**Android开发说明文档**

**一、开发环境的搭建过程**

第一步 安装JDK，并在环境变量中配置相应的参数。

第二步 安装开发工具，常用的开发工具有两种，一种是eclipse开发工具开发时需要导入ADT插件，一种是Android Studio，开发时不需要任何插件。我们这里使用Android Studio

第三步 安装SDK（android开发环境），并在环境变量中配置相应的参数。

第四步 测试环境是否搭建好，以此用命令javac、android –h进行测试，如果命令窗口中出现相应的说明，则证明开发环境搭建成功。

**二、功能需求分析**

1、使用Android原生技术打造项目所需要的UI（可以在原生的基础上，部分加入混合式技术

2、完成与百度大脑的《人脸检测》API交互，完成相应的功能

3、完成与百度大脑的《人脸对比》API交互，完成相应的功能

4、完成人脸库的建立功能

5、利用《人脸搜索》完成基于合影照片的人脸搜索并显示

**三、功能设计**

1. 数据库设计

本项目用到数据库的地方不多，数据库中只有一张USER表，表用来存放用户信息，保存用户的用户名和密码。表如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 字段名 | 类型 | 长度 | 备注 |
| 用户名 | Username | Char | 10 | 主键 |
| 密码 | Password | Char | 20 | 初始值0 |

1. 单元测试设计

# 针对图片压缩+Base64编码进行了单元测试，代码如下：

|  |
| --- |
|  |

建立Face对象的测试方法：

|  |
| --- |
| **public void** testSetFace(){   Face face =**new** Face(**"123"**,R.drawable.***qq\_icon***);  Face face2 =**new** Face(**"234"**,R.drawable.***qq\_icon***);  List<Face> users=**new** ArrayList<>();   users.add(face);  users.add(face2);   Gson gson=**new** Gson();  String s1=gson.toJson(users);  System.***out***.println(s1);  } |

测试FaceSearch方法的测试：

|  |
| --- |
| **public static void** testFaceSearch(){   FaceSearch faceSearch =**new** FaceSearch();  BitmapFactory.Options options = **new** BitmapFactory.Options();  options.**inJustDecodeBounds** =**true**;  Bitmap bitmap2 = BitmapFactory.*decodeResource*(getResources(), R.drawable.***facetest***, options);  ByteArrayOutputStream baos = **new** ByteArrayOutputStream();  bitmap2.compress(Bitmap.CompressFormat.***JPEG***, 100, baos);  **byte**[] data = baos.toByteArray();  System.***out***.println(faceSearch.*faceSearch1*(data));  } |

1. 类设计

实体类：

1. Face：人脸信息集合，属性包括裁剪后的人脸照片、人脸照片的评价
2. FriendMessage：好友动态信息集合，属性包括好友头像、昵称、文字信息、发布时间
3. User\_list: 人脸搜索1：n返回结果格式，属性包括人脸名称、相似度、人脸所在位置等
4. User\_list2: 人脸搜索m：n返回结果格式，属性包括人脸名称、相似度、人脸所在位置等
5. User\_list3: 人脸检测返回结果格式，属性包括表情、性别、年龄、人脸所在位置等
6. User\_list4: 人脸对比返回结果格式，属性包括两张人脸信息、相似度

动作类：

1. Login1 ：登录界面调用
2. Register：注册界面调用
3. First ：侧滑栏界面调用
4. MainActivity ：人脸搜索，拍照实现支持人脸1：N 与M：N 搜索，并保存在以人脸名字命名的相册
5. MainActivity2 ：发布动态，上传N张图片到服务器；动态可以打字输入或语音输入
6. MainActivity3 ：人脸检测，拍照或相册选取图片进行识别后返回结果
7. MainActivity4 ：人脸对比，第一张照片拍照获取，第二张照片从相册中选取。识别后返回结果

工具类：

1. FaceAdapter ：继承ArrayAdapter
2. FaceDetect ： 人脸检测方法
3. FaceSearch ： 人脸搜索方法，包括1：N 与M：N方法
4. FaceMatch ： 人脸对比方法
5. AuthService：自动获取token码
6. FileUtil、Base64Util、HttpUtil、GsonUtil ：调用2、3、4、5时需要的工具类
7. MyGridview ：继承GridView
8. SpeechRecognizerTool ：语音输入的工具类
9. 本地接口设计

1）支持GET和POST请求、支持跨域

2）mock文件支持 .js 文件和 .json 文件

3）json文件 用来存放固定的模拟数据，而 js文件 可更加自由的处理并返回模拟数据

4）修改mock文件后，无需重启node服务，重新调用接口即返回新的数据

1. Restful网络接口设计

1）每一个URI代表一种资源；

2）客户端和服务器之间，传递这种资源的某种表现层；

3）客户端通过四个HTTP动词，对服务器端资源进行操作，实现"表现层状态转化"。

**四、软件安装及使用说明**

软件安装：直接安装apk即安装完毕。

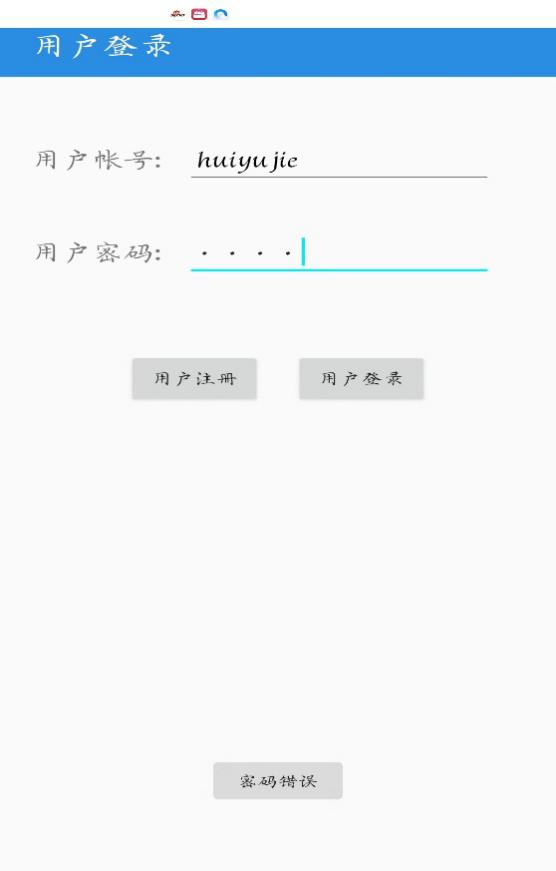
使用说明：

**一** 、用户登录及注册

1、打开主页面进行用户注册。若输入重复用户名则提示“用户已存在”，注册成功则跳转至登陆界面。



2、填好注册好的信息进行用户登录。登陆成功跳转至主界面，登陆失败弹出相关错误信息。

1. 侧滑栏功能

点击左上角按钮，即可打开左侧菜单栏，功能都包含在菜单栏中。



三、人脸搜索功能

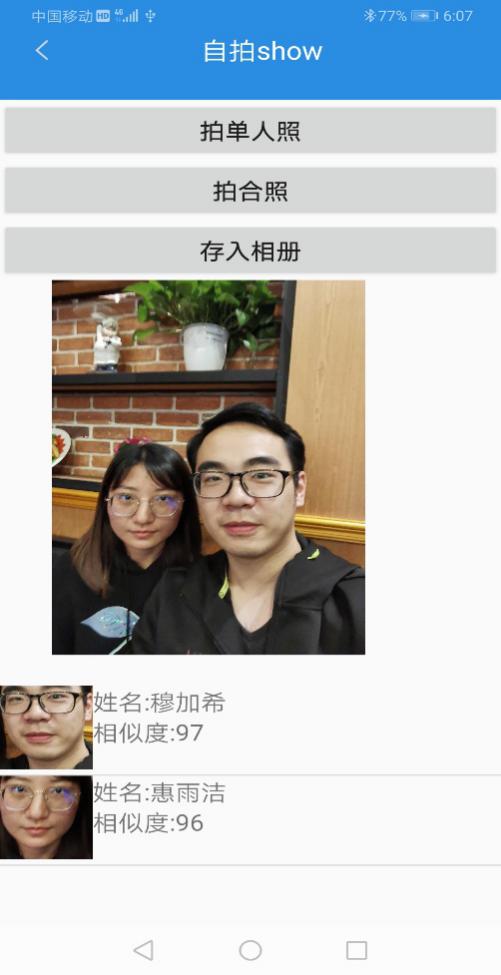
1、进入主页面，左滑侧边栏。点击“自拍show”进入拍照识别界面。



2、单人点击单人拍照识别（1：n人脸搜索），多人点击合照识别（m:n人脸搜索）。



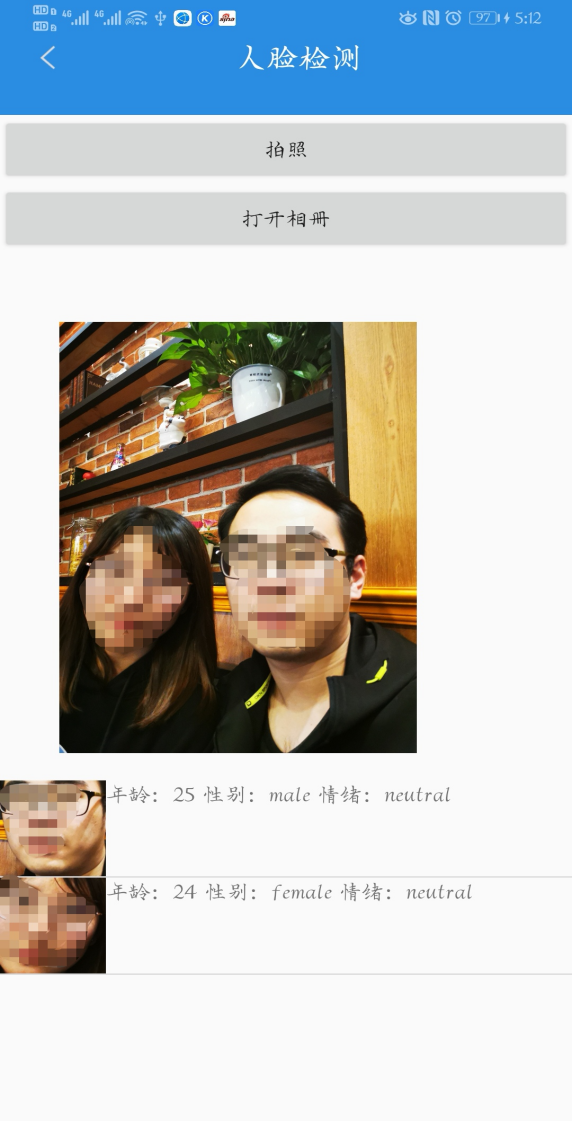
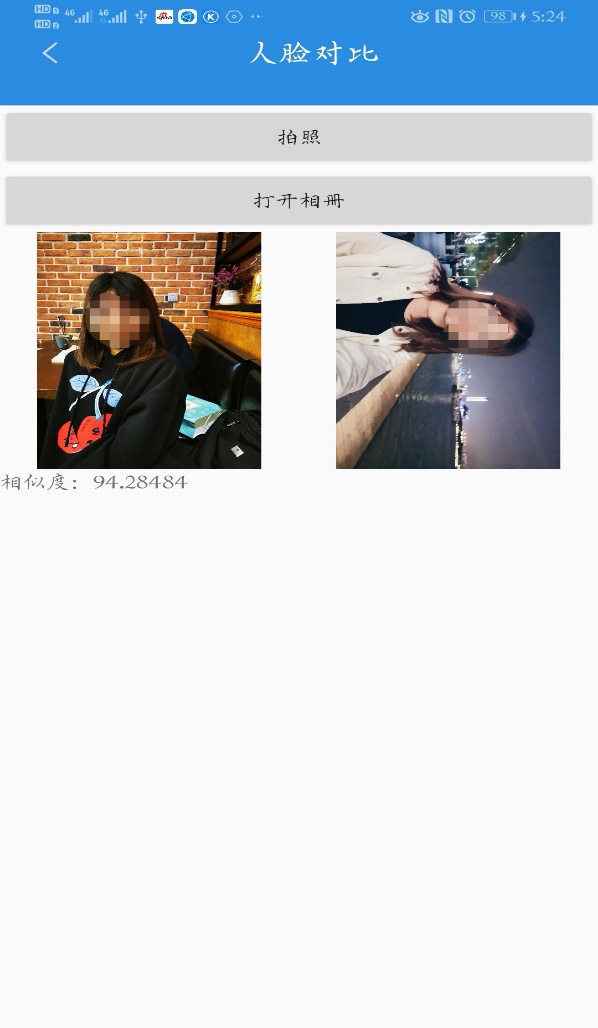
3、拍照返回数据。单人直接返回姓名和匹配度，合照返回人脸裁剪小图和数据，显示在列表中。

4、点击存入相册，即可保存到对应的相册中

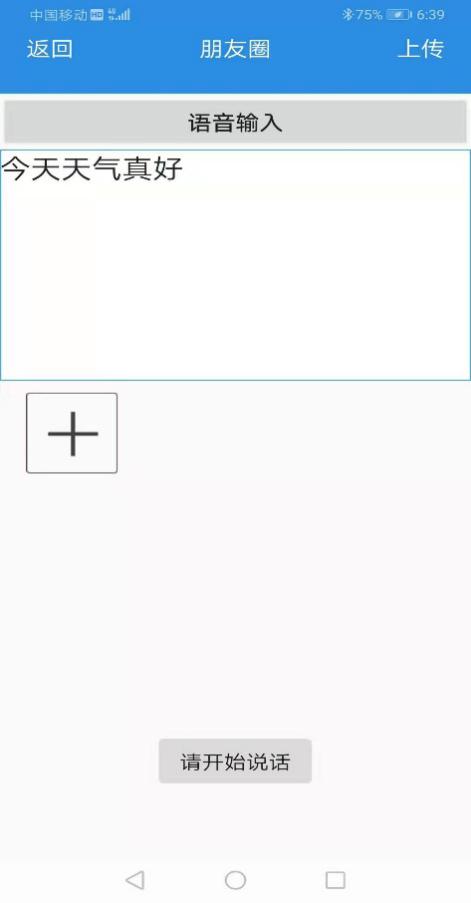


四 （拍照/相册选取）实现人脸检测/人脸对比

五 发布动态

1. 点击侧滑栏“发布动态”进入文字输入界面。可选打字输入或语音输入要发表的内容。输入完成点击‘上传’返回主界面即可显示结果。

****  ****

2.点击加号相册选取多张照片（最多9张），剪裁图片，以网格形式显示在界面，点击图片可以将已经插入的图片删除，点击上传可上传至服务器。

