

慧眼SDK集成使用文档

1、简介

1.1 功能介绍

慧眼SDK iOS 版是一种面向iOS移动设备人脸技术开发包，此版SDK包括活体识别功能。基于该方案，开发者可以轻松的构建包含活体识别的应用。在您使用SDK之前，我们首先为您介绍一下SDK本身及相关人脸能力，以便您能更方便使用。

此版SDK所包含的能力如下：

- 活体检测** 通过让用户做出指定人脸配合式的交互动作，识别当前操作者是否为活体，此功能为离线使用，可设定指定动作是否使用及应用顺序。动作包含：眨眼、张嘴、向右摇头、向左摇头、向上抬头、向下低头六个。可有效抵御高清图片、3D建模、视频等攻击。
- 授权**： SDK的授权判断，授权介质也称为license，在SDK使用中，需要通过license向授权服务器发起请求，判断SDK的使用合法性及使用有效期。

1.2 兼容性

- 系统**： 支持iOS8以上系统。需要开发者通过Deployment Target 来保证支持系统的检测。
- 机型**： 手机和平板皆可
- 网络**： 支持 WIFI 及移动网络,移动网络支持使用 NET 网关及WAP 网关(CMWAP、CTWAP、UNIWAP、3GWAP)。

1.3 开发包说明



2、集成指南

2.1 准备工作

- **申请license**： 此license用户活体检测功能使用。
- **申请认证会话Token**： 通过在慧眼管理端配置，每次进入SDK都要重新获取。

注意： license授权文件需要您工程的Bundle Id(应用包名)。license和Bundle Id是一一对应关系，填错了会导致SDK不能用。具体流程可联系壹证通商务。

2.2配置授权信息集成

```
//人脸license文件名
```

```
#define FACE_LICENSE_NAME @"idl-license"
```

```
//人脸license文件后缀
```

```
#define FACE_LICENSE_SUFFIX @"face-ios"
```

您申请的应用名称 (appname) + 「-face-ios」 后缀, 如申请的应用名称 (appname) 为 test123, 则此处填写 test123-face-ios)

```
#define FACE_LICENSE_ID @"spider-hy-identity-face-ios"
```

配置好证书, 即可运行。注意已经设置好的Bundle ID 不要随意改动

2.3 添加SDK到工程

1、SDK包中以下三个文件 拖至项目中

- LiveDetectionSDKFrame.framework
- com.baidu.idl.face.model.faceSDK.bundle
- com.baidu.idl.face.faceSDK.bundle
- com.baidu.idl.face.live.action.image.bundle

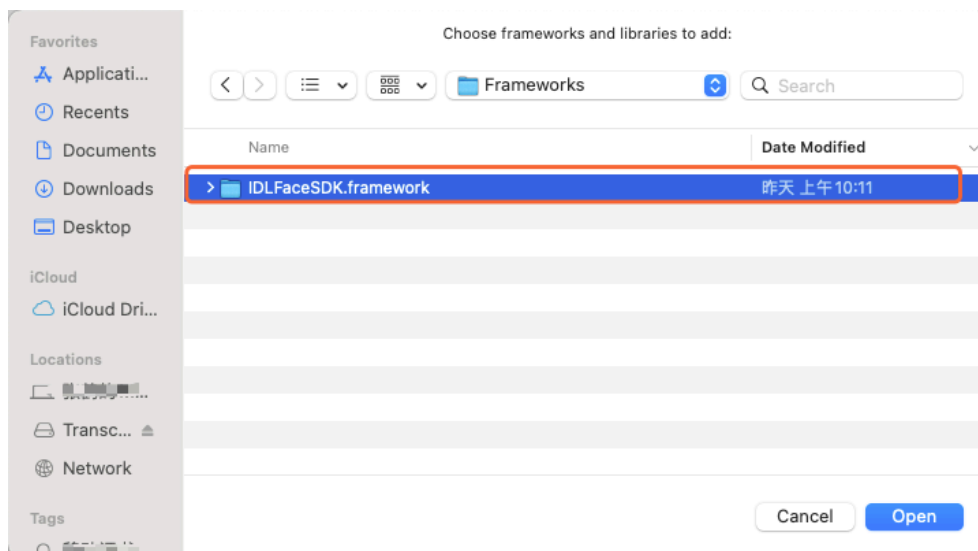
2、选择添加链接C++标准库

libc++.tbd

如果没有使用Pod管理第三方库的话, 请在请在Build Setting > Linking > Other Linker Flags 上面加入 -ObjC 选项。如果用了pod请忽略, 因为pod会自动添加上。

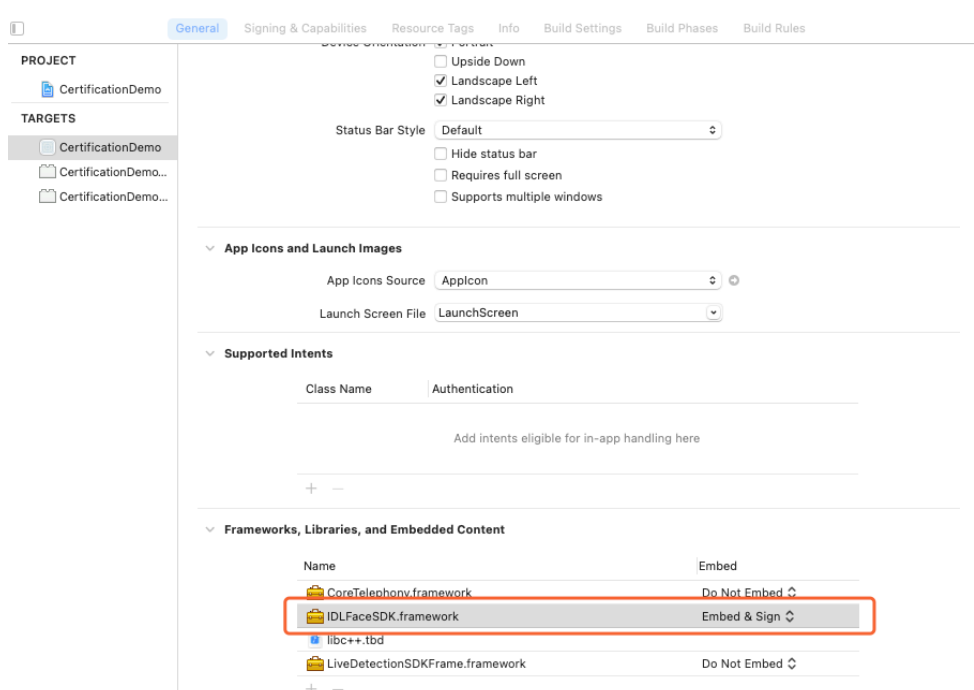
2.4 添加依赖库

• 添加依赖库, 选择 TARGETS -> Build Phases -> Link Binary With Libraries -> Add Files, 选择LiveDetectionSDKFrame.framework Frameworks文件下的IDLFaceSDK.framework 如下图所示:



• 依赖库配置

修改Xcode 的配置参数 如图



2.5 权限声明

需要使用相机权限：编辑Info.plist文件，添加

Privacy- Camera Usage Description 的Key值，Value为使用相机时候的提示语，可以填写：『使用相机』。

3、功能使用

3.1 SDK配置

在AppDelegate.m 文件 -(BOOL)application:(UIApplication

*)applicationDidFinishLaunchingWithOptions:(NSDictionary *)launchOptions 调用以下代码

```
NSString* licensePath = [NSString stringWithFormat:@"%d.%d", FACE_LICENSE_NAME, FACE_LICENSE_SUFFIX]; [LiveFaceSDKManager setLicenseID:FACE_LICENSE_ID andLocalLicenceFile:licensePath];
```

3.2 SDK调用

```
[[LiveFaceSDKManager sharedInstance] initWithToken:token];

UnitidLivenessViewController *livenessVC = [[UnitidLivenessViewController alloc] init];

_liveActionArray = [NSMutableArray arrayWithArray:@[@1,@2]];

[livenessVC livenesswithList:_livenessArray order:YES
numberOfLiveness:_livenessArray.count];

livenessVC.delegate = self; // 核验结果代理回调

[self.navigationController pushViewController:livenessVC animated:YES];
```

其中token为必填项，_livenessArray`为所选动作数组，默认为眨眼动作（为提供用户更好的用户体验，选择单个动作即可）。

3.3 SDK结果回调

```
/*

活体检测返回结果代理方法

*/

- (void)livenessDetectionControllerDidGetNetworkResult:(NSDictionary *)response;

typedef NS_ENUM(NSUInteger, SpiderIDResultCode) {

    SpiderIDCodeOK = 0, //成功

    SpiderIDCodeMANUAL_AUDIT = 1, //提交人工审核成功

    SpiderIDCodeCANCEL = 2, //取消

    SpiderIDCodeINVAILD_TOKEN = 3, //无效的token

    SpiderIDCodeINVAILD_TPAGETOKEN = 4, //获取页面会话token失败

    SpiderIDCodeINVAILD_FAILURE = 5, //其他失败

};`
```

```
/*
```

活体检测取消代理方法

```
*/
```

```
- (void)livenessDetectionControllerDidCancel:(CancelStatus)status;
```

```
typedef enum : NSUInteger {
```

```
UserCancelStatus,    //用户取消
```

```
TimeOutStatus,       //超时
```

```
TokenIllegalStatus,  //无效token
```

```
} CancelStatus;
```

3.5 问题与解决方案

1、

```
Undefined symbols for architecture arm64:
"std::exception::what() const", referenced from:
  vtable for std::__1::bad_function_call in LiveDetectionSDKFrame(IDLFaceSDK.a)
"std::__1::basic_streambuf<char, std::__1::char_traits<char> >::imbue(std::__1::locale const&)", referenced from:
  vtable for std::__1::basic_stringbuf<char, std::__1::char_traits<char>, std::__1::allocator<char> > in
  LiveDetectionSDKFrame(IDLFaceSDK.a)
"std::__1::basic_streambuf<char, std::__1::char_traits<char> >::setbuf(char*, long)", referenced from:
  vtable for std::__1::basic_stringbuf<char, std::__1::char_traits<char>, std::__1::allocator<char> > in
  LiveDetectionSDKFrame(IDLFaceSDK.a)
"std::__1::basic_streambuf<char, std::__1::char_traits<char> >::sync()", referenced from:
  vtable for std::__1::basic_stringbuf<char, std::__1::char_traits<char>, std::__1::allocator<char> > in
  LiveDetectionSDKFrame(IDLFaceSDK.a)
"typeidinfo for std::__1::basic_streambuf<char, std::__1::char_traits<char> >", referenced from:
  typeidinfo for std::__1::basic_filebuf<char, std::__1::char_traits<char> > in LiveDetectionSDKFrame(IDLFaceSDK.a)
  typeidinfo for std::__1::basic_stringbuf<char, std::__1::char_traits<char>, std::__1::allocator<char> > in
  LiveDetectionSDKFrame(IDLFaceSDK.a)
"std::__1::basic_streambuf<char, std::__1::char_traits<char> >::xsgetn(char*, long)", referenced from:
  vtable for std::__1::basic_filebuf<char, std::__1::char_traits<char> > in LiveDetectionSDKFrame(IDLFaceSDK.a)
  vtable for std::__1::basic_stringbuf<char, std::__1::char_traits<char>, std::__1::allocator<char> > in
  LiveDetectionSDKFrame(IDLFaceSDK.a)
"std::__1::basic_streambuf<char, std::__1::char_traits<char> >::uflow()", referenced from:
  vtable for std::__1::basic_filebuf<char, std::__1::char_traits<char> > in LiveDetectionSDKFrame(IDLFaceSDK.a)
  vtable for std::__1::basic_stringbuf<char, std::__1::char_traits<char>, std::__1::allocator<char> > in
  LiveDetectionSDKFrame(IDLFaceSDK.a)
"std::__1::basic_ostream<char, std::__1::char_traits<char> >::~basic_ostream()", referenced from:
  construction vtable for std::__1::basic_ostream<char, std::__1::char_traits<char> > in std::__1::basic_fstream<char,
  std::__1::char_traits<char> > in LiveDetectionSDKFrame(IDLFaceSDK.a)
  construction vtable for std::__1::basic_ostream<char, std::__1::char_traits<char> > in std::__1::basic_ostringstream<char,
  std::__1::char_traits<char>, std::__1::allocator<char> > in LiveDetectionSDKFrame(IDLFaceSDK.a)
  construction vtable for std::__1::basic_ostream<char, caffe::nullstream::char_traits<char> > in caffe::nullstream in
  LiveDetectionSDKFrame(IDLFaceSDK.a)
  construction vtable for std::__1::basic_ostream<char, std::__1::char_traits<char> > in std::__1::basic_stringstream<char,
  std::__1::char_traits<char>, std::__1::allocator<char> > in LiveDetectionSDKFrame(IDLFaceSDK.a)
"std::__1::basic_istream<char, std::__1::char_traits<char> >::~basic_istream()", referenced from:
  construction vtable for std::__1::basic_istream<char, std::__1::char_traits<char> > in std::__1::basic_fstream<char,
  std::__1::char_traits<char> > in LiveDetectionSDKFrame(IDLFaceSDK.a)
  construction vtable for std::__1::basic_istream<char, std::__1::char_traits<char> > in std::__1::basic_istream<char,
  std::__1::char_traits<char>, std::__1::allocator<char> > in LiveDetectionSDKFrame(IDLFaceSDK.a)
```

解决方案：

