**Android 软件开发报告**

**小组成员：**

**21180605张佳玲**

**21180609蔡睿欣**

**21180607闻世宇**

**21180502刘若冰**

**211805袁梓宸**

**二零二一年十一月**

**一、组员贡献和比例**

张佳玲：工程的主要功能实现与完善35%

蔡睿欣：实验报告与汇报20%

闻世宇：答辩演示材料15%

刘若冰：资料与参考15%

袁梓宸：资料与参考15%

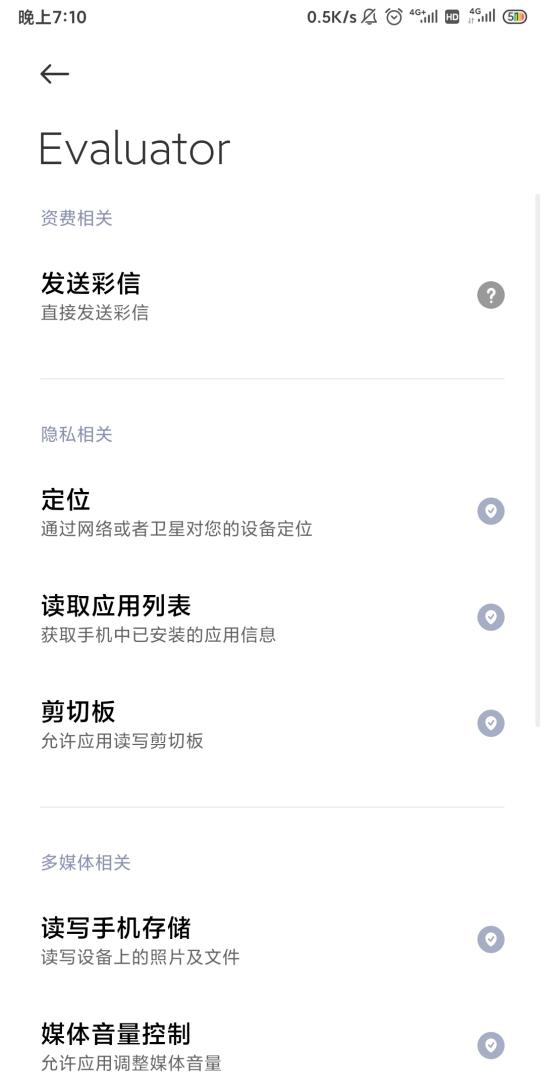
1. **使用说明**

**1.apk安装**

在工程中实现功能后导出了apk文件，可以将软件安装至安卓系统的手机或平板设备中，要求设备本身要有具有裁剪图像功能的软件，便于在软件中调用辅助。



烟雾污染检测软件-图标



烟雾污染检测软件-应用权限

**2.开始界面-软件内图库Gallery**

下左图为软件开始界面，左上角“Gallery”字样提示使用者此界面展示的是已经导入到软件中的可分析的图片，初始为空。下右图为单击最下方的摄像头按钮后，将弹出菜单询问使用哪种方式进行图片的导入，使用者根据需求可以进行选择。

**3.导入图片**

使用者拍摄图片或从系统相册/文件夹中选择图片后，软件会调用系统默认的相册软件，等待使用者手动将图片裁剪到需要进行识别的区域后才可以保存至软件，图片未经裁剪无法向后进行。

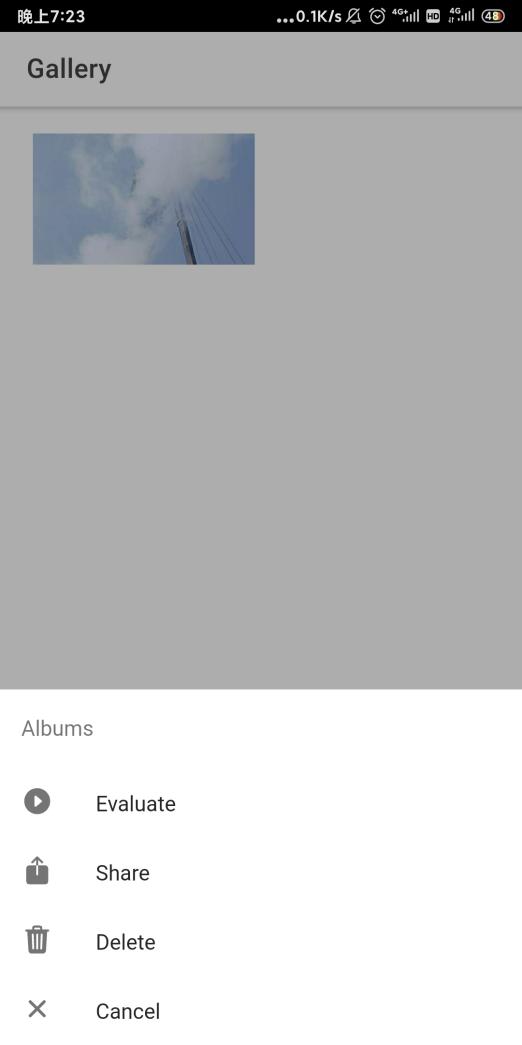


**4.图片操作**

裁剪后的图片会展示在开始界面中。



单击图片会弹出菜单询问要对图片进行何种操作：



Evaluate-计算其黑度值；

Share-将图片分享至其他app；

Delete：将图片从软件中删除；

Cancel：不进行任何操作，关闭菜单。

**三、功能简介**

**1.Photo**

拍照，要求能够调用摄像头，并把拍摄的照片存储到指定文件夹。

**2.File**

读取文件，要求可以进入手机的存储空间，读取任意文件中的图片。

**3.Calculation**

计算，图片加载到APP中，可以手动选择烟雾位置，并计算其黑度值。

**4. Share**

共享，将导入软件的图片生成链接共享到QQ，朋友圈邮件等APP。

**四、实现思路**

**1.各部分所用的主要工具**（实际中有交叉）



**2.具体设计思路**

1）View

使用Ionic脚手架生成了一个具有三个Tab页的基础项目框架，针对项目第一大拍摄照片需求，将其中一个Tab用作相册展示区域并在屏幕下方添加拍摄按钮，其他两个Tab保留原始页面架构和路由信息的基础上更改为弹出选项框和烟雾浓度结果展示页面。

页面样式和切换主要通过Vue、VueRouter和Ionic组件库搭建。

其中样式部分选择Ionic而没有选择其他UI库是因为Capacitor基本上与Ionic捆绑使用（同一家公司的产品+官方推荐），而此项目最终结果呈现需要Android APK，像拍摄分享等安卓手机的原生功能如果不使用Android SDK编写的话，需要Capacitor这样的手机平台runtime，它把安卓平台特定的功能再次打包封装成WEB开发者可以通过Javascript调用的插件。

2）Model

Vue框架本身就是对MVC的一种实现，所以一开始对数据的操作直接用了Vue的Data，但是也因此被Vue自带的数据处理方面的劣势限制住，比如说父子组件之间的数据传递非常麻烦，而且默认是单向传递，如果要实现子组件数据变化父组件自改变需要写很多重复的代码。因此后来又去了解并使用了Vuex，是Vue团队推出的专门用于项目内数据共享的方法，有点全局变量的意味在里面，如果多个组件同时对一部分数据产生影响，那么把这些数据同一放在store（一般是单个的js文件）。

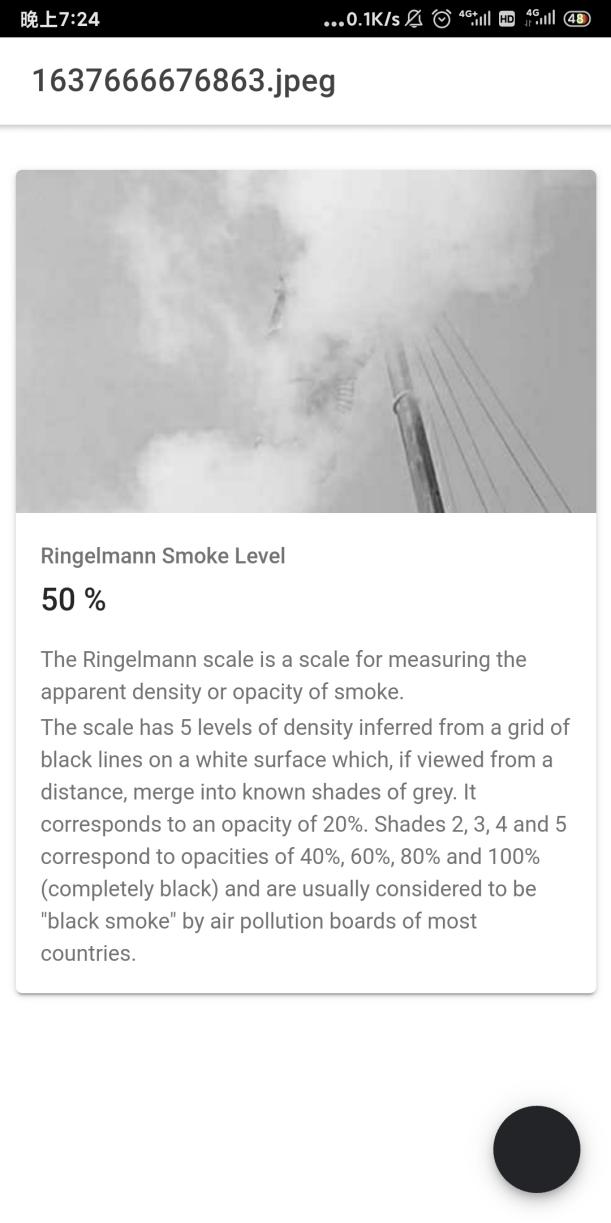
3）Controller

同样使用了Vue框架的实现，但把各个功能单独提取到js文件做了一个微小的模块化处理。其中：

* 拍照：直接调用@capacitor/camera，可以在options中指定拍照参数，拍摄结束把照片信息返回并插入在store中的imgs数组。
* 删除：将选中的img从store中保存的imgs数组里删除。
* 分享：调用了@capacitor/share和@capacitor/geolocation，想要在分享的时候加入当前位置信息。
* 烟雾浓度判定：导入了“image-js”这个图像处理库，对选中的img的webPath信息做一步转化，转化成库中的Image类型，再对该图片对象灰度化处理，提取像素总数和颜色总数求出平均值，得到图像大体上的灰度百分比作为格林曼灰度测定结果。

**五、结果展示**

对所导入的图片进行分析操作后会生成对图中烟雾进行的林格曼黑度分析结果。



**六、遇到的问题**

**1.照片读写**

前文设计思路中提及，在软件内使用vuex来保存信息，如果仅论网页端这样数据处理基本上没有问题，但最终app需要能够在安卓上进行照片存储和访问，需要额外在软件外进行文件读写操作，但由于Capacitor的官方文档对于文件操作部分的api介绍比较简略，没有示例代码，自己使用该api没有办法在网页端进行调试（需要安卓环境才能检测效果）等等诸多原因没有实现在安卓端的照片读写。

**2.代码调试**

90%的时间都是在用网页开发者工具中的响应式视图中对代码开发结果进行观测，但其实很多安卓原生功能即使代码成功实现了但是网页中也无法展示，在没有意识到这一点之前花费了很多时间在没有必要的代码修改上，而且即使在后续Capacitor的文档阅读中获悉了这一点，也还是导致开发时间成倍增长，因为网页上某个功能没有出效果而必须再用安卓模拟器跑一遍才能知道代码逻辑有没有写对。Ionic-cli虽然官方问档给出了直接跑通android的命令但多次实验没有效果。

**七、心得与展望**

很多时候最好使用平台推荐的开发方式去完成特定平台的应用。一开始直接排除掉了使用Android Studio开发，主要是想规避自己重新去学Kotlin或者Java以及相应的框架，但其实使用跨平台的开发方式没有想象中的方便，比喻来说就好像本来一部车能到的地方我偏偏选择地铁9号线转轻轨再换11路，如果之后有类似安卓平台的开发需求我会去尝试Kotlin。