

**基于机器学习的垃圾分类系统**

**项 目 详 细 设 计 书**

**Garbage Sorting System Based on Machine Learning**

**课程名称： 软件项目管理**

**指导老师： 刘 晋**

**小组成员： 202130310241 杨文豪**

**202130310242 侯同庆**

**202130310243 陈洪涛**

**日 期： 2022.03.31**

目录

[1 引言 3](#_Toc99667538)

[1.1编写目的 3](#_Toc99667539)

[1.2项目背景 3](#_Toc99667540)

[2 设计概述 4](#_Toc99667541)

[2.1 需求概述 4](#_Toc99667542)

[2.2 运行环境概述 4](#_Toc99667543)

[3 系统详细设计 5](#_Toc99667544)

[3.1总体设计 5](#_Toc99667545)

[3.2 业务模块 7](#_Toc99667546)

[3.2.1 业务框图 7](#_Toc99667547)

[3.2.2 程序逻辑 9](#_Toc99667548)

[3.2.3 接口设计 10](#_Toc99667549)

[3.3 算法模块 11](#_Toc99667550)

[3.3.1 程序逻辑 11](#_Toc99667551)

[3.3.2 接口设计 11](#_Toc99667552)

[3.4 前端模块 12](#_Toc99667553)

[3.4.1 程序逻辑 12](#_Toc99667554)

[3.4.2 界面设计 12](#_Toc99667555)

[3.4.3 接口设计 13](#_Toc99667556)

[4 风险分析 14](#_Toc99667557)

# 1 引言

## 1.1编写目的

本报告的目的是对垃圾识别分类系统进行详细设计说明，以便用户及项目开发人员了解产品详细的设计与实现。为开发人员提供开发参考书。以下叙述将结合文字描述、[伪代码](https://so.csdn.net/so/search?q=%E4%BC%AA%E4%BB%A3%E7%A0%81&spm=1001.2101.3001.7020)，图表等来描述垃圾识别分类系统的详细设计和相关的模块描述。本报告的预期读者有用户、项目经理、开发人员以及跟该项目相关的其他人员。

## 1.2项目背景

**（1）项目名称**：基于OpenCV和TensorFlow的生活垃圾图像分类识别

**（2）任务提出者**：杨文豪、侯同庆、陈洪涛

**（3）开发者**：杨文豪、侯同庆、陈洪涛

**（4）选择背景**：日常生活中的垃圾一般分为有害垃圾、可回收垃圾、干垃圾、以及湿垃圾这四类，对不同类别的垃圾应采取不同分类方法，如果投放不当，可能会导致各种环境污染问题。合理地进行垃圾分类即能提高垃圾资源处理效率，也能缓解环境污染问题，因次我们在图像识别的基础上设计一个垃圾分类系统，可以高效地实现对日常生活中常见垃圾进行智能识别分类

# 2 设计概述

## 2.1 需求概述

**功能需求：**

（1）获取图像：通过电脑摄像头或者从本地打开一张图片

（2）图像显示：在屏幕上显示当前已经获取的图像

（3）识别图像：使用预先训练好的模型对图像进行识别，得到图像中垃圾的名字

（4）垃圾分类：根据识别出来的垃圾名字，使用系统内部已经标签好的数据，对垃圾名字进行分类。

（4）信息展示：展示垃圾名字，并且显示当前垃圾的种类

（5）语音播报：播报垃圾名字和种类。

**非功能需求：**

（1）性能需求：对垃圾识别分类结果准确度达到60%以上。

（2）运行需求：用户界面在用户操作后发出相应的响应。软件接口和硬件接口提供限制。

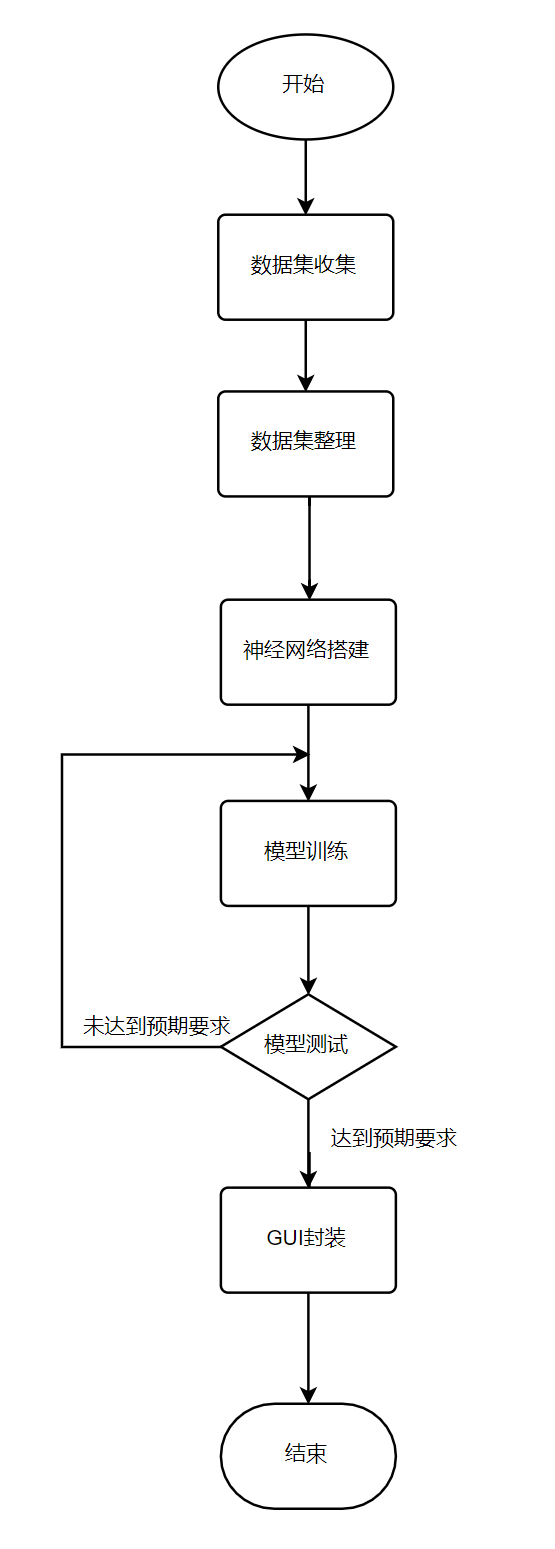
## 2.2 运行环境概述

本项目使用开源机器学习库Tensorflow 2.2.0作为训练模型、测试模型、识别图像的机器学习框架。cuda版本为9.0.1，openCV版本为4.5.4，前端界面使用pyQt5，Python使用3.8.8。

# 3 系统详细设计

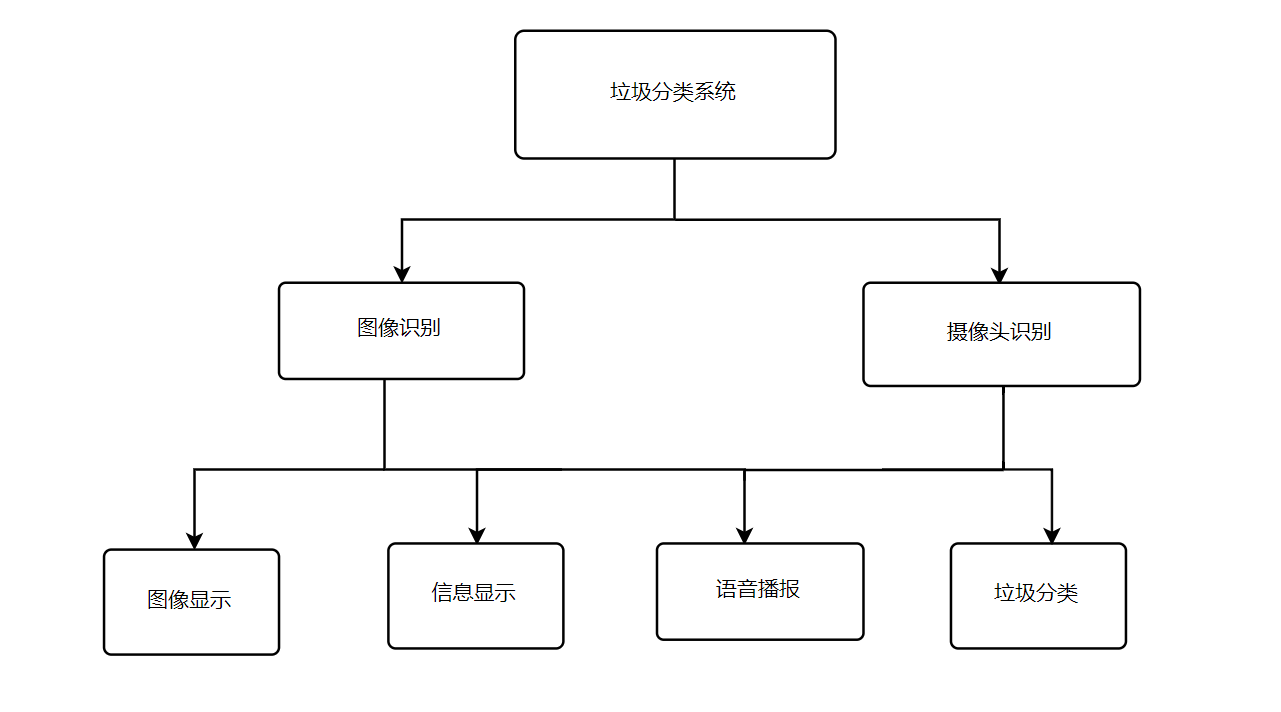
## 3.1总体设计

本设计的垃圾分类图像识别搭建在TensorFlow的环境下实现的。整个项目大体分为图像的收集、图像的处理、卷积神经网络的搭建、模型训练、模型测试、最后用一个GUI界面对项目进行封装。设计完成了一个可识别垃圾图像类别以及可调用摄像头识别垃圾物体的系统。如图3.1垃圾分类图像识别的系统构成步骤。



**图 3.1 垃圾分类图像识别的系统构成步骤**

分类是图像识别系统的中心环节，系统的最终目标是根据已训练好的模型反馈对应图像的分类类型，然后以一个直观的的图标展现给用户。系统从功能模块中可划分为两大模块，图像识别功能模块与摄像头识别功能模块。如图3.2系统的模块示意图所示:



**图3.2系统的模块示意图**

图像识别:在项目系统中用户可以使用手机拍照的垃圾图片或网上下载的垃圾图片，但凡是jpg、png格式的图片都可以传如系统对该图片进行预测得到垃圾的分类结果。

摄像头识别：若用户觉得拍照再传进系统比较麻烦，那么还可以采用直接摄像头的方法。用户可以打开摄像头，把要分类的垃圾放进摄像头拍摄区域，同样也可以得出垃圾的分类结果。注意使用摄像头进行分类预测时，最好不要让多种垃圾同时出现在摄像头中，因为系统目前还只可以在同一时间只识别一种垃圾。

垃圾分类：结果共分为四类。

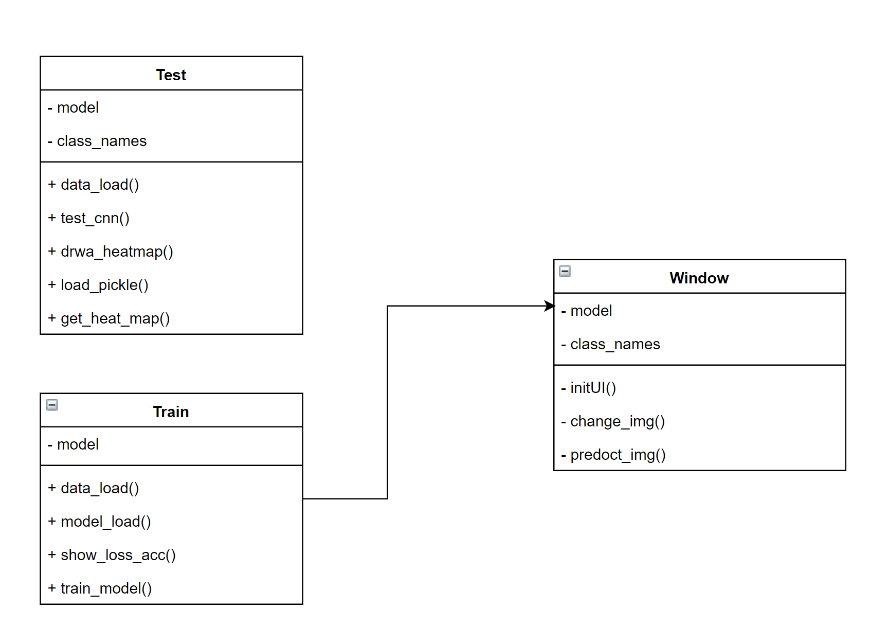
（1）可回收物：指回收后经过再加工可以成为生产原料或者经过整理可以再利用的物品，主要包括废纸类、塑料类、玻璃类、金属类、电子废弃物类、织物类等。

（2）厨余垃圾：狭义的厨余垃圾是有机垃圾的一种，分为熟厨余包括剩菜、剩饭、菜叶；生厨余垃圾包括果皮、蛋壳、茶渣、骨、贝壳；泛指家庭生活饮食中所需用的来源生料及成品(熟食)或残留物。但广义的厨余垃圾还包括用过的筷子，食品的包装材料等。

（3）有害垃圾：有毒有害垃圾是指存有对人体健康有害的重金属、有毒的物质或者对环境造成现实危害或者潜在危害的废弃物。

（4）其他垃圾：包括除上述几类垃圾之外难以回收的废弃物，通常根据垃圾特性采取焚烧或者填埋的方式处理。

图3.3是垃圾检测系统的类图。

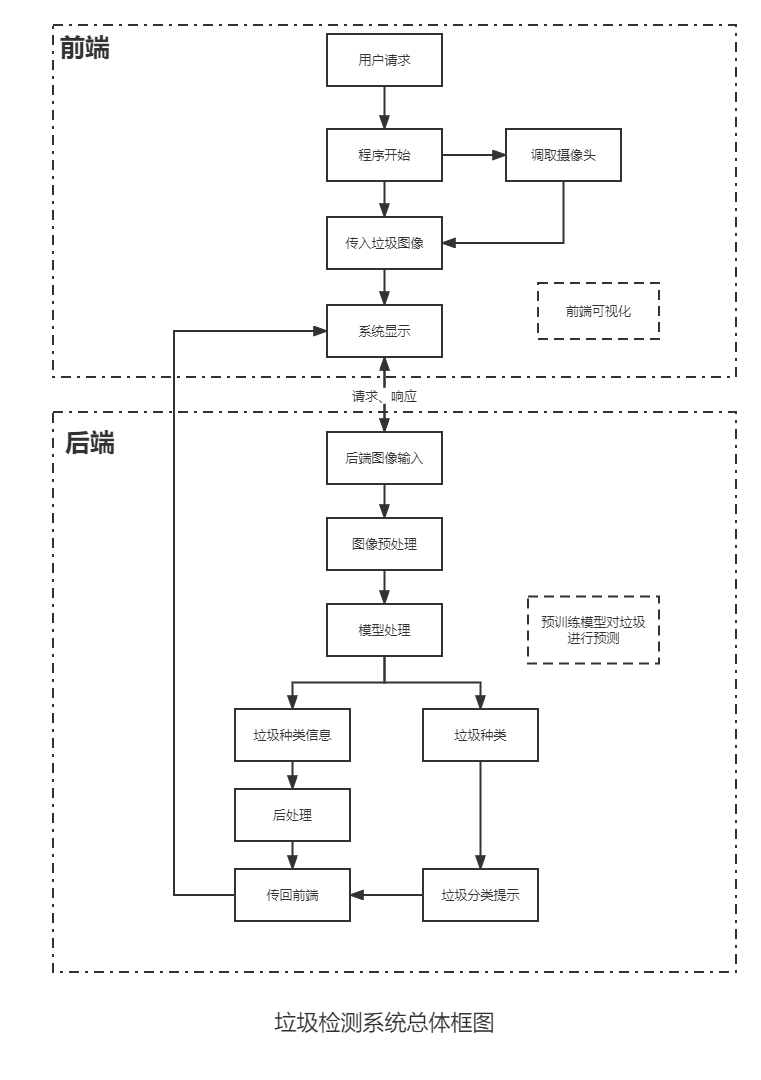


**图3.3垃圾检测系统的类图**

## 3.2 业务模块

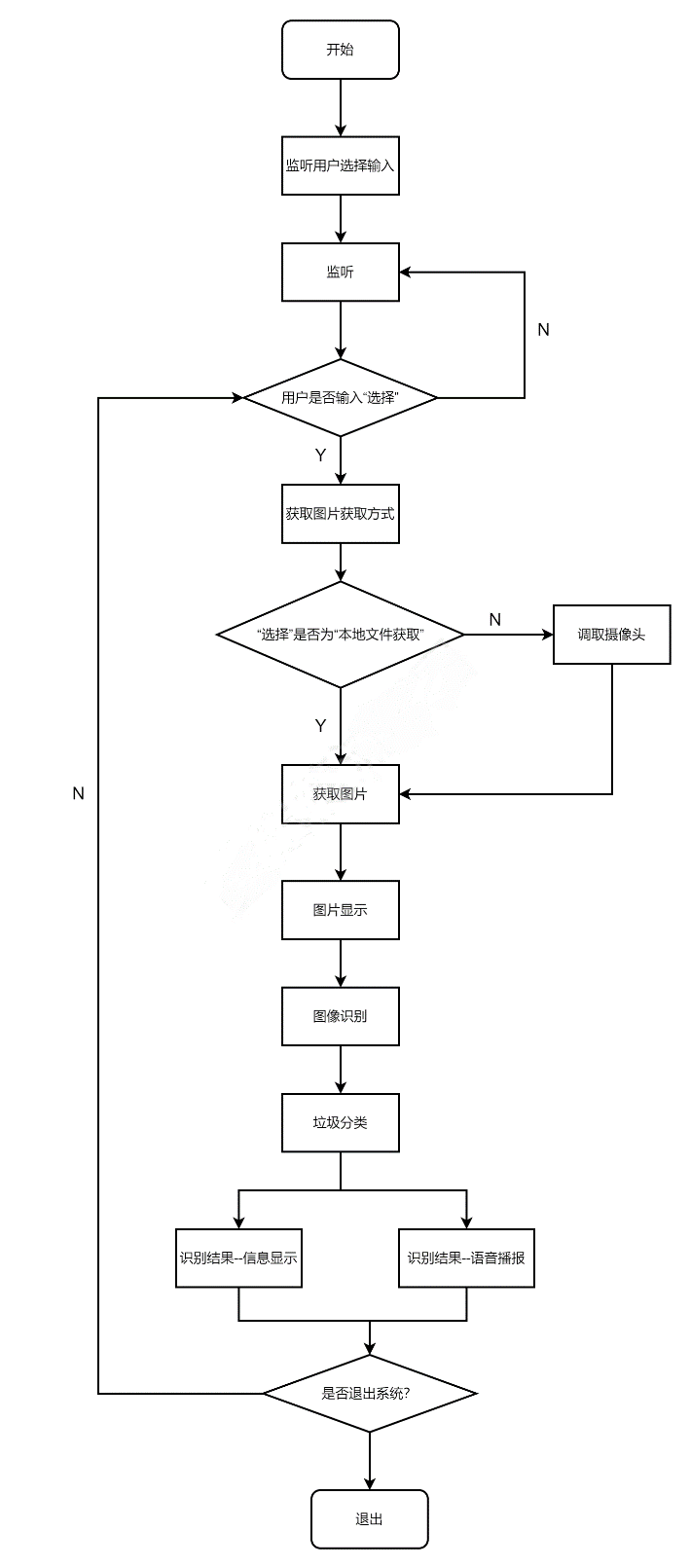
### 3.2.1 业务框图

如图3.4所示，本系统主要由用户指令、前端、后端和后台接口四部分组成。用户在发出启动本系统命令后，前端显示模块调用摄像头并将视频传递给后台处理模块。后台处理模块在对数据进行预处理后将数据输入给训练好的模型中，得出视频中垃圾的种类。得到垃圾的种类后，经过后处理将结果回传给前端显示模块用于结果展示。



**图3.4业务总体框图**

### 3.2.2 程序逻辑



**图3.5 垃圾检测业务流程图**

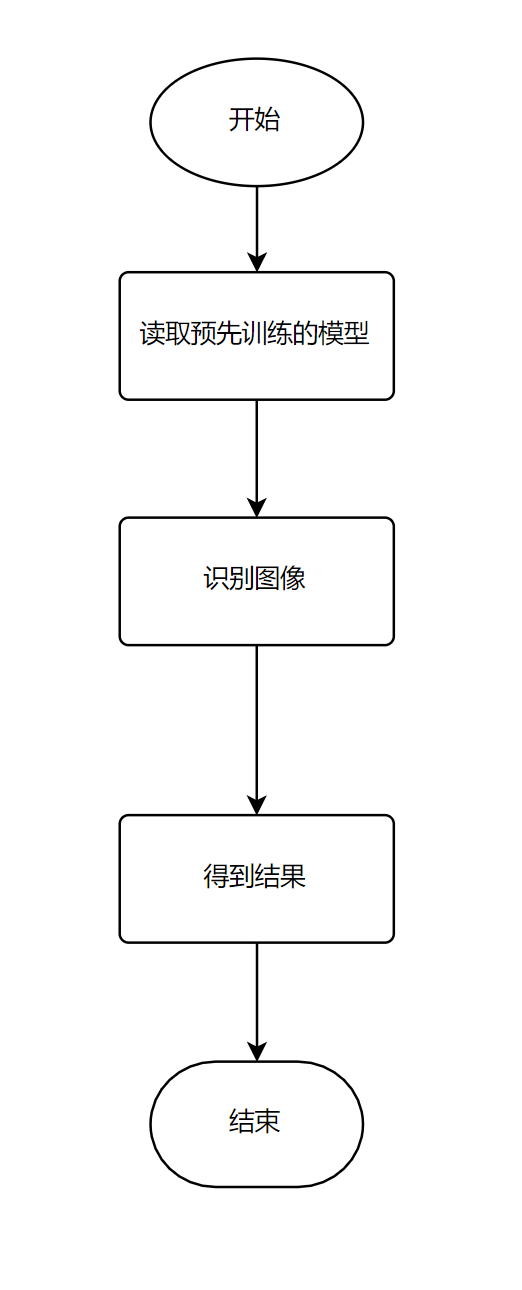
### 3.2.3 接口设计

**表3.1 业务模块接口设计表**

|  |  |  |  |
| --- | --- | --- | --- |
| **接口名** | **功能** | **参数** | **返回值** |
| InitializeEnvironment() | 初始化系统的运行环境，包括初始化硬件、加载模型、加载tensorflow、openCV运行环境 | 无 | 0成功  -1失败 |
| messageTransmitSound(message, sound) | 将文本转化为语音存放到缓存区 | message：字符串，要转化的文本  sound：Sound类型，转化完成的语音存放缓冲区 | 0成功  -1失败 |
| playSound(sound，volume) | 将一段声音进行播放 | sound：Sound类型，声音缓冲区  volume：整形， 播放的音量，0表示无声音，1--99 | 0成功  -1失败 |

## 3.3 算法模块

### 3.3.1 程序逻辑



**图3.6 算法模块流程图**

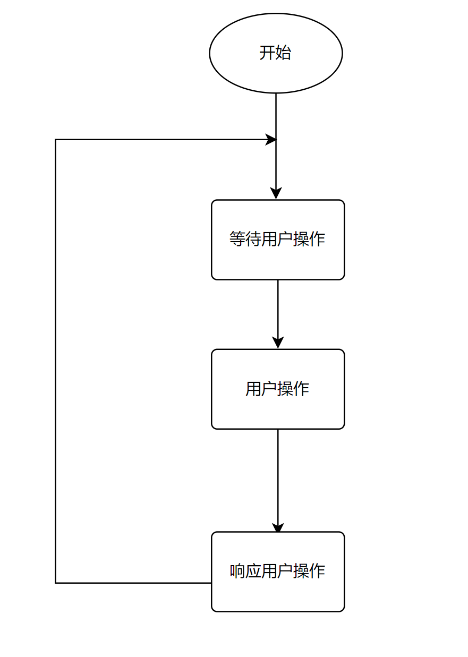
### 3.3.2 接口设计

**表3.2 算法模块接口设计表**

|  |  |  |  |
| --- | --- | --- | --- |
| **接口名** | **功能** | **参数** | **返回值** |
| trainModel(model) | 训练模型 | model，模型 | 损失函数损失率 |
| identifyImageInformation(image,message) | 对预处理过的图像进行识别 | image：Image类型,待识别的图像  message：字符串,图像中垃圾的信息 | 0成功  -1失败 |
| testModel(model) | 测试模型 | model，模型 | 模型正确率 |
| showModel(model) | 显示模型信息 | model，模型 | 无 |

## 3.4 前端模块

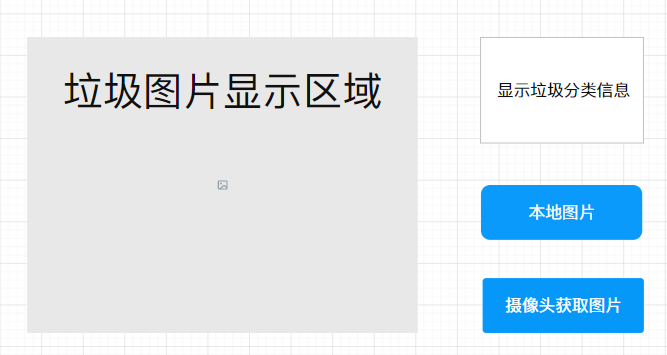
### 3.4.1 程序逻辑



**图3.7 前端模块流程图**

### 3.4.2 界面设计

本文编写的垃圾识别分类系统操作界面如图3.8所示，软件包含两个按钮，分别为打开摄像头开关、输入本地图片；软件可以显示输入的图片，并将检测结果显示在软件界面上。



**图3.8 前端设计展示图**

### 3.4.3 接口设计

**表3.3 前端模块接口设计表**

|  |  |  |  |
| --- | --- | --- | --- |
| **接口名** | **功能** | **参数** | **返回值** |
| GetImageFormCamera() | 从摄像头获取一张图片 | 无 | Image类型，从摄像头得到一个图像 |
| getImageFormFile(path) | 从本地获取一张图片 | path图像所在路径 | Image类型，返回一个图像 |
| displayMessage2Screen(message) | 将信息显示到屏幕上 | message：字符串 要显示的字符串 | 0成功  -1失败 |
| displayImage2Screen(image) | 将图像显示到屏幕上 | image：Image类型,要显示的字符串 | 0成功  -1失败 |
| openCamera() | 打开摄像头 | 无 | 0成功  -1失败 |

# 4 风险分析

**预计可能出现的风险：**

（1）开发人员水平不足，无法开发出预定的功能。

（2）开发人员之间配合出现问题，导致功能实现出现差错。

（3）管理人员与开发人员之间对项目理解出现偏差、沟通不畅、配合出现问题。

（4）开发任务繁重，时间紧张，没有足够的测试，项目出现漏洞。

**相应规避办法：**

（1）开发人员、管理人员加强沟通，对理解偏差的任务持续沟通。

（2）开发文档制作详细，定期对开发人员进行学习培训，提高开发人员开发水平。

（3）对项目进行详细、反复的测试，尽量减少项目漏洞。