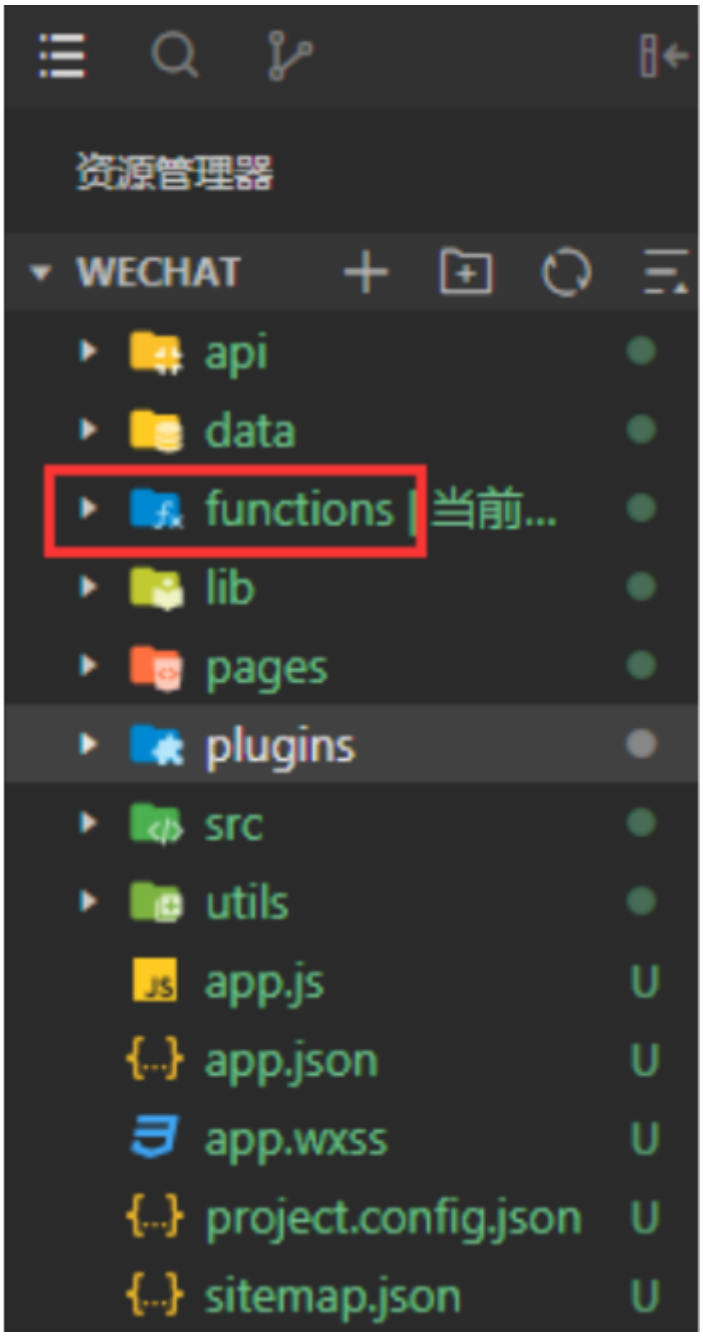
1、实现方式

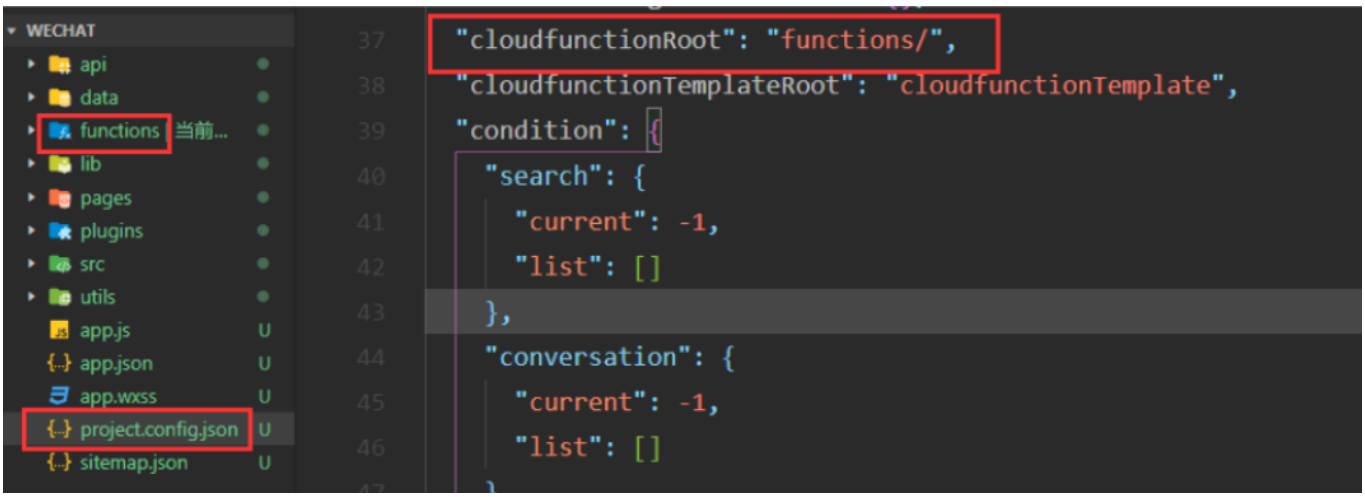
前端调用相机组件实现人脸在线采集，
通过采集到的人脸图片的base64字符串调用云开发侧实现的腾讯云人脸识别云函数，
然后将识别结果回调到小程序页面中。

2.实现流程

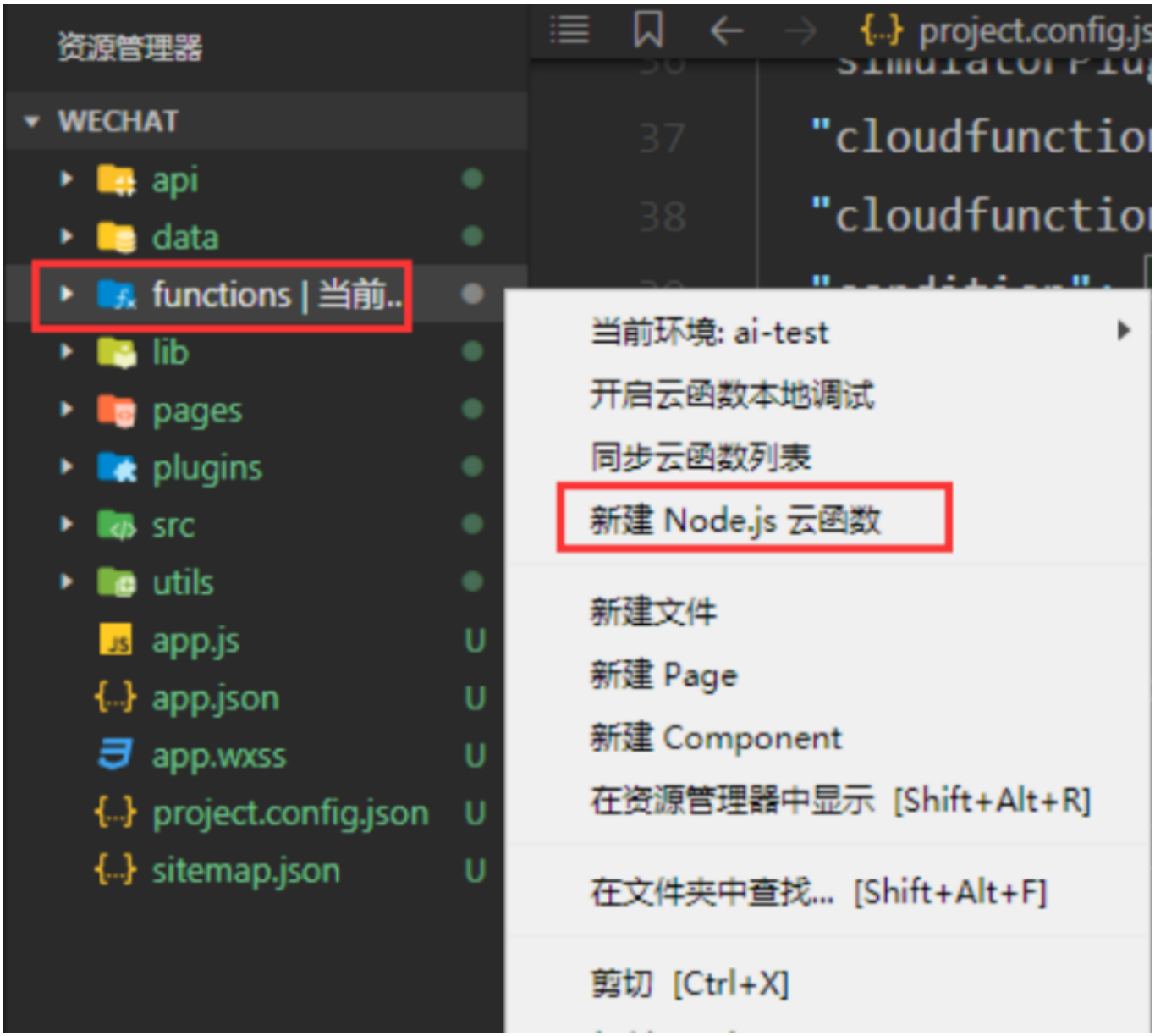
第一步：开通云开发控制台并创建云端项目环境

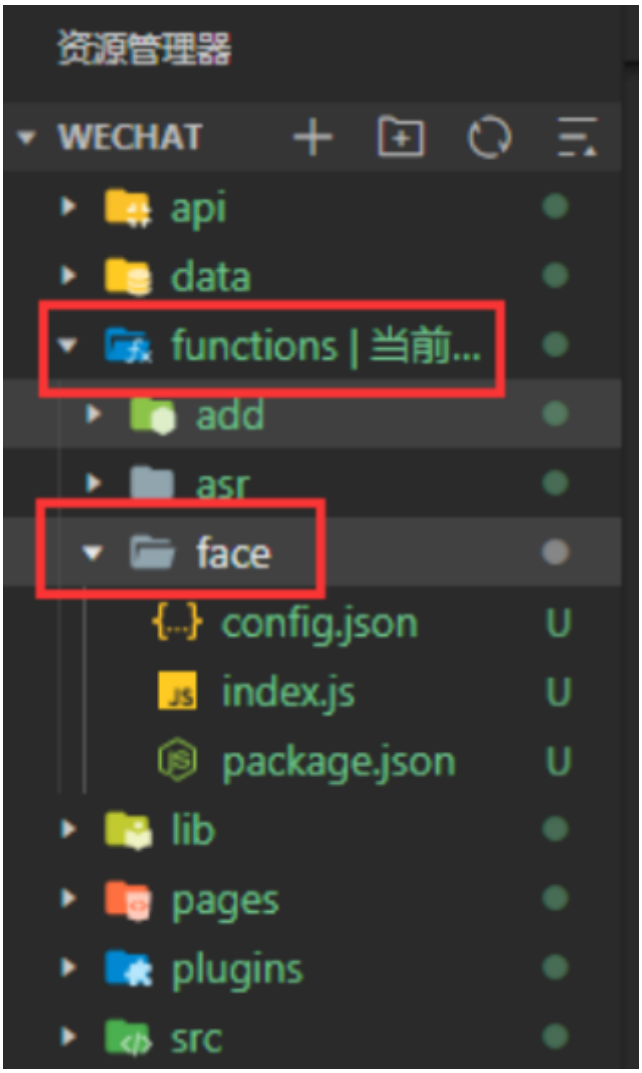
第二步：在小程序项目根目录下创建本地云函数根目录functions，在项目根目录找到project.config.json 文件，新增 cloudfunctionRoot 字段，值为刚才创建的本地云函数根目录名称

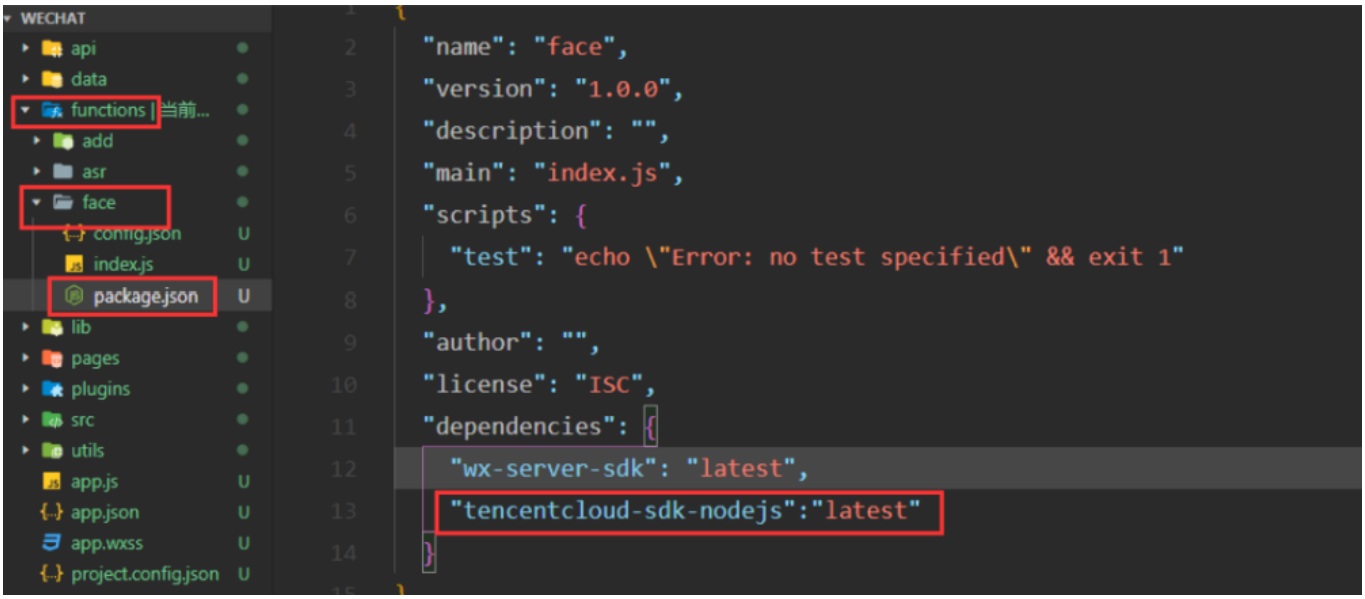




第三步：创建人脸识别云函数并配置tencentcloud-sdk-nodejs依赖


资源管理器
WECHAT
api
data
functions | 当前..
lib
pages
plugins
src
utils
app.js
app.json
app.wxss
project.config.json
sitemap.json
project.config.js
37
"cloudfunctio
38
"cloudfunctio
当前环境: ai-test
开启云函数本地调试
同步云函数列表
新建 Node.js 云函数
新建文件
新建 Page
新建 Component
在资源管理器中显示 [Shift+Alt+R]
在文件夹中查找... [Shift+Alt+F]
剪切 [Ctrl+X]

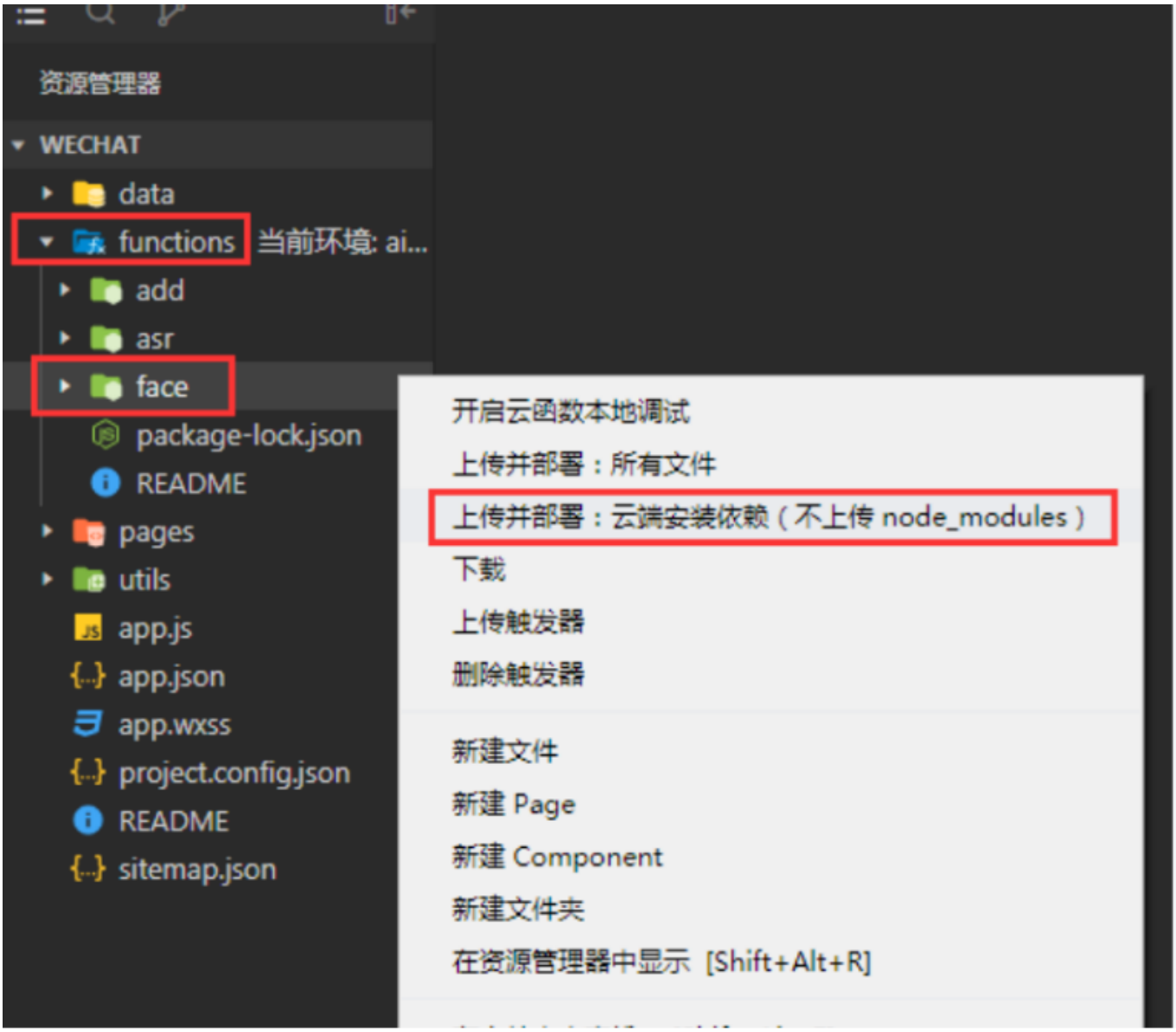




第四步：在人脸识别云函数目录下的入口文件index.js中实现人脸识别-人脸检测与分析的API调用Demo，然后上传Demo至云端

```
// 云函数入口文件
const cloud = require('wx-server-sdk') // 引入云开发服务的内核SDK
cloud.init( //初始化一个'wx-server-sdk' SDK 实例
    {
        env: 'cloud1-5gui9ks84b86e290' // 开通云开发服务后创建的云环境的环境ID（默认可以创建两个ID）
    }
```

```
)


// 云函数入口函数
exports.main = async (event, context) => {
    const tencentcloud = require("tencentcloud-sdk-nodejs");
    const IaiClient = tencentcloud.iai.v20200303.Client;
    const clientConfig = {
        credential: {
            secretId: "自己腾讯云的secretId",
            secretKey: "自己腾讯云的secretKey",
        },
        region: "ap-guangzhou",
        profile: {
            httpProfile: {
                endpoint: "iai.tencentcloudapi.com",
            },
        },
    };

    const client = new IaiClient(clientConfig);
    let base64Data = event.x //接收客户端post的x参数，值类型为base64字符串
    const params = {
        Image: base64Data,
        NeedFaceAttributes: 1
    } // 定义SDK的请求参数字典

    // params = JSON.stringify(params) // 转换为json字符串
    return new Promise((resolve, reject) => { // 通过Promise容器来接收异步API的回调，然后通过当前脚本返回给客户端
        client.DetectFace(params, function (errMsg, response) { // 此接口是异步的，那么当前脚本无法对外直接访问接口返回值
            if (errMsg) {
                resolve({
                    "Result": errMsg
                })
            }
            // resp = response.to_json_string()
            resolve({
                "Result": response
            })
        });
    })
}
```



第五步：小程序中实现人脸图片在线采集页面

在小程序公共配置文件app.json中，添加页面生成参数

```
"pages/camerac/camerac",
```

---

camerac.wxml

```
<!--pages/camerac/camerac.wxml-->
<camera device-position="front" flash="off" binderror="error" style="width:100%; height: 300px;">
</camera>
<button type="primary" bindtap="takePhoto">拍照</button>
<view wx:if="{{ FaceInfos['0']['FaceAttributesInfo']['Gender'] > 50}}">性别：男</view>
<view wx:if="{{ FaceInfos['0']['FaceAttributesInfo']['Gender'] < 50}}">性别：女</view>
```

```
<view>年龄：{{ FaceInfos['0']['FaceAttributesInfo']['Age'] }}</view>
<view wx:if="{{ FaceInfos['0']['FaceAttributesInfo']['Expression'] == 0 }}">表情：正常</view>
<view wx:if="{{ FaceInfos['0']['FaceAttributesInfo']['Expression'] < 50 }}">表情：微笑</view>
<view wx:if="{{ FaceInfos['0']['FaceAttributesInfo']['Expression'] > 50 }}">表情:大笑</view>
<view wx:if="{{ FaceInfos['0']['FaceAttributesInfo']['Beauty'] == 0 }}">魅力值：一般</view>
<view wx:if="{{ FaceInfos['0']['FaceAttributesInfo']['Beauty'] < 50 }}">魅力值：有点迷人</view>
<view wx:if="{{ FaceInfos['0']['FaceAttributesInfo']['Beauty'] > 50 }}">魅力值：偶像级</view>
<view>请求的图片宽度：{{ ImageWidth }}</view>
<view>请求的图片高度：{{ ImageHeight }}</view>
<view>人脸框左上角横坐标：{{ FaceInfos['0']['X'] }}</view>
<view>人脸框左上角纵坐标：{{ FaceInfos['0']['Y'] }}</view>
<view>人脸框宽度:{{ FaceInfos['0']['Width'] }}</view>
<view>人脸框高度：{{ FaceInfos['0']['Height'] }}</view>
<image mode="widthFix" src="{{src}}"></image>
```

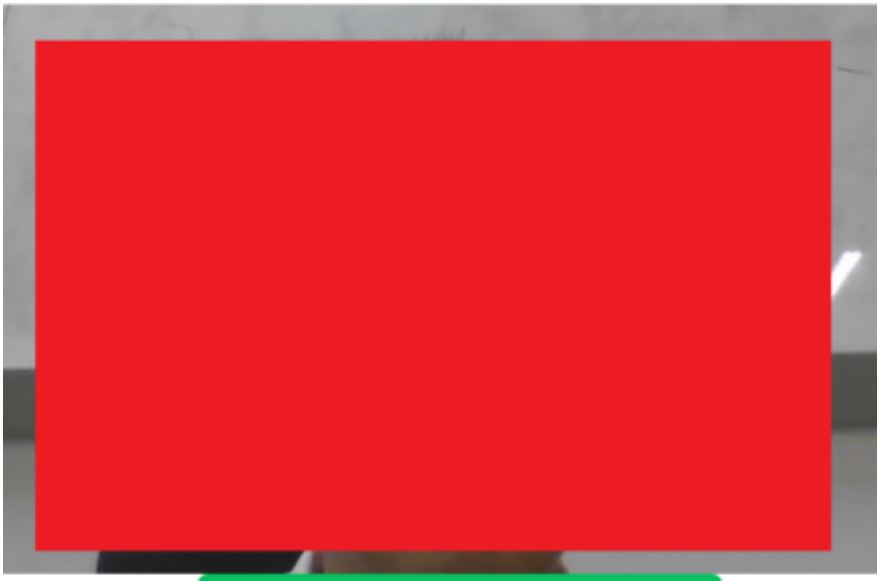
---

camerac.js

```
Page({
    takePhoto() { // 定义绑定的事件处理函数
        var that=this; // 指定this到that，避免绑定的对象改变
        const ctx = wx.createCameraContext() // 创建 camera 上下文 CameraContext 对象
        ctx.takePhoto({
            quality: 'high',
            success: (res) => {
                this.setData({
                    src: res.tempImagePath
                })
                var base64=wx.getFileSystemManager().readFileSync(res.tempImagePath,'base64')
                wx.cloud.init()
                wx.cloud.callFunction({
                    // 云函数名称
                    name: 'face',
                    // 传给云函数的参数
                    data: {
                        x:base64 // 传图片的base64字符串
                    },
                    success: function(res) {
                        that.setData({
                            ImageWidth: res.result.Result.ImageWidth+"px",
                            ImageHeight: res.result.Result.ImageHeight+"px",
                            FaceInfos: res.result.Result.FaceInfos,
                        })
                    },
                    fail: console.error
                })
            }
        })
    },
    error(e) {
        console.log(e.detail)
    }
})
```

camerac.css

```
.photo {
    display: flex;
    margin-top: 10px;
    height: 100px;
}

.ph {
    border: 1px dashed #909090;
    margin-right: 30px;
    width: 80px;
    height: 60px;
}

.phzz {
    border: 1px dashed #909090;
    margin-right: 70px;
    margin-left: 70px;
    width: 100px;
    height: 60px;
}

.phright{
    border: 1px dashed #909090;
    margin-left: 20px;
    width: 80px;
    height: 60px;
}

.textp{
    margin-left: 70px;
    font-size: 14px;
}

.text{
    margin-left: 25px;
    font-size: 14px;
}

.text2{
    margin-left: 80px;
    font-size: 14px;
}

.text3{
    margin-left: 98px;
    font-size: 14px;
}
```



拍照

年龄:
请求的图片宽度:
请求的图片高度:
人脸框左上角横坐标:
人脸框左上角纵坐标:
人脸框宽度:
人脸框高度: