

# 第 21 章

## 人脸识别案例——年龄小侦探 App



### 学习指引

移动 App 已逐渐成为技术人员必须掌握的一门技术和本领，移动 App 的学习已迫不及待。下面让我们通过一个案例来一起揭开移动 App 神秘的面纱。该案例介绍的是一个人脸识别的“年龄小侦探 App”，是一个基于 HTML 5+CSS 3+jQuery 实现的移动端 App，支持 Android 和 iOS 系统。下面将从基础和实用的角度通过基本功能梳理、开发环境准备、项目实战的三步走方针展开论述，力求每一位学习者对一个简单 App 的实现可以柳暗花明，积累一定的经验，掌握必要的实战能力！



### 重点导读

- 掌握使用 HTML 5+JavaScript+CSS 设计。
- 掌握 HBuilder 的安装与环境配置。
- 熟悉 Native.js。
- 掌握 App 打包的方法。
- 了解人脸识别的方法。

## 21.1 项目概述

### 21.1.1 功能梳理

一位优秀的开发工程师一定是一名合格的产品经理和需求分析师。一切开发和设计均是围绕业务展开和实现的。面对一个应用系统首先要做到的是明确它的需求，才能让代码在你的指尖尽情跳跃，来达到最终的完美效果。

下面就让我们简要梳理分析一下“年龄小侦探 App”需要实现哪些功能项才能呈现完美效果？

- 照片选择器：其实就是移动端文件的查询，调用文件选择器。



- 摄像头拍照：调用本地摄像头进行拍照。
- 文件上传：将从照片文件夹中选择的照片或者拍摄确定的照片上传。
- 人脸识别：调用 face++接口实现人脸识别，给出年龄判断结果。



## 21.1.2 开发环境

开发环境和第 20 章一样，可参考“记事本 App”案例开发环境。



## 21.1.3 代码结构

使用 HBuilder 打开项目之后，整个项目的代码结构如图 21-1 所示。

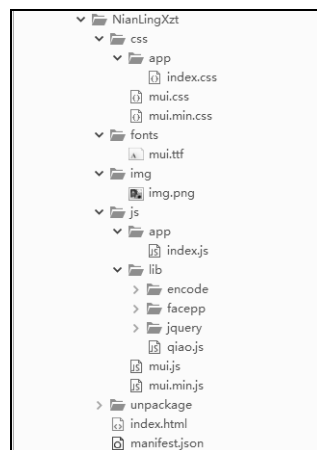


图 21-1 项目的代码结构

# 21.2 项目解析

该项目中只有一个主要的展示和操作页面，即 index.html，所有效果都是通过原生自带和自己手动编写的简单 CSS 3 展示，所有的功能实现都是通过调用原生的 Mui 和 JavaScript 控制实现的（主要的调用为 index.js 和 qiao.js）。

此项目所有的常用方法都是在 index.js 中进行调用的。看一下 index.js 和 qiao.js 文件的代码，就会让你一目了然。



## 21.2.1 index.html 文件

该代码文件直接是应用程序的主入口，定义了程序的结构和所需的 JavaScript、image 等资源，具体代码如下：

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width,initial-scale=1,minimum-scale=1,
maximum-scale=1,user-scalable=no" />
  <title>face</title>
  <link rel="stylesheet" type="text/css" href="css/mui.min.css"/>
  <link rel="stylesheet" type="text/css" href="css/app/index.css"/>
</head>
<body>
  <div class="mui-content-padded">
    <img class="face-img" id="faceImg"/>
    <button class="mui-btn mui-btn-primary face-btn" id="faceBtn">开始识别</button>
    <p id="res"></p>
  </div>
  <form method="post" action="http://upload.qiniu.com/" enctype="multipart/form-data"
id="faceForm">
    <input name="file" type="file" />
    <input name="key" type="hidden" />
    <input name="token" type="hidden" />
  </form>
```

```

<script type="text/javascript" src="js/mui.min.js"></script>
<script type="text/javascript" src="js/lib/jquery/jquery-1.11.2.min.js"></script>
<script type="text/javascript" src="js/lib/jquery/jquery.form.js"></script>
<script type="text/javascript" src="js/lib/encode/core-min.js"></script>
<script type="text/javascript" src="js/lib/encode/cipher-core-min.js"></script>
<script type="text/javascript" src="js/lib/encode/enc-base64-min.js"></script>
<script type="text/javascript" src="js/lib/encode/hmac-min.js"></script>
<script type="text/javascript" src="js/lib/encode/sha1-min.js"></script>
<script type="text/javascript" src="js/lib/facepp/facepp-sdk.min.js"></script>
<script type="text/javascript" src="js/lib/qiao.js"></script>
<script type="text/javascript" src="js/app/index.js"></script>
</body>
</html>

```

## 21.2.2 index.js 文件

具体代码如下：

```

// 初始化
mui.init({});

// 所有方法都放到这里
mui.plusReady(function() {
    if(mui.os.ios) plus.navigator.setStatusBarBackground('#E0E0E0');
    qiao.on('#faceImg', 'tap', choiceImg); // 选择照片
    qiao.on('#faceBtn', 'tap', uploadImg); // 上传照片
    // 退出
    mui.back=function() {
        qiao.h.exit();
    };
});
// 选择图片
function choiceImg() {
    qiao.h.sheet('选择照片', ['拍照', '相册'], function(e) {
        var index=e.index;
        if(index==1) choiceCamera();
        if(index==2) choicePic();
    });
}
// 调用相机
function choiceCamera() {
    var cmr=plus.camera.getCamera();
    cmr.captureImage(function (p) {
        plus.io.resolveLocalFileSystemURL(p, function(entry) {
            setImg(entry.toLocalURL());
        }, function(e) {});
    }, function(e) {}, {index:1, filename:"_doc/camera/"});
}
// 选择相册照片
function choicePic() {
    plus.gallery.pick(function (path) {setImg(path);}, function(e) {}, {filter:'image'});
}
function setImg(src) {
    $('#faceImg').attr('src', src);
}

// 上传图片
var url, tsrc;

```



```
function uploadImg(){
    var src=$('#faceImg').attr('src');
    if(src){
        beginw();// 开始上传
        if(tsrc && tsrc==src && url){
            facepp();
        }else{
            tsrc=src;
            var token=qiao.qiniu.uptoken(src);
            var filename=qiao.qiniu.file;
            qiao.h.upload({
                url:'http://upload.qiniu.com/',
                filepath:src,
                datas:[
                    {key:'key', value:filename},
                    {key:'token', value:token}
                ],
                success:function(){
                    url=qiao.qiniu.url();
                    facepp();
                },
                fail:function(s){
                    showRes('上传文件失败:' + s);
                }
            });
        }
    }else{
        showRes('请先选择要识别的照片! ');
    }
}

// 人脸识别
function facepp(){
    if(url){
        qiao.facepp.do({
            url:url,
            success:function(result){
                if(result && result.face && result.face.length){
                    var face=result.face[0].attribute;
                    var str='识别成功! 性别:' + (face.gender.value=='Male' ? '男':'女') + ',年龄:' +
face.age.value;
                    showRes(str);
                }else{
                    showRes('识别失败,请上传包含人脸的图片! ');
                }
            },
            fail:function(){
                showRes('识别失败,请重试! ');
            }
        });
    }
}

// 显示结果
function showRes(msg){
    $('#res').text(msg);
    endw();
}

// 上传开始
function beginw(){
    $('#faceBtn').attr('disabled', true);
}
```

```

        qiao.h.waiting();
    }
    // 上传结束
    function endw(){
        qiao.h.closeWaiting();
        $('#faceBtn').attr('disabled', false);
    }

```

### 21.2.3 qiao.js 文件

具体代码如下:

```

var qiao={};
qiao.on=function(obj, event, func){
    $(document).off(event, obj).on(event, obj, func);
};
// 对 Mui 以及 Native.js 相关封装
qiao.h={};
// page 相关
qiao.h.normalStyle={top:'45px',bottom:0};
qiao.h.normalPage=function(id){
    return qiao.h.page(id, {styles:qiao.h.normalStyle});
};
qiao.h.centerStyle={top:'45px',bottom:'50px'};
qiao.h.centerPage=function(id){
    return qiao.h.page(id, {styles:qiao.h.centerStyle});
};
qiao.h.page=function(id, options){
    var url=id + '.html';
    options.id=id;
    options.url=url;
    return options;
};
qiao.h.indexPage=function(){
    return plus.webview.getWebviewById(plus.runtime.appid);
};
qiao.h.currentPage=function(){
    return plus.webview.currentWebview();
};
qiao.h.getPage=function(id){
    return id ? plus.webview.getWebviewById(id):null;
};
qiao.h.show=function(id, ani, time, func){
    if(id) plus.webview.show(id, ani, time, func);
};
qiao.h.hide=function(id, ani, time){
    if(id) plus.webview.hide(id, ani, time);
};
qiao.h.fire=function(id, name, values){
    mui.fire(qiao.h.getPage(id), name, values);
};
// 以下为 UI 封装-----
// upload
qiao.h.upload=function(options){
    if(!options) return;
    var url=options.url;
    var filepath=options.filepath;
    var datas=options.datas || [];

```



```
var success=options.success;
var fail=options.fail;
if(url && filepath){
    var task=plus.uploader.createUpload(url, {
        method:"POST",
        blocksize:204800,
        priority:100
    },
    function(t, status){
        if(status==200){
            if(success) success(t);
        }else{
            if(fail) fail(status);
        }
    }
    );
    task.addFile(filepath, {key:'file'});
    if(datas && datas.length){
        for(var i=0; i<datas.length; i++){
            var data=datas[i];
            task.addData(data.key, data.value);
        }
    }
    task.start();
}
};
// nativeUI 相关
qiao.h.tip=function(msg, options){
    plus.nativeUI.toast(msg,options);
};
qiao.h.waiting=function(title, options){
    plus.nativeUI.showWaiting(title, options);
};
qiao.h.closeWaiting=function(){
    plus.nativeUI.closeWaiting();
};
// popover
qiao.h.pop=function(){
    mui('.mui-popover').popover('toggle');
};
// actionsheet
qiao.h.sheet=function(title, btns,func){
    if(title && btns && btns.length > 0){
        var btnArray=[];
        for(var i=0; i<btns.length; i++){
            btnArray.push({title:btns[i]});
        }
        plus.nativeUI.actionSheet({
            title:title,
            cancel:'取消',
            buttons:btnArray
        }, function(e){
            if(func) func(e);
        });
    }
};
// 提示框相关
```

```

qiao.h.modaloptions={
  title:'title',
  abtn:'确定',
  cbtn:['确定','取消'],
  content:'content'
};
qiao.h.alert=function(options, ok){
  var opt=$.extend({}, qiao.h.modaloptions);
  opt.title='提示';
  if(typeof options=='string'){
    opt.content=options;
  }else{
    $.extend(opt, options);
  }
  plus.nativeUI.alert(opt.content, function(e){
    if(ok) ok();
  }, opt.title, opt.abtn);
};
qiao.h.confirm=function(options, ok, cancel){
  var opt=$.extend({}, qiao.h.modaloptions);
  opt.title='确认操作';
  if(typeof options=='string'){
    opt.content=options;
  }else{
    $.extend(opt, options);
  }
  plus.nativeUI.confirm(opt.content, function(e){
    var i=e.index;
    if(i==0 && ok) ok();
    if(i==1 && cancel) cancel();
  }, opt.title, opt.cbtn);
};
qiao.h.prompt=function(options, ok, cancel){
  var opt=$.extend({}, qiao.h.modaloptions);
  opt.title='输入内容';
  if(typeof options=='string'){
    opt.content=options;
  }else{
    $.extend(opt, options);
  }
  plus.nativeUI.prompt(opt.content, function(e){
    var i=e.index;
    if(i==0 && ok) ok(e.value);
    if(i==1 && cancel) cancel(e.value);
  }, opt.title, opt.content, opt.cbtn);
};
// 以下为插件封装-----
// 本地存储相关
qiao.h.length=function(){
  return plus.storage.getLength();
};
qiao.h.key=function(i){
  return plus.storage.key(i);
};
qiao.h.getItem=function(key){
  if(key){
    for(var i=0; i<qiao.h.length(); i++) {

```

```

        if(key==plus.storage.key(i)){
            return plus.storage.getItem(key);
        }
    };
}
return null;
};
qiao.h.insertItem=function(key, value){
    plus.storage.setItem(key, value);
};
qiao.h.delItem=function(key){
    plus.storage.removeItem(key);
};
qiao.h.clear=function(){
    plus.storage.clear();
};
// web sql
qiao.h.db=function(name, size){
    var db name=name ? name:'db test';
    var db size=size ? size:2;
    return openDatabase(db_name, '1.0', 'db_test', db_size * 1025 * 1025);
};
qiao.h.update=function(db, sql){
    if(db &&sql) db.transaction(function(tx){tx.executeSql(sql)});
};
qiao.h.query=function(db, sql, func){
    if(db && sql){
        db.transaction(function(tx){
            tx.executeSql(sql, [], function(tx, results) {
                func(results);
            }, null);
        });
    }
};
// 以下为功能封装-----
// 退出
qiao.h.exit=function(){
    qiao.h.confirm('确定要退出吗?', function(){
        plus.runtime.quit();
    });
};
// 刷新
qiao.h.endDown=function(selector){
    var sel=selector ? selector:'#refreshContainer';
    mui(sel).pullRefresh().endPulldownToRefresh();
};
// qiniu (七牛云)
qiao.qiniu={
    ak:'3YhXI8s0TsYLyEv irq7aKGsQsmN6i3WoERBttnyY',
    sk:'9lWh6588LIrQcrMpTagR0f19KV BcRvtgu5ZlmFU',
    pr:'http://7x13r9.com1.z0.glb.clouddn.com/',
    scope:'uikoo9-facepp',
};
qiao.qiniu.deadline=function(){
    return Math.round(new Date().getTime() / 1000) + 3600;
};
qiao.qiniu.genScope=function(src){

```



```

var scope=qiao.qiniu.scope;
if(src){
    var ss=src.split('.');
    qiao.qiniu.file=qiao.qiniu.uid() + '.' + ss[ss.length - 1];
    scope=scope + ':' + qiao.qiniu.file;
}
return scope;
};
qiao.qiniu.uid=function(){
    return Math.floor(Math.random()*100000000+10000000).toString();
};
qiao.qiniu.uptoken=function(src) {
    //SETP 1
    var putPolicy='{"scope":"' + qiao.qiniu.genScope(src) + '","deadline":"' + qiao.qiniu.
deadline() + '"}';
    //SETP 2
    var encoded=qiao.encode.base64encode(qiao.encode.utf16to8(putPolicy));
    //SETP 3
    var hash=CryptoJS.HmacSHA1(encoded, qiao.qiniu.sk);
    var encoded signed=hash.toString(CryptoJS.enc.Base64);
    //SETP 5
    var upload token=qiao.qiniu.ak + ":" + qiao.encode.safe64(encoded signed) + ":" + encoded;
    return upload token;
};
qiao.qiniu.url=function(key){
    return qiao.qiniu.pr + qiao.qiniu.file;
};
// qiniu encode
qiao.encode={};
qiao.encode.utf16to8=function(str){
    var out, i, len, c;
    out="";
    len=str.length;
    for (i=0; i < len; i++) {
        c=str.charCodeAtAt(i);
        if ((c >=0x0001) && (c <=0x007F)) {
            out +=str.charAt(i);
        } else if (c > 0x07FF) {
            out +=String.fromCharCode(0xE0 | ((c >> 12) & 0x0F));
            out +=String.fromCharCode(0x80 | ((c >> 6) & 0x3F));
            out +=String.fromCharCode(0x80 | ((c >> 0) & 0x3F));
        } else {
            out +=String.fromCharCode(0xC0 | ((c >> 6) & 0x1F));
            out +=String.fromCharCode(0x80 | ((c >> 0) & 0x3F));
        }
    }
    return out;
};
qiao.encode.utf8to16=function(str){
    var out, i, len, c;
    var char2, char3;
    out="";
    len=str.length;
    i=0;
    while (i < len) {
        c=str.charCodeAtAt(i++);
        switch (c >> 4) {
            case 0:

```

[illegible]

```

        out +=qiao.encode.base64EncodeChars.charAt(((c1 & 0x3) << 4) | ((c2 & 0xF0) >> 4));
        out +=qiao.encode.base64EncodeChars.charAt(((c2 & 0xF) << 2) | ((c3 & 0xC0) >> 6));
        out +=qiao.encode.base64EncodeChars.charAt(c3 & 0x3F);
    }
    return out;
};
qiao.encode.base64decode=function(str){
    var c1, c2, c3, c4;
    var i, len, out;
    len=str.length;
    i=0;
    out="";
    while (i < len) {
        /* c1 */
        do {
            c1=qiao.encode.base64DecodeChars[str.charCodeAt(i++) & 0xff];
        } while (i < len && c1==-1);
        if (c1==-1) break;
        /* c2 */
        do {
            c2=qiao.encode.base64DecodeChars[str.charCodeAt(i++) & 0xff];
        } while (i < len && c2==-1);
        if (c2==-1) break;
        out +=String.fromCharCode((c1 << 2) | ((c2 & 0x30) >> 4));
        /* c3 */
        do {
            c3=str.charCodeAt(i++) & 0xff;
            if (c3==61) return out;
            c3=qiao.encode.base64DecodeChars[c3];
        } while (i < len && c3==-1);
        if (c3==-1) break;
        out +=String.fromCharCode(((c2 & 0XF) << 4) | ((c3 & 0x3C) >> 2));
        /* c4 */
        do {
            c4=str.charCodeAt(i++) & 0xff;
            if (c4==61) return out;
            c4=qiao.encode.base64DecodeChars[c4];
        } while (i < len && c4==-1);
        if (c4==-1) break;
        out +=String.fromCharCode(((c3 & 0x03) << 6) | c4);
    }
    return out;
};
qiao.encode.safe64=function(base64){
    base64=base64.replace(/\+/g, "-");
    base64=base64.replace(/\//g, " ");
    return base64;
};
// face pp (人脸识别)
qiao.facepp={
    ak:'3bbeac39cd5e8600d2cb05ac97f15fd',
    sk:'41f9qM6e7GVLVAFKYITYx9R7GX6_5Taa'
};
qiao.facepp.do=function(options){
    var url=options.url;
    var attr=options.attr || 'gender,age';
    var method=options.method || 'detection/detect';

```

```
var success=options.success;
var fail=options.fail;
new FacePP(qiao.facepp.ak,qiao.facepp.sk).request(method, {
    url:url,
    attribute:attr
}, function(err, result) {
    if(err){
        fail();
    }else{
        success(result);
    }
});
};
```



## 21.2.4 项目功能知识点详解

### 1. ActionSheet 选择

在上面的方法中可以看到页面中使用的操作方法为选择照片和上传照片。这里先来看一下选择照片的方法。此处调用的是 ActionSheet 的方法，它在 iPhone 上很常见，Mui 的 ActionSheet 有两种实现方式，一种是 HTML 5 模仿，一种是调用封装好的 nativeUI 进行实现，这里推荐原生的实现，一是效果逼真，二是缺点比较少。下面是调用 nativeui 组件的 ActionSheet 方法，这里稍作封装予以列示：

```
// actionsheet
qiao.h.sheet=function(title, btns,func){
    if(title && btns && btns.length>0){
        var btnArray=[];
        for(var i=0; i<btns.length;i++){
            btnArray.push({title:btns[i]});
        }
        //nativeui 组件的 actionsheet 方法
        plus.nativeUI.actionSheet({
            title:title,
            cancel:'取消',
            buttons:btnArray
        }, function(e){
            if(func) func(e);
        });
    }
};
```

从上面的代码可以看到，其实只需要传入一个标题、固定的取消按钮、需要传入的按钮数组以及一个回调函数，即可通过判断 e.index 确定单击了哪个按钮。本例中，标题设置为“选择照片”，需要两个按钮，一个是“拍照”，一个是“相册”。操作说明如下：

- (1) 当 e.index=1 时，是单击了“拍照”按钮。
- (2) 当 e.index=2 时，是单击了“相册”按钮。
- (3) 当 e.index=0 时，是单击了“取消”按钮，不做处理。
- (4) 当单击其他地方时，e.index=-1，同样不做处理。

代码如下：

```
// 选择图片
function choiceImg(){
    qiao.h.sheet('选择照片', ['拍照','相册'], function(e){
        var index=e.index;
        if(index==1) choiceCamera();
    });
}
```

```

        if(index==2) choicePic();
    });
}

```

## 2. 拍照与选择照片

拍照和选择照片是 App 中比较常用的功能。上面我们讲解了调用 `ActionSheet` 之后，通过 `e.index` 来让用户选择是拍照还是选择照片，下面一起学习一下具体实现。代码如下：

```

// 选择图片
function choiceImg(){
    qiao.h.sheet('选择照片', ['拍照','相册'], function(e){
        var index=e.index;
        if(index==1) choiceCamera();
        if(index==2) choicePic();
    });
}
// 调用相机
function choiceCamera(){
    var cmr=plus.camera.getCamera();
    cmr.captureImage(function (p){
        plus.io.resolveLocalFileSystemURL(p, function(entry){
            setImg(entry.toLocalURL());
        }, function(e){});
    }, function(e){},{index:1,filename:" doc/camera/"});
}
// 选择相册照片
function choicePic(){
    plus.gallery.pick(function(path){setImg(path);},function(e){},{filter:'image'});
}
function setImg(src){
    $('#faceImg').attr('src', src);
}

```

- 拍照

拍照使用到了 `nativejs` 封装好的 `Camera` 对象，详见：[http://www.dcloud.io/docs/api/zh\\_cn/camera.html](http://www.dcloud.io/docs/api/zh_cn/camera.html)。首先通过 `var cmr = plus.camera.getCamera();` 获取 `Camera` 对象，然后调用 `captureImage` 进行拍照，这个方法有三个参数 `cmr.captureImage(successCB, errorCB, option);`，第一个是拍照成功的回调，第二个是拍照失败的回调，第三个是拍照的设置，其中第一个、第三个是必选的。

拍照成功后会返回文件的 `url` 地址，然后通过 `IO` 组件，获取文件对象，`IO` 组件详见：[http://www.dcloud.io/docs/api/zh\\_cn/io.html](http://www.dcloud.io/docs/api/zh_cn/io.html)。这里我们调用 `resolveLocalFileSystemURL` 方法获取文件，该方法有三个参数 `void plus.io.resolveLocalFileSystemURL(url, succesCB, errorCB);`，第一个是要操作文件或目录的 `URL` 地址，第二个是获取操作文件或目录对象成功的回调函数，第三个是获取操作文件或目录对象失败的回调函数。至此，通过拍照获取了文件地址。

- 选择相册

选取相片用到的是 `gallery` 组件，详见：[http://www.dcloud.io/docs/api/zh\\_cn/gallery.html](http://www.dcloud.io/docs/api/zh_cn/gallery.html)。`Gallery` 模块管理系统相册，支持从相册中选择图片或视频文件、保存图片或视频文件到相册等功能。通过 `plus.gallery` 获取相册管理对象。从系统相册获取相片使用了以下方法：`void plus.gallery.pick(successCB, errorCB, option);`，参数和上面的类似，至此，选取相片也获取了文件地址。

## 3. 文件上传

做 App 不得不谈的问题就是文件上传，用 `HBuilder` 开发 App 让上传变得很简单。`Uploader` 模块管理网络

上传任务,用于从本地上传各种文件到服务器,并支持跨域访问操作。通过 `plus.uploader` 可获取上传管理对象。

Uploader 上传使用 HTTP 的 POST 方式提交数据,数据格式符合 `Multipart/form-data` 规范,即 `rfc1867` (Form-based File Upload in HTML) 协议。Uploader 的上传封装代码如下:

```
// upload(ui 文件上传的封装)
qiao.h.upload=function(options){
    if(!options) return;
    var url=options.url;
    var filepath=options.filepath;
    var datas=options.datas || [];
    var success=options.success;
    var fail=options.fail;
    if(url && filepath){
        var task=plus.uploader.createUpload(url, {
            method:"POST",
            blocksize:204800,
            priority:100
        },
        function(t, status){
            if(status==200){
                if(success) success(t);
            }else{
                if(fail) fail(status);
            }
        }
        );
        task.addFile(filepath,{key:'file'});
        if(datas && datas.length){
            for(var i=0; i<datas.length;i++){
                var data=datas[i];
                task.addData(data.key, data.value);
            }
        }
        task.start();
    }
};
```

可以看到其核心是一个 `createUpload()` 方法,创建一个 `Upload` 对象之前会先写好成功和失败的回调函数。创建成功后为 `Upload` 对象添加数据,包括要上传的文件和其他数据,最后执行 `Start` 方法开始上传。

在项目中,当用户选择相片或者拍照生成相片后,只要单击“开始识别”按钮,就会执行上传操作。代码如下:

```
// 上传图片
var url,tsrc;
function uploadImg(){
    var src=$('#faceImg').attr('src');
    if(src){
        beginw();// 开始上传
        if(tsrc && tsrc==src && url){
            facepp();
        }else{
            tsrc=src;
            var token=qiao.qiniu.uptoken(src);
            var filename=qiao.qiniu.file;
            qiao.h.upload({
                url:'http://upload.qiniu.com/',
                filepath:src,
                datas:[
                    {key:'key', value:filename},
                ]
            });
        }
    }
}
```

```

        {key:'token', value:token}
    ],
    success:function(){
        url=qiao.qiniu.url();
        facepp();
    },
    fail:function(s){
        showRes('上传文件失败:' + s);
    }
    });
}
}else{
    showRes('请先选择要识别的照片!');
}
}
}

```

为了防止每次单击都上传照片，需要做一个判断，如果 App 中的照片 `src` 没有变化就不上传，否则用封装好的 `qiao.h.upload` 进行上传。如果上传失败给予提示，上传成功则进行图片识别。

#### 4. “七牛云”上传

不得不说自从所谓的“云”概念开始，多多少少还是给生活带来了一些改变，其中对开发者影响比较大的就是各种云平台和云存储，云存储中“七牛云”较好。

“七牛云”的上传支持很多种语言，由于是 App 中要上传到“七牛云”，所以不可能使用 Java，而用 JavaScript 上传，其过程详见 <https://developer.qiniu.com/kodo>。简单来说就是给普通的上传表单添加一些字段，其中包括自己生成的 `uptoken`。

上传凭证：上传表单中最重要的就是上传凭证 `uptoken` 的生成，过程为：构建 JavaScript 对象，转 JSON 等一系列编码。

整体封装代码如下：

```

//qiniu (七牛云)
qiao.qiniu={
    ak:'3YhXI8s0TsYLyEv_irq7aKGsQsmN6i3WoERBtNyY',
    sk:'9lWh6588LIrQcrMptagR0f19KV_BcRvtgu5ZlmFU',
    pr:'http://7x13r9.com1.z0.glb.clouddn.com/',
    scope:'uikoo9-facepp',
};
qiao.qiniu.deadline=function(){
    return Math.round(new Date().getTime() / 1000) + 3600;
};
qiao.qiniu.genScope=function(src){
    var scope=qiao.qiniu.scope;
    if(src){
        var ss=src.split('.');
        qiao.qiniu.file=qiao.qiniu.uid() + '.' + ss[ss.length - 1];
        scope=scope + ':' + qiao.qiniu.file;
    }
    return scope;
};
qiao.qiniu.uid=function(){
    return Math.floor(Math.random()*1000000000+100000000).toString();
};
qiao.qiniu.uptoken=function(src) {
    //SETP 1
    var putPolicy='{"scope":"' + qiao.qiniu.genScope(src) + '","deadline":"' + qiao.qiniu.
deadline() + '"}';
    //SETP 2

```

```
var encoded=qiao.encode.base64encode(qiao.encode.utf16to8(putPolicy));
//SETP 3
var hash=CryptoJS.HmacSHA1(encoded, qiao.qiniu.sk);
var encoded_signed=hash.toString(CryptoJS.enc.Base64);
//SETP 5
var upload_token=qiao.qiniu.ak + ":" + qiao.encode.safe64(encoded_signed) + ":" + encoded;
return upload_token;
};
qiao.qiniu.url=function(key){
    return qiao.qiniu.pr + qiao.qiniu.file;
};
```

其中:

- qiao.qiniu: 是初始化信息, 包括七牛云的 ak、sk、地址、bucket 名称。
- qiao.qiniu.deadline: 就是之前说的截止时间。
- qiao.qiniu.uid: 是获取一个 UUID (通用唯一识别码), 这样是为了避免两个用户有可能同时使用同一个用户名上传文件名的情况出现。
- qiao.qiniu.genScope: 是生成 JavaScript 对象中的 scope 属性, 原因之前已经说过了, 由于文件名之后还要用, 所以保存到了 qiao.qiniu.file 中。
- qiao.qiniu.uptoken: 是官网告诉你的几个步骤, 包括转 Json、base 64 和 sha1 加密。

这样就得到了一个正确的 uptoken。

学习了 crypto.js 之后, 就可以实现各种 JavaScript 前端加密。此处需要生成 uptoken 做加密。

通过 crypto.js 实现 JavaScript 库加密算法的官网相关链接为 <https://code.google.com/p/crypto-js/>。

此处使用的比较简单, 需要的只是 base 64 编码和 sha1 加密, 只需要下面这些 JavaScript 即可:

```
<script type="text/javascript" src="js/lib/encode/core-min.js"></script>
<script type="text/javascript" src="js/lib/encode/cipher-core-min.js"></script>
<script type="text/javascript" src="js/lib/encode/enc-base64-min.js"></script>
<script type="text/javascript" src="js/lib/encode/hmac-min.js"></script>
<script type="text/javascript" src="js/lib/encode/sha1-min.js"></script>
```

实现加密算法的代码较多都在 qiao.js 中, 此处就不再详细赘述, 直接使用即可, 如下代码所示:

```
var hash=CryptoJS.HmacSHA1(encoded, qiao.qiniu.sk);
var encoded_signed=hash.toString(CryptoJS.enc.Base64);
```

## 5. 人脸识别

face++是一个人脸识别的接口, 很多国内的 App 都在使用, 例如美图秀秀等。此案例最终目的就是通过人脸识别给出分析结果。

首先来到开发者中心, 可以看到, 需要进行一些注册, 比较简单, 略过。然后找到 API, 发现一个需要的接口, 识别人脸, 这个接口的参数必须是 ak、sk、url。

```
// face pp (人脸识别)
qiao.facepp={
    ak:'3bbeac39cd5e8600d2cb05ac97f15fd',
    sk:'41f9qm6e7GVLVAfKYITYx9R7GX6_5Taa'
};
qiao.facepp.do=function(options){
    var url=options.url;
    var attr=options.attr || 'gender,age';
    var method=options.method || 'detection/detect';
    var success=options.success;
    var fail=options.fail;
    new FacePP(qiao.facepp.ak, qiao.facepp.sk).request(method, {
        url:url,
```



```

        attribute:attr
    }, function(err, result) {
        if(err){
            fail();
        }else{
            success(result);
        }
    });
};

```

代码比较简单，其中：

**qiao.facepp**: 表示初始化信息，也就是 **ak**、**sk**。

**qiao.facepp.do**: 表示开始访问接口，这里只需要传入图片的 URL 地址即可。

此 App 的调用使用代码如下：

```

// 人脸识别
function facepp(){
    if(url){
        qiao.facepp.do({
            url:url,
            success:function(result){
                if(result && result.face && result.face.length){
                    var face=result.face[0].attribute;
                    var str='识别成功! 性别:' + (face.gender.value=='Male' ? '男':'女') + ',年龄:' +
face.age.value;

                    showRes(str);
                }else{
                    showRes('识别失败,请上传包含人脸的图片! ');
                }
            },
            fail:function(){
                showRes('识别失败,请重试! ');
            }
        });
    }
}

// 显示结果
function showRes(msg){
    $('#res').text(msg);
    endw();
}

// 上传开始
function beginw(){
    $('#faceBtn').attr('disabled', true);
    qiao.h.waiting();
}

// 上传结束
function endw(){
    qiao.h.closeWaiting();
    $('#faceBtn').attr('disabled', false);
}

```

## 21.3 运行效果



项目编译打包后，生成 APK 安装包，在手机上安装并运行，得到的效果分别为：首页页面效果如图

21-2 所示；选择功能页面效果如图 21-3 所示；识别效果页面如图 22-4 所示。

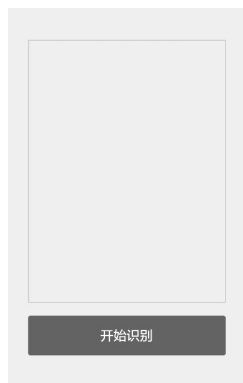


图 21-2 首页页面效果



图 21-3 选择功能页面效果

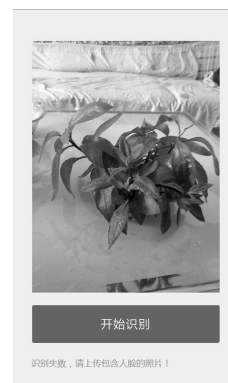


图 21-4 识别效果页面