

# 武汉大学计算机学院

## 本科生课程设计报告

### 基于 Android 平台的植物识别应用 「寻花」总体设计与实现

专 业 名 称     :    计算机科学与技术

课 程 名 称     :    嵌入式系统实验设计

指 导 教 师     :    刘树波 教授

团 队 成 员 一:    朱卉萱 (2017302180267)

团 队 成 员 二:    夏雪琴 (2017301500172)

二〇二〇年六月

# 郑 重 声 明

本团队呈交的设计报告，是在指导老师的指导下，独立进行实验工作所取得的成果，所有数据、图片资料真实可靠。尽我所知，除文中已经注明引用的内容外，本设计报告不包含他人享有著作权的内容。对本设计报告做出贡献的其他个人和集体，均已在文中以明确的方式标明。本设计报告的知识产权归属于培养单位。

团队成员签名：\_\_\_\_\_

日期：\_\_\_\_\_

# 目 录

<b>1 实验目的和问题描述 .....</b>	<b>1</b>
1.1 实验目的.....	1
1.2 软件描述.....	1
1.3 功能设计.....	2
1.3.1 界面相关.....	2
1.3.2 功能相关.....	2
<b>2 实验环境 .....</b>	<b>2</b>
<b>3 程序结构及算法实现 .....</b>	<b>3</b>
3.1 总体方案.....	3
3.1.1 项目总体设计.....	4
3.1.2 项目目录结构.....	4
3.2 界面 UI 设计 .....	7
3.2.1 欢迎界面设计.....	7
3.2.2 相机拍摄界面设计.....	8
3.2.3 结果显示页面设计.....	9
3.2.4 app 图标设计 .....	14
3.3 具体功能实现.....	14
3.3.1 界面切换.....	16
3.3.2 调用系统相机.....	16
3.3.3 从相册进行选择.....	17
3.3.4 图片裁剪.....	18
3.3.5 退出应用.....	18
3.3.6 调用图片识别 API .....	19
3.3.7 识别结果显示.....	21
3.3.8 返回并删除临时图片.....	22
<b>4 结论.....</b>	<b>23</b>
4.1 实验数据.....	23
4.1.1 欢迎界面.....	23

4.1.2 功能选择界面.....	24
4.1.3 拍摄与裁剪界面.....	24
4.1.5 结果显示界面.....	25
4.2 问题与解决方案.....	26
<b>附录.....</b>	<b>27</b>
成员贡献表.....	27
个人总结.....	27
朱卉萱.....	27
夏雪琴.....	28

# 1 实验目的和问题描述

## 1.1 实验目的

通过面向对象的分析与设计，实现一款安卓系统下 App 软件的开发。实验旨在提升同学的综合能力和编程经验，加强对于嵌入式开发的认识。

本实验中的具体目标是开发一款 Android 平台下的植物识别 App。

## 1.2 软件描述

人并不是万能的，我们所知道或认识的东西远远比我们自已自以为所了解的要少得多。据统计，全世界大概有四十几万种花，但是你认识或能叫得出名字的又有几种呢？基于此，团队设计了安卓平台下的“寻花”手机应用程序。本应用旨在减少用户生活中面对植物“相对不识”尴尬场面的出现。

“寻花”是一款帮助用户识别植物的手机应用，使用 Android Studio 平台进行开发。它通过调用百度云的智能花卉识别接口，实现了对用户拍照获取的图片或者从现有图库中选取的相片进行识别的功能。识别结果将会以友好的界面向用户反馈。

## **1.3 功能设计**

### **1.3.1 界面相关**

- (1) 欢迎界面
- (2) 相机主界面
- (3) 识别结果显示界面

### **1.3.2 功能相关**

- (1) 调用本地相机接口获取图片
- (2) 调用本地相册访问接口选择图片
- (3) 图片的裁剪与显示
- (4) 调用网络接口
- (5) 接口返回结果的展示

## **2 实验环境**

操作系统：Windows10

代码语言：Java

开发平台：Android Studio 3.5.2

调试与演示平台：Huawei P10 - Android 9.0

界面设计工具：Adobe Photoshop CC2019

## 3 程序结构及算法实现

### 3.1 总体方案

分析问题可知，本软件实现需要有方便的界面构建以及程序模拟，故本实验采用 Android Studio 进行开发。与 Eclipse 相比，Android Studio 在安卓开发方面功能更强大，提供了集成的 Android 开发工具用于开发和调试。

下面是 Android Studio 下的系统架构图。

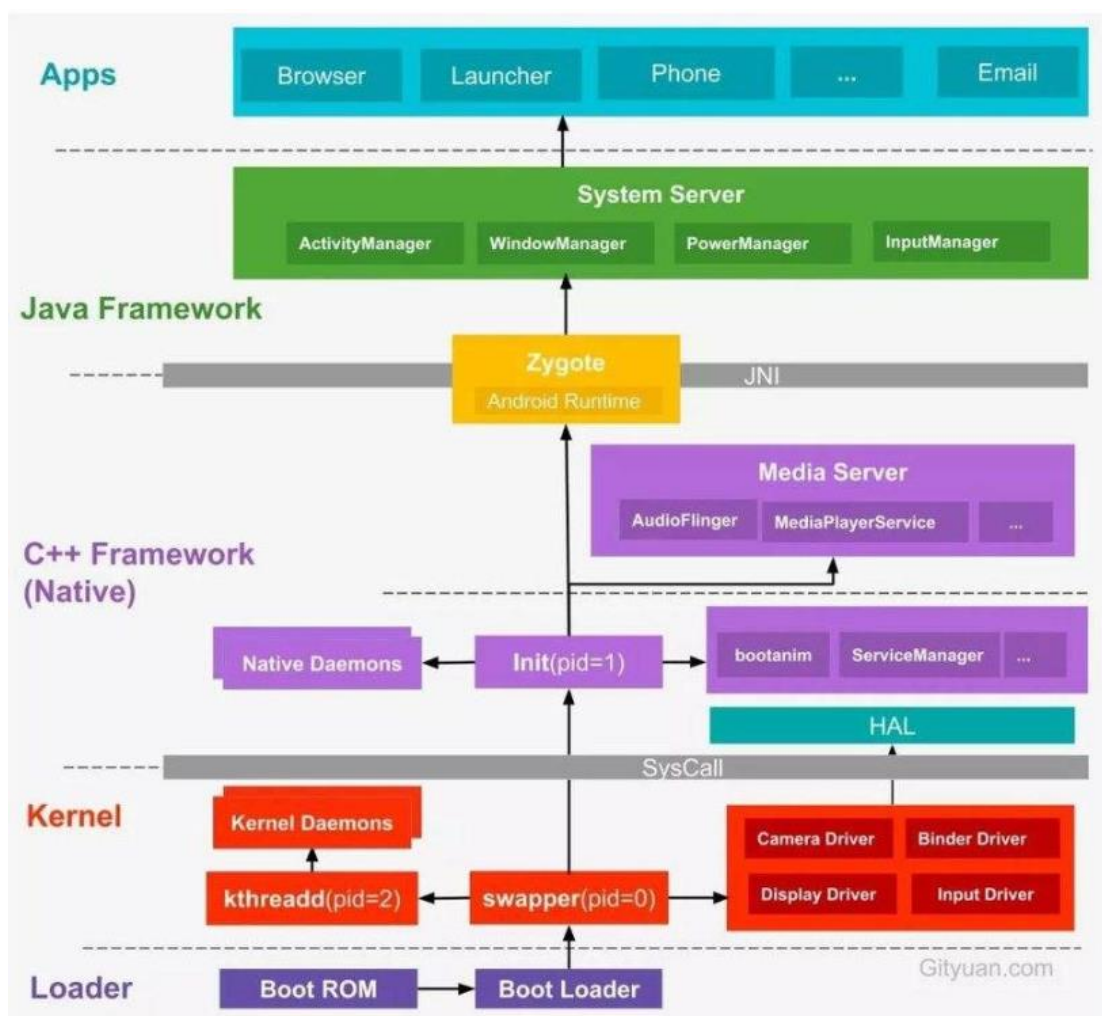


图 3-1-1 系统架构图

3.1.1 项目总体设计

根据项目需求，做出本项目状态图如下：

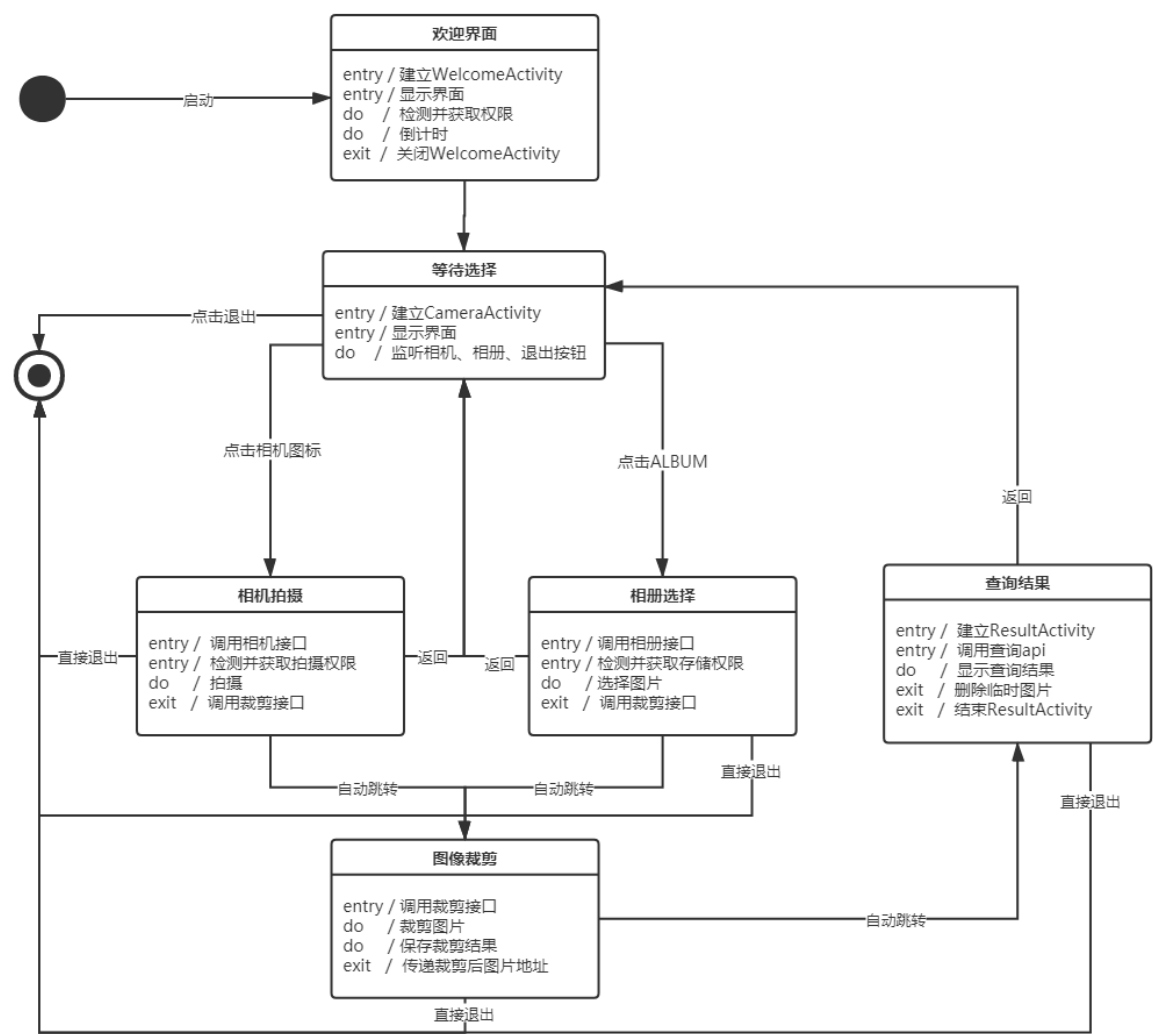


图 3-1-1-1 项目状态转移图

3.1.2 项目目录结构

根据状态转移图，构建本项目基本目录结构。Android Studio 下项目目录文件较多，此处着重介绍本团队开发过程中需自行修改的文件。下两图为项目目录结构图与主要代码文件结构。



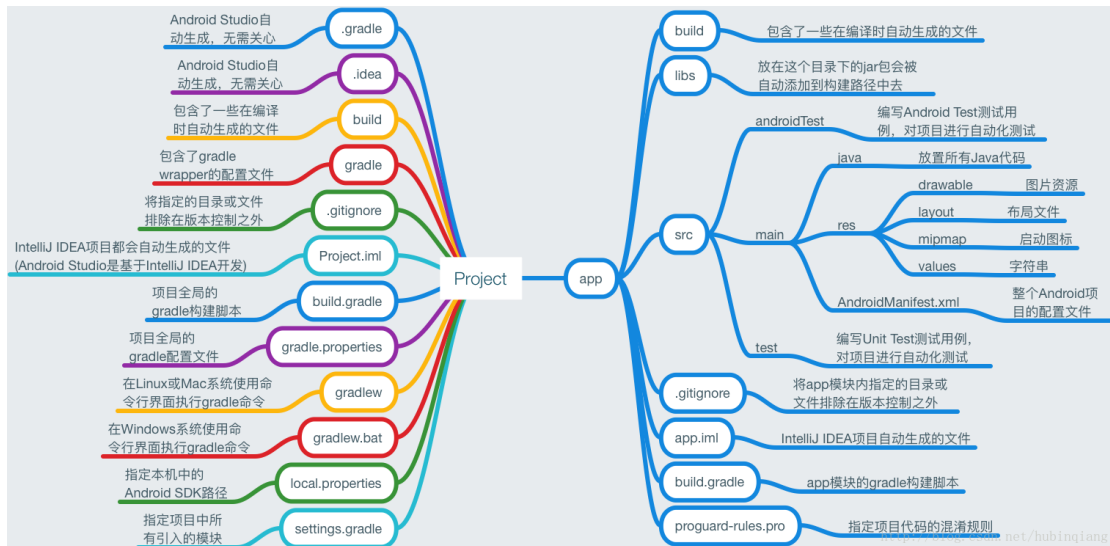


图 3-1-2-1 项目目录结构图

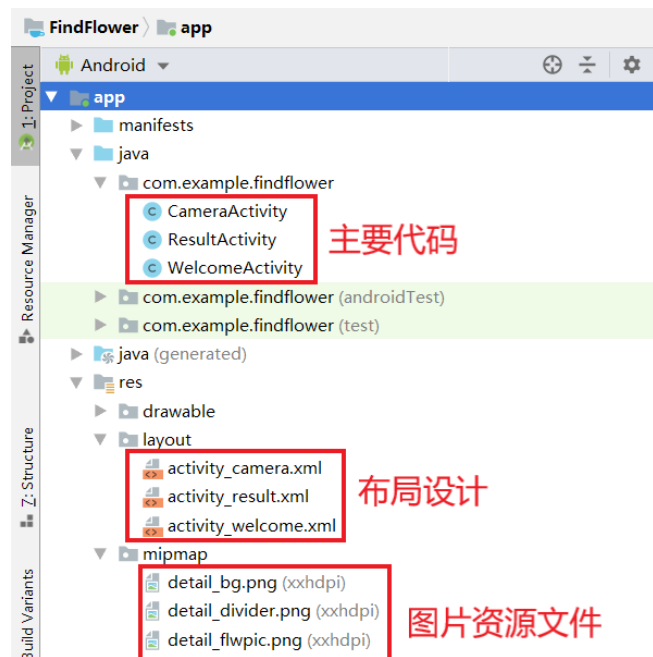


图 3-1-2-2 主要代码文件结构

## 1. main/java

此目录下面放置的是所有与项目相关的 Java 代码。项目中根据项目的功能模块创建相对应的 Java 类。本实验中根据项目需要，分别创建了 WelcomeActivity（欢迎页活动代码）、CameraActivity（相机拍摄页活动代码）、ResultActivity（结果展示页活动代码）

## 2. layout 布局文件

布局文件与界面显示相关，在.xml 文件中对界面设计进行操作。实验中与活动代码对应，分别建立了 activity\_welcome（欢迎页界面代码）、activity\_camera（相机拍摄页界面代码）、activity\_result（结果展示页界面代码）

对于.xml 文件，可以使用编译器内置的 Design 模式通过拖拉控件及属性选择进行界面设计并自动生成代码；或选择 Split 模式手动编写代码，并实时预览界面效果。

## 3. AndroidManifest 文件

该文件包含了项目的配置信息。其中保存了项目的部分基本信息，并具有决定项目入口，标记项目所需权限等功能。

## 4. 资源文件

项目中用到的资源文件主要为图片文件。为了满足目前主流安卓机型的显示，将文件统一放置在 mipmap/xxhdpi 文件夹下。资源文件列表如下：

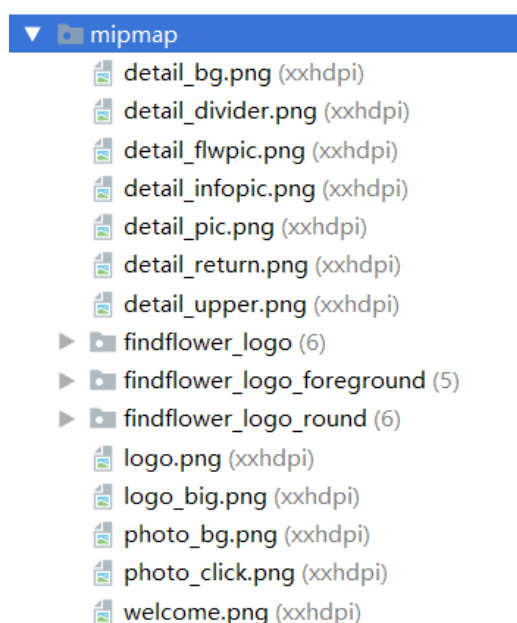


图 3-1-2-3 资源文件目录

## 3.2 界面 UI 设计

一款 App 的 UI 设计直接影响用户的第一体验，故 UI 设计一定要美观且便于操作。本项目采用了简约化、扁平化的设计风格，在满足用户需求的同时，带给用户愉悦的使用体验。界面展示大小为  $1080 \times 1920\text{px}$ 。界面设计中，要注意尽量多使用相对约束关系，减少使用绝对数值，以达到多机型兼容显示的目标。

### 3.2.1 欢迎界面设计

欢迎界面向用户展示应用 logo，三秒后自动跳转至拍照功能页面。

以下为欢迎界面的 layout 布局文件代码：

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@mipmap/welcome"
    tools:context=".WelcomeActivity"/>
```

效果展示：



图 3-2-1-1 欢迎界面设计图

### 3.2.2 相机拍摄界面设计

相机拍摄界面是项目的主界面，在欢迎界面计时结束后自动出现。在这个界面中，用户可以选择相机进行拍照获取图片，也可以选择 ALBUM 在相册里面进行图片选择，还可以选择退出按钮，退出应用程序。以下为此界面的布局文件代码：

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@mipmap/photo_bg"
    tools:context=".CameraActivity">

    <Button
        android:id="@+id/startCameraButton"
        style="@style/Widget.AppCompat.Button.Borderless"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:background="@mipmap/photo_click"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.52"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/choiceFromAlbumButton"
        style="@style/Widget.AppCompat.Button.Borderless"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="ALBUM"
        android:textColor="#2F4F4F"
        android:textSize="18sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/startCameraButton"
        app:layout_constraintVertical_bias="0.0" />

    <Button
        android:id="@+id/exitAppButton"
        style="@style/Widget.AppCompat.Button.Borderless"
        android:layout_width="90dp"
        android:layout_height="62dp"
        android:text="退出"
        android:textColor="#2F4F4F"
        android:textSize="18sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

效果展示：



图 3-2-2-1 相机拍摄界面设计图

### 3.2.3 结果显示页面设计

结果显示界面用于向用户展示用户拍摄图片的查询结果。界面内容包括展示用户拍摄的图片、植物名、判断准确度、植物简介以及植物示例图片。界面主要采用可滚动的纵向线性布局，嵌套约束布局及横向线性布局以达到放置各种组件的效果。

以下为此页面布局代码：

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@mipmap/detail_bg"
    tools:context=".ResultActivity">
```

<ScrollView

```
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent">
```

<LinearLayout

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:orientation="vertical">
```

<androidx.constraintlayout.widget.ConstraintLayout

```
    android:id="@+id/constraintLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
```

<ImageView

```
    android:id="@+id/upper_bg"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:contentDescription="0"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@mipmap/detail_upper" />
```

<Button

```
    android:id="@+id/rtn_button"
    android:layout_width="33dp"
    android:layout_height="22dp"
    android:layout_marginStart="16dp"
    android:layout_marginLeft="16dp"
    android:layout_marginTop="16dp"
    android:background="@mipmap/detail_return"
    android:contentDescription="0"
    app:layout_constraintStart_toStartOf="@+id/upper_bg"
    app:layout_constraintTop_toTopOf="@+id/upper_bg"
    app:srcCompat="@mipmap/detail_return" />
```

<ImageView

```
    android:id="@+id/pic_bg"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="#00FFFFFF"
    android:contentDescription="0"
    app:layout_constraintBottom_toBottomOf="@+id/upper_bg"
    app:layout_constraintEnd_toEndOf="@+id/upper_bg"
    app:layout_constraintStart_toStartOf="@+id/upper_bg"
    app:layout_constraintTop_toTopOf="@+id/upper_bg"
    app:layout_constraintVertical_bias="0.35"
    app:srcCompat="@mipmap/detail_pic" />
```

```

<ImageView
    android:id="@+id/flw_pic"
    android:layout_width="wrap_content"
    android:layout_height="0dp"
    android:contentDescription="0"
    app:layout_constraintBottom_toBottomOf="@+id/pic_bg"
    app:layout_constraintEnd_toEndOf="@+id/pic_bg"
    app:layout_constraintStart_toStartOf="@+id/pic_bg"
    app:layout_constraintTop_toTopOf="@+id/pic_bg"
    tools:srcCompat="@mipmap/detail_flwpic" />

<TextView
    android:id="@+id/name_ch"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:contentDescription="0"
    android:text="花名"
    android:textColor="#FFFFFF"
    android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="@+id/upper_bg"
    app:layout_constraintEnd_toEndOf="@+id/upper_bg"
    app:layout_constraintStart_toStartOf="@+id/upper_bg"
    app:layout_constraintTop_toBottomOf="@+id/pic_bg"
    app:layout_constraintVertical_bias="0.10000024" />

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="@+id/upper_bg"
    app:layout_constraintStart_toStartOf="@+id/upper_bg"
    app:layout_constraintTop_toBottomOf="@+id/name_ch"
    app:layout_constraintVertical_bias="0.0">

    <TextView
        android:id="@+id/score"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="准确度"
        android:textColor="#FFFFFF"
        android:textSize="18sp" />

    <Space
        android:layout_width="10dp"
        android:layout_height="match_parent"
        android:layout_weight="4" />

```

```

<TextView
    android:id="@+id/score_txt"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="0"
    android:textColor="#FFFFFF"
    android:textSize="18sp" />

<Space
    android:layout_width="10dp"
    android:layout_height="match_parent"
    android:layout_weight="4" />

<TextView
    android:id="@+id/textView5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="%"
    android:textColor="#FFFFFF"
    android:textSize="18sp" />
</LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

```

```

<TextView
    android:id="@+id/description"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:text="简介"
    android:textSize="18sp"
    android:textStyle="bold" />

<ImageView
    android:id="@+id/divider1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:srcCompat="@mipmap/detail_divider" />

<TextView
    android:id="@+id/description_txt"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:maxEms="14"
    android:text="Oops, 搜索结果暂时没有简介呢"
    android:textSize="18sp" />

<ImageView
    android:id="@+id/divider2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:srcCompat="@mipmap/detail_divider" />

```



```

<TextView
    android:id="@+id/info_pic_txt"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:text="参考图片"
    android:textSize="18sp"
    android:textStyle="bold" />

<ImageView
    android:id="@+id/info_pic"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:contentDescription="name of the flower"
    app:srcCompat="@mipmap/detail_infopic"
    tools:layout_editor_absoluteX="72dp"
    tools:layout_editor_absoluteY="229dp" />

</LinearLayout>
</ScrollView>
</androidx.constraintlayout.widget.ConstraintLayout>

```

效果展示:



图 3-2-3-1 结果展示界面设计图

### 3.2.4 app 图标设计



图 3-2-4-1 app 图标设计图

## 3.3 具体功能实现

这款应用软件所包含的功能有界面切换，功能选择界面的调用系统相机，从图库进行选择，退出 App,以及调用百度云的智能花卉识别接口对图片进行识别，以及识别之后的结果显示（图片显示和文字介绍）。

在对这些功能的实现过程中，我们常用到 Android 里面的 Intent。Intent 是 Android 非常常用的一个用于组件间互相通信的信息对象，常用于启动组件和传递数据。Intent 不仅可用于应用程序之间，也可用于应用程序内部的 activity, service 和 broadcast receiver 之间的交互。Intent 是一种运行时绑定（runtime binding)机制，它能在程序运行的过程中连接两个不同的组件。通过 Intent，我们的程序可以向 Android 表达某种请求或者意愿，Android 会根据意愿的内容选择适当的组件来响应。activity、service 和 broadcast receiver 之间是通过 Intent 进行通信的。如下图所示：

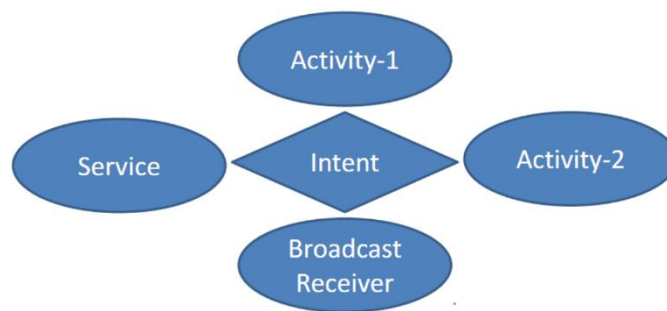


图 3-3-1 Intent 功能图

Intent 发出的时候，系统对应的行为正是由 Intent 所包含信息的组合决定。一个 Intent 所包含的信息如下图：

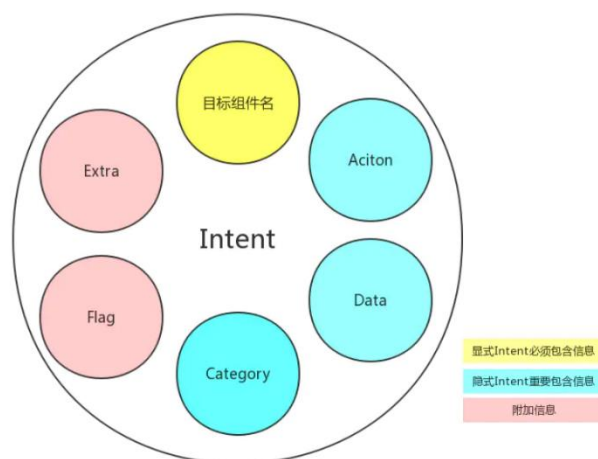


图 3-3-2 Intent 包含信息图

### 3.3.1 界面切换

通过开启倒计时线程，使程序休眠，以达到展示欢迎界面的目的。休眠结束后，启动主界面活动，并关闭当前界面活动。下面是相关的代码：

*//开启倒计时线程，实现启动界面自动跳转*

```
Thread myThread=new Thread() {//创建子线程

    @Override
    public void run() {
        try{
            sleep(3000);//使程序休眠三秒
            Intent it=new Intent(getApplicationContext(),CameraActivity.class);
            startActivity(it);//启动 CameraActivity
            WelcomeActivity.this.finish();//关闭当前活动，保证之后不会 home 返回
        }catch (Exception e){
            e.printStackTrace();
        }
    }
};
myThread.start();//启动线程
```

### 3.3.2 调用系统相机

一个 Button 按钮，点击则触发事件，在这个事件中实现调用系统相机拍照的功能。我们要设置拍照得到的照片的储存目录，也要检测安卓系统的版本是否能够使用 File，同时因为我们要调用系统相机，故需要对清单文件进行配置，即对相关权限的允许。我们通过 Intent 调用 MediaStore.ACTION\_IMAGE\_CAPTURE 这个 Action 来打开系统相机。方法中，我们使用了 FileProvider，它用来在不同程序之间分享数据(相对于 ContentProvider 来说)。下面是调用系统相机的相关代码：

```
private static final int TAKE_PHOTO_PERMISSION_REQUEST_CODE = 0; // 拍照的权限处理返回码
private static final int WRITE_SDCARD_PERMISSION_REQUEST_CODE = 1; // 读储存卡内容的权限处理返回码

private static final int TAKE_PHOTO_REQUEST_CODE = 3; // 拍照返回的 requestCode
private Uri photoUri = null;
private Uri photoOutputUri = null; // 图片最终的输出文件的 Uri
public void startCamera() {
    File file = new File(getExternalCacheDir(), "image.jpg");
```

```

try {
    if(file.exists()) {
        file.delete();
    }
    file.createNewFile();
} catch (IOException e) {
    e.printStackTrace();
}

if(Build.VERSION.SDK_INT >= 24) {
    photoUri = FileProvider.getUriForFile(this,
        "com.example.findflower.provider", file);
} else {
    photoUri = Uri.fromFile(file); // Android 7.0 以前使用原来的方法来获取文件的 Uri
}

// 打开系统相机的 Action, 等同于: "android.media.action.IMAGE_CAPTURE"
Intent takePhotoIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
// 设置拍照所得照片的输出目录
takePhotoIntent.putExtra(MediaStore.EXTRA_OUTPUT, photoUri);
startActivityForResult(takePhotoIntent, TAKE_PHOTO_REQUEST_CODE);
}

```

### 3.3.3 从相册进行选择

从相册进行图片选择，创建一个 Intent，调用 ACTION\_GET\_CONTENT 来获取本地图库中的图片，ACTION\_GET\_CONTENT 也能获取本地的音频，视频等，故我们需要设置获取文件的数据类型。用 startActivityForResult(intent,int)打开新的 Activity choiceFromAlbumIntent，并在其中执行选择图片的操作。下面是从相册选择的相关代码：

```

public void choiceFromAlbum() {
    // 打开系统图库的 Action, 等同于: "android.intent.action.GET_CONTENT"
    Intent choiceFromAlbumIntent = new Intent(Intent.ACTION_GET_CONTENT);
    // 设置数据类型为图片类型
    choiceFromAlbumIntent.setType("image/*");
    startActivityForResult(choiceFromAlbumIntent, CHOICE_FROM_ALBUM_REQUEST_CODE);
}

```

### 3.3.4 图片裁剪

创建一个 `Intent cropPhotoIntent`,调用系统裁剪功能。对要进行裁剪的数据进行类型设置,同时需要我们为这个 `intent` 进行允许授权,最后设置我们最终图片的输出目录。下面是图片进行裁剪的代码:

```
private void cropPhoto(Uri inputUri) {  
    // 调用系统裁剪图片的 Action  
    Intent cropPhotoIntent = new Intent("com.android.camera.action.CROP");  
    // 设置数据 Uri 和类型  
    cropPhotoIntent.setDataAndType(inputUri, "image/*");  
    // 授权应用读取 Uri, 这一步要有, 不然裁剪程序会崩溃  
    cropPhotoIntent.addFlags(Intent.FLAG_GRANT_READ_URI_PERMISSION);  
    // 设置图片的最终输出目录  
    // 由于 glide 使用缓存, 为了保证显示的图片正确, 每次生成的图片名必须不同  
    String fileName = Environment.getExternalStorageDirectory().toString()  
        + "/" + Calendar.getInstance().getTimeInMillis() + ".jpg";  
    cropPhotoIntent.putExtra(MediaStore.EXTRA_OUTPUT,  
        photoOutputUri = Uri.parse("file:////" + fileName));  
    startActivityForResult(cropPhotoIntent, CROP_PHOTO_REQUEST_CODE);  
}
```

### 3.3.5 退出应用

一个 `Button` 按钮, 在 `onCreate` 方法中对它进行绑定, 并调用 `setOnClickListener()`函数, 添加点击事件。退出应用方法中, 创建一个 `intent`, 并调用 `ACTION_MAIN` 这个程序入口操作, 然后重写返回键回到桌面, 结束进程。

下面是退出应用的相关代码:

```
private void exit() {  
    Intent intent = new Intent(Intent.ACTION_MAIN);  
    intent.addCategory(Intent.CATEGORY_HOME);  
    intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);  
    startActivity(intent);  
    finish();  
    System.exit(0);  
}
```

### 3.3.6 调用图片识别 API

本项目采用了百度云提供的植物识别 API，该接口通过检测用户上传的植物图片，返回植物名称、置信度信息。在调用 api 前，首先需要在项目中加载 API 提供的 ImageClassify Java SDK 工具包，并设置云服务密钥。API 请求与返回参数如下：

参数名称	是否必选	类型	说明	
image	是	string	图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式。 <b>注意：图片需要base64编码、去掉编码头后再进行urlencode。</b>	
baike_num	否	integer	0	返回百科信息的结果数，默认不返回

图 3-3-6-1 百度云植物识别 API 请求参数表

参数	类型	是否必须	说明
log_id	uint64	是	唯一的log id，用于问题定位
result	array(object)	是	植物识别结果数组
+name	string	是	植物名称，示例：吉娃娃
+score	uint32	是	置信度，示例：0.5321
+baike_info	object	否	对应识别结果的百科词条名称
++baike_url	string	否	对应识别结果百度百科页面链接
++image_url	string	否	对应识别结果百科图片链接
++description	string	否	对应识别结果百科内容描述

图 3-3-6-2 百度云植物识别 API 返回参数表

由于调用网络 API 服务需要使用到网络访问，为保证主线程安全，该访问操作需要在一个子线程中进行。因此定义一个子线程 `runnable`，并在活动的 `onCreate()` 函数重写中开始进程。相关代码如下

```
private Runnable runnable = new Runnable() {
    @Override
    public void run() { //联网获取结果子线程。访问网络需要单独在线程内进行
        // TODO: http request.
        getFlowerInfo();
    }
};

//开启联网获取结果子线程
Thread getInfo = new Thread(runnable);
getInfo.start();
```

进入子线程后，建立网络链接，向 API 传输必要的参数（图片数据及服务密钥）。图片数据来源于裁剪结束后传递给当前活动的图片地址值。当 API 调用正确时，可以接收到云服务器返回的识别结果。相关代码如下：

```
public void getFlowerInfo() {
    AipImageClassify client = new AipImageClassify(APP_ID, API_KEY, SECRET_KEY);

    // 传入可选参数调用接口
    HashMap<String, String> options = new HashMap<>();
    options.put("baike_num", "5");

    // 参数为本地路径
    JSONObject res = client.plantDetect(pic_path, options);

    //从 api 传回的结果中提取结果信息
    try {
        result = res.getJSONArray("result");
    } catch (JSONException e) {
        e.printStackTrace();
    }
}
```



### 3.3.7 识别结果显示

为了保证 UI 界面的一致性，子线程中不可对 UI 界面进行操作。因此在获得识别结果后，需要返回主线程，在主线程中调用界面更新函数。相关代码如下：

```
//保证获取结果线程执行完后返回主线程
try {
    getInfo.join();
} catch (InterruptedException e) {
    e.printStackTrace();
}

//返回主线程显示结果
setFlowerInfo();
```

界面更新函数根据识别返回的值更新界面元件的显示。其中对于图片的更新使用了图片加载框架 **Glide**，该框架提供了简便平滑的图片加载服务，并能通过预置的函数简便地对图片进行裁切显示。本应用中，两次调用了 **Glide** 服务，分别对于本地图片与网络图片进行显示。

本地图片展示用于向用户展示其拍摄/选择的用于检索的照片。该图片被裁剪为圆形后，显示在搜索结果的最上方。网络图片则根据搜索结果中返回的百科图片链接，显示在搜索结果的尾端作为参考图片。在结果没有提供参考图片时，该位置将显示默认的暂无图片提示图。相关代码如下：

```
//显示裁剪后的目标图片
ImageView flw_pic = findViewById(R.id.flw_pic);
Glide.with(context).load(pic_path)
    .apply(RequestOptions.bitmapTransform(new CircleCrop()))
    .into(flw_pic);

//显示结果示例图片
imgUrl = baike_info.getString("image_url");
Glide.with(context).load(imgUrl)
    .into(info_pic);
```

其他文字信息分别使用组件的 `setText` 函数进行设置，需要根据显示的精度和格式对其中一些参数进行处理。相关代码如下：

```
//依次设置结果名 准确度 简介内容
name_ch.setText(result.getJSONObject(0).getString("name"));
String s = result.getJSONObject(0).getString("score");
score_txt.setText(String.valueOf(Math.round(Double.parseDouble(s) * 1000) / 10.0));

JSONObject baike_info = result.getJSONObject(0).getJSONObject("baike_info");
s = "\t\t\t" + baike_info.getString("description");
description_txt.setText(s);
```

### 3.3.8 返回并删除临时图片

使用界面上的返回键或使用手机 `home` 键操作均可达到返回上级的效果。

界面返回键使用 `Button` 组件实现。在活动的 `onCreate()` 函数中需要对其设置监听。当按钮被点击时，执行返回操作。返回操作的实现为结束当前活动，这将使程序自动跳转回上一级活动界面。相关代码如下：

```
//返回按钮监听
Button rtn_button = findViewById(R.id.rtn_button);
rtn_button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        ResultActivity.this.finish();
    }
});
```

由于用户裁剪的图片被临时保存在手机内存中，为了保证内存的有序管理，需要重写本活动的退出函数。在将要退出本界面时，删除临时保存的图片，并调用超类的退出函数实现界面的退出。相关代码如下：

```
@Override
public void finish() {
    File temp_img = new File(pic_path);
    temp_img.delete();

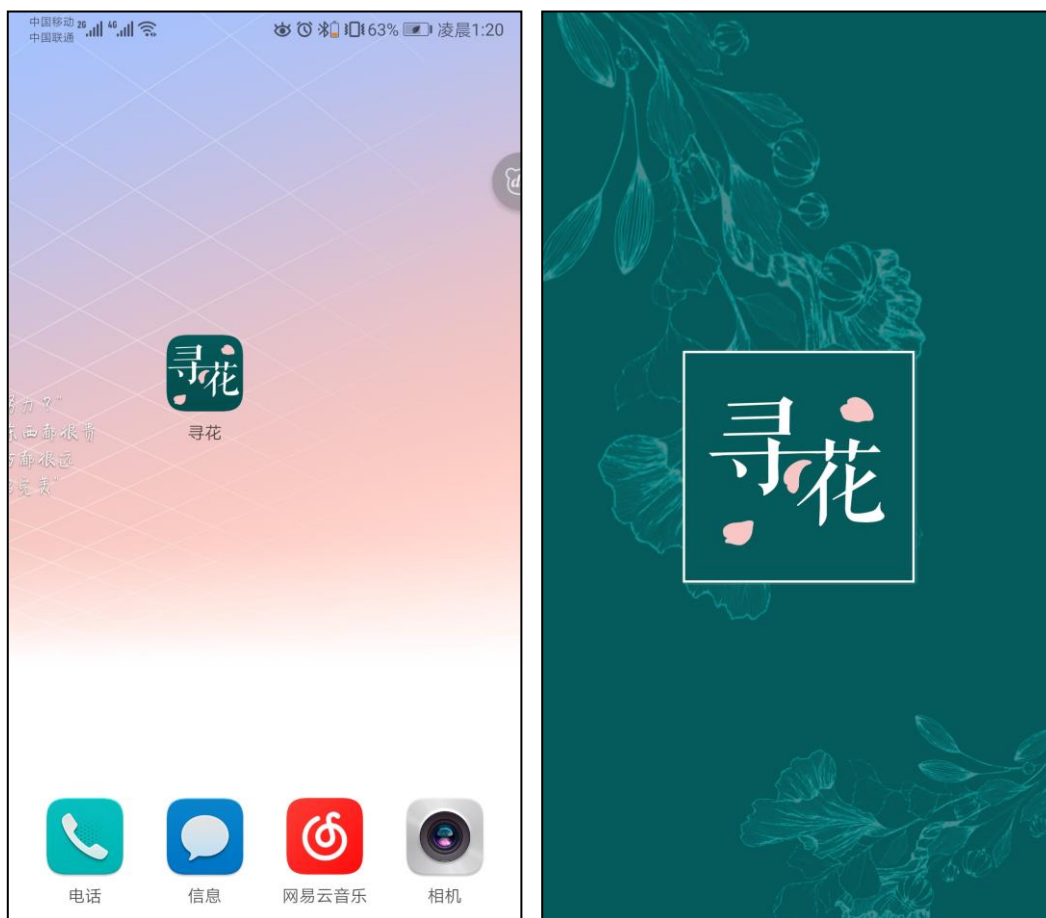
    //调用超类退出函数。若无则无法正确退出
    super.finish();
}
```

## 4 结论

### 4.1 实验数据

以下为演示应用部分功能时的部分截图。

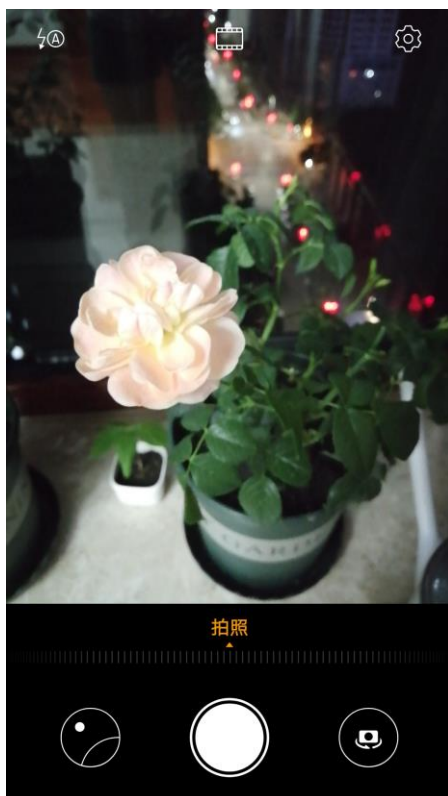
#### 4.1.1 欢迎界面



#### 4.1.2 功能选择界面



#### 4.1.3 拍摄与裁剪界面



4.1.5 结果显示界面

月季花

准确度 78.3 %

简介

月季花(学名：Rosa chinensis Jacq.)：被称为花中皇后，又称“月月红”，是常绿、半常绿低矮灌木，四季开花，一般为红色，或粉色、偶有白色和黄色，可作为观赏植物，也可作为药用植物，亦称月季。有三个自然变种，现代月季花型多样，有单瓣和重瓣，还有高心卷边等优美花型；其色彩艳丽、丰富，不仅有红、粉、黄、白等单色，还有混色、银边等品种；多数品种有芳香。月季的品种繁多，世界上已有近万种，中国也有千种以上。自然花期8月到次年4月，花成大型，由内向外，呈散发型，有浓郁香气，可广泛用于园艺栽培和切花。月季的适应性强，耐寒、抗旱，不论地栽、盆栽均可，适用于美化庭院、装点园林、布置花坛、配植花篱、花架，月季栽培容易，可作切花，用于做花束和各种花篮，月季花朵可提取香精，并可入药。也有较好的抗真菌及协同抗耐药真菌活性。红色切花更成为情人节必送的礼物之一，并成为爱情诗歌的主题。中国是月季的原产地之一。月季花荣秀美，姿色多样，四时常开，深受人们的喜爱，中国有52个城市将它选为市花，1985年5月月季被评为中国十大名花第五名。(概述图片参考资料来源：)

非植物

准确度 11.6 %

简介

Oops，搜索结果暂时没有简介呢

参考图片

暂无图片

月季花

准确度 78.3 %

简介

月季花(学名：Rosa chinensis Jacq.)：被称为花中皇后，又称“月月红”，是常绿、半常绿低矮灌木，四季开花，一般为红色，或粉色、偶有白色和黄色，可作为观赏植物，也可作为药用植物，亦称月季。有三个自然变种，现代月季花型多样，有单瓣和重瓣，还有高心卷边等优美花型；其色彩艳丽、丰富，不仅有红、粉、黄、白等单色，还有混色、银边等品种；多数品种有芳香。月季的品种繁多，世界上已有近万种，中国也有千种以上。自然花期8月到次年4月，花成大型，由内向外，呈散发型，有浓郁香气，可广泛用于园艺栽培和切花。月季的适应性强，耐寒、抗旱，不论地栽、盆栽均可，适用于美化庭院、装点园林、布置花坛、配植花篱、花架，月季栽培容易，可作切花，用于做花束和各种花篮，月季花朵可提取香精，并可入药。也有较好的抗真菌及协同抗耐药真菌活性。红色切花更成为情人节必送的礼物之一，并成为爱情诗歌的主题。中国是月季的原产地之一。月季花荣秀美，姿色多样，四时常开，深受人们的喜爱，中国有52个城市将它选为市花，1985年5月月季被评为中国十大名花第五名。(概述图片参考资料来源：)

参考图片



25

## 4.2 问题与解决方案

1. 问题: `android:name="android.support.FILE_PROVIDER_PATHS"`不能使用

解决方案: 网上搜索后得知,其进行了更新,需要使用如下内容对其进行替换,  
`android:name="androidx.core.content.FileProvider"`, 同时 `import` 也要更改。

2. 问题: 项目真机调试一次之后, 无法进行第二次

解决方案: 网上的所有方法尝试之后, 对本机适用的是关闭项目, 关闭 Android Studio, 冷却一会之后重新打开, 便可解决。

3. 问题: 拍照返回图片很模糊

解决方案: 不直接使用 `Bitmap bitmap = (Bitmap) bundle.get("data")` 返回 `data`, 因为 `data` 容易溢出, 会导致像素变低, 而是创建文件路径, 先将图片进行保存, 然后在从相应目录下面获取原图。

4. 问题: 在编译器预览中排布正确的布局, 真机调试后全部混乱

解决方案: 在布局时全部使用相对布局, 采用百分比的方式计算图片位置, 减少直接使用 `px` 或 `dp` 为单位的元件属性。

5. 问题: 创建 API 请求出现 `android.os.NetworkOnMainThreadException` 报错

解决方案: 将网络请求由主线程移入子线程中, 保证主线程不会因为网络请求时间过长而导致界面假死。

6. 问题: 用户图片始终只显示软件第一次运行、第一次拍照之后的图片, 但查询时使用的同一地址的图片是正确的

解决方案: 由于 `Glide` 调用的图片是缓存中的图片, 因此在图片名不变的情况下, 同一句 `Glide` 操作调用的总是缓存中已经保存的第一张图片。因此将裁剪后图片的保存名修改成使用当前时间生成的文件名, 以保证每次调用的图片文件名不同。为了防止多次生成图片占用内存, 在退出界面时加入删除临时图片的操作。

## 附录

### 成员贡献表

姓名	角色	可验证贡献
朱卉萱	PO SM Developer	产品功能设计；界面 UI 设计；API 调用；结果界面实现；图片显示；查询结果显示
夏雪琴	Developer	相机调用；相册选择；裁剪；退出；欢迎界面及功能选择界面

### 个人总结

#### 朱卉萱

本次嵌入式程序设计实验是我第一次接触手机应用开发。相比于此前使用通用型 IDE Eclipse 进行的 Java 开发项目，Android Studio 更精更专于安卓应用的开发，因此集成了很多实用且有趣的功能，如可视化的界面设计模块、自动多分辨率的图片管理等等。这些新的功能让我更加体会到了程序设计的乐趣。在实验过程中，从对 IDE 的一无所知，到反复查阅资料、面向搜索引擎编程，再到完全掌握项目结构、可以独立进行模块开发，这个过程加深了我对于 java 开发的理解，并强化了自我的学习能力。实验中小组成员配合默契，进度顺利。

另外，相比于传统的电脑软件开发，本次的手机应用开发具有嵌入式开发的一个重要特点，即需要考虑到兼容性。在开发时，无论是权限的处理，还是界面的安排，都需要注意到在不同的嵌入式设备上运行的效果。这是非常难得的经历。

本次实验还有很多可以扩展的内容值得在未来进一步探究。如可以加入用户数据库系统，实现用户的登陆、搜索结果的保存等功能。还可以进一步完善兼容性，保证其在更多版本的安卓系统上均可运行等。这些功能留待进一步的学习与丰富。



## 夏雪琴

这次的“寻花”应用软件嵌入式实验设计项目，在和组长的带领以及我们的配合之下顺利完成了。对我来说，这是一次全新的体验，因为我第一次使用 Android Studio 进行 App 开发。环境配置好之后，在 GitHub 上面下载了一个项目在其上运行，但不是很成功。所以就找了一个精短的学习视频，跟着完成了一个简单的项目案例。之后便对一个项目的开发有了一定的了解，然后就正式完成自己的相关部分。当然，由于不懂的东西太多，在完成相应功能之前都会查找相关资料进行学习。

当我们整个项目完成且运行成功之后，还是很兴奋的，同时也有那么一点点小成就感。当然，我欠缺的东西还很多，没有对 Android 的系统学习，Java 的一些相关知识也渐忘了，都只能要用时查找资料。这次的开发经验激起了我对 Android 的兴趣，希望自己后面有时间对其进行系统的学习，特别是要巩固 Java 的知识板块。



## 教师评语评分

评语： \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

评分： \_\_\_\_\_

评阅人：

年      月      日

（备注：对该实验报告给予优点和不足的评价，并给出百分之评分。）