uniapp 安卓百度离线人脸识别UTS原生组件

目　录

使用方法

介绍

安卓百度离线人脸识别UTS原生组件，支持在线激活，单个人脸注册，批量注册，相机人脸注册，包含多个识别组件，支付模式识别组件，闸机模式识别组件，考勤模式识别组件，金融活检模式识别组件，人证核检组件等，识别结果支持返回当前识别的人脸图片base64格式，插件UTS开发，支持uniapp x

使用流程

1. 在线激活
2. 初始化SDK
3. 注册人脸
4. 识别人脸

SDK版本

FaceSDK_8.2_20240308

联系作者

关注微信公众号可联系作者

官方文档

https://ai.baidu.com/ai-doc/FACE/pk37c1mqu

插件地址

https://ext.dcloud.net.cn/plugin?id=20343

权限

1. android.permission.CAMERA
2. android.permission.WRITE_EXTERNAL_STORAGE
3. android.permission.READ_EXTERNAL_STORAGE

示例文件下载

1. 插件首页下载（推荐）

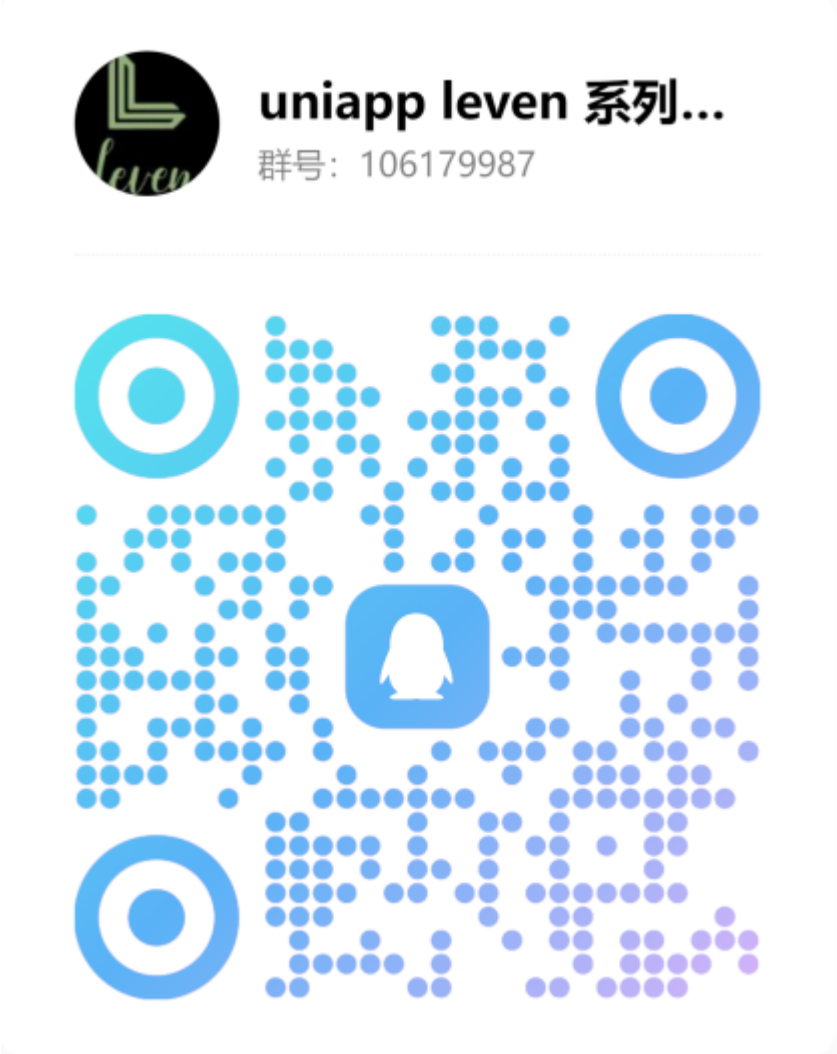
   > 通过“使用 HBuilderX 导入示例项目”可直接创建uniapp项目

2. 本地下载，点击这里即可下载

演示程序

使用方法

加群下载演示程序



API用法

- 用法

  在需要使用插件的页面加载以下代码

```
import * as module from "@/uni_modules/leven-uts-baiduFace"
```

- 示例

```
<template>
  <view>
    <uni-card title="安卓百度人脸识别UTS原生插件-人脸管理">
      <button type="primary" @click="requestPermissions">申请插件权限</button>
      <button type="primary" @click="getDeviceId">获取设备指纹数据</button>
      <button type="primary" @click="activeOnline">在线激活</button>
      <button type="primary" @click="initSdk">初始化SDK</button>
      <button type="primary" @click="getLicenseData">获取激活码信息</button>
      <button type="primary" @click="register">单个人脸注册</button>
      <button type="primary" @click="batchRegister">批量注册</button>
```

```
      <button type="primary" @click="deleteFace">删除人脸</button>
      <button type="primary" @click="getAllFace">获取所有的人脸</button>
      <button type="primary" @click="getFaceById">根据ID获取人脸</button>
      <button type="primary" @click="getFaceCount">获取注册的人脸数量</button>
      <button type="primary" @click="clearFace">清空人脸</button>
      <button type="primary" @click="logStr = ''">清空日志</button>
    </uni-card>
    <uni-card class="uni-card-box" title="日志">
      <view><text style="font-size: 14px; flex-wrap: wrap;">{{logStr}}</text><
    </uni-card>
  </view>
</template>

<script>
  // 批量注册人脸的文件
  import {
    faceList
  } from "@/utils/face.js"
  import * as module from "@/uni_modules/leven-uts-baiduFace"
  export default {
    data() {
      return {
        logStr: "",
      }
    },
    methods: {
      // 清空人脸
      clearFace() {
        module.clearFace(res => {
          this.writeLog(JSON.stringify(res))
        })
      },
      // 获取注册的人脸数量
      getFaceCount() {
        module.getFaceCount(res => {
          this.writeLog(JSON.stringify(res))
        })
      },
      // 根据ID获取人脸
      getFaceById() {
        module.getFaceById({
          //用户id
          id: "1234"
        }, res => {
          this.writeLog(JSON.stringify(res))
        })
      },
      // 获取所有的人脸
      getAllFace() {
        module.getAllFace(res => {
```

```
        this.writeLog(JSON.stringify(res))
      })
    },
    // 删除人脸
    deleteFace() {
      module.deleteFace({
        //用户id
        id: "1234"
      }, res => {
        this.writeLog(JSON.stringify(res))
      })
    },
    //批量注册
    batchRegister() {
      let list = faceList.map(item => {
        let obj = {
          id: item._id,
          name: item.name,
          url: item.pic
        }
        return obj;
      })
      module.batchRegister({
        list: list
      }, res => {
        this.writeLog(JSON.stringify(res))
      })
    },
    //单个注册人脸
    register() {
      module.register({
        //注册地址，支持本地和网络地址
        // url: "http://www.yeyuboke.com/svga/b.jpg",
        url: "/storage/emulated/0/DCIM/Camera/IMG_20240902_141732.jpg",
        //用户id
        id: "1234",
        //用户名称
        name: "leven1"
      }, res => {
        this.writeLog(JSON.stringify(res))
      })
    },
    //获取激活码信息
    getLicenseData() {
      module.getLicenseData(res => {
        this.writeLog(JSON.stringify(res))
      });
    },
    //初始化人脸库
    initSdk() {
```

```
          module.initSdk(res => {
            this.writeLog(JSON.stringify(res))
          });
        },
        //在线激活
        activeOnline() {
          module.activeOnline({
            //激活码
            licenseKey: "9XHX-VKLY-3XBT-XVSE"
            // licenseKey: "XWSX-CCFB-47CX-LS9R"
          }, res => {
            this.writeLog(JSON.stringify(res))
          });
        },
        //获取设备的指纹数据，用于离线激活
        getDeviceId() {
          module.getDeviceId(res => {
            this.writeLog(JSON.stringify(res))
          });
        },
        //申请文件权限
        requestPermissions() {
          module.requestPermissions(res => {
            console.log(JSON.stringify(res))
          });
        },
        // 写日志
        writeLog(str) {
          console.log(str)
        }
      }
    }
</script>

<style>

</style>
```

组件用法

> 组件用法请参考各章节的使用方法

更新日志

- [优化] 组件属性新增参数 识别位图旋转角度
- [优化] 组件属性新增参数 识别结果是否裁剪人脸图片
- [新增] 新增接口【离线激活】

首次发布

申请插件权限

方法名

`requestPermissions`

用法

- 用法如下：

```
module.requestPermissions(res => {
  console.log(JSON.stringify(res))
});
```

- 参数说明

  无

回调

- 示例

```
{
  "data": {
    "grantedList": [
      "android.permission.READ_EXTERNAL_STORAGE",
      "android.permission.WRITE_EXTERNAL_STORAGE"
    ]
  },
  "message": "权限被拒绝",
  "code": -1
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象

申请插件权限

参数名	参数类型	参数描述
data.grantedList	Array	申请的权限列表，**被拒绝的时候会返回**
code	Integer	返回类型，0.成功，其他：失败

获取设备指纹数据

方法名

`getDeviceId`

用法

- 用法如下：

```
module.getDeviceId(res => {
  this.writeLog(JSON.stringify(res))
});
```

- 参数说明

  无

回调

- 示例

```
{
  "data": {
    "deviceId": "11140C1F12FEEB2C52DFBEA35335271512"
  },
  "message": "",
  "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
data.deviceId	String	指纹数据
code	Integer	返回类型，0.成功，其他：失败

在线激活

方法名

`activeOnline`

用法

- 用法如下：

```
module.activeOnline({
  //激活码
  licenseKey: "9XHX-VKLY-3XBT-XVSE"
}, res => {
  this.writeLog(JSON.stringify(res))
});
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
licenseKey	String	是	无	百度申请的激活码

回调

- 示例

```
{
  "data": {},
  "message": "",
  "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象

参数名	参数类型	参数描述
code	Integer	返回类型，0.成功，其他：失败

初始化SDK

方法名

`initSdk`

用法

- 用法如下：

```
module.initSdk(res => {
  this.writeLog(JSON.stringify(res))
});
```

- 参数说明

  无

回调

- 示例

```
{
  "data": {
    "successCount": 2,
    "type": "initDbProgress",
    "progress": 1,
    "finishCount": 2
  },
  "message": "",
  "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
data.successCount	Integer	初始化成功人脸数量

参数名	参数类型	参数描述
data.type	String	初始化类型， startInitDBLoad：开始初始化数据库，initDbProgress：初始化数据库进度， initDbComplete：初始化数据库完成
data.progress	float	初始化进度
data.finishCount	Integer	完成初始化人脸数量
code	Integer	返回类型，0.成功，其他：失败

获取激活码信息

- 用法如下：

- 参数说明

  无

- 示例

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
data.algorithmId	Integer	算法Id
data.deviceId	String	指纹数据
data.expireTime	Integer	过期时间
data.functionList	String	可用函数列表
data.packageName	String	应用包名
data.md5	String	md5值
data.licenseKey	String	秘钥
code	Integer	返回类型，0.成功，其他：失败

单个人脸注册

方法名

用法

- 用法如下：

```
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
url	String	是	无	注册的文件地址，支持网络地址
id	String	是	无	注册用户的id
name	String	是	无	注册用户姓名

回调

- 示例

```
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
data.imagePath	String	注册后的人脸图片路径
code	Integer	返回类型，0.成功，其他：失败

批量注册

方法名

`batchRegister`

用法

- 用法如下：

```
let list = faceList.map(item => {
  let obj = {
    id: item._id,
    name: item.name,
    url: item.pic
  }
  return obj;
})
module.batchRegister({
  list: list
}, res => {
  this.writeLog(JSON.stringify(res))
})
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
list	Array	是	无	注册的人脸数据列表
list.url	String	是	无	注册的文件地址，支持网络地址
list.id	String	是	无	注册用户的id
list.name	String	是	无	注册用户姓名

回调

- 示例

批量注册


```
{
  "data": {
    "isSuccess": true,
    "user": {
      "name": "李凯迪",
      "url": "http://sjxxbp.yach-sh.cn/5a7ba5c157e9418ea143be1ed326e94d.jpg",
      "id": "a5782af7653f223b012439c74a5b503c",
      "imagePath": "/storage/emulated/0/levenBaiduFace/a5782af7653f223b012439c
    },
    "failed": 0,
    "type": "registerProgress",
    "progress": 1,
    "count": 33,
    "success": 33
  },
  "message": "",
  "code": 0
}
```


- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
data.isSuccess	Boolean	是否注册成功
data.user	Object	注册后的用户信息
data.user.name	String	注册后的用户姓名
data.user.url	String	注册的地址
data.user.id	String	注册的id
data.user.imagePath	String	注册后的用户人脸图片路径
data.failed	Integer	失败数量
data.type	Integer	注册类型：registerStart：开始注册，registerProgress：注册中，registerComplete：注册完成
data.progress	float	注册进度
data.count	Integer	总数量
data.success	Integer	成功数量

参数名	参数类型	参数描述
code	Integer	返回类型，0.成功，其他：失败

删除人脸

方法名

用法

- 用法如下：

```
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
id	String	是	无	注册用户的id

回调

- 示例

```
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

获取所有的人脸

方法名

`getAllFace`

用法

- 用法如下：

```
module.getAllFace(res => {
  this.writeLog(JSON.stringify(res))
});
```

- 参数说明

  无

回调

- 示例


```
{
  "data": {
    "list": [
      {
        "userName": "李凯迪",
        "userId": "a5782af7653f223b012439c74a5b503c",
        "imagePath": "/storage/emulated/0/levenBaiduFace/a5782af7653f223b01243
      },
      {
        "userName": "鑫",
        "userId": "a5782af7653f223b012439c74a5b503b",
        "imagePath": "/storage/emulated/0/levenBaiduFace/a5782af7653f223b01243
      },
      {
        "userName": "邓海秀",
        "userId": "a5782af7653f223b012439c627451c6c",
        "imagePath": "/storage/emulated/0/levenBaiduFace/a5782af7653f223b01243
      },
      {
        "userName": "郭宁",
```

```
        "userId": "a5782af7653f223b012439c50cc29584",
        "imagePath": "/storage/emulated/0/levenBaiduFace/a5782af7653f223b01243
      },
      {
        "userName": "谭晓岚",
        "userId": "a5782af7653f223b012439c4211bc292",
        "imagePath": "/storage/emulated/0/levenBaiduFace/a5782af7653f223b01243
      },
      {
        "userName": "虾滑",
        "userId": "a5782af7653f223b012439c34424dd09",
        "imagePath": "/storage/emulated/0/levenBaiduFace/a5782af7653f223b01243
      },
      {
        "userName": "盐西",
        "userId": "a5782af7653f223b012439c21d454eb5",
        "imagePath": "/storage/emulated/0/levenBaiduFace/a5782af7653f223b01243
      },
      {
        "userName": "张愈靓",
        "userId": "a5782af7653f223b012439c10868d744",
        "imagePath": "/storage/emulated/0/levenBaiduFace/a5782af7653f223b01243
      },
      {
        "userName": "钟华",
        "userId": "a5782af7653f223b012439c03c0f9671",
        "imagePath": "/storage/emulated/0/levenBaiduFace/a5782af7653f223b01243
      },
      {
        "userName": "张权红",
        "userId": "a5782af7653f223b012439bf6e399e52",
        "imagePath": "/storage/emulated/0/levenBaiduFace/a5782af7653f223b01243
      },
      {
        "userName": "饶应颖",
        "userId": "a5782af7653f223b012439be0925236b",
        "imagePath": "/storage/emulated/0/levenBaiduFace/a5782af7653f223b01243
      },
      {
        "userName": "飞鱼",
        "userId": "a5782af7653f223b012439bd413f195b",
        "imagePath": "/storage/emulated/0/levenBaiduFace/a5782af7653f223b01243
      },
      {
        "userName": "吴",
        "userId": "a5782af7653f223b012439bc512f7a51",
        "imagePath": "/storage/emulated/0/levenBaiduFace/a5782af7653f223b01243
      },
      {
        "userName": "穆建华",
```

获取所有的人脸


```
        "userId": "a5782af7653f223b012439bb1df53133",
        "imagePath": "/storage/emulated/0/levenBaiduFace/a5782af7653f223b01243
      },
      {
        "userName": "赵丽彤",
        "userId": "a5782af7653f223b012439ba436bc7d3",
        "imagePath": "/storage/emulated/0/levenBaiduFace/a5782af7653f223b01243
      },
      {
        "userName": "黄静",
        "userId": "a5782af7653f223b012439b93323a098",
        "imagePath": "/storage/emulated/0/levenBaiduFace/a5782af7653f223b01243
      },
      {
        "userName": "卜丽霞",
        "userId": "a5782af7653f223b012439b84ffc079e",
        "imagePath": "/storage/emulated/0/levenBaiduFace/a5782af7653f223b01243
      },
      {
        "userName": "刘李",
        "userId": "a5782af7653f223b012439b773461c0d",
        "imagePath": "/storage/emulated/0/levenBaiduFace/a5782af7653f223b01243
      },
      {
        "userName": "陈浩",
        "userId": "a5782af7653f223b012439b616653f7e",
        "imagePath": "/storage/emulated/0/levenBaiduFace/a5782af7653f223b01243
      },
      {
        "userName": "王成斌",
        "userId": "a5782af7653f223b012439b5148e5ed9",
        "imagePath": "/storage/emulated/0/levenBaiduFace/a5782af7653f223b01243
      },
      {
        "userName": "吴思丽",
        "userId": "a5782af7653f223b012439b4787a4930",
        "imagePath": "/storage/emulated/0/levenBaiduFace/a5782af7653f223b01243
      },
      {
        "userName": "何小金",
        "userId": "a5782af7653f223b012439b32233f27d",
        "imagePath": "/storage/emulated/0/levenBaiduFace/a5782af7653f223b01243
      },
      {
        "userName": "陈虹帆",
        "userId": "a5782af7653f223b012439b22739efac",
        "imagePath": "/storage/emulated/0/levenBaiduFace/a5782af7653f223b01243
      },
      {
        "userName": "詹泽林",
```

```
        "userId": "a5782af7653f223b012439b14a612af1",
        "imagePath": "/storage/emulated/0/levenBaiduFace/a5782af7653f223b01243
      },
      {
        "userName": "刘彧聪",
        "userId": "a5782af7653f223b012439b06d533bbd",
        "imagePath": "/storage/emulated/0/levenBaiduFace/a5782af7653f223b01243
      },
      {
        "userName": "蒋孟楠",
        "userId": "a5782af7653f223b012439af26a2c8db",
        "imagePath": "/storage/emulated/0/levenBaiduFace/a5782af7653f223b01243
      },
      {
        "userName": "周文斌",
        "userId": "a5782af7653f223b012439ae4b5ff039",
        "imagePath": "/storage/emulated/0/levenBaiduFace/a5782af7653f223b01243
      },
      {
        "userName": "程利琴",
        "userId": "a5782af7653f223b012439ad71918e81",
        "imagePath": "/storage/emulated/0/levenBaiduFace/a5782af7653f223b01243
      },
      {
        "userName": "郑佳萍",
        "userId": "a5782af7653f223b012439ac107297ac",
        "imagePath": "/storage/emulated/0/levenBaiduFace/a5782af7653f223b01243
      },
      {
        "userName": "谯竹君",
        "userId": "a5782af7653f223b012439ab034669e8",
        "imagePath": "/storage/emulated/0/levenBaiduFace/a5782af7653f223b01243
      },
      {
        "userName": "邱晓蓉",
        "userId": "a5782af7653f223b012439aa0ab0ca2e",
        "imagePath": "/storage/emulated/0/levenBaiduFace/a5782af7653f223b01243
      },
      {
        "userName": "杨光",
        "userId": "a5782af7653f223b012439a95390b872",
        "imagePath": "/storage/emulated/0/levenBaiduFace/a5782af7653f223b01243
      },
      {
        "userName": "刘爽",
        "userId": "a5782af7653f223b012439a81f6f3a2c",
        "imagePath": "/storage/emulated/0/levenBaiduFace/a5782af7653f223b01243
      },
      {
        "userName": "leven1",
```

```
        "userId": "1234",
        "imagePath": "/storage/emulated/0/levenBaiduFace/1234.jpg"
      },
      {
        "userName": "leven1",
        "userId": "123456",
        "imagePath": "/storage/emulated/0/levenBaiduFace/123456.jpg"
      }
    ]
  },
  "message": "",
  "code": 0
}
```


- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
data.list	Array	人脸列表
data.list.userName	String	用户名
data.list.userId	Integer	用户id
data.imagePath	String	人脸图片保存地址
code	Integer	返回类型，0.成功，其他：失败

根据ID获取人脸

方法名

`getFaceById`

用法

- 用法如下：

```
module.getFaceById({
  //用户id
  id: "1234"
}, res => {
  this.writeLog(JSON.stringify(res))
})
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
id	String	是	无	注册用户的id

回调

- 示例

```
{
  "data": {
    "userName": "leven1",
    "userId": "1234",
    "imagePath": "/storage/emulated/0/levenBaiduFace/1234.jpg"
  },
  "message": "",
  "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示

根据ID获取人脸

参数名	参数类型	参数描述
data	Object	数据对象
data.userName	String	用户名
data.userId	String	用户id
data.imagePath	String	人脸图片地址
code	Integer	返回类型，0.成功，其他：失败

获取注册的人脸数量

方法名

`getFaceCount`

用法

- 用法如下：

```
module.getFaceCount(res => {
  this.writeLog(JSON.stringify(res))
});
```

- 参数说明

  无

回调

- 示例

```
{
  "data": {
    "count": 35
  },
  "message": "",
  "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
data.count	Integer	人脸数量
code	Integer	返回类型，0.成功，其他：失败

清空人脸

方法名

`clearFace`

用法

- 用法如下：

```
module.clearFace(res => {
  this.writeLog(JSON.stringify(res))
});
```

- 参数说明

  无

回调

- 示例

```
{
  "data": {},
  "message": "",
  "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

离线激活 v1.1.0

方法名

`activeOffline`

> 离线激活文件需放置在sd卡根目录

用法

- 用法如下：

```
module.activeOffline(res => {
  this.writeLog(JSON.stringify(res))
});
```

- 参数说明

  无

回调

- 示例

```
{
  "data": {},
  "message": "",
  "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

使用方法

组件使用

在需要使用插件的页面加载以下代码

```
<leven-uts-baiduFace-CameraRegister ref="refLevenCameraRegister" style="flex:1; he
  @onError="onError">
</leven-uts-baiduFace-CameraRegister>
```

> 组件只能在nvue/uvue页面中使用，不支持vue页面

页面示例代码

```
<template>
  <view>
    <uni-card title="安卓百度人脸识别UTS原生插件-相机人脸注册">
      <view style="flex:1; height: 500px; position: relative;">
        <leven-uts-baiduFace-CameraRegister ref="refLevenCameraRegister" style="f
          @onError="onError">
        </leven-uts-baiduFace-CameraRegister>
      </view>
      <view>
        <button type="primary" @click="register">注册人脸</button>
      </view>
    </uni-card>
    <uni-card class="uni-card-box" title="日志">
      <view><text style="font-size: 14px; flex-wrap: wrap;">{{logStr}}</text></vie
    </uni-card>
  </view>
</template>

<script>
  export default {
    data() {
      return {
        //日志
        logStr: "",
        //组件配置
        config: {
          //RGB预览Y轴转向falese为0，true为180
          rgbRevert: false,
```

```
            //可传入大于50px的数值，小于此大小的人脸不予检测
            minFaceSize: 50,
            //人脸置信度用于表征被检测到的物体是人脸的概率，该阈值设置越高检测越严格，建议在0.3-0
            faceThreshold: 0.5,
            //是否开启属性检测，默认：false
            attribute: false,
            //是否设置质量检测，默认：true
            qualityControl: false,
            //质量检测参数设置，qualityControl为true时有效
            qualityConfig: {
              // 模糊度设置，默认0.8。取值范围[0~1]，0是最清晰，1是最模糊
              blur: 0.8,
              // 光照设置，默认0.8.取值范围[0~1]， 数值越大，光线越强
              illum: 0.8,
              //姿态阈值，默认：15
              gesture: 15,
              // 左眼被遮挡的阈值，默认0.8
              leftEye: 0.8,
              // 右眼被遮挡的阈值，默认0.8
              rightEye: 0.8,
              // 鼻子被遮挡的阈值，默认0.8
              nose: 0.8,
              // 嘴巴被遮挡的阈值，默认0.8
              mouth: 0.8,
              // 左脸颊被遮挡的阈值，默认0.8
              leftCheek: 0.8,
              // 右脸颊被遮挡的阈值，默认0.8
              rightCheek: 0.8,
              // 下巴被遮挡阈值，默认为0.8
              chinContour: 0.8
            },
            //识别阈值，0-1，默认为0.8
            liveScoreThreshold: 0.6,
            //摄像头显示位置,-1:usb,0.后置，1.前置，默认：1
            rbgCameraId: 1,
            //摄像头旋转角度，默认：0，可传入0、90、180、270四个选项
            videoDirection: 90,
            //人脸检测角度，默认：0，可传入0、90、180、270四个选项。
            rgbDetectDirection: 270,
            //是否开启最优人脸检测，默认：false
            usingBestImage: false,
            //多人采用检测，不能跟踪
            isMultiIdentify: true
          }
        }
      },
      methods: {
        //注册人脸
        register() {
          if (this.$refs.refLevenCameraRegister) {
```

使用方法

```
            this.$refs.refLevenCameraRegister.register({
              id: "123456",
              name: "leven1"
            }, res => {
              this.writeLog("register:" + JSON.stringify(res))
            })
          }
        },
        //错误事件
        onInit(e) {
          this.writeLog("onInit:" + JSON.stringify(e))
        },
        //错误事件
        onError(e) {
          this.writeLog("onError:" + JSON.stringify(e))
        },
        // 写日志
        writeLog(str) {
          console.log(str);
          let logStr = uni.$lv.date.format(null, "yyyy-mm-dd hh:MM:ss") + " " + str
          // let logStr = str + "\n";
          this.logStr = logStr + this.logStr;
        }
      }
    }
</script>

<style>

</style>
```

组件方法

注册人脸

方法名

`register`

用法

- 用法如下：

```
if (this.$refs.refLevenCameraRegister) {
  this.$refs.refLevenCameraRegister.register({
    id: "123456",
    name: "leven1"
  }, res => {
    this.writeLog("register:" + JSON.stringify(res))
  })
}
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
id	String	否	时间戳	注册后保存的id
name	String	否	时间戳	注册后保存的名字

回调

- 示例

```
{
  "data": {
    "type": "complete",
    "imagePath": "/storage/emulated/0/levenBaiduFace/123456.jpg"
  },
  "message": "",
  "code": 0
}
```

- 回调说明：

注册人脸

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
data.type	String	注册类型，complete：注册完成，success：注册成功
data.imagePath	String	注册后本地保存的头像路径
code	Integer	返回类型，0.成功，其他：失败

组件属性

config

> 组件初始化配置

参数说明

参数名	参数类型	是否必填	默认值	参数描述
rgbRevert	Boolean	否	false	RGB预览Y轴转向falese为0，true为180
minFaceSize	Integer	否	50	可传入大于50px的数值，小于此大小的人脸不予检测
faceThreshold	float	否	0.5	人脸置信度用于表征被检测到的物体是人脸的概率，该阈值设置越高检测越严格，建议在0.3-0.8区间内调整阈值，默认：0.5
attribute	Boolean	否	false	是否开启属性检测
qualityControl	Boolean	否	true	是否设置质量检测
qualityConfig	Object	否	无	质量检测参数设置，qualityControl为true时有效
qualityConfig.blur	float	否	0.8	模糊度设置，默认0.8。取值范围[0~1]，0是最清晰，1是最模糊

组件属性

参数名	参数类型	是否必填	默认值	参数描述
qualityConfig.illum	float	否	0.8	光照设置，默认0.8.取值范围[0~1], 数值越大，光线越强
qualityConfig.gesture	Integer	否	15	姿态阈值，默认：15
qualityConfig.leftEye	float	否	0.8	左眼被遮挡的阈值
qualityConfig.rightEye	float	否	0.8	右眼被遮挡的阈值
qualityConfig.nose	float	否	0.8	鼻子被遮挡的阈值
qualityConfig.mouth	float	否	0.8	嘴巴被遮挡的阈值
qualityConfig.leftCheek	float	否	0.8	左脸颊被遮挡的阈值
qualityConfig.rightCheek	float	否	0.8	右脸颊被遮挡的阈值
qualityConfig.chinContour	float	否	0.8	下巴被遮挡阈值
liveScoreThreshold	float	否	0.8	识别阈值
rbgCameraId	Integer	否	1	摄像头显示位置,-1：usb，0：后置，1：前置
videoDirection	Integer	否	0	摄像头旋转角度，可传入0、90、180、270四个选项
rgbDetectDirection	Integer	否	0	人脸检测角度，可传入0、90、180、270四个选项
usingBestImage	Boolean	否	false	是否开启最优人脸检测

组件属性

参数名	参数类型	是否必填	默认值	参数描述
isMultiIdentify	Boolean	否	false	多人采用检测，不能跟踪

组件事件

onError

事件名称

`onError`

> 错误事件

返回示例

```
onError(e) {
    console.log(e.detail)
  }
```

回调说明：

参数名	参数类型	参数描述
message	String	消息提示
code	Integer	错误码

onInit

事件名称

`onInit`

> 初始化事件

返回示例

```
onInit(e) {
    console.log(e.detail)
  }
```

回调说明：

参数名	参数类型	参数描述
message	String	消息提示
code	Integer	错误码

使用方法

组件使用

在需要使用插件的页面加载以下代码

```
<leven-uts-baiduFace-paymentFace ref="refLevenPayment" style="flex:1; height: 500
  @onResultFail="onResultFail" @onResultSuccess="onResultSuccess" @onError="onErr
</leven-uts-baiduFace-paymentFace>
```

> 组件只能在nvue/uvue页面中使用，不支持vue页面

页面示例代码

```
<template>
  <view>
    <uni-card title="安卓百度人脸识别UTS原生插件-支付模式人脸识别">
      <view style="flex:1; height: 500px; position: relative;">
        <leven-uts-baiduFace-paymentFace ref="refLevenPayment" style="flex:1; hei
          @onResultFail="onResultFail" @onResultSuccess="onResultSuccess" @onErro
        </leven-uts-baiduFace-paymentFace>
        <!-- 组件内部自定义内容 -->
        <cover-view v-if="imageBase64" style="position: absolute; right: 0; botto
          <image :src="imageBase64" style="width: 170px; height: 170px;" mode="sc
        </cover-view>
      </view>
      <view>
        <button type="primary" @click="checkRecognitionStatus">当前是否可进行人脸识别
        <button type="primary" @click="openRecognition">开启人脸识别</button>
        <button type="primary" @click="closeRecognition">关闭人脸识别</button>
      </view>
    </uni-card>
    <uni-card class="uni-card-box" title="日志">
      <view><text style="font-size: 14px; flex-wrap: wrap;">{{logStr}}</text></vie
    </uni-card>
  </view>
</template>

<script>
  export default {
    data() {
      return {
```

```
        //日志
        logStr: "",
        //组件配置
        config: {
          //默认为80px。可传入大于30px的数值，小于此大小的人脸不予检测
          minFaceSize: 50,
          //人脸置信度用于表征被检测到的物体是人脸的概率，该阈值设置越高检测越严格，建议在0.3-0
          faceThreshold: 0.5,
          //是否开启属性检测，默认：false
          attribute: false,
          //是否设置质量检测，默认：true
          qualityControl: false,
          //质量检测参数设置，qualityControl为true时有效
          qualityConfig: {
            // 模糊度设置，默认0.8。取值范围[0~1]，0是最清晰，1是最模糊
            blur: 0.8,
            // 光照设置，默认0.8.取值范围[0~1]， 数值越大，光线越强
            illum: 0.8,
            //姿态阈值，默认：15
            gesture: 15,
            // 左眼被遮挡的阈值，默认0.8
            leftEye: 0.8,
            // 右眼被遮挡的阈值，默认0.8
            rightEye: 0.8,
            // 鼻子被遮挡的阈值，默认0.8
            nose: 0.8,
            // 嘴巴被遮挡的阈值，默认0.8
            mouth: 0.8,
            // 左脸颊被遮挡的阈值，默认0.8
            leftCheek: 0.8,
            // 右脸颊被遮挡的阈值，默认0.8
            rightCheek: 0.8,
            // 下巴被遮挡阈值，默认为0.8
            chinContour: 0.8
          },
          //识别阈值，0-1，默认为0.8
          liveScoreThreshold: 0.6,
          //识别框下方默认文本，默认：请保持面部在取景框内
          defaultText: "请保持面部在取景框内",
          //识别成功后是否显示识别文本内容，默认：true
          isShowSuccessText: true,
          //识别成功后识别文本返回默认文本的延迟时间，单位：秒，默认：3
          successTextToDefaultTime: 5,
          //摄像头旋转角度，默认：0，可传入0、90、180、270四个选项
          videoDirection: 90,
          //人脸检测角度，默认：0，可传入0、90、180、270四个选项。
          rgbDetectDirection: 90,
          //是否开启最优人脸检测，默认：false
          usingBestImage: false,
          //识别成功后是否可以继续识别，默认：false
```

```
            successContinue: false,
            //RGB预览Y轴转向falese为0，true为180
            rgbRevert: false,
            //摄像头显示位置,-1:usb,0.后置，1.前置，默认：1
            rbgCameraId: 1,
            //是否开启openGL渲染，默认：false
            isOpenGl: false
          },
          //当前识别的人脸图片
          imageBase64: ""
        }
      },
      methods: {
        //关闭人脸识别
        closeRecognition() {
          if (this.$refs.refLevenPayment) {
            this.$refs.refLevenPayment.closeRecognition(res => {
              this.writeLog(JSON.stringify(res))
            })
          }
        },
        //开启人脸识别
        openRecognition() {
          if (this.$refs.refLevenPayment) {
            this.imageBase64 = "";
            this.$refs.refLevenPayment.openRecognition(res => {
              this.writeLog(JSON.stringify(res))
            })
          }
        },
        //当前是否可进行人脸识别状态
        checkRecognitionStatus() {
          if (this.$refs.refLevenPayment) {
            this.$refs.refLevenPayment.checkRecognitionStatus(res => {
              this.writeLog(JSON.stringify(res))
            })
          }
        },
        //错误事件
        onError(e) {
          this.writeLog(JSON.stringify(e))
        },
        //初始化事件
        onInit(e) {
          this.writeLog(JSON.stringify(e))
        },
        //识别成功
        onResultSuccess(e) {
          console.log("识别成功");
          let detail = e.detail;
```

```
          if (detail.recognitionImageBase64) {
            this.imageBase64 = "" + detail.recognitionImageBa
          }
          //这里不能打印日志，因为base64的原因程序容易挂掉
          // this.writeLog(JSON.stringify(e))
        },
        //识别失败
        onResultFail(e) {
          this.writeLog(JSON.stringify(e))
        },
        //识别超时
        onOverTime(e) {
          this.writeLog(JSON.stringify(e))
        },
        // 写日志
        writeLog(str) {
          console.log(str)
          let logStr = uni.$lv.date.format(null, "yyyy-mm-dd hh:MM:ss") + " " + str
          // let logStr = str + "\n";
          this.logStr = logStr + this.logStr;
        }
      }
    }
</script>

<style>

</style>
```

组件方法

当前是否可进行人脸识别

方法名

`checkRecognitionStatus`

用法

- 用法如下：

```
if (this.$refs.refLevenPayment) {
  this.$refs.refLevenPayment.checkRecognitionStatus(res => {
    this.writeLog(JSON.stringify(res))
  })
}
```

- 参数说明

  无

回调

- 示例

```
{
  "data": {
    "status": false
},
  "message": "",
  "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
data.status	Boolean	当前可识别状态
code	Integer	返回类型，0.成功，其他：失败

开启人脸识别

- 用法如下：

```
```

- 参数说明

  无

- 示例

```
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

关闭人脸识别

方法名

`closeRecognition`

用法

- 用法如下：

```
if (this.$refs.refLevenPayment) {
  this.$refs.refLevenPayment.closeRecognition(res => {
    this.writeLog(JSON.stringify(res))
  })
}
```

- 参数说明

  无

回调

- 示例

```
{
  "data": {},
  "message": "",
  "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

组件属性

config

> 组件初始化配置

参数说明

参数名	参数类型	是否必填	默认值	参数描述
minFaceSize	Integer	否	50	可传入大于50px的数值，小于此大小的人脸不予检测
faceThreshold	float	否	0.5	人脸置信度用于表征被检测到的物体是人脸的概率，该阈值设置越高检测越严格，建议在0.3-0.8区间内调整阈值，默认：0.5
attribute	Boolean	否	false	是否开启属性检测
qualityControl	Boolean	否	true	是否设置质量检测
qualityConfig	Object	否	无	质量检测参数设置，qualityControl为true时有效
qualityConfig.blur	float	否	0.8	模糊度设置，默认0.8。取值范围[0~1]，0是最清晰，1是最模糊
qualityConfig.illum	float	否	0.8	光照设置，默认0.8.取值范围[0~1], 数值越大，光线越强

组件属性

参数名	参数类型	是否必填	默认值	参数描述
qualityConfig.gesture	Integer	否	15	姿态阈值，默认：15
qualityConfig.leftEye	float	否	0.8	左眼被遮挡的阈值
qualityConfig.rightEye	float	否	0.8	右眼被遮挡的阈值
qualityConfig.nose	float	否	0.8	鼻子被遮挡的阈值
qualityConfig.mouth	float	否	0.8	嘴巴被遮挡的阈值
qualityConfig.leftCheek	float	否	0.8	左脸颊被遮挡的阈值
qualityConfig.rightCheek	float	否	0.8	右脸颊被遮挡的阈值
qualityConfig.chinContour	float	否	0.8	下巴被遮挡阈值
liveScoreThreshold	float	否	0.8	识别阈值
defaultText	String	否	请保持面部在取景框内	识别框下方默认文本
isShowSuccessText	Boolean	否	true	识别成功后是否显示识别文本内容
successTextToDefaultTime	Integer	否	3	识别成功后识别文本返回默认文本的延迟时间
videoDirection	Integer	否	0	摄像头旋转角度，可传入0、90、180、270四个选项
rgbDetectDirection	Integer	否	0	人脸检测角度，可传入0、90、180、270四个选项

组件属性

参数名	参数类型	是否必填	默认值	参数描述
usingBestImage	Boolean	否	false	是否开启最优人脸检测
successContinue	Boolean	否	false	多人采用检测，不能跟踪
rgbRevert	Boolean	否	false	RGB预览Y轴转向falese为0，true为180
rbgCameraId	Integer	否	1	摄像头显示位置,-1：usb，0：后置，1：前置
isOpenGl	Boolean	否	false	是否开启openGL渲染
isCropFace `v1.1.2`	Boolean	否	true	识别结果是否裁剪人脸图片
rotateBitmap `v1.0.10`	Integer	否	180	识别位图旋转角度

组件事件

onOverTime

事件名称

`onOverTime`

> 识别超时

返回示例

```
onOverTime(e) {
    console.log(e.detail)
  }
```

回调说明：

参数名	参数类型	参数描述
message	String	消息提示
code	Integer	错误码

onResultFail

事件名称

`onResultFail`

> 识别失败

返回示例

```
onResultFail(e) {
    console.log(e.detail)
  }
```

回调说明：

参数名	参数类型	参数描述
message	String	消息提示
code	Integer	错误码

onResultSuccess

事件名称

`onResultSuccess`

> 识别成功

返回示例

```
onResultSuccess(e) {
    console.log(e.detail)
  }
```

回调示例

```
{
  "recognitionImageBase64": "",
  "registerImagePath": "/storage/emulated/0/levenBaiduFace/123456.jpg",
  "userName": "leven1",
  "userId": "123456",
  "registerTime": 1726799590442,
  "similar": 0.9583905339241028
}
```

回调说明：

参数名	参数类型	参数描述
recognitionImageBase64	String	当前识别的人脸图片base64数据
registerImagePath	String	注册的人脸图片路径
userName	String	注册的用户姓名
userId	String	注册的用户id
registerTime	Integer	注册时间
similar	Float	相似度

onError

事件名称

`onError`

> 错误事件

返回示例

```
onError(e) {
    console.log(e.detail)
  }
```

回调说明：

参数名	参数类型	参数描述
message	String	消息提示
code	Integer	错误码

onInit

事件名称

`onInit`

> 初始化事件

返回示例

```
onInit(e) {
    console.log(e.detail)
  }
```

回调说明：

参数名	参数类型	参数描述
message	String	消息提示
code	Integer	错误码

使用方法

组件使用

在需要使用插件的页面加载以下代码

```
<leven-uts-baiduFace-GateFace ref="refLevenGateFace" style="flex:1; height: 500px
  @onResultSuccess="onResultSuccess" @onError="onError" @onInit="onInit">
</leven-uts-baiduFace-GateFace>
```

> 组件只能在nvue/uvue页面中使用，不支持vue页面

页面示例代码

```
<template>
  <view>
    <uni-card title="安卓百度人脸识别UTS原生插件-闸机模式人脸识别">
      <view style="flex:1; height: 500px; position: relative;">
        <leven-uts-baiduFace-GateFace ref="refLevenGateFace" style="flex:1; height
          @onResultSuccess="onResultSuccess" @onError="onError" @onInit="onInit">
        </leven-uts-baiduFace-GateFace>
        <!-- 组件内部自定义内容 -->
        <cover-view v-if="imageBase64" style="position: absolute; right: 0; bottom
          <image :src="imageBase64" style="width: 170px; height: 170px;" mode="sca
        </cover-view>
      </view>
    </uni-card>
    <uni-card class="uni-card-box" title="日志">
      <view><text style="font-size: 14px; flex-wrap: wrap;">{{logStr}}</text></vie
    </uni-card>
  </view>
</template>

<script>
  export default {
    data() {
      return {
        //日志
        logStr: "",
        //组件配置
        config: {
          //默认为80px。可传入大于30px的数值，小于此大小的人脸不予检测
```

```
            minFaceSize: 50,
            //人脸置信度用于表征被检测到的物体是人脸的概率，该阈值设置越高检测越严格，建议在0.3-0
            faceThreshold: 0.5,
            //是否开启属性检测，默认：false
            attribute: false,
            //是否设置质量检测，默认：true
            qualityControl: false,
            //质量检测参数设置，qualityControl为true时有效
            qualityConfig: {
              // 模糊度设置，默认0.8。取值范围[0~1]，0是最清晰，1是最模糊
              blur: 0.8,
              // 光照设置，默认0.8.取值范围[0~1]， 数值越大，光线越强
              illum: 0.8,
              //姿态阈值，默认：15
              gesture: 15,
              // 左眼被遮挡的阈值，默认0.8
              leftEye: 0.8,
              // 右眼被遮挡的阈值，默认0.8
              rightEye: 0.8,
              // 鼻子被遮挡的阈值，默认0.8
              nose: 0.8,
              // 嘴巴被遮挡的阈值，默认0.8
              mouth: 0.8,
              // 左脸颊被遮挡的阈值，默认0.8
              leftCheek: 0.8,
              // 右脸颊被遮挡的阈值，默认0.8
              rightCheek: 0.8,
              // 下巴被遮挡阈值，默认为0.8
              chinContour: 0.8
            },
            //识别阈值，0-1，默认为0.8
            liveScoreThreshold: 0.6,
            //摄像头旋转角度，默认：0，可传入0、90、180、270四个选项
            videoDirection: 90,
            //人脸检测角度，默认：0，可传入0、90、180、270四个选项。
            rgbDetectDirection: 90,
            //是否开启最优人脸检测，默认：false
            usingBestImage: false,
            //RGB预览Y轴转向falese为0，true为180
            rgbRevert: false,
            //摄像头显示位置,-1:usb,0.后置，1.前置，默认：1
            rbgCameraId: 1,
            //是否开启openGL渲染，默认：false
            isOpenGl: false,
            //识别成功后延迟识别的时间：单位：秒，默认：5
            successContinueTime: 5
          },
          //当前识别的人脸图片
          imageBase64: ""
        }
```

```
    },
    methods: {
      //错误事件
      onError(e) {
        this.writeLog(JSON.stringify(e))
      },
      //初始化事件
      onInit(e) {
        this.writeLog(JSON.stringify(e))
      },
      //识别成功
      onResultSuccess(e) {
        let detail = e.detail;
        if (detail.recognitionImageBase64) {
          this.imageBase64 = "" + detail.recognitionImageBa
        }
        //这里不能打印日志，因为base64的原因程序容易挂掉
        // this.writeLog(JSON.stringify(e))
      },
      //识别失败
      onResultFail(e) {
        this.writeLog(JSON.stringify(e))
      },
    }
  }
</script>

<style>

</style>
```

组件属性

config

> 组件初始化配置

参数说明

参数名	参数类型	是否必填	默认值	参数描述
minFaceSize	Integer	否	50	可传入大于50px的数值，小于此大小的人脸不予检测
faceThreshold	float	否	0.5	人脸置信度用于表征被检测到的物体是人脸的概率，该阈值设置越高检测越严格，建议在0.3-0.8区间内调整阈值，默认：0.5
attribute	Boolean	否	false	是否开启属性检测
qualityControl	Boolean	否	true	是否设置质量检测
qualityConfig	Object	否	无	质量检测参数设置，qualityControl为true时有效
qualityConfig.blur	float	否	0.8	模糊度设置，默认0.8。取值范围[0~1]，0是最清晰，1是最模糊
qualityConfig.illum	float	否	0.8	光照设置，默认0.8.取值范围[0~1], 数值越大，光线越强

参数名	参数类型	是否必填	默认值	参数描述
qualityConfig.gesture	Integer	否	15	姿态阈值，默认：15
qualityConfig.leftEye	float	否	0.8	左眼被遮挡的阈值
qualityConfig.rightEye	float	否	0.8	右眼被遮挡的阈值
qualityConfig.nose	float	否	0.8	鼻子被遮挡的阈值
qualityConfig.mouth	float	否	0.8	嘴巴被遮挡的阈值
qualityConfig.leftCheek	float	否	0.8	左脸颊被遮挡的阈值
qualityConfig.rightCheek	float	否	0.8	右脸颊被遮挡的阈值
qualityConfig.chinContour	float	否	0.8	下巴被遮挡阈值
liveScoreThreshold	float	否	0.8	识别阈值
videoDirection	Integer	否	0	摄像头旋转角度，可传入0、90、180、270四个选项
rgbDetectDirection	Integer	否	0	人脸检测角度，可传入0、90、180、270四个选项
usingBestImage	Boolean	否	false	是否开启最优人脸检测
successContinue	Boolean	否	false	多人采用检测，不能跟踪
rgbRevert	Boolean	否	false	RGB预览Y轴转向falese为0，true为180

组件属性

参数名	参数类型	是否必填	默认值	参数描述
rbgCameraId	Integer	否	1	摄像头显示位置,-1：usb，0：后置，1：前置
isOpenGl	Boolean	否	false	是否开启openGL渲染
successContinueTime	Integer	否	5	识别成功后延迟识别的时间
isCropFace `v1.1.2`	Boolean	否	true	识别结果是否裁剪人脸图片
rotateBitmap `v1.0.10`	Integer	否	180	识别位图旋转角度

组件事件

onResultFail

事件名称

`onResultFail`

> 识别失败

返回示例

```
onResultFail(e) {
    console.log(e.detail)
  }
```

回调说明：

参数名	参数类型	参数描述
message	String	消息提示
code	Integer	错误码

onResultSuccess

事件名称

`onResultSuccess`

> 识别成功

返回示例

```
onResultSuccess(e) {
    console.log(e.detail)
  }
```

回调示例

```
{
  "recognitionImageBase64": "",
  "registerImagePath": "/storage/emulated/0/levenBaiduFace/123456.jpg",
  "userName": "leven1",
  "userId": "123456",
  "registerTime": 1726799590442,
  "similar": 0.9583905339241028
}
```

回调说明：

参数名	参数类型	参数描述
recognitionImageBase64	String	当前识别的人脸图片base64数据
registerImagePath	String	注册的人脸图片路径
userName	String	注册的用户姓名
userId	String	注册的用户id
registerTime	Integer	注册时间
similar	Float	相似度

onError

事件名称

`onError`

> 错误事件

返回示例

```
onError(e) {
    console.log(e.detail)
  }
```

回调说明：

参数名	参数类型	参数描述
message	String	消息提示
code	Integer	错误码

onInit

事件名称

`onInit`

> 初始化事件

返回示例

```
onInit(e) {
    console.log(e.detail)
  }
```

回调说明：

参数名	参数类型	参数描述
message	String	消息提示
code	Integer	错误码

使用方法

组件使用

在需要使用插件的页面加载以下代码

```
<leven-uts-baiduFace-Attendance ref="refLevenGateFace" style="flex:1; height: 500
  @onResultSuccess="onResultSuccess" @onError="onError" @onInit="onInit">
</leven-uts-baiduFace-Attendance>
```

> 组件只能在nvue/uvue页面中使用，不支持vue页面

页面示例代码

```
<template>
  <view>
    <uni-card title="安卓百度人脸识别UTS原生插件-考勤模式人脸识别">
      <view style="flex:1; height: 500px; position: relative;">
        <leven-uts-baiduFace-Attendance ref="refLevenGateFace" style="flex:1; hei
          @onResultSuccess="onResultSuccess" @onError="onError" @onInit="onInit">
        </leven-uts-baiduFace-Attendance>
        <!-- 组件内部自定义内容 -->
        <cover-view v-if="imageBase64" style="position: absolute; right: 0; botto
          <image :src="imageBase64" style="width: 170px; height: 170px;" mode="sc
        </cover-view>
      </view>
    </uni-card>
    <uni-card class="uni-card-box" title="日志">
      <view><text style="font-size: 14px; flex-wrap: wrap;">{{logStr}}</text></vi
    </uni-card>
  </view>
</template>

<script>
  export default {
    data() {
      return {
        //日志
        logStr: "",
        //组件配置
        config: {
          //默认为80px。可传入大于30px的数值，小于此大小的人脸不予检测
```

```
            minFaceSize: 50,
            //人脸置信度用于表征被检测到的物体是人脸的概率，该阈值设置越高检测越严格，建议在0.3-0
            faceThreshold: 0.5,
            //是否开启属性检测，默认：false
            attribute: false,
            //是否设置质量检测，默认：true
            qualityControl: false,
            //质量检测参数设置，qualityControl为true时有效
            qualityConfig: {
              // 模糊度设置，默认0.8。取值范围[0~1]，0是最清晰，1是最模糊
              blur: 0.8,
              // 光照设置，默认0.8.取值范围[0~1]， 数值越大，光线越强
              illum: 0.8,
              //姿态阈值，默认：15
              gesture: 15,
              // 左眼被遮挡的阈值，默认0.8
              leftEye: 0.8,
              // 右眼被遮挡的阈值，默认0.8
              rightEye: 0.8,
              // 鼻子被遮挡的阈值，默认0.8
              nose: 0.8,
              // 嘴巴被遮挡的阈值，默认0.8
              mouth: 0.8,
              // 左脸颊被遮挡的阈值，默认0.8
              leftCheek: 0.8,
              // 右脸颊被遮挡的阈值，默认0.8
              rightCheek: 0.8,
              // 下巴被遮挡阈值，默认为0.8
              chinContour: 0.8
            },
            //识别阈值，0-1，默认为0.8
            liveScoreThreshold: 0.6,
            //摄像头旋转角度，默认：0，可传入0、90、180、270四个选项
            videoDirection: 90,
            //人脸检测角度，默认：0，可传入0、90、180、270四个选项。
            rgbDetectDirection: 90,
            //是否开启最优人脸检测，默认：false
            usingBestImage: false,
            //RGB预览Y轴转向falese为0，true为180
            rgbRevert: false,
            //摄像头显示位置,-1:usb,0.后置，1.前置，默认：1
            rbgCameraId: 1,
            //是否开启openGL渲染，默认：false
            isOpenGl: false,
            //识别成功后延迟识别的时间：单位：秒，默认：5
            successContinueTime: 5,
            //戴口罩是否检测
            checkMouthMask: false,
            //多人识别
            isMultiIdentify: true
```

```
        },
        //当前识别的人脸图片
        imageBase64: ""
      }
    },
    methods: {
      //错误事件
      onError(e) {
        this.writeLog(JSON.stringify(e))
      },
      //初始化事件
      onInit(e) {
        this.writeLog(JSON.stringify(e))
      },
      //识别成功
      onResultSuccess(e) {
        let detail = e.detail;
        if (detail.recognitionImageBase64) {
          this.imageBase64 = "" + detail.recognitionImageBa
        }
        //这里不能打印日志，因为base64的原因程序容易挂掉
        // this.writeLog(JSON.stringify(e))
      },
      //识别失败
      onResultFail(e) {
        this.writeLog(JSON.stringify(e))
      },
    }
  }
</script>

<style>

</style>
```

组件属性

config

> 组件初始化配置

参数说明

参数名	参数类型	是否必填	默认值	参数描述
minFaceSize	Integer	否	50	可传入大于50px的数值，小于此大小的人脸不予检测
faceThreshold	float	否	0.5	人脸置信度用于表征被检测到的物体是人脸的概率，该阈值设置越高检测越严格，建议在0.3-0.8区间内调整阈值，默认：0.5
attribute	Boolean	否	false	是否开启属性检测
qualityControl	Boolean	否	true	是否设置质量检测
qualityConfig	Object	否	无	质量检测参数设置，qualityControl为true时有效
qualityConfig.blur	float	否	0.8	模糊度设置，默认0.8。取值范围[0~1]，0是最清晰，1是最模糊
qualityConfig.illum	float	否	0.8	光照设置，默认0.8.取值范围[0~1], 数值越大，光线越强

参数名	参数类型	是否必填	默认值	参数描述
qualityConfig.gesture	Integer	否	15	姿态阈值，默认：15
qualityConfig.leftEye	float	否	0.8	左眼被遮挡的阈值
qualityConfig.rightEye	float	否	0.8	右眼被遮挡的阈值
qualityConfig.nose	float	否	0.8	鼻子被遮挡的阈值
qualityConfig.mouth	float	否	0.8	嘴巴被遮挡的阈值
qualityConfig.leftCheek	float	否	0.8	左脸颊被遮挡的阈值
qualityConfig.rightCheek	float	否	0.8	右脸颊被遮挡的阈值
qualityConfig.chinContour	float	否	0.8	下巴被遮挡阈值
liveScoreThreshold	float	否	0.8	识别阈值
videoDirection	Integer	否	0	摄像头旋转角度，可传入0、90、180、270四个选项
rgbDetectDirection	Integer	否	0	人脸检测角度，可传入0、90、180、270四个选项
usingBestImage	Boolean	否	false	是否开启最优人脸检测
successContinue	Boolean	否	false	多人采用检测，不能跟踪
rgbRevert	Boolean	否	false	RGB预览Y轴转向falese为0，true为180

组件属性

参数名	参数类型	是否必填	默认值	参数描述
rbgCameraId	Integer	否	1	摄像头显示位置,-1：usb，0：后置，1：前置
isOpenGl	Boolean	否	false	是否开启openGL渲染
successContinueTime	Integer	否	5	识别成功后延迟识别的时间
checkMouthMask	Boolean	否	false	戴口罩是否检测
isMultiIdentify	Boolean	否	false	多人识别
isCropFace `v1.1.2`	Boolean	否	true	识别结果是否裁剪人脸图片
rotateBitmap `v1.0.10`	Integer	否	180	识别位图旋转角度

组件事件

onResultFail

> 识别失败

```
```

参数名	参数类型	参数描述
message	String	消息提示
code	Integer	错误码

onResultSuccess

事件名称

`onResultSuccess`

> 识别成功

返回示例

```
onResultSuccess(e) {
    console.log(e.detail)
  }
```

回调示例

```
{
  "recognitionImageBase64": "",
  "registerImagePath": "/storage/emulated/0/levenBaiduFace/123456.jpg",
  "userName": "leven1",
  "userId": "123456",
  "registerTime": 1726799590442,
  "similar": 0.9583905339241028
}
```

回调说明：

参数名	参数类型	参数描述
recognitionImageBase64	String	当前识别的人脸图片base64数据
registerImagePath	String	注册的人脸图片路径
userName	String	注册的用户姓名
userId	String	注册的用户id
registerTime	Integer	注册时间
similar	Float	相似度

onError

事件名称

`onError`

> 错误事件

返回示例

```
onError(e) {
    console.log(e.detail)
  }
```

回调说明：

参数名	参数类型	参数描述
message	String	消息提示
code	Integer	错误码

onInit

事件名称

`onInit`

> 初始化事件

返回示例

```
onInit(e) {
    console.log(e.detail)
  }
```

回调说明：

参数名	参数类型	参数描述
message	String	消息提示
code	Integer	错误码

使用方法

组件使用

在需要使用插件的页面加载以下代码

```
<leven-uts-baiduFace-Finance ref="refLevenFinance" style="flex:1; height: 500px;"
  @onResultFail="onResultFail" @onResultSuccess="onResultSuccess" @onError="onErro
</leven-uts-baiduFace-Finance>
```

> 组件只能在nvue/uvue页面中使用，不支持vue页面

页面示例代码

```
<template>
  <view>
    <uni-card title="安卓百度人脸识别UTS原生插件-金融活检模式人脸识别">
      <view style="flex:1; height: 500px; position: relative;">
        <leven-uts-baiduFace-Finance ref="refLevenFinance" style="flex:1; height:
          @onResultFail="onResultFail" @onResultSuccess="onResultSuccess" @onErro
        </leven-uts-baiduFace-Finance>
        <!-- 组件内部自定义内容 -->
        <cover-view v-if="imageBase64" style="position: absolute; right: 0; botto
          <image :src="imageBase64" style="width: 170px; height: 170px;" mode="sc
        </cover-view>
      </view>
      <view>
        <button type="primary" @click="checkRecognitionStatus">当前是否可进行人脸识别
        <button type="primary" @click="openRecognition">开启人脸识别</button>
        <button type="primary" @click="closeRecognition">关闭人脸识别</button>
      </view>
    </uni-card>
    <uni-card class="uni-card-box" title="日志">
      <view><text style="font-size: 14px; flex-wrap: wrap;">{{logStr}}</text></vie
    </uni-card>
  </view>
</template>

<script>
  export default {
    data() {
      return {
```

```
        //日志
        logStr: "",
        //组件配置
        config: {
          //默认为80px。可传入大于30px的数值，小于此大小的人脸不予检测
          minFaceSize: 50,
          //人脸置信度用于表征被检测到的物体是人脸的概率，该阈值设置越高检测越严格，建议在0.3-0
          faceThreshold: 0.5,
          //是否开启属性检测，默认：false
          attribute: false,
          //是否设置质量检测，默认：true
          qualityControl: false,
          //质量检测参数设置，qualityControl为true时有效
          qualityConfig: {
            // 模糊度设置，默认0.8。取值范围[0~1]，0是最清晰，1是最模糊
            blur: 0.8,
            // 光照设置，默认0.8.取值范围[0~1]， 数值越大，光线越强
            illum: 0.8,
            //姿态阈值，默认：15
            gesture: 15,
            // 左眼被遮挡的阈值，默认0.8
            leftEye: 0.8,
            // 右眼被遮挡的阈值，默认0.8
            rightEye: 0.8,
            // 鼻子被遮挡的阈值，默认0.8
            nose: 0.8,
            // 嘴巴被遮挡的阈值，默认0.8
            mouth: 0.8,
            // 左脸颊被遮挡的阈值，默认0.8
            leftCheek: 0.8,
            // 右脸颊被遮挡的阈值，默认0.8
            rightCheek: 0.8,
            // 下巴被遮挡阈值，默认为0.8
            chinContour: 0.8
          },
          //识别阈值，0-1，默认为0.8
          liveScoreThreshold: 0.6,
          //识别框下方默认文本，默认：请保持面部在取景框内
          defaultText: "请保持面部在取景框内",
          //识别成功后是否显示识别文本内容，默认：true
          isShowSuccessText: true,
          //识别成功后识别文本返回默认文本的延迟时间，单位：秒，默认：3
          successTextToDefaultTime: 5,
          //摄像头旋转角度，默认：0，可传入0、90、180、270四个选项
          videoDirection: 90,
          //人脸检测角度，默认：0，可传入0、90、180、270四个选项。
          rgbDetectDirection: 90,
          //是否开启最优人脸检测，默认：false
          usingBestImage: false,
          //识别成功后是否可以继续识别，默认：false
```

```
            successContinue: false,
            //RGB预览Y轴转向falese为0，true为180
            rgbRevert: false,
            //摄像头显示位置,-1:usb,0.后置，1.前置，默认：1
            rbgCameraId: 1,
            //是否开启openGL渲染，默认：false
            isOpenGl: false
          },
          //当前识别的人脸图片
          imageBase64: ""
        }
      },
      methods: {
        //关闭人脸识别
        closeRecognition() {
          if (this.$refs.refLevenFinance) {
            this.$refs.refLevenFinance.closeRecognition(res => {
              this.writeLog(JSON.stringify(res))
            })
          }
        },
        //开启人脸识别
        openRecognition() {
          if (this.$refs.refLevenFinance) {
            this.imageBase64 = "";
            this.$refs.refLevenFinance.openRecognition(res => {
              this.writeLog(JSON.stringify(res))
            })
          }
        },
        //当前是否可进行人脸识别状态
        checkRecognitionStatus() {
          if (this.$refs.refLevenFinance) {
            this.$refs.refLevenFinance.checkRecognitionStatus(res => {
              this.writeLog(JSON.stringify(res))
            })
          }
        },
        //错误事件
        onError(e) {
          this.writeLog(JSON.stringify(e))
        },
        //初始化事件
        onInit(e) {
          this.writeLog(JSON.stringify(e))
        },
        //识别成功
        onResultSuccess(e) {
          console.log("识别成功")
          let detail = e.detail;
```

```
          if (detail.recognitionImageBase64) {
            this.imageBase64 = "" + detail.recognitionImageBa
          }
          // this.writeLog(JSON.stringify(e))
        },
        //识别失败
        onResultFail(e) {
          this.writeLog(JSON.stringify(e))
        },
        //识别超时
        onOverTime(e) {
          this.writeLog(JSON.stringify(e))
        },
        // 写日志
        writeLog(str) {
          console.log(str)
          let logStr = uni.$lv.date.format(null, "yyyy-mm-dd hh:MM:ss") + " " + str
          // let logStr = str + "\n";
          this.logStr = logStr + this.logStr;
        }
      }
    }
</script>

<style>

</style>
```

组件方法

当前是否可进行人脸识别

方法名

`checkRecognitionStatus`

用法

- 用法如下：

```
if (this.$refs.refLevenFinance) {
  this.$refs.refLevenFinance.checkRecognitionStatus(res => {
    this.writeLog(JSON.stringify(res))
  })
}
```

- 参数说明

  无

回调

- 示例

```
{
  "data": {
    "status": false
},
  "message": "",
  "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
data.status	Boolean	当前可识别状态
code	Integer	返回类型，0.成功，其他：失败

开启人脸识别

方法名

`openRecognition`

用法

- 用法如下：

```
if (this.$refs.refLevenFinance) {
  this.$refs.refLevenFinance.openRecognition(res => {
    this.writeLog(JSON.stringify(res))
  })
}
```

- 参数说明

  无

回调

- 示例

```
{
  "data": {},
  "message": "",
  "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

关闭人脸识别

方法名

`closeRecognition`

用法

- 用法如下：

```
if (this.$refs.refLevenFinance) {
  this.$refs.refLevenFinance.closeRecognition(res => {
    this.writeLog(JSON.stringify(res))
  })
}
```

- 参数说明

  无

回调

- 示例

```
{
  "data": {},
  "message": "",
  "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

组件属性

config

> 组件初始化配置

参数说明

参数名	参数类型	是否必填	默认值	参数描述
minFaceSize	Integer	否	50	可传入大于50px的数值，小于此大小的人脸不予检测
faceThreshold	float	否	0.5	人脸置信度用于表征被检测到的物体是人脸的概率，该阈值设置越高检测越严格，建议在0.3-0.8区间内调整阈值，默认：0.5
attribute	Boolean	否	false	是否开启属性检测
qualityControl	Boolean	否	true	是否设置质量检测
qualityConfig	Object	否	无	质量检测参数设置，qualityControl为true时有效
qualityConfig.blur	float	否	0.8	模糊度设置，默认0.8。取值范围[0~1]，0是最清晰，1是最模糊
qualityConfig.illum	float	否	0.8	光照设置，默认0.8.取值范围[0~1], 数值越大，光线越强

参数名	参数类型	是否必填	默认值	参数描述
qualityConfig.gesture	Integer	否	15	姿态阈值，默认：15
qualityConfig.leftEye	float	否	0.8	左眼被遮挡的阈值
qualityConfig.rightEye	float	否	0.8	右眼被遮挡的阈值
qualityConfig.nose	float	否	0.8	鼻子被遮挡的阈值
qualityConfig.mouth	float	否	0.8	嘴巴被遮挡的阈值
qualityConfig.leftCheek	float	否	0.8	左脸颊被遮挡的阈值
qualityConfig.rightCheek	float	否	0.8	右脸颊被遮挡的阈值
qualityConfig.chinContour	float	否	0.8	下巴被遮挡阈值
liveScoreThreshold	float	否	0.8	识别阈值
defaultText	String	否	请保持面部在取景框内	识别框下方默认文本
isShowSuccessText	Boolean	否	true	识别成功后是否显示识别文本内容
successTextToDefaultTime	Integer	否	3	识别成功后识别文本返回默认文本的延迟时间
videoDirection	Integer	否	0	摄像头旋转角度，可传入0、90、180、270四个选项
rgbDetectDirection	Integer	否	0	人脸检测角度，可传入0、90、180、270四个选项

组件属性

参数名	参数类型	是否必填	默认值	参数描述
usingBestImage	Boolean	否	false	是否开启最优人脸检测
successContinue	Boolean	否	false	多人采用检测，不能跟踪
rgbRevert	Boolean	否	false	RGB预览Y轴转向falese为0，true为180
rbgCameraId	Integer	否	1	摄像头显示位置,-1：usb，0：后置，1：前置
isOpenGl	Boolean	否	false	是否开启openGL渲染

组件事件

onOverTime

事件名称

`onOverTime`

> 识别超时

返回示例

```
onOverTime(e) {
    console.log(e.detail)
  }
```

回调说明：

参数名	参数类型	参数描述
message	String	消息提示
code	Integer	错误码

onResultFail

事件名称

`onResultFail`

> 识别失败

返回示例

```
onResultFail(e) {
    console.log(e.detail)
  }
```

回调说明：

参数名	参数类型	参数描述
message	String	消息提示
code	Integer	错误码

onResultSuccess

事件名称

`onResultSuccess`

> 识别成功

返回示例

```
onResultSuccess(e) {
    console.log(e.detail)
  }
```

回调示例

```
{}
```

回调说明：

无

onError

事件名称

`onError`

> 错误事件

返回示例

```
onError(e) {
    console.log(e.detail)
  }
```

回调说明：

参数名	参数类型	参数描述
message	String	消息提示
code	Integer	错误码

onInit

事件名称

`onInit`

> 初始化事件

返回示例

```
onInit(e) {
    console.log(e.detail)
  }
```

回调说明：

参数名	参数类型	参数描述
message	String	消息提示
code	Integer	错误码

使用方法

组件使用

在需要使用插件的页面加载以下代码

```
<leven-uts-baiduFace-Person ref="refLevenPerson" style="flex:1; height: 500px;" :
</leven-uts-baiduFace-Person>
```

> 组件只能在nvue/uvue页面中使用，不支持vue页面

页面示例代码

```
<template>
  <view>
    <uni-card title="安卓百度人脸识别UTS原生插件-认证核检">
      <view style="flex:1; height: 500px; position: relative;">
        <leven-uts-baiduFace-Person ref="refLevenPerson" style="flex:1; height: 5
        </leven-uts-baiduFace-Person>
      </view>
      <view>
        <button type="primary" @click="start">开始人证核检</button>
      </view>
    </uni-card>
    <uni-card class="uni-card-box" title="日志">
      <view><text style="font-size: 14px; flex-wrap: wrap;">{{logStr}}</text></vie
    </uni-card>
  </view>
</template>

<script>
  export default {
    data() {
      return {
        //日志
        logStr: "",
        //组件配置
        config: {
          //默认为80px。可传入大于30px的数值，小于此大小的人脸不予检测
          minFaceSize: 50,
          //人脸置信度用于表征被检测到的物体是人脸的概率，该阈值设置越高检测越严格，建议在0.3-0
          faceThreshold: 0.5,
```

```
          //是否开启属性检测，默认：false
          attribute: false,
          //是否设置质量检测，默认：true
          qualityControl: false,
          //质量检测参数设置，qualityControl为true时有效
          qualityConfig: {
            // 模糊度设置，默认0.8。取值范围[0~1]，0是最清晰，1是最模糊
            blur: 0.8,
            // 光照设置，默认0.8.取值范围[0~1]， 数值越大，光线越强
            illum: 0.8,
            //姿态阈值，默认：15
            gesture: 15,
            // 左眼被遮挡的阈值，默认0.8
            leftEye: 0.8,
            // 右眼被遮挡的阈值，默认0.8
            rightEye: 0.8,
            // 鼻子被遮挡的阈值，默认0.8
            nose: 0.8,
            // 嘴巴被遮挡的阈值，默认0.8
            mouth: 0.8,
            // 左脸颊被遮挡的阈值，默认0.8
            leftCheek: 0.8,
            // 右脸颊被遮挡的阈值，默认0.8
            rightCheek: 0.8,
            // 下巴被遮挡阈值，默认为0.8
            chinContour: 0.8
          },
          //识别阈值，0-1，默认为0.8
          liveScoreThreshold: 0.6,
          //摄像头旋转角度，默认：0，可传入0、90、180、270四个选项
          videoDirection: 90,
          //人脸检测角度，默认：0，可传入0、90、180、270四个选项。
          rgbDetectDirection: 90,
          //是否开启最优人脸检测，默认：false
          usingBestImage: false,
          //RGB预览Y轴转向falese为0，true为180
          rgbRevert: false,
          //摄像头显示位置,-1:usb,0.后置，1.前置，默认：1
          rbgCameraId: 1,
          //是否开启openGL渲染，默认：false
          isOpenGl: false
        },
        //当前识别的人脸图片
        imageBase64: ""
      }
    },
    methods: {
      //关闭人脸识别
      start() {
        if (this.$refs.refLevenPerson) {
```

```
          this.$refs.refLevenPerson.start({
            //核检图片，支持本地和网络地址
            // url: "http://www.yeyuboke.com/svga/b.jpg",
            url: "/storage/emulated/0/DCIM/Camera/IMG_20240902_141732.jpg",
          }, res => {
            this.writeLog(JSON.stringify(res))
          })
        }
      },
      //错误事件
      onError(e) {
        this.writeLog(JSON.stringify(e))
      },
      //初始化事件
      onInit(e) {
        this.writeLog(JSON.stringify(e))
      },
      // 写日志
      writeLog(str) {
        console.log(str)
        let logStr = uni.$lv.date.format(null, "yyyy-mm-dd hh:MM:ss") + " " + str
        // let logStr = str + "\n";
        this.logStr = logStr + this.logStr;
      }
    }
  }
</script>

<style>

</style>
```

组件方法

开始人证核检

方法名

`start`

用法

- 用法如下：

```
if (this.$refs.refLevenPerson) {
  this.$refs.refLevenPerson.start({
    //核检图片，支持本地和网络地址
    // url: "http://www.yeyuboke.com/svga/b.jpg",
    url: "/storage/emulated/0/DCIM/Camera/IMG_20240902_141732.jpg",
  }, res => {
    this.writeLog(JSON.stringify(res))
  })
}
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
url	String	否	无	核检图片的地址或路径，支持网络地址

回调

- 示例

```
{
  "data": {},
  "message": "核检成功",
  "code": 0
}
```

- 回调说明：

开始人证核检

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

组件属性

config

> 组件初始化配置

参数说明

参数名	参数类型	是否必填	默认值	参数描述
minFaceSize	Integer	否	50	可传入大于50px的数值，小于此大小的人脸不予检测
faceThreshold	float	否	0.5	人脸置信度用于表征被检测到的物体是人脸的概率，该阈值设置越高检测越严格，建议在0.3-0.8区间内调整阈值，默认：0.5
attribute	Boolean	否	false	是否开启属性检测
qualityControl	Boolean	否	true	是否设置质量检测
qualityConfig	Object	否	无	质量检测参数设置，qualityControl为true时有效
qualityConfig.blur	float	否	0.8	模糊度设置，默认0.8。取值范围[0~1]，0是最清晰，1是最模糊
qualityConfig.illum	float	否	0.8	光照设置，默认0.8.取值范围[0~1], 数值越大，光线越强

组件属性

参数名	参数类型	是否必填	默认值	参数描述
qualityConfig.gesture	Integer	否	15	姿态阈值，默认：15
qualityConfig.leftEye	float	否	0.8	左眼被遮挡的阈值
qualityConfig.rightEye	float	否	0.8	右眼被遮挡的阈值
qualityConfig.nose	float	否	0.8	鼻子被遮挡的阈值
qualityConfig.mouth	float	否	0.8	嘴巴被遮挡的阈值
qualityConfig.leftCheek	float	否	0.8	左脸颊被遮挡的阈值
qualityConfig.rightCheek	float	否	0.8	右脸颊被遮挡的阈值
qualityConfig.chinContour	float	否	0.8	下巴被遮挡阈值
liveScoreThreshold	float	否	0.8	识别阈值
videoDirection	Integer	否	0	摄像头旋转角度，可传入0、90、180、270四个选项
rgbDetectDirection	Integer	否	0	人脸检测角度，可传入0、90、180、270四个选项
usingBestImage	Boolean	否	false	是否开启最优人脸检测
rgbRevert	Boolean	否	false	RGB预览Y轴转向falese为0，true为180
rbgCameraId	Integer	否	1	摄像头显示位置,-1：usb，0：后置，1：前置

组件属性

参数名	参数类型	是否必填	默认值	参数描述
isOpenGl	Boolean	否	false	是否开启openGL渲染

组件事件

onError

事件名称

`onError`

> 错误事件

返回示例

```
onError(e) {
    console.log(e.detail)
  }
```

回调说明：

参数名	参数类型	参数描述
message	String	消息提示
code	Integer	错误码

onInit

事件名称

`onInit`

> 初始化事件

返回示例

```
onInit(e) {
    console.log(e.detail)
  }
```

回调说明：

参数名	参数类型	参数描述
message	String	消息提示
code	Integer	错误码