

智能垃圾分类及投放点定位系统 设计与实现

冯昊昊

北京林业大学, 北京

收稿日期: 2022年6月20日; 录用日期: 2022年7月19日; 发布日期: 2022年7月27日

摘要

为实现帮助人们辨识垃圾类别及定位对应垃圾桶投放点的位置, 该文开发出一套基于深度学习及百度地图API的垃圾分类与投放的系统。系统的主要功能如下: 将垃圾置于手机相机镜头内, 应用捕捉到画面并进行实时识别, 并导航至离用户当前位置最近的垃圾投放点; 通过语音询问垃圾类别或当前最近垃圾投放点位置, APP进行垃圾详细信息展示或导航至最近垃圾投放点; 手动输入垃圾桶类别, 应用自动获取当前位置信息并上传至服务器数据库保存, APP连网获取位置及类别信息, 并调用百度地图API进行覆盖物标记。

关键词

垃圾分类, 深度学习, APP开发, 定位与导航

Research on the Design and Implementation of Intelligent Garbage Classification and Positioning System

Haohao Feng

Beijing Forestry University, Beijing

Received: Jun. 20th, 2022; accepted: Jul. 19th, 2022; published: Jul. 27th, 2022

Abstract

To help people identify the type of garbage and locate the location of the corresponding trash

can, this article developed a system of garbage classification and positioning based on deep learning and Baidu Map API. The main functions of the system are as follows: place the garbage in the camera lens of the phone, the application captures the screen and recognizes it in real-time, and navigates to the garbage disposal point closest to the user's current location; asks the garbage category or the current location of the nearest garbage disposal point through voice, The APP displays detailed information about the garbage or navigates to the nearest garbage disposal point; manually enter the garbage bin category, the application automatically obtains the current location information and uploads it to the server database for storage, the APP connects to the Internet to obtain the location and category information, and calls the Baidu map API for coverage mark.

Keywords

Garbage Classification, Deep Learning, App Development, Positioning and Navigation

Copyright © 2022 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

随着我国经济水平的高速发展，人们的物质消费水平不断提升，相对应的垃圾产生量也在迅速增长，由垃圾产生的问题日益突出，人们的反映越来越强烈，推行垃圾分类势在必行，推行垃圾分类意义重大 [1] [2] [3]。

近些年来深度学习技术的发展，为垃圾分类工作提供了新的手段。周滢慤[4]基于机器视觉技术，设计了一种基于 Faster-RCNN 卷积神经网络的目标检测模型改进算法，并采用了基于 Hyper-Column 像素分类的方案解决小尺寸目标分类难的问题，同时也改进了 VGG16 神经网络结构以提升识别精度与速度，而人机交互纠错机制的引入则提高了该垃圾分类模型识别的精度。黄国维[5]提出基于深度学习的城市智能分类垃圾桶设计，通过在 NB-IOT 智能物联网技术基础上搭建智能垃圾桶实时检测网络，在卷积神经网络基础上提出了垃圾智能分类算法与模型，仿真试验结果表明，该分类算法反应时间快，识别准确率高。岳晓明等[6]提出了一种基于 CenterNet 网络的垃圾分类模型，该模型可在各种复杂情况下对垃圾目标进行快而准的识别，并拥有 25 ms 的低耗时以及 98.2% 的高精度，解决了目前垃圾分类检测方法适用性较差、错误率较高、无法实时检测等问题，有望应用在工业输送过程中。王小燕等[7]对 Faster RCNN、YOLOv3 和 RetinaNet 三种网络模型在垃圾分类中的应用进行了研究，并依托于所搭建的分类识别实验平台设计了对照实验，结果表明采用基于 Faster RCNN 的垃圾分类识别算法，达成了垃圾分类检测的目标，极大地节省了人工成本，提升了不同种类垃圾分类检测效率，从而降低了环境受到垃圾污染的可能性。

然而在推行垃圾分类和研究智能化垃圾分类方法的过程中，遇到了许多现实问题以及局限性，本文的研究内容便来源于人们在实际生活中遇到的垃圾分类处理问题以及对目前已有的垃圾分类研究成果应用局限性的思考，具体问题以及局限性如下。

1) 垃圾分类仅依赖工作人员进行分拣与回收，存在着许多问题，如劳动强度过高、分拣效率较低、工作环境恶劣，影响工作人员身体健康等。

- 2) 目前国内还存在一些群众对垃圾分类的意识较为淡薄, 不理解垃圾分类的重大意义, 如老人和小孩等人群, 由于主观学习能力较差, 导致对垃圾的类别认识不够清晰, 较容易错误投放垃圾。
- 3) 人们在外出时, 既存在因为未随身携带垃圾分类手册而不了解所携带垃圾所属类别, 导致无法进行准确投放, 也存在因为不了解垃圾投放点的位置而无法及时将身边的垃圾准确投放至对应类别垃圾桶内, 导致垃圾乱扔等问题。
- 4) 目前已有的研究内容主要分为基于硬件的分类系统以及基于深度学习的分类算法, 前者基于大型分类机搭建而成, 需要用户主动前往设备, 后者主要是针对算法模型的研究, 需要提前搜集垃圾图像进行检测, 两者均具有无法随时随地进行垃圾分类识别和便捷性较低等缺点。
- 5) 截止本文确定研究内容前, 利用移动端设备进行垃圾分类的相关研究非常少, 且在各软件应用商店内均无相关类型的 APP, 人们对使用更加便捷的方式进行垃圾分类与投放有一定的需求。

基于上述分析可知, 智能垃圾分类及投放点定位 APP 的开发, 可有效提升人们对垃圾种类识别以及垃圾投放的高效性和便利性, 也可提高人们进行垃圾分类的兴趣与意愿, 同时也具有一定的探索意义。

2. 系统开发主要技术选型

2.1. Java

Java 是一种开发语言, 具有面向对象的特点, 具有封装、继承、多态性的特性[8], 因此可以设计出低耦合的系统, 使系统更加灵活、更加易于维护。广泛使用在 Web 后端开发以及客户端开发中[9]。由于 Java 虚拟机 JVM 的存在, 使其开发的应用具有良好的可移植性。

2.2. SpringBoot

SpringBoot 框架是 Java 平台上的一个开源应用程序框架, 它提供了一个具有反转控制特征的容器。尽管 SpringBoot 框架本身对编程模型没有任何限制, 但它在 Java 应用程序中的频繁使用使其非常流行, 后来被用作 EJB (EnterpriseJavaBeans) 模型的补充, 甚至是替代品。SpringBoot 框架提供了一系列开发解决方案, 如利用控制反转的核心特性, 通过依赖注入实现控制反转, 实现管理对象生命周期的容器化, 各种持久化技术管理数据访问[10], 提供大量优秀的 Web 框架来促进开发等。

2.3. MyBatis-Plus

MyBatis-Plus (简称 MP) 是一个 MyBatis 的增强工具, 在 MyBatis 的基础上只做增强不做改变, 为简化开发、提高效率而生[11]。甚至是替代品。它具有以下特性:

- 1) 无侵入: 只做增强不做改变, 引入它不会对现有工程产生影响, 如丝般顺滑。
- 2) 损耗小: 启动即会自动注入基本 CURD, 性能基本无损耗, 直接面向对象操作。
- 3) 强大的 CRUD 操作: 内置通用 Mapper、通用 Service, 仅仅通过少量配置即可实现单表大部分 CRUD 操作, 更有强大的条件构造器, 满足各类使用需求。
- 4) 支持主键自动生成: 支持多达 4 种主键策略(内含分布式唯一 ID 生成器——Sequence), 可自由配置, 完美解决主键问题。

2.4. MySQL

MySQL 是一个关系型数据库管理系统[12], 由瑞典 MySQL AB 公司开发, 属于 Oracle 旗下产品。MySQL 是最流行的关系型数据库管理系统之一, 在 WEB 应用方面, MySQL 是最好的 RDBMS (Relational Database Management System, 关系数据库管理系统) 应用软件之一。MySQL 可以在各种平台上

运行 UNIX, Linux, Windows 等。可以将其安装在服务器甚至桌面系统上。此外, MySQL 是可靠, 可扩展和快速的。

2.5. Android

安卓(Android)是一种基于 Linux 内核(不包含 GNU 组件)的自由及开放源代码的操作系统[13]。主要用于移动设备, 如智能手机和平板电脑, 由美国 Google 公司和开放手机联盟领导及开发。Android 分为四个层, 从高层到低层分别是应用程序层、应用程序框架层、系统运行库层和 Linux 内核层。

2.6. XUI

XUI 是一个简洁而又优雅的 Android 原生 UI 框架。其主要具有以下特点:

- 1) 简洁优雅, 尽可能少得引用资源文件的数量, 项目库整体大小不足 1 M (打包后大约 644 k)。
- 2) 组件丰富, 提供了绝大多数我们在开发者常用的功能组件。
- 3) 兼容性高, 框架还提供了 3 种不同尺寸设备的样式(4.5 英寸、7 英寸和 10 英寸), 并且最低兼容到 Android SDK 17, 让 UI 兼容性更强。
- 4) 扩展性强, 各组件提供了丰富的属性和样式 API, 可以通过设置不同的样式属性, 构建不同风格的 UI。

3. 系统功能需求分析

智能垃圾分类及投放点定位系统主要包括垃圾名称模糊搜索、垃圾信息上传、投放点信息上传、垃圾分类图鉴、语音识别、智能垃圾分类、投放点定位及导航等功能。该系统分为前端(即 Android 移动端 APP)和后端(即服务器端应用)。

移动端功能需求简要说明如下。

- 1) 垃圾信息上传。用户点击主界面的垃圾信息上传按钮, 进入垃圾信息上传界面, 输入垃圾相关信息后, 点击上传, 即可将垃圾信息数据上传至数据库中。
- 2) 投放点信息上传。用户点击主界面的垃圾投放点信息上传按钮, 进入垃圾投放点信息上传界面, 输入垃圾投放点相关信息后, 点击上传, 即可将垃圾投放点信息数据上传至数据库中。
- 3) 垃圾分类图鉴。请求数据库中保存的所有已上传的垃圾信息, 将垃圾信息数据展示给用户。
- 4) 语音识别。当用户使用语音识别功能时, APP 会根据识别结果进行判断, 若用户是询问最近的指定垃圾投放点的位置信息, 则会跳转自可视化界面并自动进行导航; 若用户是询问某一垃圾所属的分类类别, 则会跳转到垃圾分类图鉴界面, 并展示该垃圾的相关信息。
- 5) 智能垃圾分类器。用户可以通过操作界面的按钮打开手机相机, 然后将所要进行检测的垃圾物放置在相机镜头内, APP 将会自动检测镜头内垃圾物的名称以及所属类别, 并将识别结果输出在操作界面上。
- 6) 投放点定位及导航。用户可通过主界面按钮进入地图界面, APP 在后台自动向服务端发送请求, 获取保存在数据库里的垃圾投放点的经纬度信息, 然后调用百度地图 API 将其显示在地图上。若用户发起导航请求, APP 则调用百度地图 API 通过获取用户当前所在位置, 并计算离当前位置最近的投放点坐标, 进而自动规划最佳导航路线。

服务器端功能主要是对垃圾、垃圾投放点等详细信息进行增、删、改、查等管理。

4. 技术架构

用户通过安卓端 APP 实时识别垃圾种类以及下载垃圾详情和投放点详情信息, 而客户端 APP 内的垃圾与投放点信息数据均来自服务端, 为了完成客户端与服务端的数据传输以及信息交互, 本系统采用

MySQL 数据库进行垃圾与投放点信息数据和各种信息的存储,然后通过 HTTP 网络协议实现了客户端服务端的交互。为了提升系统的使用性能以及可维护性,开发该 APP 时使用了 MVC 架构,将掌上电力 APP 划分为多层,通过各个层来完成不同的任务,各层任务如下:

表现层:该层即系统运行所在的移动设备,用户通过该层与智能垃圾分类及投放点定位 APP 进行交互。

业务层:该层即 APP 内各个功能具体实现所在层次,用户进行操作后,APP 通过该层进行逻辑处理以及通过 HTTP 与服务端进行网络交互,包括发送请求上传或获取垃圾与投放点信息等。

服务层:即服务端所在层次,该层主要对用户数据进行相关操作以及业务的核心处理,为用户提供服务。

数据层:大量的用户数据需要进行操作以及管理,该层主要利用数据库对用户数据进行管理以及维护。

图 1 展示了智能垃圾分类及投放点定位系统所采用的技术架构。

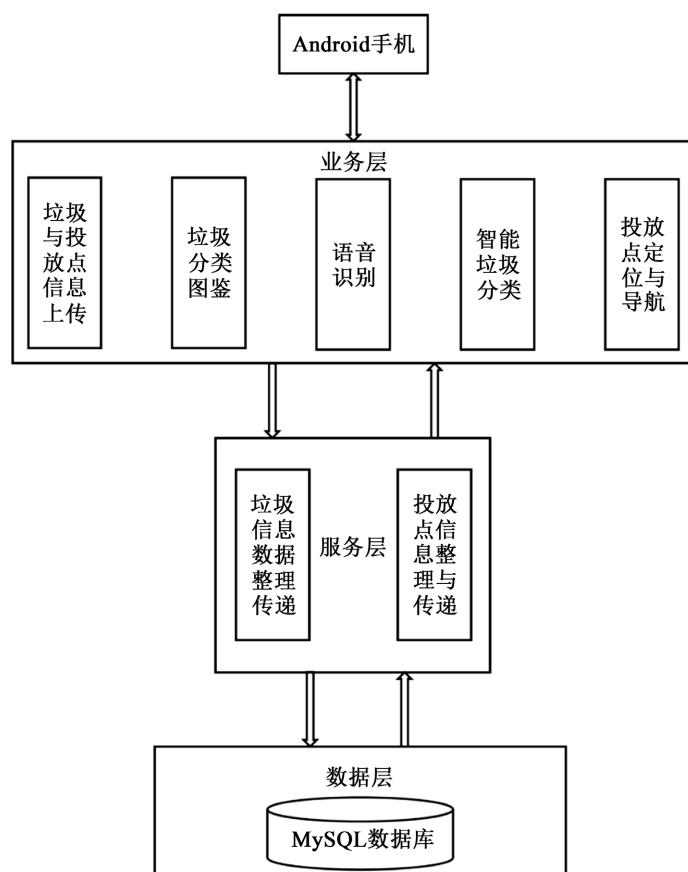


Figure 1. Technical architecture diagram of the system

图 1. 系统技术架构图

5. 系统设计与实现

5.1. 关键问题解决方案

在开发智能垃圾分类及投放点定位 APP 过程中,需要解决的关键问题主要有两点,分别为 APP 调用垃圾分类模型对手机相机视频流中的垃圾物体进行分类识别以及将投放点位置相关信息展示在地图界面上并可以对其发起导航,下面将针对上述开发过程中所需要解决的关键问题进行具体说明并提出相应的解决方案。

5.1.1. MobileNetV2 轻量化卷积神经网络

在开发 APP 中的智能垃圾分类模块时，需要对手机相机视频流中的垃圾物体进行分类识别，考虑到目前手机的处理器已发展完备具备较高的计算能力，同时为了给用户带来良好的体验，需保证识别的高效性和准确性，因此本文采用图像处理的方法对视频流中的帧图像进行分析处理，从而确定相机镜头下垃圾的名称以及所属种类并将结果信息直观展示在手机界面上。

在计算机视觉领域中，图像分类是非常重要的组成部分也是基本问题之一，行为检测、目标识别等复杂的视觉任务也是以它为基础进行研究的。传统的图像分类算法一般需要对图像进行手工特征提取来对图像进行描述，在进行分类时，需要将整幅图像作为输入，导致算法的计算量非常大，计算速度较慢，分类效率以及正确率也会因为图像的背景等冗余信息的干扰而降低，其次分类效果随着特征提取算法的不同而各有优劣，算法对不同图像的自适应程度较差。目前，基于深度学习的图像分类算法融合了特征提取及分类，无需人工进行图像特征提取，算法实现简单明了，实验结果精度较高，模型的鲁棒性较好。

在 MobileNetV1 网络结构基础上对其进行了改进并提出了 MobileNetV2 网络。相较于 MobileNetV1，MobileNetV2 的改进主要体现在如下四个方面。

- 1) 引入残差结构，先升维再降维，增强梯度的传播，显著减少推理期间所需的内存占用(Inverted Residuals)。
- 2) 去掉 Narrow layer (low dimension or depth)后的 ReLU，保留特征多样性，增强网络的表达能力 (Linear Bottlenecks)。
- 3) 网络为全卷积的，使得模型可以适应不同尺寸的图像，使用 RELU6 (最高输出为 6)激活函数，使得模型在低精度计算下具有更强的鲁棒性。
- 4) 增加了 MobileNetV2 building block，如下图 2 所示，若需要下采样，可在 DWise 时采用步长为 2 的卷积；小网络使用小的扩张系数(expansion factor)，大网络使用大一点的扩张系数。

改进后的 MobileNetV2 网络完整结构如下图 2 所示，第一层是标准卷积，然后后面是前述的 bottleneck 结构。其中 t 是扩展因子， c 是输出通道数， n 是重复次数， s 代表步长。如果步长为 2，代表当前重复结构的第一个块步长为 2，其余的步长为 1，步长为 2 时则没有跳跃连接。

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	-

Figure 2. Structure of MobileNetV2 network

图 2. MobileNetV2 网络结构

经过使用 ImageNet 上的数据集进行模型训练和测试，MobileNetV2 相较于 MobileNetV1 在降低参数量，缩小网络结构体积的前提下，Top1 准确率有所提升，并且单帧图像检测耗时由 113 ms 下降到 75 ms，分类识别速度显著提高。

针对手机相机视频流中的垃圾图像，采用 MobileNetV2 模型进行垃圾分类识别。MobileNetV2 模型既可以满足高准确率要求，又可以进行对视频流中的垃圾图像进行实时分类，获得了分类速率以及分类效果之间的平衡。

5.1.2. 百度地图 Android SDK

在开发 APP 中的投放点定位及导航模块时，首先需要在手机界面上显示网络地图图层，然后需要将投放点位置信息显示在网络地图上，并且能够对投放点位置发起导航。为了给用户带来良好的使用体验，在实现该功能时需要考虑网络地图操作的流畅性、投放点位置信息展示的明显性、导航路线的实时性等问题。

随着移动互联网的快速发展和人们对 APP 便利性的需求，旅游导航类、物流类、生活服务类 APP 在应用中标配了地图功能，这些功能越来越受到人们的欢迎，发挥着越来越重要的作用。另一方面，基于 Android 系统的移动终端设备，由于其便携性、稳定性和应用丰富性，拥有着大量的用户群体。基于此，国内对网络地图的研究取得了快速发展，出现了百度地图、腾讯地图、高德地图等地图应用，其中百度地图由于研究开展较早，用户基数较大，属于其中发展较好的。百度地图为安卓应用开发者提供了基于 Android 系统的地图 SDK，SDK 包含了丰富的功能接口，包括地图创建、与地图交互、在地图上绘制、导航、路线规划等。

在创建地图方面，基于百度地图 Android SDK 所创建的地图可清晰显示行政区划等信息并可以在普通线路地图、卫星地图和空白地图三种类型之间进行切换。所创建的地图支持双指操作进行放大和缩小，且操作流畅无卡顿，对道路、建筑物、绿地以及河流等重要的自然特征都有良好的展示。

在对地图进行绘制方面，百度地图 Android SDK 支持开发人员在网络地图界面上放置覆盖物，可支持放置的覆盖物种类丰富，包括绘制点标记、线、弧线和面等。其中点标记所使用的覆盖物图标可进行自定义修改，以满足不同场景下的不同需求。

在导航方面，百度地图 Android SDK 为开发人员提供了三种导航功能，包括步行导航、市内步行导航和骑行导航，并为每一种导航提供了调用接口。此外在步行导航功能里，SDK 还额外支持了步行 AR 导航以及偏航纠正等功能，大大提升了用户体验。

5.2. Android 端设计与实现

5.2.1. 垃圾信息上传功能模块与实现

垃圾信息填写与上传界面分为四个部分，分别为垃圾名称填写、垃圾种类选择、垃圾描述信息填写、垃圾示例图像采集以及垃圾信息确认上传。其中，垃圾名称填写输入框功能通过安卓原生自带的 EditText 组件实现，用户可在输入框内输入具体垃圾名称，系统会将输入或改动内容自动保存；垃圾种类选择功能是由安卓原生自带的 CheckedTextView 组件实现，用户点击该组件，则会弹出下拉选择框，共四种类型可选择，分别为厨余垃圾、可回收垃圾、其它垃圾、有害垃圾，系统会自动保存用户最新选择的结果；垃圾描述信息填写同垃圾名称填写功能一样，基于安卓原生自带的 EditText 组件实现，系统同样会将输入或改动内容自动保存；垃圾示例图像采集功能分为两种情况，第一种情况是选择相册内已存在的照片为示例图像，首先需要创建安卓自带的 activity 跳转工具类 Intent，并为其配置 Intent.ACTION_PICK 属性，然后调用 XUI 框架里的 XPageFragment.java 类里的 startActivityForResult 方法并传入上面创建的工具类 Intent，便可进入相册选择照片，选择完成后系统将保存结果图像并弹窗展示；另外一种情况则是选择使用手机相机进行拍照获取示例图像，首先同样需要创建安卓自带的 activity 跳转工具类 Intent，并为其配置 MediaStore.ACTION_IMAGE_CAPTURE 属性，然后调用安卓自带 FileProvider 类中的 getUriForFile

方法传入当前上下文、APP 文件包路径、照片占位文件等参数并获取返回结果，然后将该返回结果以及创建的 Intent 类传入 XPageFragment.java 类里的 startActivityForResult 方法作为参数，即可进入相机拍摄界面，拍摄完成后系统将保存结果图像并弹窗展示，此外，两种情况所获得的图像均可在弹窗中进行裁剪，该操作是基于 UCrop 开源的安卓端图像裁剪工具中的 UCropActivity 实现的，可对图像进行裁剪以及放大缩小等操作；垃圾信息上传确认由安卓原生提供的 Button 组件实现，并为其添加点击监听事件，当用户点击确认上传按钮时，系统将用户之前所填写或选择的垃圾名称、垃圾种类、垃圾描述信息数据进行封装，然后通过网络传输，将数据上传至云端服务器应用程序，并保存至数据库中，而垃圾示例图像数据则会由系统调用阿里云 OSS 所提供的 Android SDK，将图像数据上传保存至阿里云 OSS，即阿里云提供的对象存储服务。本功能模块实现效果如图 3 所示。

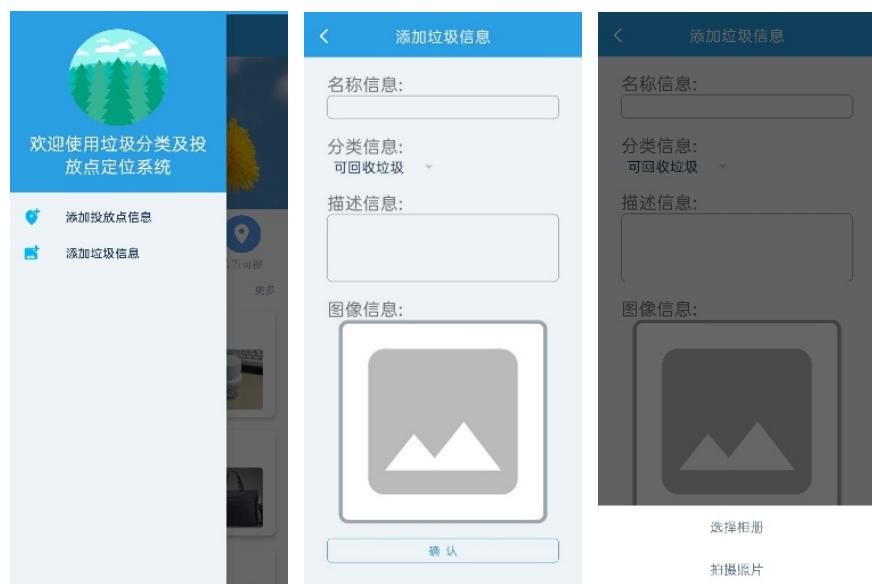


Figure 3. Design of garbage information upload function module

图 3. 垃圾信息上传功能模块设计

5.2.2. 投放点信息上传功能模块设计与实现

投放点信息上传界面分为五个部分，分别为地理信息、位置描述、周围环境图像信息、备注信息以及确认上传按钮，其中地理信息是对当前位置的省市区乡镇街道信息进行描述，位置描述是对当前位置参考描述(如北京林业大学五号楼附近)，两者都是通过调用 BaiduMap Android SDK 获取到的，在 APP 内调用 BaiduMap Android SDK 获取地理位置信息的步骤如下：① 位置信息数据的获取可通过新建子类 MyListener 并使之继承抽象类 BDAbstractListener，然后重写子类中的 onReceiveLocation 方法；② onReceiveLocation 方法会传入一个 BDLocation 类参数，该参数包含了获取当前位置相关信息的方法；③ 通过 BDLocation 的 getProvince 方法可以获取到当前所在省份，getCity 方法可以获取到当前所在市，getDistrict 方法可以获取到当前所在区县，getStreet 方法可以获取到当前所在街道，getTown 方法可以获取到当前所在乡镇，getLongitude 方法可以获取到当前所在位置的经度，getLatitude 方法可以获取到当前所在位置的纬度，getLocationDescribe 方法可以获取到当前位置描述信息。

另外三个部分，即周围环境图像信息的采集、备注信息的填写以及确认上传按钮的功能与垃圾信息上传功能模块的垃圾示例图像采集、垃圾描述信息填写、垃圾示例图像采集所使用的技术基本一致，此处不再赘述。本功能模块实现效果如图 4 所示。



Figure 4. Design of the function module for uploading information on garbage disposal points
图 4. 垃圾投放点信息上传功能模块设计

5.2.3. 垃圾分类图鉴功能模块设计与实现

垃圾类型选择列表分为四项，分别为厨余垃圾、可回收垃圾、其它垃圾以及有害垃圾，每个选项对应一个 Button 组件，点击 Button 即可跳转至对应类型的垃圾图谱卡片列表。该界面的设计实现过程如下：首先创建该界面对应的 XML 布局文件，父布局使用嵌套的双层 LinearLayout，内部配置四个由 XUI 框架提供的基于安卓原生 Button 美化的 RippleView 用于展示四种垃圾类型按钮，并为其分别添加不同颜色，采用的是北京市推行的四大垃圾种类代表色，即绿色为厨余垃圾、灰色为其它垃圾、蓝色为可回收垃圾、红色为有害垃圾；然后创建对应的 Fragment 用于定义具体的执行逻辑，在该 Fragment 内为上述四个 RippleView 按钮配置具体的点击监听事件，即点击不同类型的 RippleView 按钮跳转至不同类型的垃圾图谱卡片列表界面。

垃圾图谱卡片列表界面用于展示所选择垃圾类型下，系统所有已存储的垃圾信息卡片列表，该界面的实现方式主要是通过如下方式：首先创建该界面的 XML 布局文件，父布局使用 LinearLayout，在父布局内部嵌入在 GitHub 上开源的 scwangUI 框架中的 SmartRefreshLayout 布局，在 SmartRefreshLayout 布局中配置 MaterialHeader、RecyclerView、MaterialFooter 三个组件，分别用于下拉刷新样式、卡片样式、列表触底展示样式，由此便组成了垃圾图谱卡片列表界面整体布局；然后创建垃圾图谱卡片列表界面对应的 Fragment 用于定义具体的执行逻辑，在该 Fragment 内为 RecyclerView 内的每一项(即垃圾图谱卡片)配置具体的点击监听事件，即点击不同的卡片则会跳转至具体垃圾的垃圾图谱卡片界面。当用户由垃圾类型选择列表界面进入到该界面时，系统会自动向服务端应用程序发送获取指定种类垃圾数据的请求，应用程序收到请求后，从数据库中读取出所有该类型的垃圾数据，封装成 JSON 数据格式并返回给 APP，当 APP 获取到该数据之后，解析成一个数据集合 List，List 中每一项数据代表一种垃圾，并将每一项数据对应到 RecyclerView 内的每一项数据，进行最终的展示。

垃圾图谱卡片界面用于展示具体某一垃圾的详细信息，该界面包含四个部分，从上至下，依次为垃圾名称、垃圾示例图像、所述类别、垃圾描述信息。此界面所采用的技术原理为：首先，创建此界面的 XML 布局文件，在父布局 LinearLayout 中嵌套一个 ConstraintLayout，然后在 ConstraintLayout 内部由上至下，分别配置一个 XUI 框架提供的图像展示组件 XUIAlphaImageView 用于展示垃圾示例图像，一个

TextView 用于展示垃圾名称，一个 TextView 用于展示垃圾所属种类，一个 TextView 用于展示垃圾描述信息；其次，创建该界面的逻辑处理类 Fragment.java，然后在该类内部为垃圾示例图像展示组件添加点击事件，具体为当首次点击该图像时，图片可放大，再次点击即可缩小至原来大小，并可重复点击。本功能模块实现效果如图 5 所示。



Figure 5. Design of function module of garbage classification picture book

图 5. 垃圾分类图鉴功能模块设计

5.2.4. 语音识别功能模块设计与实现

为实现讯飞语音识别 SDK 的下载与导入，需进入科大讯飞开发者平台的控制台，在控制台里创建新应用，在创建新应用里输入所要创建的应用的基本信息，平台选择 Android，创建完成后可在我的应用里看到 APPID，此 ID 唯一且在 APP 开发中会使用到。点击添加新服务可以选择所要使用的服务，选择完成后点击下载即可进行下载，下载完成后进行解压，解压后的文件夹包括 assets、libs、res 等文件，将这些文件夹内的文件移至 APP 项目根目录下相同名称文件下内即可。

完成 SDK 的导入后，即可进行语音识别模块的开发，主要过程如下：创建该界面的 XML 布局文件，使用三层 LinearLayout 布局进行嵌套以保证内部组件位于屏幕中央，然后在内部添加 TextView 组件用于展示语音输入内容和 Button 按钮组件用于开始语音识别的点击；在语音识别逻辑实现类 SpeechFragment.java 中定义 InitVoice 方法，并在该方法内初始化 SDK 提供的语音识别监听器类 RecognizerListener，在该 RecognizerListener 类的 onResult 方法(即监听结果回调方法)中编写逻辑判断代码，逻辑为当用户询问的问题包含某一垃圾名称时，则弹窗询问用户是否要查看该垃圾的详细信息，用户点击是便跳转至该垃圾的图谱界面，点击否弹窗则会消失；当用户询问的问题包含投放点、垃圾桶、垃圾箱等关键词时，则弹窗询问用户是否导航至离当前位置最近的垃圾投放点，用户点击是便跳转至导航界面，点击否则弹窗消失；当 APP 无法识别用户的语音输入时，则会提示用户匹配不到语音识别内容对应结果，请使用标准普通话重试。弹窗界面使用的是 XUI 框架内的 DialogLoader 类实现。本功能模块实现效果如图 6 所示。

5.2.5. 智能垃圾分类功能模块设计与实现

本模块的开发过程主要分为模型训练以及模型部署两个部分，接下来将对每部分的主要流程进行介绍。



Figure 6. Design of speech recognition function module
图 6. 语音识别功能模块设计

1) 模型训练

进行图像分类模型的训练，首先需要安装深度学习环境，本文所使用的模型训练机器主要配置如表 1 所示。

完成环境配置后，需要准备数据集进行训练，本文所使用的垃圾图像数据集均通过 Python 爬虫脚本从网络上爬取得到，通过人工筛选爬取结果，最终确定了包含 4 种垃圾类型，119 种垃圾，8 万多张图像的数据集。

完成数据集的收集与整理之后，即可开始模型的训练，模型训练主要过程如下：

① 下载 models-1.12.0 并解压，为 GitHub 上谷歌开源的一个 TensorFlow 框架研究项目，其包含了模型训练所需要用到的脚本以及网络结构文件等；

Table 1. Table structure of model training machine configuration
表 1. 模型训练机器配置表

基本配置	型号及版本号
CPU	Intel® Core™ i7-9700K CPU @ 3.60GHz×8
GPU	GeForce GTX 1080 Ti
OS	Ubuntu18.04
Python	3.6.6

- ② 将解压后的 models 文件夹中的 slim 文件夹加入环境变量；
- ③ 下载预训练的量化模型 MobileNet_v2_0.75_224_quant.pb；
- ④ 修改 MobileNetV2 结构参数，进入 slim/nets/MobileNet_v1.py 文件，修改 MobileNetV2 以及 MobileNet_v2_base 方法中的 depth_multiplier 为 0.75；
- ⑤ 将③中下载的预训练模型解压至 slim 文件夹里，并将该预训练模型的绝对路径作为训练脚本中 checkpoint_path 参数的值，在 slim 中新建 train_dir 文件夹，用来保存训练模型后生成的 ckpt 节点文件；

⑥ 开始训练，调用 slim 文件夹内的 `train_image_classifier.py` 训练脚本并将准备完成的数据集路径传入。训练过程中的模型准确率以及 Loss 变化分别如图 7 和图 8 所示，模型训练完成经验证集测试，准确率为 92%；

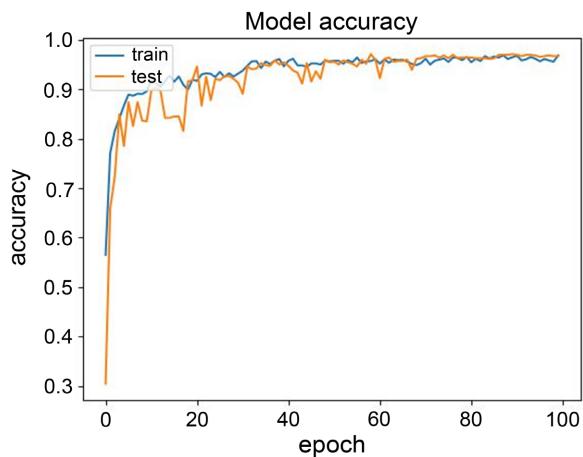


Figure 7. Graph of model training accuracy change
图 7. 模型训练准确率变化图

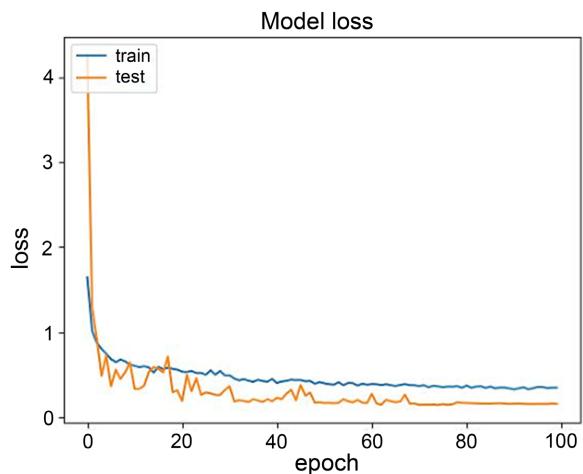


Figure 8. Diagram of model training Loss change
图 8. 模型训练 Loss 变化图

⑦ 导出 pb 模型文件，通过调用 slim 文件夹内的 `export_inference_graph.py` 脚本实现；
 ⑧ 转换 pb 模型为 tflite 模型，调用 tensorflow 依赖包下的 `lite/toco.py` 实现。

2) 模型部署

将上述得到的 `graph.tflite` 模型，分类标签文件 `label.txt` 文件，以及存有 119 种垃圾中英文名称对照和所属类别的 `data.txt` 文件移动至 APP 项目中的 `assets` 文件夹内。然后修改图像分类逻辑类 `ImageClassifier.java` 中的相关属性配置，将 `MODEL_PATH` 的值设置为模型文件 `graph.tflite` 的路径，将 `LABEL_PATH` 设置为标签文件 `labels.txt`，将识别结果现实个数修改为 1，即仅显示可能性最高的垃圾名称。在主界面逻辑处理类 `MainActivity.java` 中为智能识别图标添加点击事件，当用户点击该图标时，APP 跳转至 `ImageClassifier.java`，开启手机相机镜头，并将模型以及标签文件读取进内存对相机视频流图像中的垃圾物体进行分类识别，实现效果如下图 9 所示。



Figure 9. Design of functional modules for intelligent garbage classification

图 9. 智能垃圾分类功能模块设计

5.2.6. 投放点定位及导航功能模块设计与实现

投放点定位功能开发主要过程如下，首先在 `AndroidManifest.xml` 中配置 service，service 内容包括三个属性：name、enabled、process，值分别为：`com.baidu.location.f`、`true`、`:remote`，目的是开启间隔定位功能(每隔 n 秒定位一次)；然后创建地图界面的 XML 布局文件，布局内容为 `RelativeLayout` 父布局内嵌套一个 `com.baidu.mapapi.map.MapView` 组件，百度地图将会在此组件上进行展示，布局文件编辑完成后，创建地图逻辑处理类 `MapFragment.java`，后续操作均在该类中进行。在 `initViews` 方法内加载刚创建的地图 XML 布局文件以及标记图标文件；创建继承自安卓原生自带的 `AsyncTask` 的子类 `MyTask` 进行异步编程(之所以采用异步编程，主要因为在程序开发中，相较于 CPU 处理的速度，网络请求的速度比之要慢很多，若用户处在网络状态较差的环境且 APP 在主线程中进行了网络请求操作，那么主线程就会进入阻塞等待网络返回，达到 5 秒之后应用便会崩溃重启)并重写 `doInBackground` 方法，在该方法内编写向服务端应用程序发送投放点信息请求，并将返回结果中每一投放点的经纬度信息封装为 SDK 中的 `MarkerOptions` 类集合，然后将该集合传入 SDK 中 `BaiduMap` 类里的 `addOverlays` 方法，即可在地图上指定经纬度位置进行定位与标记；在上述定位功能开发中所设置的标记图标添加点击监听事件，点击标记物会在屏幕底部进行弹窗，弹窗内容包括该投放点的位置描述信息、环境示例图像以及两个按钮——发起导航和取消，该弹窗由安卓原生自带的 `BottomSheetDialog` 类为基础进行设计实现，投放点的位置描述信息、环境示例图像均为投放点信息上传功能模块上传的数据，当用户点击发起导航按钮，即跳转至该投放点的步行导航界面，下面将对投放点导航功能开发进行简要介绍。

投放点导航功能开发主要过程如下，首先在地图逻辑处理类 `MapFragment.java` 中创建 `routePlanWithWalkParam` 方法，并以经纬度为参数，在该方法内部将传入的经纬度封装为 SDK 中的 `WalkNaviLaunchParam` 类实例，然后以 `WalkNaviLaunchParam` 类实例为参数实例化 SDK 中的 `WalkNavigateHelper` 类的实例，即导航控制类，并重写 `WalkNavigateHelper` 类中的 `onRoutePlanSuccess` 方法，在该方法内部实现导航界面的跳转；接着实现导航界面开发，创建导航界面逻辑处理类 `NavigateActivity.java`，主要逻辑处理是在其 `onCreate` 方法中设置步行导航状态监听。本模块实现效果如下图 10 所示。

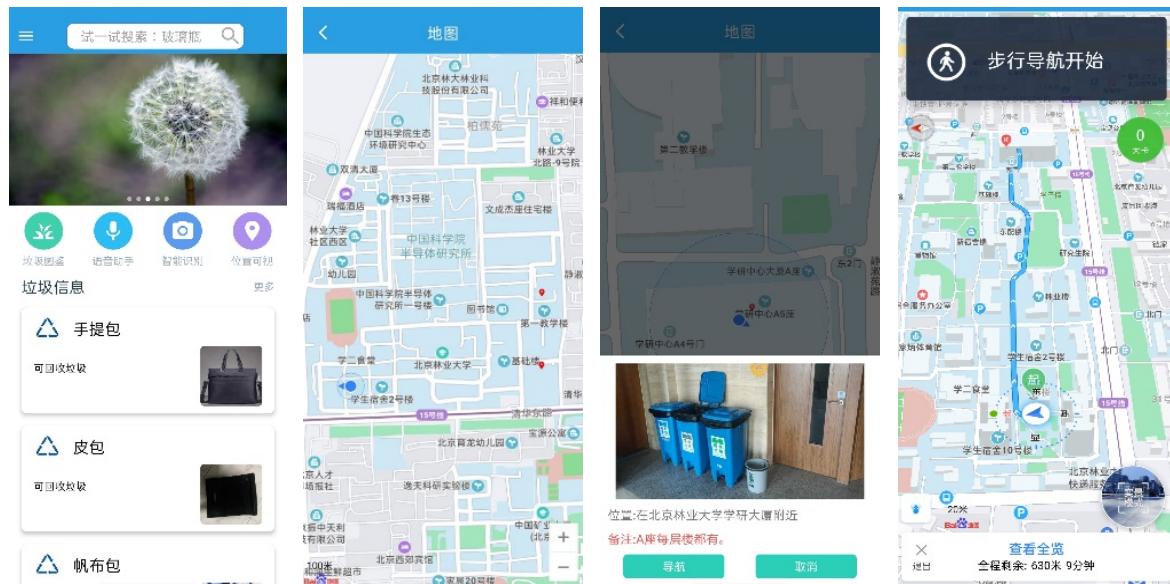


Figure 10. The design of location and navigation function module of the placement point
图 10. 投放点定位及导航功能模块设计

5.3. 服务端设计

服务端应用程序搭建方面，使用 IntelliJ IDEA 编译器搭建基于 SpringBoot 框架的后台应用程序项目，项目分为三层架构，即 Controller 接口层、Service 业务层以及 Mapper 持久层。开发阶段使用 Swagger-2 框架进行后台接口测试。持久层使用 Mybatis-plus 框架与 MySQL 数据库进行交互。同时，在基于 Centos7 系统的阿里云服务器上安装 docker，在 docker 内安装 Nginx 及 MySQL 和 Redis 镜像并配置端口映射。项目开发完成后将代码打包为 jar 格式文件，再以此文件为基础制作 docker 镜像，创建项目的 docker 容器，将服务端应用程序运行于 docker 环境中，便于后续的更新与管理。

5.4. 数据库设计

智能垃圾分类及投放点定位系统使用了 MySQL 关系型数据库管理系统以及 Redis 非关系型数据库。MySQL 数据库保用于保存用户上传的垃圾信息数据以及投放点信息数据。Redis 数据库用于缓存服务端从数据库中取出的数据，加快访问数据，缓解 MySQL 数据库的读写压力。根据系统需求共有 2 张 MySQL 数据库表格。

垃圾信息表：记录所有用户上传的垃圾种类相关信息，其表结构如下表 2 所示。

Table 2. Table structure of garbage information

表 2. 垃圾信息表结构

名称	类型	长度	不为 NULL	键	注释
id	varchar	19	<input checked="" type="checkbox"/>	主键	主键 ID
name	varchar	30	<input checked="" type="checkbox"/>	-	垃圾名称
type	varchar	15	<input checked="" type="checkbox"/>	-	所属分类
description	varchar	255	<input type="checkbox"/>	-	描述信息
image	varchar	255	<input checked="" type="checkbox"/>	-	示例图像
created_time	datetime	-	<input checked="" type="checkbox"/>	-	创建时间

投放点信息表：记录所有用户上传的投放点相关信息，其表结构如下表 3 所示。

Table 3. Table structure of placement point information
表 3. 投放点信息表结构

名称	类型	长度	不为 NULL	键	注释
id	varchar	19	<input checked="" type="checkbox"/>	主键	主键 ID
longitude	varchar	15	<input checked="" type="checkbox"/>	-	经度
latitude	varchar	15	<input checked="" type="checkbox"/>	-	纬度
province	varchar	30	<input checked="" type="checkbox"/>	-	所在省份
city	varchar	30	<input checked="" type="checkbox"/>	-	所在城市
district	varchar	30	<input checked="" type="checkbox"/>	-	所在县区
street	varchar	30	<input type="checkbox"/>	-	所在街道
town	varchar	30	<input type="checkbox"/>	-	所在乡镇
address	varchar	100	<input checked="" type="checkbox"/>	-	详细地址
location_describe	varchar	100	<input checked="" type="checkbox"/>	-	位置描述
remarks	varchar	255	<input type="checkbox"/>	-	备注信息
image	varchar	255	<input checked="" type="checkbox"/>	-	环境图像
create_time	datetime	-	<input checked="" type="checkbox"/>	-	创建时间

6. 系统测试

能否正确实现用户的功能需求，是评价 APP 实用性的关键，因此对智能垃圾分类及投放点定位系统的各个功能模块的测试是至关重要的，以此来确保功能的正确运行，下面将通过设计测试用例来对本文系统进行综合性测试。

服务端是系统的重要组成部分，对系统数据的存取与共享起到关键性作用。为测试服务端功能，表 4 为服务端应用程序的测试用例。

Table 4. Table of test cases for server applications
表 4. 服务端应用程序测试用例表

测试编号	测试功能	测试具体内容	测试结果
1	新增数据	将 APP 传输来的垃圾或投放点信息数据进行整理和存储	应用程序可正常接收数据并保存至数据库中
2	删除数据	接收删除数据的请求，并根据请求内容删除对应的数据	应用程序可根据请求内容删除指定的垃圾或投放点信息数据
3	查询数据	接收 APP 获取数据请求并根据请求内容从数据库中取出对应数据并返回	应用程序可根据请求内容从数据库中取出对应的垃圾或投放点信息并整理返回给 APP
4	文件上传	上传图像文件	垃圾与投放点的示例图像可以正常上传
5	维护升级	关闭服务端应用程序并上传新的应用程序文件启动，实现应用升级	应用程序可被正常关闭并被替换，完成升级

客户端 APP 是软件的使用关键, 由于其是直接面向用户的, 因此对其测试需要更加的严密和周到。测试客户端是否能正常工作, 其测试用例如表 5。

Table 5. Table of test cases for client APP
表 5. 客户端 APP 测试用例表

测试编号	测试功能	测试具体内容	测试结果
1	上传垃圾与投放点信息	输入空的或者正确的垃圾信息, 然后点击上传按钮	当用户输入正确时, 点击上传, APP 提示用户上传成功, 数据库表中多出一条对应数据, 输入空数据时, APP 会提示用户输入信息有误, 请重新确认
2	查看垃圾分类图鉴	从主界面进入垃圾分类图鉴模块, 选择一种类型垃圾进入集合页, 再选择一种垃圾查看详细信息	用户选择一种垃圾类型进入集合页, APP 展示该类型下所有垃圾信息卡片, 用户点击卡片进入垃圾信息详情页, 显示垃圾的名称、类别、描述等信息
3	语音询问	从主界面进入语音识别模块并进行语音询问	若用户询问某一垃圾信息, APP 则自动跳转至该垃圾的详情页, 若用户询问投放点信息, APP 则自动向最近的投放点发起导航, 否则 APP 提示无法识别语音内容
4	垃圾分类识别	从主界面进入智能垃圾分类模块并将垃圾物放置于手机相机镜头内	APP 可识别相机镜头内的垃圾物并对其进行分类, 并将分类识别的结果显示在手机界面上
5	投放点定位	从主界面进入位置可视模块	APP 加载地图界面, 并且投放点位置信息以标记物的形式展现在地图上
6	投放点信息展示	从主界面进入位置可视模块, 并点击地图上的标记物	用户点击标记物, APP 底部弹窗显示该投放点的示例图像、位置描述等信息, 以及导航和取消两个按钮
7	投放点导航	从主界面进入位置可视模块, 并点击地图上的标记物, 在弹出的窗口上点击导航按钮	APP 跳转至步行导航界面, 向该投放点所在位置发起导航

7. 结束语

为了响应国家推行垃圾分类的措施, 开发了一个智能垃圾分类及投放点定位系统, 系统主要包括垃圾信息上传、投放点信息上传、垃圾分类图鉴、语音识别、智能垃圾分类、垃圾投放点定位及导航等功能, 分为 Android 端 APP 以及服务器后端应用程序。经过实际测试, 本系统可以满足人们的日常垃圾分类需求。相比较于目前已有相似的系统或平台, 该系统设计垃圾图鉴功能模块, 收集大量的垃圾种类信息, 帮助人们查找了解生活中遇到的垃圾及其对应的分类信息; 设计语音识别模块, 帮助人们便捷地查询所需要的垃圾种类信息或投放点信息, 对于老人小孩等文字操作不便的人群, 语音识别功能很好地解决了这一不便; 设计投放点定位及导航模块, 帮助人们了解所在地理位置周围的投放点信息, 将身边的垃圾及时投放进对应分类的垃圾桶内; 设计智能垃圾分类模块, 在使人们在感受人工智能技术带来的便捷之外, 提高了人们对于垃圾分类的兴趣, 也使得人们可以对生活中的垃圾进行快速的识别与分类。

参考文献

- [1] 堵锡忠, 李娟. 生活垃圾分类“北京实践”的特点分析[J]. 城市管理与科技, 2021, 22(3): 19-21.
- [2] 任玉霞. 人民日报社: 做好垃圾分类共建美丽家园[J]. 城市管理与科技, 2021, 22(3): 57.
- [3] 薛存科. 基于绿色发展的城市垃圾分类及其污染的法律治理研究[J]. 环境科学与管理, 2021, 46(4): 48-52.
- [4] 周滢慤. 基于机器视觉的生活垃圾智能分拣系统的设计与实现[D]: [硕士学位论文]. 哈尔滨: 哈尔滨工业大学, 2018.
- [5] 黄国维. 基于深度学习的城市垃圾桶智能分类研究[D]: [硕士学位论文]. 淮南: 安徽理工大学, 2019.
- [6] 岳晓明, 李俊, 侯言旭, 林志诚. 基于 CenterNet 的垃圾分类检测方法[J]. 工业控制计算机, 2020, 33(8): 78-79+82.
- [7] 王小燕, 谢文昊, 杨艺芳, 胡瑞. 基于深度学习的垃圾分类检测方法[J]. 现代电子技术, 2021, 44(21): 110-113.
<https://doi.org/10.16652/j.issn.1004-373x.2021.21.023>
- [8] 柯采. 基于 J2EE 和 MVC 模式的 Web 应用研究[J]. 软件, 2020, 41(3): 165-167.
- [9] 张恒铭. 基于 SpringBoot + MyBatis 框架的高校人事信息发布系统设计与实现[J]. 电脑知识与技术, 2020, 16(18): 68-70.
- [10] 叶刚, 王立河, 王英明, 谷国栋. 基于 Mybatis Plus 的动态生成代码设计与实现[J]. 电脑编程技巧与维护, 2019(7): 7-8.
- [11] 石怡. 基于 MySQL 数据库的查询性能优化研究[J]. 四川职业技术学院学报, 2021, 31(1): 164-168.
- [12] 李登科, 王聪华. 基于 Android 操作系统的移动终端 APP 开发研究[J]. 电脑编程技巧与维护, 2020(9): 69-70.
- [13] 高俊, 牛艺霏. Android 应用架构技术分析和设计[J]. 信息技术与信息化, 2020(3): 13-15.