

文章编号: 2095-2163(2022)06-0146-09

中图分类号: U491.17

文献标志码: A

基于深度学习与百度 AI 的菜谱推荐与健康分析微信小程序设计

陈沁仪, 冲 蕾, 于泽昊, 张泽权, 王展纯, 邹星雨

(上海工程技术大学 城市轨道交通学院, 上海 201620)

摘要: 本作品贴合当下时代潮流,以深度学习为核心,建立能够实现菜谱推荐与健康分析的微信小程序。使用基于 TensorFlow 搭建的神经网络模型,上传部署至腾讯云云函数,用以对用户在前端页面的选择进行预测并排序;调用微信相关接口,实现基于百度 AI 的菜品识别功能对用户的卡路里摄入动态监控,在页面中显示今日摄入的卡路里总和;使用 WeUI 相关组件显示近一周卡路里摄入变化曲线,更加简洁。综上,本作品能够对用户的健康起到一定的监督作用。

关键词: 微信小程序; 云函数; 深度学习; TensorFlow

Miniprogram design of recipe recommendation and health analysis based on deep learning and Baidu AI

CHEN Qinyi, CHONG Lei, YU Zehao, ZHANG Zequan, WANG Zhanchun, ZOU Xingyu

(School of Urban Railway Transportation, Shanghai University of Engineering Science, Shanghai 201620, China)

[Abstract] This work is in line with the current trend of the times, with deep learning as the core, to establish a WeChat applet that can realize recipe recommendation and health analysis. This paper uses the neural network model based on TensorFlow, which is uploaded and deployed to cloud function of Tencent Cloud, in order to predict and sort the user's selection on the front-end page. Meanwhile, the paper calls WeChat related interfaces to realize the dynamic monitoring of user's calorie intake based on Baidu AI's dish recognition function, and displays the total calorie intake today on the page. Therefore, WeUI related components are used to display the change curve of calorie intake in the past week. The display effect is more concise. The conclusion of the paper is the work can play a certain role in monitoring the health of users.

[Key words] miniprogram; cloud function; deep learning; TensorFlow

0 引言

受到 2020 年新冠疫情的影响,大部分校园、企业食堂采用了无人化、自动化的相关餐饮服务,智能化、数字化迎来了一个新的发展阶段。已有研究表明,食物中的一些成分可以改变血液中某些神经递质的浓度水平,传递各种各样的情绪信息。由此可见,食物,是影响神经递质浓度水平的关键,是控制情绪的重要因素,直接影响了人的感情中枢^[1]。因此对菜谱推荐和健康分析进行研究的迫切性日益凸显。本文拟对此展开探述与论述。

1 深度学习

1.1 深度学习原理

深度学习泛指一类用复杂多重非线性结构分析数据的抽象算法,现在多用深度学习为基础研发的深度神经网络作为工具的机器学习模型。在本作品中,主要体现在全连接的人工神经网络上,示意图如

图 1 所示。

神经网络(NN)是以模仿大脑神经结构为出发点的计算模型。大多数的神经网络可以被用来解决多维特征/多元变量问题。根据已有结论,具有 2 个隐藏层的神经网络可以拟合任意函数。因此,传统的神经网络采用 2 个隐藏层。在数学上,可以被表示成一种函数模型: $(y_1, y_2, \dots, y_m) = F(x_1, x_2, \dots, x_n)$ 。其中,输入 n 与输出 m 可以是任意长度的向量,同时可以将函数 f 看作是黑盒,不需要了解其内部逻辑。

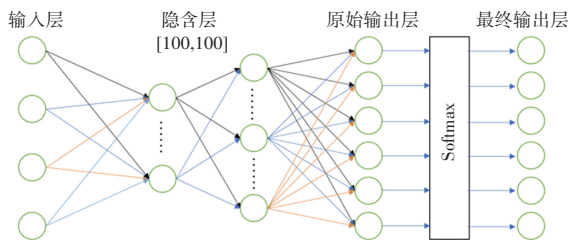


图 1 神经网络示意图

Fig. 1 Schematic diagram of neural network

人工神经网络拥有自组织、自学习、自适应、强

作者简介: 陈沁仪(2000-),女,本科生,主要研究方向:微信小程序开发;冲 蕾(1982-),女,硕士,讲师,主要研究方向:城市轨道交通信号自动控制。

收稿日期: 2021-10-04

非线性函数逼近能力以及强容错能力。目前,包括图像识别、预测和模糊控制研究这些新兴领域,人工神经网络都有涉足。人工神经网络通过对已知信息的反复学习和训练,同时逐渐调整和更改神经元连接的权重,以此完成信息处理和模拟输入输出关系的目标。网络的每一层都包含多个神经元,用变权

的有向弧来描述。

1.2 搭建模型

本作品设计的基于 TensorFlow 的神经网络算法流程如图 2 所示,主要分为 3 个部分,即:定义模型、训练与保存模型、加载模型并预测,分别设置在 2 个可执行的 Python 文件中。

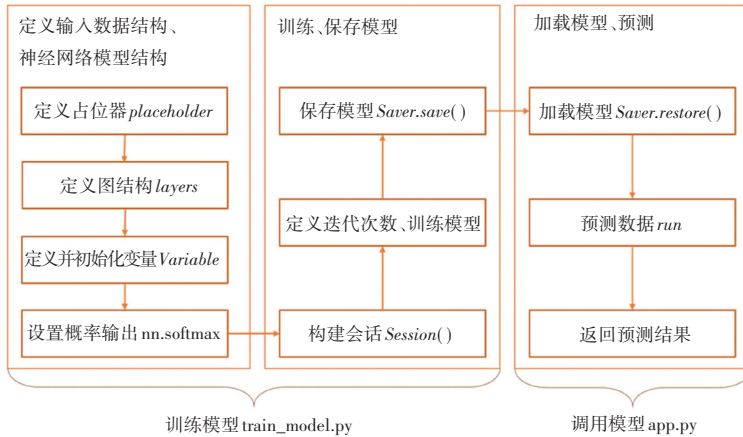


图 2 神经网络算法设计流程

Fig. 2 Design process of neural network algorithm

定义模型的部分主要工作有:

- (1) 定义占位器 *placeholder* 用以规定输入数据的结构。
- (2) 定义图结构 *layers* 用以规定神经网络模型的结构,如连接方式、神经元个数、激活函数等,在本作品中含 2 层隐藏层,各 100 个神经元,采用 *sigmoid* 函数为激活函数。
- (3) 定义变量 *Variable*, 如权重、偏移率等指标用以检测与分析神经网络模型的合理性。
- (4) 定义损失函数、优化函数等用以约束并优化构建的神经网络模型。

训练、保存模型的部分主要工作有:

- (1) 实例化 *saver* 对象 *train.Saver()*,用以保存模型。
- (2) 构建并打开会话 *Session()*。

- (3) 模型内所有参数初始化 *global_variables_initializer()*,如模型优化函数、损失函数等参数。本作品采用梯度下降法进行优化,通过计算可知,设置学习率为 0.15 时较优。

- (4) 定义迭代次数并训练模型 *sess.run(train, feed_dict = {X: x_data, y_ture: y_data})*,*sess.run(loss, feed_dict = {X: x_data, y_ture: y_data})*,本作品设置迭代次数 3 000 次,最终的模型损失为 2.04,可以收敛。

- (5) 保存模型 *saver.save (sess, "./src/tmp/model.ckpt"*),会在指定目录下生成一系列“.ckpt”文件用以恢复模型。

在此基础上,研究进一步给出了第三方库调用说明如图 3 所示。

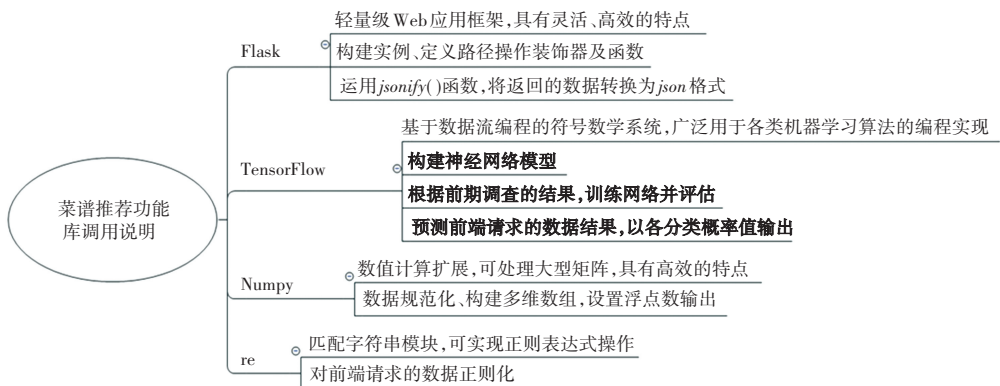


图 3 第三方库调用说明

Fig. 3 Third-party library call instructions

1.3 在腾讯云云函数的部署

在腾讯云平台上部署后端代码,定义执行入口文件 `sl.handler.py` 执行 `app.py` 的内容 `severless_wsgi.handle_request(app.app, event, context)`。`app.py` 的内容包括执行服务器请求部分与加载模型并预测部分。

执行服务器请求部分的主要内容有:

- (1) 构建 Flask 实例 `app = Flask(__name__)`。
- (2) 定义路径操作修饰器 `@app.route()`, 这里的请求方式为 ANY, 即接收任意请求方式。
- (3) 定义路径操作函数 `def index()`。
- (4) 将获取的待预测参数正则化并转换成 list 形式, 这里只需提取其中的有效数字部分, 即 `data = list(re.findall(r"\d + \.?\d *", user_selection))`, 并转换为整形 `list(map(int, data))`。
- (5) 运行加载模型并预测部分的程序, 得到预测结果。

(6) 运用 `jsonify()` 函数, 将得到的结果 JSON 化作为返回值。

加载模型并预测部分的主要内容有:

- (1) 清除图的结构 `reset_default_graph()`。

(2) 定义占位器 `placeholder` 用以规定输入数据的格式。

(3) 构建并打开会话 `Session()`。

(4) 创建 saver 实例 `tf.train.Saver()` 并恢复模型 `saver.restore(sess, "./tmp/model.ckpt")`。

(5) 预测数据 `sess.run(hypothesis, feed_dict = {X: selection})`。

(6) 将预测结果与所属类别匹配, 构成字典格式。

(7) 将该字典按照概率值降序排列 `sorted(list1, key = lambda e:e.getitem_('probability'), reverse = True)`, 并返回。

设置该云函数的触发方式为 API 网关触发, 采用 API 网关触发的优势在于可以选择同步或异步调用, 且可以通过定义不同的路径操作修饰器和函数用以实现多个功能, 最大化利用一个函数解决多个功能。在小程序端仅需要通过 (`wx.request()`) 进行 HTTPS 访问即可, 流程如图 4 所示, (`wx.request()`) 方式的参数设置见表 1。预测成功后的页面显示如图 5 所示。

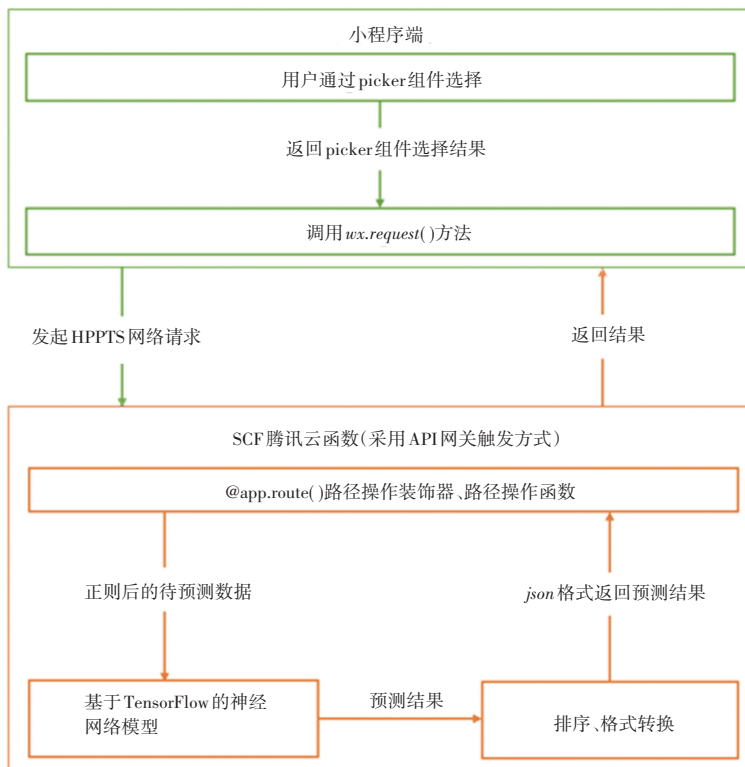


图 4 菜谱推荐的前后端交互流程示意图

Fig. 4 Schematic diagram of the front-end and back-end interaction process of recipe recommendation

表 1 wx.request() 参数说明

Tab. 1 wx.request() parameters description

属性	类型	说明
url	string	开发者服务器接口地址
data	string/object/ArrayBuffer	请求的参数
header	Object	设置请求的 header content-type 默认为 application/json
method	string	HTTP 请求办法
dataType	string	返回的数据格式
success	function	接口调用成功的回调函数
fail	function	接口调用失败的回调函数



图 5 预测成功后的页面显示

Fig. 5 The page after the successful prediction

2 基于百度 AI 的菜品识别功能

2.1 百度 AI

百度 AI 作为国内最大的人工智能平台之一, 提供了很多方便快捷可被调用的 API 接口和具体应用, 目前百度 AI 的语音、图像、文字、客流量等识别方向都取得了相当显著的效果, 在识别精度和速度上达到了较高的状态^[2]。

2.2 实现菜品识别并录入

通过调用百度 AI 的智能图像分析功能, 进而实现

菜品识别, 识别速度快、准确率较高。具体过程如下:

(1) 调用微信提供的 (*wx.chooseImage()*) 接口, 引导用户选择拍照或相册上传图片, 返回的结果为用户上传图片的临时路径, 即不会再次占用用户的内存。

(2) 使用 (*wx.getFileSystemManager().readFile()*) 获取上传图片的临时路径, 并使用 *base64* 格式转码。

(3) 使用在百度 AI 官网注册申请到的 *client_id* 和 *client_secret*, 调用 (*wx.request()*) 接口以 *HTTPS* 协议形式, 以 *GET* 请求方式, 请求百度 AI 提供的 *api* 接口, 由此可获得到可被调用的唯一 *access_token*。

(4) 调用 (*wx.showLoading()*) 加载框, 提示用户正在进行识别。

(5) 再次调用 *wx.request()* 接口, 以 *POST* 的请求方式, 请求百度 AI 提供的 *API* 接口, 在 *URL* 中添加此前获取的 *access_token*, *DATA* 为待识别图片的 *base64* 编码, 调用成功后可获得菜品识别结果、卡路里值及置信度, 并调用 (*wx.hideLoading()*) 接口关闭加载提示。

基于百度 AI 的菜品识别及卡路里录入功能示意如图 6 所示。研究中, 在页面显示菜品识别的相关结果后, 设置录入按钮, 用户可根据真实的菜品摄入情况选择录入。调用云函数 (*wx.cloud.callFunction()*), 通过 *db.collection().add()* 在数据库的对应集合中添加用户 *openid*、菜品名称、数据来源、卡路里值、创建时间等信息, 并在个人中心页面显示。菜品识别成功后的界面显示、录入后的历史记录可见图 7。

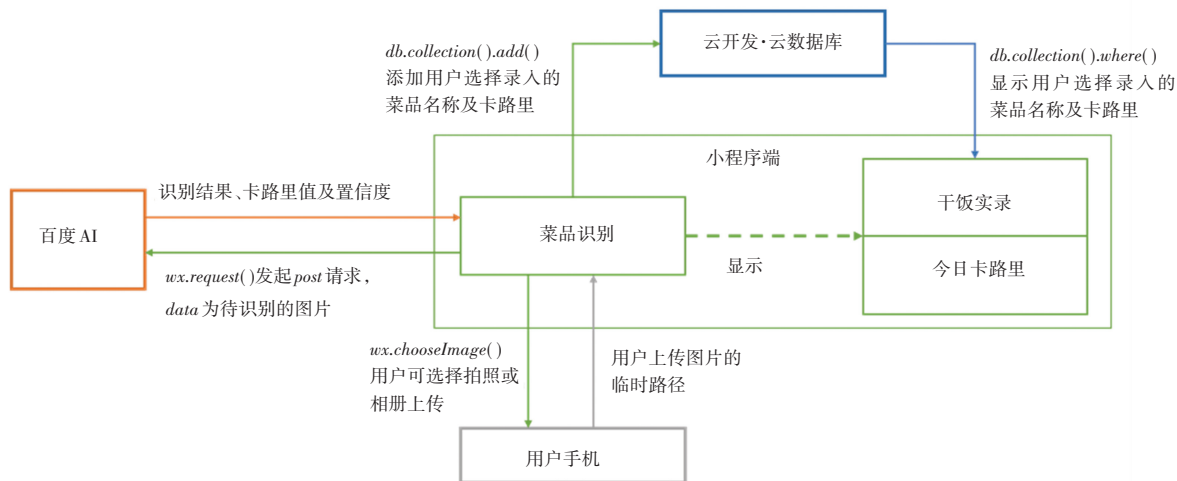


图 6 基于百度 AI 的菜品识别及卡路里录入功能示意图

Fig. 6 Schematic diagram of dish recognition and calorie entry function based on Baidu AI

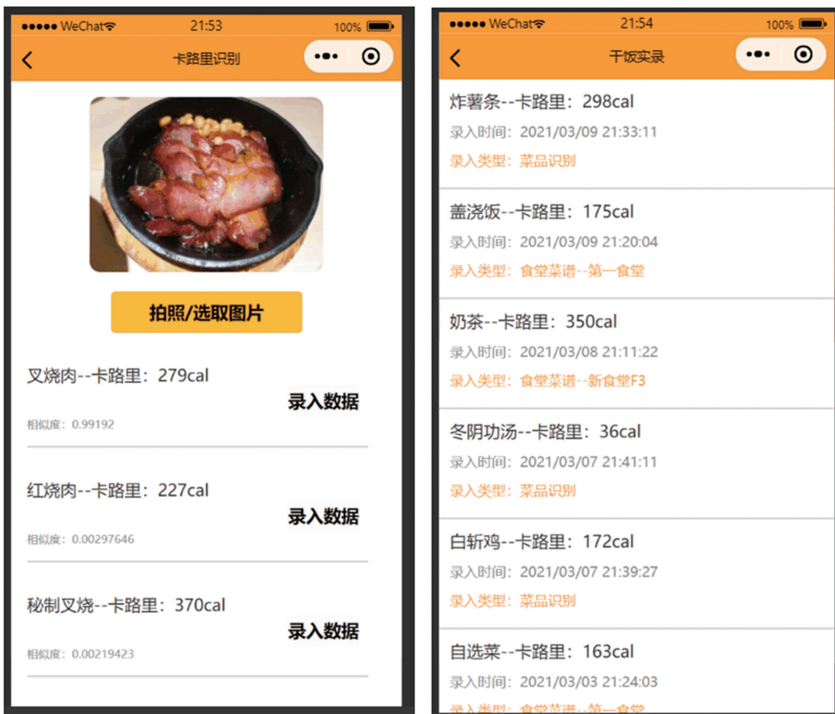


图 7 菜品识别成功后的界面显示、录入后的历史记录

Fig. 7 The interface display after the dish recognized and the history record after entry

3 微信小程序

3.1 微信小程序概述

微信小程序是一种全新的连接用户与服务的方式,可以在微信内被便捷地获取和传播,具有出色的使用体验。故本文采用微信小程序作为前端,有较高的灵活性与兼容性。可以最大限度调用微信提供的开放接口,提高开发效率,缩短代码运行时间。

微信小程序基于当前的 Web 技术,通过独立的

运行环境感知平台,提供接近原生应用的用户体验。开发方法类似于 WebAPP 的开发方法。微信小程序框架系统称为 MINA 应用框架,是一个双线程模型,分为逻辑层(App Service)和展示层(View)两部分。小程序的显示层描述语言包括 WXML 和 WXSS,这是一个基于 JavaScript 的逻辑层框架,提供了一个在显示层和逻辑层之间传输数据和事件的系统。框架如图 8 所示。框架的核心是一个响应式数据链接系统,可以保持和显示数据同步^[3]。

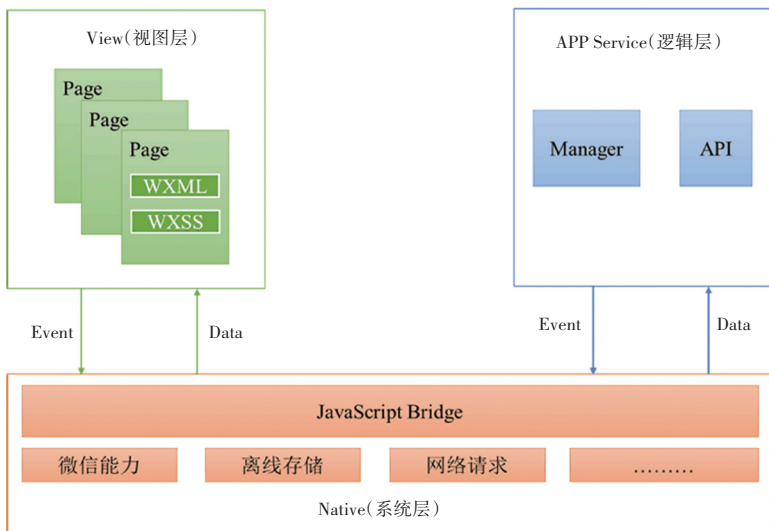


图 8 微信小程序框架系统概览

Fig. 8 Overview of WeChat miniprogram framework system

在小程序启动时,微信会在后台完成以下工作: 下载小程序代码包、加载小程序代码包、初始化小程序首页,如图 9 所示。同时,在小程序启动或一个新

的页面被打开时,页面的初始数据(*data*)和路径等相关信息会从逻辑层发送给视图层,用于视图层的初始渲染^[3],详见图 9。

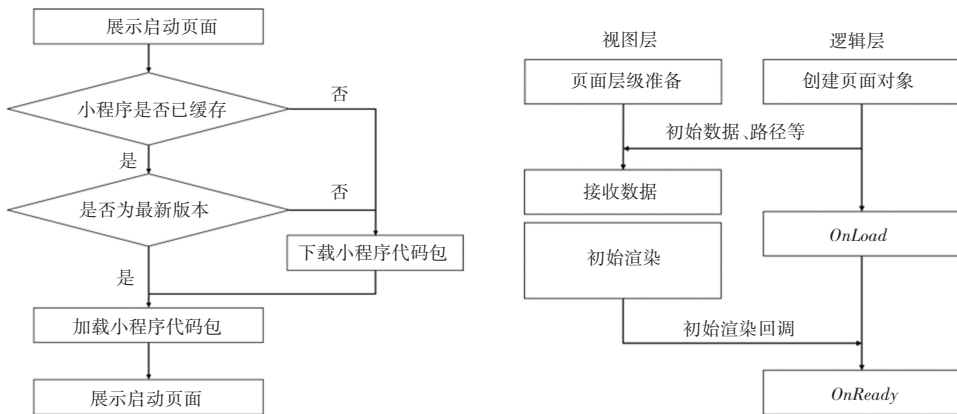


图 9 微信小程序启动流程图、初始通信流程图

Fig. 9 WeChat applet startup flowchart and initial communication flowchart

3.2 小程序·云开发

本设计采用了小程序·云开发中云数据库与云函数功能,小程序·云开发是微信团队联合腾讯云推出的专业的小程序开发服务,其优势在于无需搭建服务器,免登录、免鉴权调用微信开放服务、不限开发语言与框架等,本设计采用 JavaScript 作为小程序·云函数的主要开发语言。

云数据库中的每条记录都是一个 JSON 格式的对象,一个数据库可以有多个集合,集合可以看作一个 JSON 数组,数组中的每个对象都是一条记录,记录的格式是 JSON。常见的 MySQL 数据库为关系型数据库,但是不便于在小程序内获取和调用。关系型数据库和 JSON 数据库的概念对应关系见表 2。

表 2 关系型数据库和 JSON 数据库的概念对应关系

Tab. 2 The conceptual correspondence between relational databases and JSON databases

关系型	文档型
数据库(database)	数据库(database)
表(table)	集合(collection)
行(row)	记录(record/doc)
列(column)	字段(field)

云函数,即在云端(服务器端)运行的函数,其写法与在本地定义的 JavaScript 方法相同,代码运行在云端 Node.js 中。小程序内提供了专用于云函数调用的 API,可以在云函数中使用 wx-server-sdk 提供的 *getWXContext* 方法获取到每次调用的上下文

(*appid*, *openid* 等),无需维护复杂的鉴权机制,即可获得天然可信任的用户登录态(*openid*)。

3.3 小程序·个人中心

本设计的个人中心作为丰富用户体验的一大重点模块,为满足用户的各种应用需求,设计了如下功能模块。

(1)“干饭实录”功能。该功能主要通过云开发·云数据库实现。首先获取已被设为全局变量的用户 *openid*,再通过 *db.collection.where().get()* 在云数据库的对应集合中获取对应用户 *openid* 所录入的菜品信息;通过 *orderby('createTime', 'desc')* 对菜品按录入时间倒序排列以便于用户查看最新的菜品录入信息。同时,也运用了 *onload()* 生命周期函数以及 *onPulldownFresh()* 函数以使用户实时获取最新录入的菜品信息,并能够实现下拉刷新。在“个人中心”界面通过调取 *util.js* 获取今日日期,再通过 *db.collection.where().get().count()* 获取今日录入的菜品个数,从而完成在个人中心“干饭实录”的前端页面反馈。

(2) *BMI* 计算功能。本作品的设计旨在帮助用户对自己的身体健康状况有实时的了解。计算时通过获取用户输入的身高、体重的值,在本文的研究中设置了 *calculate()*、*weight StandardCalculate()* 以及 *physical Conditional Calculate()* 分别进行 *BMI*、标准体重以及身体状况的计算。与此同时,本页面还为用户提供了标准 *BMI* 对照表,以此让用户对 *BMI*

所反映的自身的身体状况有了一个全面的了解。

(3) 意见反馈功能。本作品注重用户体验,重视收集用户的使用意见,故将用户反馈分为两大板块。第一大板块设立在菜单搜索界面,用户可在此输入菜品名、对应食堂以及数量来反馈菜品缺失;第二大板块设立在个人中心界面,用户可在此反馈当使用本程序时所遇到的问题和反馈。当用户输入信息未输入完整时,调用(*wx.showToast()*)弹出框对用户进行提示。用户输入完成后可点击底部的反馈按钮,通过 *db.collection().add()* 在云数据库的对应集合中添加用户的问题种类和具体情况,后台开发人员将对集合中用户反馈的问题进行解决。

(4) 菜品收藏功能。该功能主要运用云函数·云数据库来实现。云函数的独特优势在于与微信登录鉴权的无缝整合。当小程序端调用云函数时,云函数的传入参数中会被注入小程序端用户的 *openid*,开发者无需校验 *openid* 的正确性,直接使用该 *openid*。对此拟给出剖析详述如下。

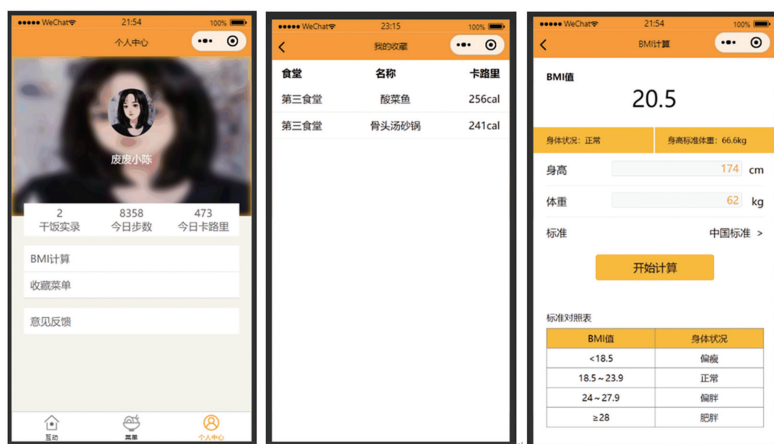
① 用户在“菜品详情”界面中点击收藏图标后,将会调用云函数判断目前云数据库的对应集合中是否有对应的菜品收藏信息。

② 若对应集合中没有相对应的收藏信息,将会使用云函数添加对应的菜品信息至云数据库中;若对应集合中有相对应的收藏信息,将会删除云数据库中相应的菜品信息。

③ 在调用云函数的过程中会使用 (*wx.showLoading()*) 加载框,提示用户正在进行加载。

④ 在判断完毕后将会更改 *icon* 图标为已收藏或未收藏,这一过程中将会调用 (*wx.showToast()*) 弹出框对用户进行提示。

个人中心界面、收藏界面、及 *BMI* 指数计算界面如图 10 所示。研究中,用户将通过云函数在“个人中心”界面的“收藏的菜单”页面中获取到自己收藏的菜品信息,同时 *onload()* 生命周期函数以及 *onPulldownFresh()* 函数帮助用户实时获取收藏信息的更新。



(a) 个人中心界面

(b) 收藏界面

(c) *BMI* 指数计算界面图 10 个人中心界面、收藏界面、及 *BMI* 指数计算界面Fig. 10 Personal center interface, collection interface and *BMI* index calculation interface

3.4 小程序·菜单界面的卡路里录入及收藏功能

本作品在前期走访了校内所有食堂,整理出各个食堂所包含的餐厅、菜品名称,运用 *scroll-view* 可滚动视图组件清晰地将食堂与菜品的所属关系展示给用户。通过后文的基于百度 AI 的菜品识别功能得到这些菜品的卡路里值,并将名称、所属餐厅、卡路里值显示在菜单详情页面供用户参考。

将云数据库读取并调用的部分程序放在生命周期函数中,以此减少用户等待时间。菜单界面的卡路里录入及收藏功能示意如图 11 所示。研究中,点击菜品名称可根据页面参数 *id*,跳转至对应的菜单

详情页面,在该页面用户可以实现菜品收藏功能和卡路里录入功能,能够在个人中心查看,菜品收藏功能和卡路里录入功能均主要通过云函数实现,详细算法将会在后文个人中心与可视化进行阐述,在进行上述操作均有明显的 (*wx.showToast()*) 提示用户操作。为照顾到用户需求,在顶部放置搜索栏,可实现模糊搜索功能,原理是使用 *db.RegExp()* 方法来构造正则对象,然后将用户输入与数据库该集合内的所有菜品名称做字符串匹配,匹配成功则显示在页面上。

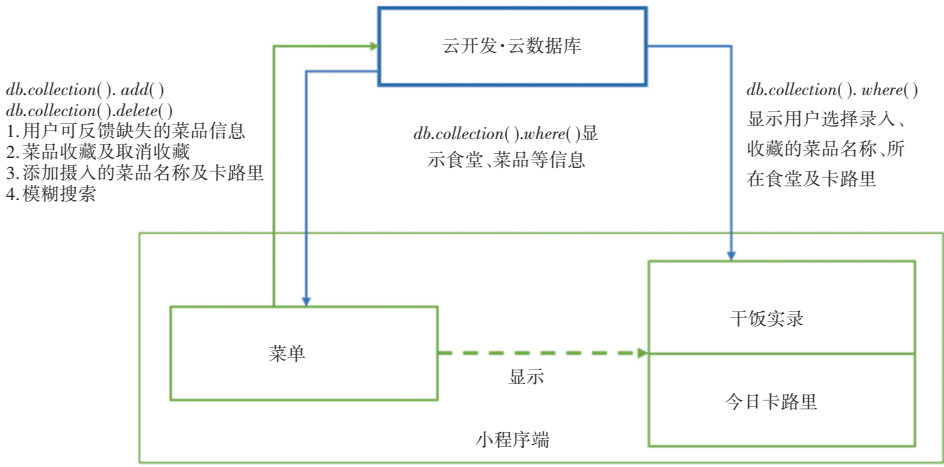


图 11 菜单界面的卡路里录入及收藏功能示意图

Fig. 11 Schematic diagram of calorie entry and collection functions in the menu interface

3.5 小程序·微信步数的获取

本作品的微信步数功能基于微信官方 API 改造实现,获取流程如图 12 所示。具体流程如下:

- (1) 调用 (*wx.request()*) 接口以 HTTPS 协议形式,以 GET 请求方式,发送 code,登录凭证校验接口,调用成功后微信接口服务会反馈 *openid*、*session_key* 等信息。
- (2) 通过 (*wx.getSetting()*) 请求用户开启微信步数权限,并通过(*wx.authorize()*)判断用户是否关注微信步数以便于获取信息。
- (3) 通过 (*wx.getWeRunData*) 请求微信官方 API 接口来获取用户近 30 天的微信步数信息,微信接口服务反馈小程序端 *encryptedData*、*iv*、*cloudID* 等信

- 息。
- (4) 通过(*wx.cloud.callFunction()*)调用云函数 *get_WeRun_data*,将获取到的信息解密后以微信步数和时间戳的形式反馈给小程序端。
- (5) 在完成微信步数的基本流程后用生命周期函数 (*wx.login()*) 获取 code 的自动更新步数信息,以此来保证微信步数信息的时效性,并服务于步数信息的可视化。
- (6) 微信步数的可视化是在 js 中将今日步数参数定义在 *array_time_step* 上,直接通过 *wxml* 中的 *text* 实现显示,并利用生命周期函数 *wx.login()* 实现数据的实时更新。

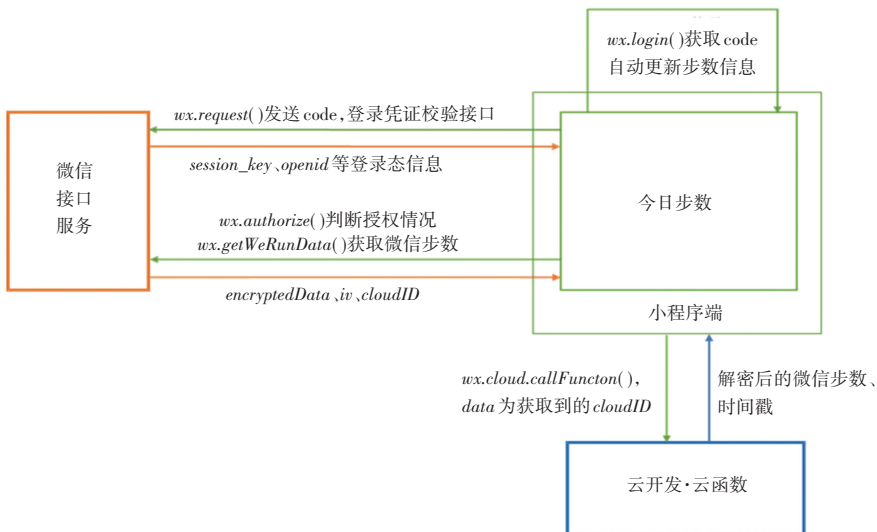


图 12 今日步数显示功能流程图

Fig. 12 Flow chart of today's steps count display function

3.6 小程序·卡路里获取与曲线显示

该功能通过云函数·云数据库·ec-canvas 组件实现。其大致流程如图 13 所示,具体步骤如下:

(1) 用户登录后,自动调用云函数判断云数据库中记录的数据日期是否与用户登录日期一致,若不一致则在云数据库的相应集合中新增一条卡路里为 0 的数据。

(2) 在“菜品识别”页面设置录入数据图标与“详细菜品”页面中设置 icon 符号。通过读取对应的菜品信息,使用 `db.collection.add()` 将菜品信息及卡路里数据添加至云数据库中。

(3) 将获取到的对应卡路里数据通过

`parseInt()` 转化为整数型、从而调取云函数对卡路里数据进行相加,然后对数据库对应集合的数据进行更新。

(4) 调用 `(wx.showToast())` 弹出框对用户进行提示。

(5) 将数据库中的数据反馈到“个人中心”界面前端显示。

(6) 在“今日卡路里”页面中调用 `ec-canvas` 组件获取折线图的模板,再通过调用云函数获取云数据库相对应集合中的数据在折线图中显示,以使用户更直观地看到自己近一周内卡路里摄取情况。

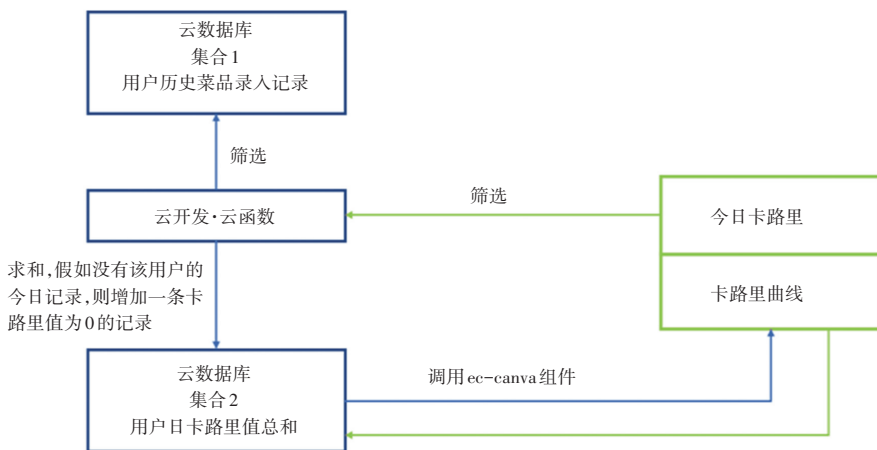


图 13 卡路里曲线获取和显示流程图

Fig. 13 Flow chart of calorie curve acquisition and display

4 结束语

本作品贴合时代潮流,以学校为试点,旨在建立一个能够监控自身良好饮食习惯的小程序,能够实现菜谱推荐与健康分析两大功能。例如通过用户选择自身当前的状态进行菜谱推荐、拍照识别菜品及卡路里、动态显示近一周卡路里摄入变化曲线连同记录在内的历史录入菜品、微信步数历史记录等功能。在开发方面,多使用云函数、云数据库等云资源,部署于线上,简化开发步骤,减少程序代码量;在菜谱推荐功能上,采用基于 TensorFlow 搭建的神经网络,并上传部署至 SCF 腾讯云函数,设置其为 API 网关触发,以减少小程序端的调用时间。

参考文献

- [1] 黄琳. 揭秘饮食与情绪的心理机制—评《食物与情绪:食欲背后的心理学》[J]. 中国酿造,2020,39(03):219.
- [2] 百度. 百度 AI 开发平台[EB/OL]. [2021-06-14]. <https://ai.baidu.com/>.
- [3] 微信开放文档. 小程序开发[EB/OL]. [2021-06-14]. <https://developers.weixin.qq.com/miniprogram/dev/framework/>.
- [4] 吴莉莉,王健庆,杜宇翔,等. 基于百度 AI 的校园服务小程序的设计与实现[J]. 现代信息科技,2019,3(18):8-10.
- [5] 雷少玲. 基于 Node.js 为后端的微信小程序授权登录的探究与实现[J]. 电子制作,2020(22):42-44.
- [6] 武思怡,解佳,张雨,等. 微信小程序开发研究[J]. 无线互联科技,2018,15(11):52-54.
- [7] 杨光,吴明芬. 多种技术融合的 API 接口数据采集策略[J]. 计算机科学与应用,2020(11):1927-1937.