TransUnion

香港身份证识别 SDK-IOS 使用文档

2019年8月

版本 1.1.6

transunion.com

历史版本

版本	描述	准备者	更新时间
1.0	起始版本	TU	2019 年 6 月 5 日
1.1	文档新增对系统和 CPU 架构支持的描述 对代码结构进行描述 对详细代码逻辑进行说明	TU	2019 年 7 月 23 日
1.1.6	更新状态处理	TU	2019 年 8 月 6 日

目录

1. 引言

1.1. 目的

方便第三方调用香港身份证验证SDK。

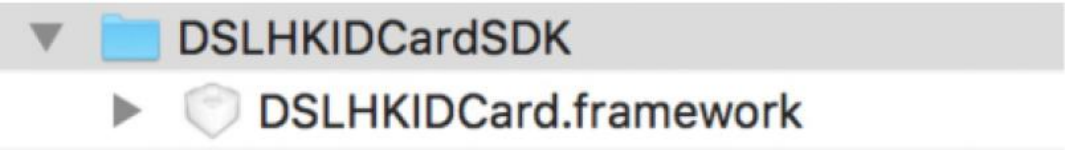
1.2. 文档范围

iOS 系统必须是 ios8.0 及以上。

2. SDK 接入

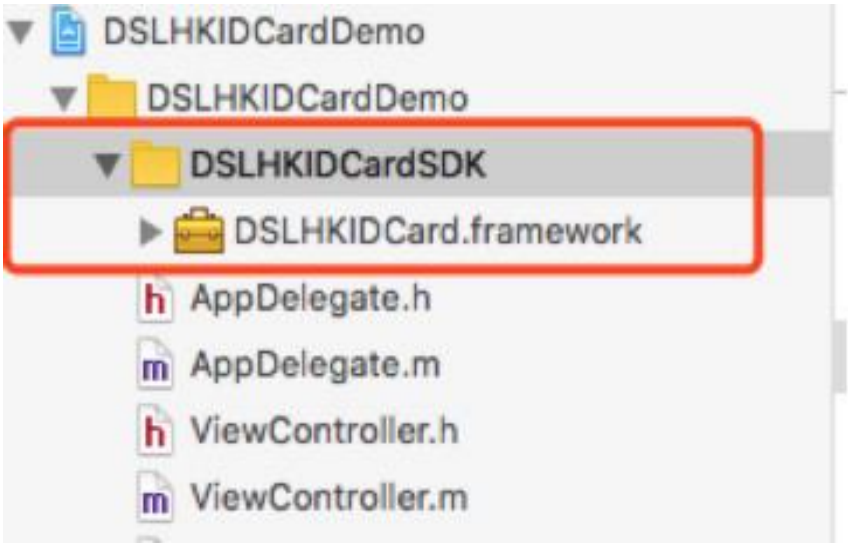
2.1. 解压 SDK 包

- 解压DSLHKIDCardSDK.zip：



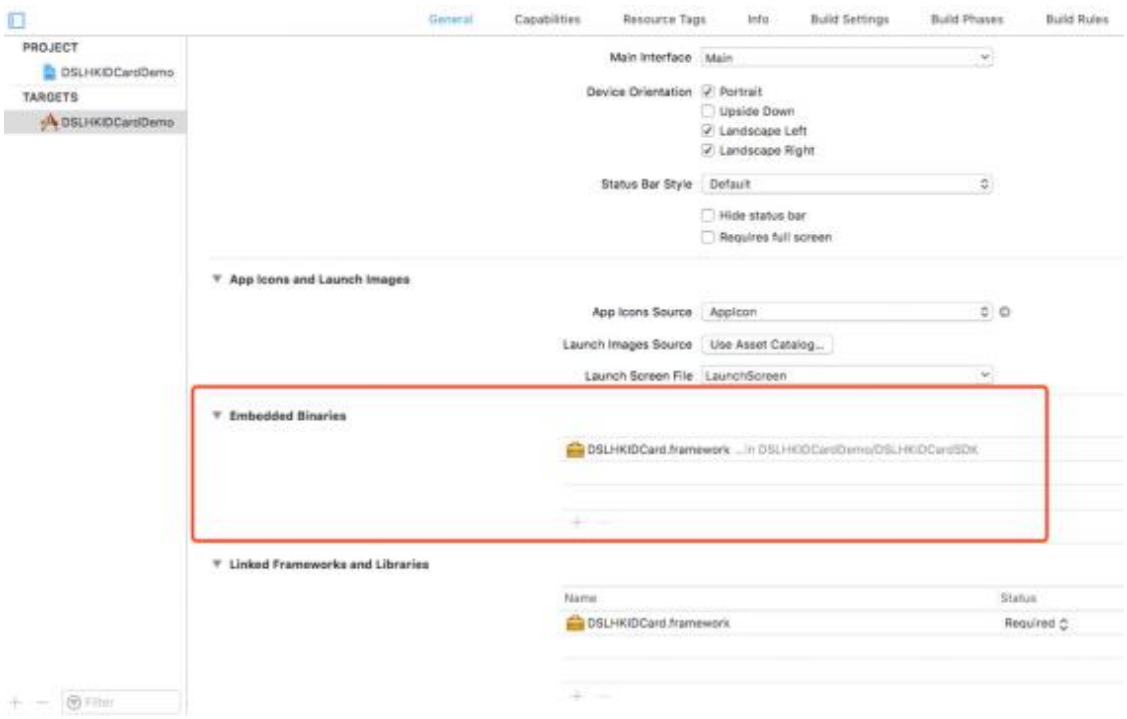
2.2. 导入包

- 把解压的DSLHKIDCardSDK包拖入到项目中：



2.3. 添加必要 framework

- 在General中添加：



3. SDK 接入

3.1. 头文件说明

3.1.1. DSLHKIDCardSDK.h

说明：所有对外的接口都是在这个类

- 接口描述：

```
/**
 开始检测
 */
+(void)startDetect;

/**
 识别结束时重置状态
 注意:目前不需要外部调用重置状态,SDK内部会重置
 */
+(void)resetStatus;

/**
 退出(离开页面的时候)时调用，把对象销毁
 */
+(void)destroyOCR;
```



```
/**
 回调对象注册接口
 @param observer 回调对象
 */
+(void)registerObserver:(id<DSLHKIDCardDetectObserverDelegate>)observer;

/**
 回调对象注销接口
 @param observer 回调对象
 */
+(void)unregisterObserver:(id<DSLHKIDCardDetectObserverDelegate>)observer;

/**
 图片识别
 @param imageData 图片
 @param lensPosition 焦距
 */
+(void)addDetectBufferData:(NSData *_Nonnull)imageData LensPosition:(float)lensPosition;

/**
 设置证件类型
 @param isNewIDCard YES:新证件；NO:老证件
 */
+(void)setIDCardType:(BOOL)isNewIDCard;

/**
 获取当前SDK版本号
 @return 当前版本号
 */
+(NSString *)getVersion;

/**
 是否开启旋转速度过快检测
 默认是关闭状态
 @param overspeed YES: 开启；NO: 关闭
 */
+(void)setOpenCheckOverspeed:(BOOL)overspeed;

/**
 是否开启光线过亮检测
 默认是关闭状态
```



```
 @param exposure YES: 开启；NO: 关闭
 */
+(void)setOpenCheckExposure:(BOOL)exposure;

/**
 设置识别超时时间
 @param times 单位:秒; 默认是30s
 */
+ (void)setOverTimes:(int)times;


/**
 视频安全签名
 @param videoPath 已经录制好的视频路径
 @param imgIDCard 静态证件图片(识别成功后，DSLHKIDCardResult结果返回的静态证件图片)
 @param videoTimeStamp 视频时间戳拼接字符串(如:视频开始时间|第一个动作识别结束时间|第二个动作识别结束时间(打开闪光灯录制视频时间)|视频结束时间)
 @return 一个字典结构，包括:signature和custom
 */
+ (NSMutableDictionary* _Nonnull)signatureWithVideoFile:(NSString* _Nonnull)videoPath
TemplateImage:(NSString* _Nonnull) imgIDCard VideoTimeStamp:(NSString* _Nonnull)videoTimeStamp;

/**
 静态图片安全签名
 @param imgIDCard 静态证件图片(识别成功后，DSLHKIDCardResult结果返回的静态证件图片)
 @return 一个字典结构，包括:signature和custom
 */
+ (NSMutableDictionary* _Nonnull)signatureWithImg:(NSData* _Nonnull)imgIDCard;

/**
 SDK日志是否需要输出
 默认值: Debug模式是YES; Release模式是NO
 @param enable YES: 输出; NO: 不输出
 */
+(void)printLog:(BOOL)enable;
```

3.1.2. DSLHKIDCardResult.h

说明：识别过程及识别结束返回的结果对象

- 接口描述：


```
// 是否识别成功
@property(nonatomic,assign)BOOL success;

/**
 retCode和message对应的值如下:

      retCode    message          说明描述

   0                        识别成功
   1        first           当前返回第一个动作的静态证件图片
   2        exposure        表示当前返回反光过强，重新开始识别;
   3        overspeed       表示当前返回移动速度太快，重新开始识别;
      4        lost          目标(证件)丢失,在识别证件过程中，发现某些证件特征持
                                续捕获不到导致识别失败
   5        overtime        识别超时,超时时间可以设置
 */
@property(nonatomic,copy)NSString *message;
@property(nonatomic, assign) int retCode;

// 校验数据,预留字段，目前不需要处理
@property(nonatomic,copy)NSString *delta;

// 图片数据
//1.当success为NO,retCode为1时，则数组的第一张图片为静态证件图片，其他情况为空
//2.当success为YES,retCode为0时,则数组的第一张图片为静态证件图片，最后一张为人脸比对的证件头像图片

@property(nonatomic,copy)NSArray<NSData *> *imageDataArray;
```

3.1.3. DSLHKIDCardNextOperation.h

说明：识别过程中，当前操作动作返回

- 接口描述：

```
typedef NS_ENUM(NSInteger,DSLHKIDCardOperationStatus){
   DSLHKIDCardOperation_BEGIN, // 开始识别
   DSLHKIDCardOperation_ORTH, // 正对识别完成
   DSLHKIDCardOperation_UP,  // 向上翻转识别完成
   DSLHKIDCardOperation_DOWN, // 向下翻转识别完成
   DSLHKIDCardOperation_COMPLETE, //识别完成
   DSLHKIDCardOperation_NOCARD,    //请对准身份证件
   DSLHKIDCardOperation_EXPOSURE,  //光线太亮
```



```
    DSLHKIDCardOperation_RESET      //复位(把证件正面放置在框内)
    DSLHKIDCardOperation_FAR,       //距离过远(注意: 仅仅是提示，与动作流程无关，当距离恢复正常后，需要继续显示当前动作的提示语)
    DSLHKIDCardOperation_NEAR,      //距离过近(注意: 仅仅是提示，与动作流程无关，当距离恢复正常后，需要继续显示当前动作的提示语)
    DSLHKIDCardOperation_OVERSPEED, //转速过快(注意: 仅仅是提示，与动作流程无关，当转速恢复正常后，需要继续显示当前动作的提示语)
    DSLHKIDCardOperation_VALID      //当【距离过远，距离过近，转速过快】恢复正常时，会发送VALID消息，UI需要恢复显示当前动作的提示语
};
```

- 结果说明：

```
// 当前识别状态
@property(nonatomic,assign)DSLHKIDCardOperationStatus currentStatus;

// 为nil,可以不关注; 提示语为了方便管理和支持多语言统一放到Localizable.strings文件，在外部配置
@property(nonatomic,copy)NSString *nextOperationHint;
```

3.1.4. DSLHKIDCardDetectObserverDelegate.h

说明：在识别过程中及识别结束，返回识别结果代理类

- 代理方法说明：

```
/**
 必须实现这个代理，当前操作动作返回

 @param command DSLHKIDCardNextOperation对象,其中的操作动作对应如下
 DSLHKIDCardOperation_BEGIN, // 开始识别
 DSLHKIDCardOperation_ORTH, // 正对识别完成
 DSLHKIDCardOperation_UP,  // 向上翻转识别完成
 DSLHKIDCardOperation_DOWN, // 向下翻转识别完成
 DSLHKIDCardOperation_COMPLETE, //识别完成
 DSLHKIDCardOperation_NOCARD,   //请对准身份证件
 DSLHKIDCardOperation_EXPOSURE,  //光线太亮
 DSLHKIDCardOperation_RESET      //复位(把证件正面放置在框内)
 DSLHKIDCardOperation_FAR,       //距离过远(注意: 仅仅是提示，与动作流程无关，当距离恢复正常后，需要继续显示当前动作的提示语)
 DSLHKIDCardOperation_NEAR,      //距离过近(注意: 仅仅是提示，与动作流程无关，当距离恢复正常后，需要继续显示当前动作的提示语)
 DSLHKIDCardOperation_OVERSPEED, //转速过快(注意: 仅仅是提示，与动作流程无关，当转速恢复正常后，需要继续显示当前动作的提示语)
```


DSLHKIDCardOperation_VALID //当【距离过远，距离过近，转速过快】恢复正常时，会发送VALID消息，UI需要恢复显示当前动作的提示语

```
*/
-(void)didUpdateOperationCommand:(DSLHKIDCardNextOperation *)command;

/**
 必须实现这个代理，识别结果返回

 @param result 识别结果返回,其中各种结果对应的code及message如下
 retCode和message对应的值如下:

 retCode    message          说明描述
 0                    识别成功
 1          first          当前返回第一个动作的静态证件图片
 2          exposure       表示当前返回反光过强，重新开始识别;
 3          overspeed      表示当前返回移动速度太快，重新开始识别;
 4          lost           目标(证件)丢失,在识别证件过程中，发现某些证件特征持续捕
                           获不到导致识别失败
 5          overtime       识别超时,超时时间可以设置
 */

-(void)didDetectResult:(DSLHKIDCardResult *)result;
```

4. SDK 使用方法

4.1. SDK 初始化

说明：只需要初始化一次即可

- 在 viewDidLoad 方法中初始化 SDK，如下面的代码片段:

```
/**
 初始化SDK
 */
- (void)viewDidLoad {
    [super viewDidLoad];
    // Do any additional setup after loading the view.

    //SDK初始化
```



```
    [self initHKIDCardSDK];

    …

}

- (void)initHKIDCardSDK
{
    //必需, 设置证件类型2018版或者2003版
    [DSLHKIDCardSDK setIDCardType:!self.recType];
    //必需, 注册回调对象
    [DSLHKIDCardSDK registerObserver:self];

    //可选, 设置是否开启旋转速度过快检测
    [DSLHKIDCardSDK setOpenCheckOverspeed:[[[NSUserDefaults standardUserDefaults]
objectForKey:OVERSPEEDSTATUS] boolValue]];
    //可选, 设置是否开启光线过亮检测
    [DSLHKIDCardSDK setOpenCheckExposure:[[[NSUserDefaults standardUserDefaults]
objectForKey:EXPSURESTATUS] boolValue]];
    //可选, 设置识别超时时间
    [DSLHKIDCardSDK setOverTimes:30];

}
```

4.2. SDK 启动识别

说明：在需要重新开始一次识别之前都需要重新启动 SDK 识别，如第一次进入 ViewController，在 viewDidLoad 方法中；在识别失败(目标丢失，超时等)需要重新再识别等，如下面的代码片段:

```
- (void)viewDidLoad {
    [super viewDidLoad];
    // Do any additional setup after loading the view.
    ...
    //开始启动识别
    [self startRecognize];
}

- (void)startRecognize
{
```



```
    ...
    [DSLHKIDCardSDK startDetect];
}
```

4.3. SDK 视频帧数据处理理

说明：在系统的摄像头回调接口中需要调用 SDK 的 addDetectBufferData 方法处理视频帧数据，如下面的代码片段：

```
#pragma mark -- AVCaptureVideoDataOutputSampleBufferDelegate
- (void)captureOutput:(AVCaptureOutput *)output didOutputSampleBuffer:(CMSampleBufferRef)sampleBuffer
fromConnection:(AVCaptureConnection *)connection {

    if(ALL_Save_Video)
    {
        [self recordingCompressionSession:sampleBuffer];

        if(self.bStopRecording || !self.bStartRec)
        {
            return;
        }
    }
    else
    {
        if(self.bStopRecording || !self.bStartRec)
        {
            return;
        }

        if(self.bRecStaticImgOk)
        {
            [self recordingCompressionSession:sampleBuffer];
        }
    }

    self.curImgData = [self imageFromSampleBuffer:sampleBuffer];

    dispatch_async(self.vedioProcessQueue, ^{
```



```
        //进行数据分析

        static NSInteger nProcessFrame = 0;
        if (nProcessFrame > 0 || !self.bStartRec) {
            return;    //last frame not finished
        }
        nProcessFrame ++;

        //必需,把视频帧传给SDK方法(addDetectBufferData)，供SDK分析识别
        [DSLHKIDCardSDK addDetectBufferData:self.curImgData];
        nProcessFrame--;

    });

}
```

4.4. SDK 回调结果处理

说明：主要是

- 提示消息(didUpdateOperationCommand:)，识别过程中，当前操作动作返回提示
- 识别结果(didDetectResult:)，识别失败(因超时，目标丢失，光线太强，移动速度过快)及识别成功的回调结果,必需实现，如下面的代码片段：
  - 提示消息回调：

```
#pragma mark -- DSLHKIDCardDetectObserverDelegate
-(void)didUpdateOperationCommand:(DSLHKIDCardNextOperation *)command {

    dispatch_main_safe(^{

        [self.nextOperationView setNextOpStatus:command.currentStatus Text:[self
getOpTip:command.currentStatus] IDCardType:self.recType];

        if(command.currentStatus == DSLHKIDCardOperation_ORTH)
        {
            if(self.dRecFirstActionOverTimeStamp < 0.01)
            {
                self.dRecFirstActionOverTimeStamp = [[NSDate date] timeIntervalSince1970] * 1000;
            }
            [self setRecStatus:1];
```



```
}
else if(command.currentStatus == DSLHKIDCardOperation_UP)
{
    if(self.dRecStartExposureTimeStamp < 0.01)
    {
        self.dRecStartExposureTimeStamp = [[NSDate date] timeIntervalSince1970] * 1000;
    }

    //已经识别到证件正面，设置拍摄有效区域证件背景框为识别状态
    self.effectiveRectImgView.image = [UIImage imageNamed:@"rec_new_card_bk"];

    [self setRecStatus:2];
}
else if(command.currentStatus == DSLHKIDCardOperation_RESET)
{
    //提示用户把证件复位，设置拍摄有效区域证件背景框为未识别状态
    self.effectiveRectImgView.image = [UIImage imageNamed:@"rec_card_un_bk"];

    [self setRecStatus:3];
}
else if(command.currentStatus == DSLHKIDCardOperation_EXPOSURE)
{
    self.nextOperationView.hidden = NO;
}
else if(command.currentStatus == DSLHKIDCardOperation_NOCARD)
{
     self.nextOperationView.hidden = NO;
}
else if(command.currentStatus != DSLHKIDCardOperation_FAR
        &&
        command.currentStatus != DSLHKIDCardOperation_NEAR
        &&
        command.currentStatus != DSLHKIDCardOperation_OVERSPEED
        &&
        command.currentStatus != DSLHKIDCardOperation_VALID)
{
    [self setRecStatus:0];
}
```



```
    });
}
```

- 识别结果回调：

```
-(void)didDetectResult:(DSLHKIDCardResult *)result
{

    NSLog(@"didDetectResult result = %@, retCode=%i, message=%@, 焦距: %f",@(result.success),
result.retCode, result.message, self.inputDevice.lensPosition);

    dispatch_main_safe((^{

        if(result.retCode == 1)
        {
            self.bRecStaticImgOk = YES;
            self.dRecStartTimeStamp = [[NSDate date] timeIntervalSince1970] * 1000;

            //已经识别到证件正面，设置拍摄有效区域证件背景框为识别状态
            self.effectiveRectImgView.image = [UIImage imageNamed:@"rec_new_card_bk"];
            //获取第一个动作的静态图片
            //可以在这向服务器请求静态图片信息
            [self uploadStaticImg:[result.imageDataArray firstObject]];
        }
        else
        {

            self.bStopRecording = YES;

            if(!ALL_Save_Video)
            {
                if ([self.captureSession isRunning])
                {
                    [self.captureSession stopRunning];
                }
            }

            if(self.recType == 0)
            {
```



```
    [self turnTorchOn:NO];
}

if (result.success == NO)
{
    //识别失败
    [self proccessRecFail:result];
    self.curIDCardResult = nil;
}
else
{
    //识别成功关闭视频流
    if(ALL_Save_Video)
    {
        if ([self.captureSession isRunning])
        {
            [self.captureSession stopRunning];
        }
    }

    self.dRecEndTimeStamp = [[NSDate date] timeIntervalSince1970] * 1000;

    self.curIDCardResult = result;

    NSLog(@"delta string = %@",(result.delta));

    //识别成功才关闭视频录制
    if(ALL_Save_Video)
    {
        [self finishVideoWriter];
    }
}

if(!ALL_Save_Video)
{
   [self finishVideoWriter];
}

if(result.success)
{
```



```
                //save图片
                if(Is_Save_Video_Picture)
                {
                    for(int i = 0; i < (int)[result.imageDataArray count]; ++i)
                    {
                        UIImage *image = [UIImage imageWithData:[result.imageDataArray objectAtIndex:i]];
                        UIImageWriteToSavedPhotosAlbum(image, self, nil, nil);

                    }
                }
            }
        }

    }));

}
```

4.5. SDK 销毁处理

说明：当页面消失的时候或者退出页面的时候 ，需要调用 destroyOCR 把 SDK 内部的资源销毁

```
-(void)viewWillDisappear:(BOOL)animated{
    [super viewWillDisappear:animated];

        …

    if ([self.captureSession isRunning])
    {
        [self.captureSession stopRunning];
    }
    //识别停止
    self.bStartRec = NO;
    //停止录制视频
    [self stopMovieRecorder];

    //必需，当页面消失的时候，需要调用destroyOCR把SDK内部的资源销毁
    [DSLHKIDCardSDK destroyOCR];


        …
```



```
}
```

4.6. SDK 回调对象注销处理

说明：当 ViewController 销毁(dealloc 方法)的时候需要注销回调设置的对象，如下面的代码片段：

```
- (void)dealloc
{
    NSLog(@"%@ dealloc!",NSStringFromClass([self class]));
    //必需, 注销回调设置的对象
    [DSLHKIDCardSDK unregisterObserver:self];

}
```

5. 工程项目介绍

5.1. 基类：DSLHKBaseIDCardViewController

说明：

- 视频采集和识别过程中的操作提示动画相关的 UI 部分统一在 DSLHKBaseIDCardViewController 类实现， 暴露了必要的接口及属性，并在视频回调接口- (void)captureOutput:(AVCaptureOutput *)output didOutputSampleBuffer :(CMSampleBufferRef)sampleBuffer fromConnection:(AVCaptureConnection *)connection 中调用 SDK 的 addDetectBufferData 方法，把视频帧传给 SDK 方法(addDetectBufferData)，供 SDK 算法模型分析识别
- 第三方需要继承这个基类实现以下功能：
    - SDK 相关方法的调用，回调处理(addDetectBufferData 方法除外)
    - 与服务器端交互的接口实现
    - 从服务器端返回的结果显示
    - 自定义 UI

5.2. 配置文件

在 DSLHKIDCardConfig.h 文件中可以配置：

- 连接的服务器
- 是否显示手持证件提示等

- Is_Save_Video_Picture(是否保存视频及图片到相册)
- ALL_Save_Video(把识别失败的视频也录制下来)
- 实际开发过程中注意切换服务器地址。

5.3. 多语言处理

说明：

- 按照苹果开发规范支持多语言功能，在工程的 Localizable.strings 文件下
- 目前只有繁体中文(香港)(简体中文，英文里面的文本也都是繁体中文)，里面具体的文本描述可以根据客户要求做修改
- 目前已有的文本描述如下：

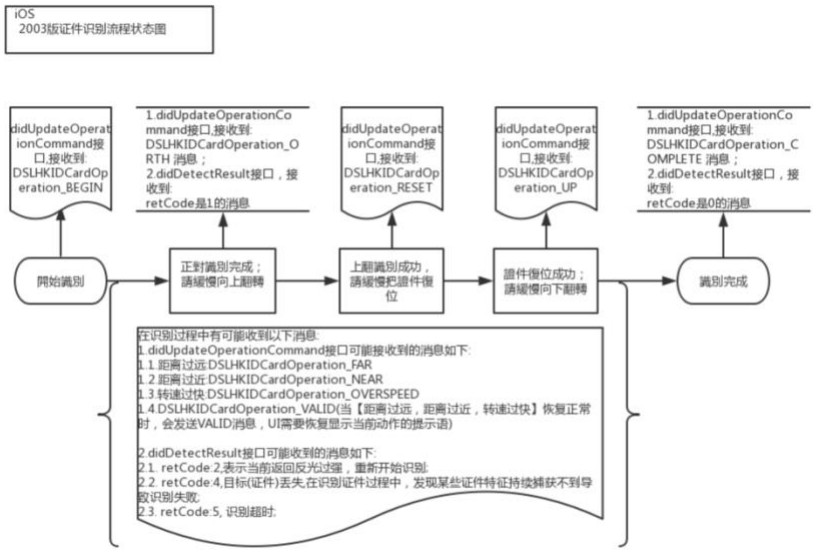
```
//主页面相关
"main_vc_title1" = "手機IP地址 請稍等";
"main_vc_title2" = "進入掃描";
"main_vc_title3" = "重播";
"main_vc_title4" = "設定";
"main_vc_title5" = "新版證件";
"main_vc_title6" = "舊版證件";
"main_vc_title7" = "退出";
"main_vc_title8" = "跳過";

//选择是否上传页面相关
"upload_video_tip1" = "上傳視頻可以讓檢測更加準確！\n我們將保障您的個人信息安全。";
"upload_video_tip2" = "沒問題，同意上傳";
"upload_video_tip3" = "殘忍拒絕";
"upload_video_tip4" = "不再提示";
"upload_video_tip5" = "確定";

//设置页面相关
"setting_view_title1" = "轉速檢測";
"setting_view_title2" = "光亮檢測";
"setting_view_title3" = "播放視頻";

//识别过程中提示相关
"idcard_operation_title1"  = "請將身份證正面置於掃描框內";
"idcard_operation_title2"  = "請緩慢向右翻轉";
"idcard_operation_title3"  = "請緩慢向左翻轉";
```



```
"idcard_operation_title4"  = "請緩慢向上翻轉";
"idcard_operation_title5"  = "請緩慢向下翻轉";
"idcard_operation_title6"  = "光線過亮，請調整";
"idcard_operation_title7"  = "識別完成";
"idcard_operation_title8"  = "向右翻識別成功，請把證件復位";
"idcard_operation_title9"  = "向上翻識別成功，請把證件復位";
"idcard_operation_title10"  = "證件捕捉成功，開始識別";
"idcard_operation_title11"  = "超時，請重新開始";
"idcard_operation_title12"  = "證件復位成功";

//手持证件提示相关
"hand_idcard_tip1"  = "手持證件，將證件置於掃描框內\n按照提示完成翻轉動作";
"hand_idcard_tip2"  = "證件識別";
"hand_idcard_tip3"  = "識別規範";
"hand_idcard_tip4"  = "開始掃描";

"result_fail_title1"  = "再試一次";

//识别中等待页面相关

"loading_rec_title1"  = "識別中，請稍等";
```


6. 证件识别流程状态图

6.1. 2003 版



6.2. 2018 版