

软件需求分析

可行性分析

技术可行性

该项目的几个模块分别为拍照上传，本地图片上传以及进行图片的处理

拍照上传

通过微信小程序开放平台 `wx.chooseMedia(Object object)` 实现拍摄图片上传

实例：

```
wx.chooseMedia({
  count: 9,
  mediaType: ['image'],
  sourceType: ['camera'],
  maxDuration: 30,
  camera: 'back',
  success(res) {
    console.log(res.tempFiles.tempFilePath)
    console.log(res.tempFiles.size)
  }
})
```

本地上传

通过微信小程序开放平台 `wx.chooseMedia(Object object)` 实现本地图片上传

实例：

```
wx.chooseMedia({
  count: 9,
  mediaType: ['image'],
  sourceType: ['album'],
  maxDuration: 30,
  camera: 'back',
  success(res) {
    console.log(res.tempFiles.tempFilePath)
    console.log(res.tempFiles.size)
  }
})
```

图片处理

通过CNN实现，已经通过初步测试，可以实现功能。

市场可行性

根据调查问卷显示，垃圾分类推广目前主要的问题是人们无法准确的把垃圾进行分类，给人们带来一定的困扰，从而影响群众积极性。目前市场上有一个迫切的需求：通过简单的拍照即可辅助人们实现垃圾分类。

功能性需求分析

用户登录

用户登录，如果是微信注册用户，自动获取微信账号的个人信息进行登录，如果是手机号码注册，则通过手机号码和验证码登录。

图像获取

用户对想要查询的垃圾进行拍照搜索，搜索后系统会给予用户相应的信息提示。本功能借助手机终端调用设备摄像头采集区域内的图像，则会开启图像识别功能。

本地相册上传图像

用户可以提前使用手机拍摄无法确定分类的垃圾的图像，存储在手机相册内。选择“图片上传”功能，系统申请访问相册的权限，之后直接对用户相册所选的照片进行类别分析。

拍照上传

用户直接选择“拍照识别”功能，调用手机设备摄像头，现场对未分类的垃圾拍照，然后由用户确定是否上传刚刚所拍照片，待上传后，由系统进行垃圾类别分析。

图像识别

用户点击“图像识别”功能，系统开启设备摄像头。该功能持续获取镜头所对区域的图像并进行实时分析，与前述功能不同点在于该功能是对动态变化的图像中的垃圾进行分析，而非静止的照片。与本功能类似的是人脸识别的功能。

类型识别

部分城市对于垃圾分类较为严格，为了避免用户分类错误造成罚款，系统提供垃圾分类功能，用户只需选取合适的图像获取方式将图像呈现给系统，即可对其进行分类。

类别

类别	特点	内容
可回收物	有利用价值、清洁干燥	纸类、金属类、塑料、玻璃
有害垃圾	危害人体健康和自然环境	废电池、废日光灯管、废水银温度计、过期药品
湿垃圾	厨余垃圾居多	剩菜剩饭、瓜皮、果核、菜叶碎骨、过期食品、湿纸巾
干垃圾	危害小、无利用价值	餐盒、毛发、尘土、带胶制品等

识别反馈

用户使用系统识别过后，由系统返回本次分析的结果，包括：**垃圾类别、正确处理方式**

非功能性需求分析

UI界面优化

UI界面在用户体验中占据很大一部分，在首版本开发过程中，允许以实现基本功能为目的的UI界面开发，但后续需要对UI界面进行用户体验层面的优化。例如优化过渡动画的流畅度、UI界面配色的合理性等。

用户操作逻辑优化

软件操作要符合人类的一般逻辑，要避免反直觉的操作流程。例如拍照或选择图片后要有确认提示，而不是直接上传，要给用户留有操作的余地。诸如此类的操作逻辑优化在首版本开发过程中就必须考虑。

扩展性分析

垃圾类别扩展

由于各地的垃圾分类政策有所区别，且随着时间的变化，政策也可能随之变化。故在开发过程中，要注意数据结构的合理性，为日后垃圾政策变动或垃圾类别变动留有扩展的余地。

识别源扩展

要为除图片以外的识别源留有扩展余地，方便日后加入语音识别，文本识别等功能。