《人脸识别》项目说明

1. 项目概况
   1. 简介：按照《企业移动云开发》课程设计要求，实现人脸识别的人脸注册、人脸搜索、人脸对比基本功能，并进行一定功能扩展，本项目还实现了与云数据库进行数据交互。产品可用于人脸打卡、趣味娱乐等。
   2. 开发环境：Android Studio 3.5.2，SDK 29，JDK 1.8
   3. 运行环境：Android 5 以上
2. 模块功能

2.1模块功能图：

人脸识别

人脸对比

人脸注册

人脸检测

人脸搜索

上传云库

2.2各模块介绍：

2.2.1人脸注册：

1. 注册+上传云库：

**new** Thread(**new** Runnable() {  
 @Override  
 **public void** run() {  
  
 *// 获取token对象* String accessToken = FaceRecogniseUtil.*getAuth*();

//先进行一个搜索验证  
 String search = FaceRecogniseUtil.*facesSearch*(accessToken, base64, **"group01"**);  
 ...  
   
   
  
 **if** (searchMsg.equals(**"SUCCESS"**)){  
 Toast.*makeText*(RegisterPageActivity.**this**, **"用户已注册，无需再注册"**, Toast.***LENGTH\_LONG***).show();  
 }**else if** (searchMsg.equals(**"match user is not found"**)) {

*// 注册到百度大脑*

String add = FaceRecogniseUtil.*add*(accessToken, base64, map.toString());  
 ...  
 **if** (errorMsg.equals(**"SUCCESS"**)) {

*// 用户信息上传到云服务器的MongoDB*

Document document = **new** Document(map);  
 document.append(**"photo"**, base64);  
 MongoDatabase face\_library = MongoDBUtill.*getConnection*(**"39.105.102.158"**,  
 27017, **"face\_library"**);  
 MongoCollection<Document> user\_info = face\_library.getCollection(**"user\_info"**);  
 user\_info.insertOne(document);  
   
 } **else** {  
 Toast.*makeText*(RegisterPageActivity.**this**,  
 **new** StringBuffer(**"注册失败："**).append(errorMsg).toString(),  
 Toast.***LENGTH\_LONG***).show();  
 }  
 }  
 }**else** {  
 Toast.*makeText*(RegisterPageActivity.**this**, **"注册失败："** + searchMsg, Toast.***LENGTH\_LONG***).show();  
 }  
}).start();

2.2.2人脸搜索：

1. M:N搜索

**switch** (**Show\_Choice**) {  
 *//如果传递为TAKE\_PHOTO* **case *TAKE\_PHOTO***: {  
 AuthorityUtils.*getAlbumAuthority*(Image\_album\_showActivity.**this**);

...

...  
 *// 启动相机程序* Intent intent = **new** Intent();  
 *// 指定开启系统相机的Action* intent.setAction(MediaStore.***ACTION\_IMAGE\_CAPTURE***);  
 intent.putExtra(MediaStore.***EXTRA\_OUTPUT***, **imageUri**);  
 System.***out***.println(**"看着路："** + intent);  
 startActivityForResult(intent, ***TAKE\_PHOTO***);  
  
 }  
 **break**;

*//如果传递为CHOOSE\_PHOTO* **case *CHOOSE\_PHOTO***: {  
 AuthorityUtils.*getAlbumAuthority*(Image\_album\_showActivity.**this**);  
 Intent intent = **new** Intent(**"android.intent.action.GET\_CONTENT"**);  
 intent.setType(**"image/\*"**);  
 startActivityForResult(intent, ***CHOOSE\_PHOTO***);  
 }  
 **break**;  
  
 **default**:  
 **break**;  
 }  
}  
  
@Override  
**protected void** onActivityResult(**int** requestCode, **int** resultCode, @Nullable Intent data) {...}

**public void** onClickUpload(View view) {...}

2.2.3人脸检测：

/\* MultiImageSelectorActivity的使用 \*/

**choose**.setOnClickListener(i -> {  
 Intent intent = **new** Intent(FaceDetectActivity.**this**, MultiImageSelectorActivity.**class**);  
 *// 存储照片的list* ArrayList<String> imgList = **new** ArrayList<>();  
 *// 是否显示调用相机拍照* intent.putExtra(MultiImageSelectorActivity.***EXTRA\_SHOW\_CAMERA***, **true**);  
 *// 最大图片选择数量* intent.putExtra(MultiImageSelectorActivity.***EXTRA\_SELECT\_COUNT***, 9);  
 *// 设置模式 (支持 单选/MultiImageSelectorActivity.MODE\_SINGLE 或者 多选/MultiImageSelectorActivity.MODE\_MULTI)* intent.putExtra(MultiImageSelectorActivity.***EXTRA\_SELECT\_MODE***, MultiImageSelectorActivity.***MODE\_MULTI***);  
 *// 默认选择图片,回填选项(支持String ArrayList)* intent.putStringArrayListExtra(MultiImageSelectorActivity.***EXTRA\_DEFAULT\_SELECTED\_LIST***, imgList);  
 startActivityForResult(intent, **REQUEST\_IMAGE**);  
});

/\* RecyclerView的使用 \*/

@Override  
**protected void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) {  
 **super**.onActivityResult(requestCode, resultCode, data);  
 **if** (resultCode == ***RESULT\_OK***){  
 **if** (requestCode == **REQUEST\_IMAGE**){  
 *// 获取返回的图片列表* List<String> imgs = data.getStringArrayListExtra(MultiImageSelectorActivity.***EXTRA\_RESULT***);  
 System.***out***.println(imgs);  
 LinearLayoutManager layoutManager = **new** LinearLayoutManager(**this**);  
 **rcv**.setLayoutManager(layoutManager);  
 PhotoAdapter adapter = **new** PhotoAdapter(imgs);  
 **rcv**.setAdapter(adapter);  
 }  
 }  
}

2.2.4人脸对比：

整体思路如上，此处略去细节

1. 其它设计内容

3.1 数据库设计：

使用阿里云MongoDB数据库存储用户信息（包括压缩后的base64类型图片）

使用百度大脑提供的人脸库

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| user\_id | user\_name | birthday | Phone\_num | E\_mail | image |
| Int(6) | String | Datetime | Int(11) | String | Base64 |

3.2 类设计

百度接口

阿里云接口

BaiduUtils

DBUtils

相机接口

各种Activity

相册接口

3.3 单元测试设计：

通过更改启动页面进行以一个Activity为单位的单元测试

3.4 本地接口设计：

主要通过点击事件进行activity之间及各事件之间的跳转转换及相关数据传递

3.5 网络接口设计：

1）利用百度智能云提供的URL及接口方法与百度大脑进行交互，例如：

String url = **"https://aip.baidubce.com/rest/2.0/face/v3/detect"**;  
 ...

...  
 String result = HttpUtil.*post*(url, accessToken, **"application/json"**, param)

1. 在阿里云服务器上部署MongoDB，暴露其所用端口号之后开启mongodb server，

再利用MongoDB的 Driver在本地通过云地址+端口号进行连接，如：

MongoDatabase face\_library = MongoDBUtill.*getConnection*(**"39.105.102.158"**,  
 27017, **"face\_library"**);

1. 使用说明

见PPT