图片进行识别和分类的模型。

### 流程图

模型的定义与训练是一个线性的过程。

根据我们的估计，我们所收集到的数据集可能不会很大，所以我们考虑使用数据增强技术来生成各类增强后的图片，以此提高正确率与降低过拟合，因此，我们的工作从定义数据生成器开始。在这一步，我们同时会从文件读入数据到数据生成器中。

在模型方面，我们主要需要敲定模型的结构与参数。我们考虑使用善于处理图像的CNN网络，并且辅以预训练模型进行迁移学习。我们使用线性的模型结构，而对于其他的超参数，比如使用哪个预训练模型、具体每层的结构、优化器和损失函数的选择等等，则需要具体实现后视效果决定。

定义好数据生成器与模型后，我们就可以进行模型训练与测试；如果模型在测试集上的表现可以接受，我们会将模型保存为h5文件，以供后端加载使用。

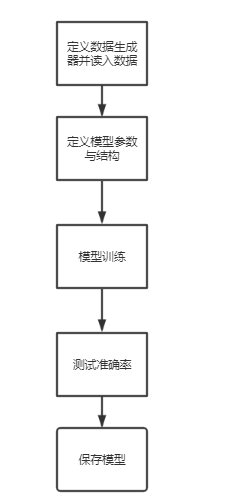


图3.5.2 模型训练流程图

### 时序图

在模型训练的过程中，主要进行交互的对象有四类：

用户：负责定义模型、数据生成器，导入并处理数据集，在模型训练前的过程不断与三者交互，调节结构、参数等等。

模型：神经网络的主要部分，负责从数据生成器中读入数据、抽取特征、更新权重。

数据生成器：数据处理的主要工具，负责读入数据并进行归一化、数据增强等工作。

数据集：模型训练的基础，由负责爬取数据的同学收集、清洗、切分。

大体概括模型训练的时序过程，用户首先定义数据生成器的各类参数，令数据生成器读入图像并准备进行处理；然后，用户定义训练模型，规定模型的输入规模、结构等参数与超参数；然后，模型不断从数据生成器中取出数据进行训练与验证，直到训练周期达到上限；最后用户使用测试集对模型的正确率进行评估。这就是模型训练的全过程。

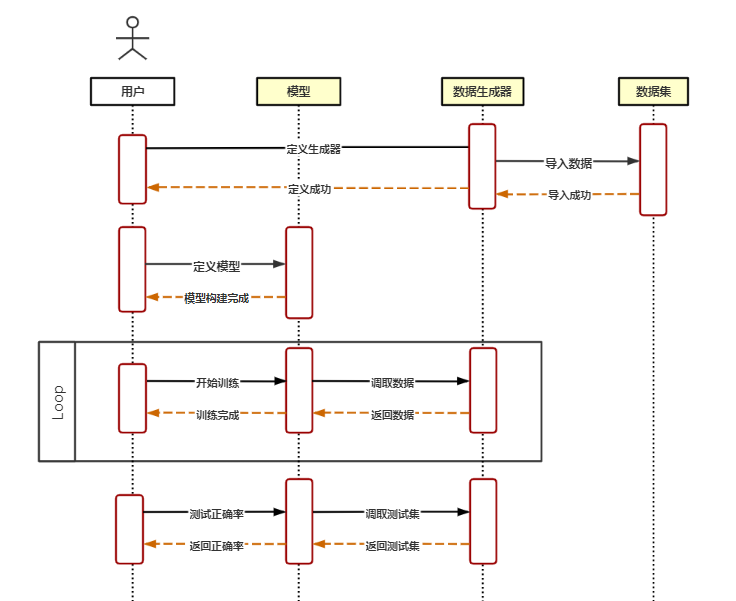


图3.5.3 模型训练时序图

### 相关数据表

|  |  |  |
| --- | --- | --- |
| 数据名 | 类型 | 说明 |
| train\_generator | ImageDataGenerator | 训练集数据生成器，负责数据增强与训练数据规范化 |
| validation\_generator | ImageDataGenerator | 验证集数据生成器，负责生成验证数据与验证数据规范化 |
| test\_generator | ImageDataGenerator | 测试集数据生成器，负责生成测试数据与测试数据规范化 |
| conv\_base | VGG16 | VGG16的卷积基 |
| model | models | CNN神经网络模型 |
| history | 字典 | 保存训练过程的数据 |

### 设计说明

在本项目中，我们选择了VGG16预训练模型与CNN神经网络来对我们的食物数据集进行训练。

VGG16是一个基于Imagenet的预训练模型，包含了大量的图像特征，此外，目前VGG16的开发与使用已经十分成熟，有大量的案例可供借鉴学习，因此，我们暂时选择VGG16作为我们的卷积基。InceptionV3和ResNet50作为后续尝试。

CNN神经网络对于图像数据有着十分强大的处理能力。我们没有使用传统的机器学习方法，因为，仅凭单一层次的机器学习，很难学习到图像的全部特征而做出正确的预测；我们也没有使用RNN等网络，因为我们的数据并不涉及时序信息，图像之间的预测可以认为在时间上是独立的。

我们使用Keras作为模型的训练框架，原因在于，Keras框架十分轻量且友好，对于新手来说易于上手；此外，Keras内部封装了大量关于数据处理的包与各式各样的模型，这允许开发者专注于模型的结构，而不是模型的内部实现。

# 环境与可持续发展 （负责人：王蕊、武泳桦）

2021年2月02日，《国务院关于加快建立健全绿色低碳循环发展经济体系的指导意见》，意见指出：要深入贯彻党的十九大和十九届二中、三中、四中、五中全会精神，全面贯彻生态文明思想，认真落实党中央、国务院决策部署，坚定不移贯彻新发展理念，全方位全过程推行绿色规划、绿色设计、绿色投资、绿色建设、绿色生产、绿色流通、绿色生活、绿色消费，使发展建立在高效利用资源、严格保护生态环境、有效控制温室气体排放的基础上，统筹推进高质量发展和高水平保护，建立健全绿色低碳循环发展的经济体系，确保实现碳达峰、碳中和目标，推动我国绿色发展迈上新台阶。

# 在人工智能领域，人们过度依赖大模型、大数据去解决精度问题，尤其是在深度学习领域。深度学习为人工智能的发展带来了诸多好处的同时，也带来诸多问题，其中一点是耗电量巨大。许多大模型训练时很多都需要数十小时，更有甚者以天计数。

# 在过去，人工智能常被比作石油行业：一旦对（数据/石油）进行开采、精炼，就能变为高利润的商品。二者的相似之处不止于此，深度学习在能量消耗方面也与石油行业一样骇人。马萨诸塞大学阿默斯特校区的研究人员公布的论文《Energy and Policy Considerations for Deep Learning in NLP》以常见的几种大型 AI 模型的训练周期为例研究，发现训练一个 AI 模型可排放超过 626,000 磅二氧化碳，这相当于五辆汽车一生排放的碳总量。训练的计算环境成本与模型大小成正比，在使用附加的调整步骤以提高模型的最终精度时成本更是呈爆炸式增长，随着 AI 算力的提升，资源消耗会更加严重。 尽管目前使用服务器集群可以加快速度，但耗电量和机器散发的热量也是不可忽视。

本次实训中，我们的项目美食实时识别系统从设计到实现都非常注重保护环境与可持续发展。环境与可持续发展是一个非常重要的主题，它与人类的生存和发展息息相关。随着工业的发展和科技的进步，环境遭到了严重的破坏与污染，保护环境、确保社会的可持续发展至关重要。在训练模型的过程之中，因为神经网络是高度并行的，而GPU非常适合并行计算，所以我们小组成员都采用GPU训练，GPU的计算速度比CPU快得多，减少电脑散热量，节省了耗电量。同时，如果每个大模型都要重新训练，那将造成资源的许多浪费。于是我们采用了迁移学习方法，利用他人已经训练好的模型，通过微调来训练出一个适合我们问题领域的模型。使用迁移学习，不需要再从头训练模型，从而节省大量资源，减少耗电量、散热量，将环境与可持续发展的原则落到实处。