

---

# 行驶证预览识别

iOS\_SDK\_说明文档\_v2.1.1.20180710

## 本文目录

---

简介	2
SDK集成配置说明	3
SDK调用IntSig 相机模块 API说明	4
SDK调用系统自定义相机模块 API说明	6
ISOpenSDKStatus错误码表	8
识别证件图像质量要求	9
调用示例代码	9

## 简介

通过合合信息全球领先的 OCR 技术，对相机预览返回视频帧数据中的行驶证图片进行识别，返回行驶证的号牌号码、所有人、住址、使用性质、品牌型号、车辆识别代号、发动机号、注册日期和发证日期等信息，可以省去用户手动录入的过程，给用户带来极大的便利。为了给用户提供行驶证识别体验，合合信息支持提供 Android 与 iOS 系统行驶证识别 SDK。用户只需在 APP 中集成合合信息提供的行驶证识别 SDK，就可以给用户 提供本地行驶证的识别功能。

本文档主要介绍OCR iOS SDK的安装和使用。在使用本文档前，您需要先了解 Optical Character Recognition(OCR)的基础知识，并已经开通了OCR服务。

支持的系统和硬件版本

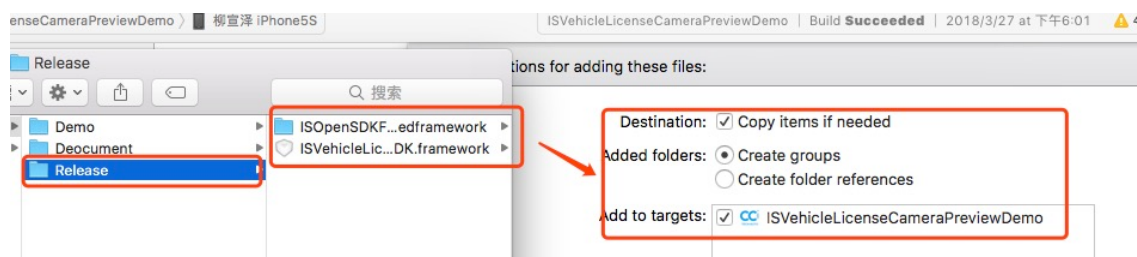
- iOS: 7.0 以上
- 架构: i386 x86\_64 armv7 armv7s arm64

行驶证预览识别功能列表如下：

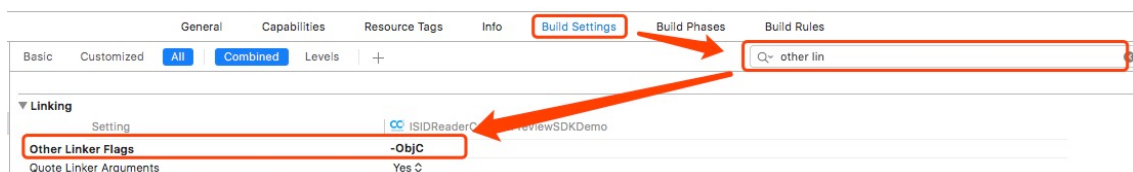
功能列表	预览版本
车牌号码	支持
车辆类型	支持
所有人	支持
地址	支持
使用性质	支持
品牌型号	支持
车辆识别代号	支持
发动机号码	支持
注册日期	支持
发证日期	支持

## SDK集成配置说明

1. 将Release文件夹下的 ISVehicleLicensePreviewSDK.framework和 ISOpenSDKFoundation.embeddedframework 添加到您的工程项目中；
2. 集成步骤：Xcode -> File -> Add Files to “Your Project”，在弹出Panel选中所有Release文件组件包->Add。（注：选择“Copy items if needed”、“Create groups”）



3. 在您的工程项目中“TARGETS -> Build Phases ->”添加如下系统库配置：
  - 3.1.“libc++.tbd”
  - 3.2.“AVFoundation.framework”
4. 在您的工程项目中配置“TARGETS -> Build Settings -> Other Linker Flags”
  - 4.1.“Other Linker Flags = -ObjC”



5. 其它功能库
  - 5.1.iOS10以后相机权限信息：在“Your Project.plist”文件中添加如下配置：

```
<key>NSCameraUsageDescription</key>
<string></string>
<key>NSPhotoLibraryUsageDescription</key>
<string></string>
```

# SDK调用IntSig 相机模块 API说明

## 调用IntSig SDK 相机模块API识别

SDK 中的相机模块封装在 `ISOpenSDKCameraViewController` 类中,可以完成拍照和识别过程,具体定义在 `<ISOpenSDKFoundation.framework/ISOpenSDKCameraViewController.h>`中,详细说明如下:

### 1. ISOpenSDKCameraViewController类成员变量说明如下:

类成员变量	说明
<code>needShowBackButton</code>	是否显示右上角返回按钮
<code>shouldHightLightCorners</code>	图片找边成功之后是否高亮显示
<code>coverView</code>	相机界面出现之前的遮罩界面，默认为空，不显示
<code>delegate</code>	相机界面的代理方法
<code>customInfo</code>	相机拍摄界面提示信息，可以自定义文字内容
<code>frameInterval</code>	相机识别帧数间隔，默认为系统每秒帧数/6；如果每秒帧数是30帧，则每秒帧数/6=5，即每5帧识别一次
<code>continousRecognize</code>	识别完成时候继续返回识别结果，默认为NO；设置为YES后将会在每次识别成功时调用识别结果回调方法

### 2. 初始化相机模块ISOpenSDKCameraViewController

```
- (ISOpenSDKCameraViewController *)cameraViewControllerWithAppkey:(NSString *)appKey  
subAppkey:(NSString *)subAppKey;
```

成员变量	说明
<code>appkey</code>	上海合合信息授权提供的Appkey
<code>subAppKey</code>	暂时填空

3. IOpenSDKCameraViewController相机模块delegate回调方法均在主线程进行返回

3.1)授权失败SDK回调

```
- (void)accessCameraDidFailed;
```

3.2)点击X按钮返回回调

```
- (void)cameraViewController:(UIViewController *)viewController  
    didClickCancelButton:(id)sender;
```

3.3)授权Appkey状态查询，关于AppKey授权情况说明

```
- (void)constructResourcesDidFinishedWithStatusCode:(ISOpenSDKStatus)status;
```

注： [关于ISOpenSDKStatus请参考”错误码表”栏目](#)

3.4)识别图片边缘检测回调

```
- (void)cameraViewController:(UIViewController *)viewController  
    didFinishDetectCardWithResult:(int)result  
        borderPoints:(NSArray *)borderPoints;
```

成员变量	说明
ViewController	相机界面
Result	检测结果，>0表示图片边缘检测成功
borderPoints	表示识别图片四个角点位置；包含四个NSValue对象，通过CGPointValue方法获取对应的CGPoint；对应为顺时针：左上、右上、右下、左下

3.5)授权成功，初始化SDK识别结果返回回调函数

```
- (void)cameraViewController:(UIViewController *)viewController  
    didFinishRecognizeCard:(NSDictionary *)resultInfo  
        cardSDKType:(ISOpenPreviewSDKType)sdkType;
```

备

注：请确保调用接口前，首先调用[授权接口](#)成功，并且返回ISOpenSDKStatus=ISOpenSDKStatusSuccess时才会进行识别结果返回。识别回调为RecognizeCardFinishHandler，其参数cardInfo中包含了识别结果

成员变量	说明
viewController	相机界面
resultInfo	resultInfo 是证件OCR识别结果信息; 通过 <b>kOpenSDKCardResultTypeCardItemInfo</b> 中的 key 获取返回相应的识别结果信息; kVehicleCardItemNumber: 号牌号码; kVehicleCardItemType: 车辆类型; kVehicleCardItemOwner: 所有人; kVehicleCardItemAddress: 地址; kVehicleCardItemUseProperty: 使用性质; kVehicleCardItemBrandModel: 品牌型号; kVehicleCardItemVIN: 车辆识别代号; kVehicleCardItemEngineNumber: 发动机号码; kVehicleCardItemRegisterDate: 注册日期; kVehicleCardItemIssueDate: 发证日期; kOpenSDKCardResultTypeOriginImage: 行驶证原图; kOpenSDKCardResultTypeImage: 行驶证切边后的图;
sdkType	调用的识别SDK种类

## SDK调用系统自定义相机模块 API说明

直接调用SDK中的方法识别接口，主要调用步骤如下：

1. 初始化SDK资源，授权Appkey；
2. 将自定义相机获取图像的视频流CMSamplebufferRef传入到识别方法中；
3. 释放SDK资源；

注：具体的接口定义在<ISVehicleLicensePreviewSDK/

ISVehicleLicensePreviewController.h>中，在集成调用类或是.pch文件中添加以下链接：

**#import <ISVehicleLicensePreviewSDK/ISVehicleLicensePreviewController.h>**

**#import <ISOpenSDKFoundation/ISOpenSDKFoundation.h>**

(1) 初始化SDK资源，授权Appkey

- (ISOpenSDKCameraViewController *)cameraViewControllerWithAppkey:(NSString *)appKey subAppkey:(NSString *)subAppKey;	
成员变量	说明
appkey	上海合合信息授权提供的Appkey
subAppKey	暂时填空

(2) 将自定义相机获取图像的视频流CMSamplebufferRef传入到识别方法中

```
- (ISOpenSDKStatus)detectCardWithOutputSampleBuffer:(CMSampleBufferRef)sampleBuffer
    cardRect:(CGRect)rect
    detectCardFinishHandler:(DetectCardFinishHandler)detectCardFinishHandler
    recognizeCardFinishHandler:(RecognizeCardFinishHandler)recognizeFinishHandler;
```

备注：请确保调用接口前，首先调用授权接口成功，并且返回ISOpenSDKStatus=ISOpenSDKStatusSuccess时才会进行识别结果返回。识别回调为RecognizeCardFinishHandler，其参数cardInfo中包含了识别结果

成员变量	说明
sampleBuffer	识别视频流缓冲区数据，格式务必为YUV格式；
rect	识别图像在视频流中矩形框位置坐标，请注意需要把矩形框的坐标系从Camera Preview Layer 的坐标系转化成实际视频流中图像的坐标系位置；
detectCardFinishHandler	识别图像边缘检测回调； 如果返回值result>0表示检测到识别图像成功；borderPointsArray表示检测到图像边界点，包含4个NSValue的对象，可以通过CGPointValue方法获取对应的CGPoint值，顺序为顺时针左上、右上、右下、左下；
recognizeFinisHandler	resultInfo 是证件OCR识别结果信息； 通过 kOpenSDKCardResultTypeCardItemInfo 中的 key 获取返回相应的识别结果信息； kVehicleCardItemNumber: 号牌号码； kVehicleCardItemType: 车辆类型； kVehicleCardItemOwner: 所有人； kVehicleCardItemAddress: 地址； kVehicleCardItemUseProperty: 使用性质； kVehicleCardItemBrandModel: 品牌型号； kVehicleCardItemVIN: 车辆识别代号； kVehicleCardItemEngineNumber: 发动机号码； kVehicleCardItemRegisterDate: 注册日期； kVehicleCardItemIssueDate: 发证日期； kOpenSDKCardResultTypeOriginImage: 行驶证原图； kOpenSDKCardResultTypeImage: 行驶证切边后的图；

(3) 释放SDK资源

```
- (void)destructResources;
```

## ISOpenSDKStatus错误码表

ISOpenSDKStatus成员变量	参数说明	
ISOpenSDKStatusUnauthorized = -1	错误说明	SDK未授权
	解决方案	请联系技术支持提供正确Appkey
ISOpenSDKStatusSuccess = 0	SDK授权成功，可以正常使用	
ISOpenSDKStatusDeviceIDError = 100	错误说明	当前设备ID错误
	解决方案	请联系技术支持提供正确Appkey
ISOpenSDKStatusAppIDError = 101	错误说明	BundleID错误，授权Appkey与绑定的BundleID不一致；
	解决方案	请检查BundleID是否同我司提供的Appkey一致
ISOpenSDKStatusAppKeyError = 102	错误说明	APPKEY错误，传递的APPKEY填写错误
	解决方案	请检查工程传入的 appKey 是否与合合信息授权的 APP_KEY 一致
ISOpenSDKStatusAuthExpiredError = 103	错误说明	超过时间限制,授权的APPKEY超出使用时间限制
	解决方案	授权到期，如需延长，请与合合信息技术支持联系
ISOpenSDKStatusDeviceCappeError = 104	错误说明	达到设备上限,授权的APPKEY使用设备数量达到限制
	解决方案	设备超限，请与合合信息技术支持联系新增设备的授权
ISOpenSDKStatusDetectCappeError = 105	错误说明	达到识别次数上限，授权的APPKEY使用次数达到限制
	解决方案	识别次数超限。请与合合信息技术支持联系新增设备的授权
ISOpenSDKStatusSubAppKeyError = 106	错误说明	错误的sub app key，暂时无用
	解决方案	请您联系我们的技术支持进行解决
ISOpenSDKStatusUnsupportedAuthError = 202	错误说明	其他未知错误
	解决方案	请您联系我们的技术支持进行解决
ISOpenSDKStatusAuthorizeInfoError = 203	错误说明	服务器错误，第一次联网验证时，因服务器问题，导致验证失败
	解决方案	请您联系我们的技术支持进行解决
ISOpenSDKStatusUnreachable = 204	错误说明	SDK无法获取授权信息，可能由于网络问题；SDK第一次联网验证时，没有网络连接，导致没有验证通过；
	解决方案	请您联系我们的技术支持进行解决
ISOpenSDKStatusConstructResourceError = 205	错误说明	初始化资源失败，暂时无用
	解决方案	请您联系我们的技术支持进行解决
ISOpenSDKStatusVersionError = 206	错误说明	错误的SDK版本
	解决方案	请您联系我们的技术支持进行解决



## 识别证件图像质量要求

1. 如果使用IntSig相机模块识别，请保证光线充足情况下即可识别；
2. 如果使用系统自定义相机，则需要在定义相机界面预览像素不低于1280 \* 720，实际上也就是在iPhone4s以上设备都可以满足；

## 调用示例代码

1. 调用IntSig相机模块API示例代码：

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
    // Override point for customization after application launch.
    self.window = [[UIWindow alloc] initWithFrame:[UIScreen mainScreen].bounds];
    //error Please fill in your appKey
    NSString *appKey = @"8699ae79cc79af62049026f851-vagfvt";
    NSString *subAppKey = nil; //reserved for future use
    ISOpenSDKCameraViewController *cameraVC = [[[ISVehicleLicensePreviewController sharedISOpenSDKController] cameraViewControllerWithAppKey:appKey
                                                subAppKey:subAppKey needCompleteness:YES];
    cameraVC.needShowBackButton = NO;
    cameraVC.customInfo = NSLocalizedString(@"请将行驶证放在框内识别", @"请将行驶证放在框内识别");
    cameraVC.shouldHighlightCorners = YES;
    cameraVC.delegate = self;
    self.baseViewController = cameraVC;
    [self.window setRootViewController:cameraVC];
    return YES;
}

#pragma mark ISOpenSDKCameraViewControllerDelegate
- (void)constructResourcesDidFinishedWithStatusCode:(ISOpenSDKStatus)status
{
    if (status != ISOpenSDKStatusSuccess)
    {
        NSLog(@"Init SDK failed with code:%ld", (long)status);
    }
}

- (void)cameraViewController:(UIViewController *)viewController didFinishRecognizeCard:(NSDictionary *)resultInfo cardSDKType:
    (ISOpenPreviewSDKType)sdkType
{
    if (resultInfo != nil)
    {
        ISViewController *resultVC = [[ISViewController alloc] initWithDictionary:resultInfo loadImageBool:YES];
        UINavigationController *navGCR = [[UINavigationController alloc] initWithRootViewController:resultVC];
        [self.baseViewController presentViewController:navGCR animated:YES completion:^
        {
            //授权成功：返回识别结果
        }];
    }
}
```

2. 调用系统自定义相机模块API示例代码：

```
- (void)viewDidAppear:(BOOL)animated
{
    [super viewDidAppear:animated];
    dispatch_async(dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_HIGH, 0), ^
    {
        NSString *appKey = @"8a7c1f2a08111c93192326f851-vagfvt";
        NSString *subAppKey = nil; //reserved for future use
        [[[ISVehicleLicensePreviewController sharedISOpenSDKController] constructResourcesWithAppKey:appKey subAppKey:subAppKey finishHandler:^(ISOpenSDKStatus status) {
            if (status == ISOpenSDKStatusSuccess)
            {
                NSLog(@"授权成功，调用自定义相机代码");
            }
            else
            {
                NSLog(@"Authorize error");
            }
        }]];
    });
}

- (void)captureOutput:(AVCaptureOutput *)captureOutput didOutputSampleBuffer:(CMSampleBufferRef)sampleBuffer fromConnection:(AVCaptureConnection *)connection
{
    CFRetain(sampleBuffer);
    CGFloat scale = self.imageFrameWidth / self.previewLayer.frame.size.height;
    CGRect rect = CGRectMake(self.borderView.frame.origin.x * scale, self.borderView.frame.origin.y * scale, self.borderView.frame.size.height * scale,
                             self.borderView.frame.size.width * scale);
    [[[ISVehicleLicensePreviewController sharedISOpenSDKController] detectCardWithOutputSampleBuffer:sampleBuffer cardRect:rect detectCardFinishHandler:^(int result, NSArray
    *borderPointsArray) {
        NSLog(@"识别图像区域坐标方位: %@", borderPointsArray);
        recognizeCardFinishHandler:^(NSDictionary *cardInfo) {
            [self.captureSession stopRunning];
            NSLog(@"识别结果: %@", cardInfo);
        }];
    }];
    CFRelease(sampleBuffer);
}
```