

活体动作检测SDK1.2.0接口文档_iOS版

----金融壹账通

一、三步接入活体动作检测SDK

1.兼容C++配置

由于内部实现使用了C++，凡有用到OCFTFaceDetector的类，都需要把后缀.m改成.mm。

2.导入SKD

2.1系统动态库导入

需要的系统库有以下五个，导入到“Linked Frameworks and Libraries”类目下：

```
AssetsLibrary.framework
CoreMedia.framework
AVFoundation.framework
CoreImage.framework
libc++.tbd
```

2.2活体动作检测SDK接入

将活体动作检测库导入到“Embedded Binaries”类目下：

```
OCFTFaceDetect.framework
PAFaceDetectSDK.framework
```

▼ Embedded Binaries

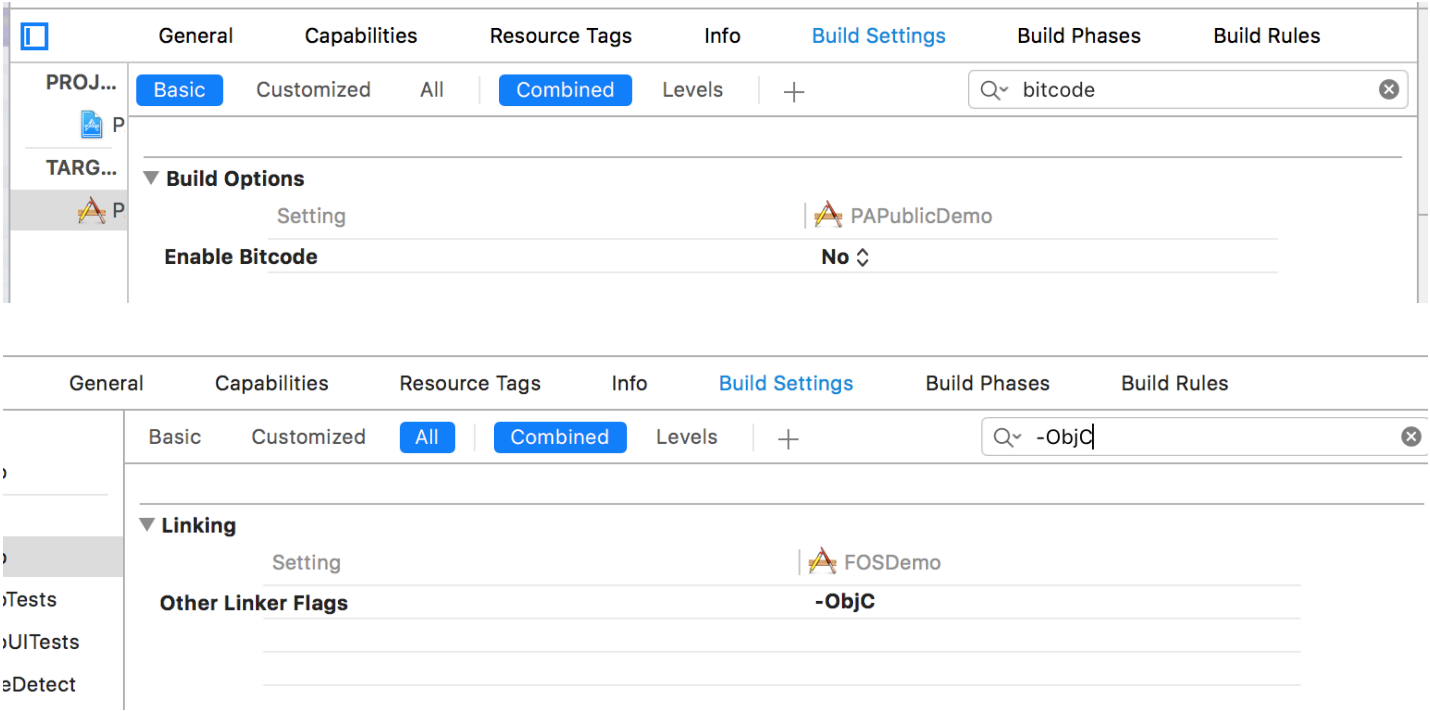
 OCFTFaceDetect.framework

 PAFaceDetectSDK.framework

+ -

2.3关闭bitcode选项，添加-ObjC

如下图所示，请将“Enable Bitcode”项目设置为“NO”，在Other Linker Flags加入-ObjC编译项：



3.开始使用活体动作检测功能

使用说明：动作检测中共包含 无动作/张嘴/眨眼/单一动作随机/全部动作随机 共5种模式。无动作模式会直接检测正脸，其他四种动作模式在进行前都会先进行无动作模式（SDK已实现）。接入时注意：如果正在检测时当前APP进入后台，此时应该停止当前的检测，待APP再次回到前台时，重新开始检测。以保证动作检测流程的完整。(具体可参照demo实现方式)

3.1初始化SDK并获得视屏预览界面

使用“OCFTFaceDetector”类的初始化接口，对活体动作检测SDK进行初始化，接口的详细使用方法可参照接口文档。

```
+ (instancetype)getDetectorWithDetectType:(OCFTFaceDetectActionType)detectType delegate:(id<OCFTFaceDetectProtocol>)delegate;
```

3.2获得视频预览界面并将界面加到业务场景中

可以通过“OCFTFaceDetector”的“videoPreview”方法获取视频预览界面，该界面为iOS提供的“AVCaptureVideoPreviewLayer”，默认全屏展示，支持业务场景定制开发。

```
-(AVCaptureVideoPreviewLayer *)videoPreview;
```

3.3实现OCFTFaceDetectProtocol

为了得到人脸活体图片，以及SDK运作过程中的事件回调，需要按照OCFTFaceDetectProtocol的约定，实现一个回调对象用于接收SDK事件。需要实现的方法如下：

实现用于接收活体动作图片的方法：

```
-(void)onDetectionSuccess:(OCFTFaceDetectionFrame *)faceInfo;
```

实现用于接收检测失败的方法：

```
-(void)onDetectionFailed:(OCFTFaceDetectFailedType)failedType;
```

实现指导用户做活体动作的方法：

```
-(void)onDetectionChangeAnimation:(OCFTFaceDetectActionType)type options:(NSDictionary*)options;
```

实现用于提示用户改善拍摄条件的方法：

```
-(void)onSuggestingOptimization:(OCFTFaceDetectOptimizationType)type;
```

二、接口文档

1.初始化接口：

初始化接口，该接口初始化活体动作检测SDK，并返回OCFTFaceDetector对象，OCFTFaceDetector用于提供管理活体动作检测的接口。

```
+ (instancetype)getDetectorWithDetectType:(OCFTFaceDetectActionType)detectType delegate:(id<OCFTFaceDetectProtocol>)delegate;
```

参数详解：

1) detectType是一个枚举，其用途是指定动作类型：

枚举	value	说明
OCFT_COLLECTFACE	200	无动作
OCFT_MOUTH	201	张嘴提示
OCFT_EYEBLINK	202	眨眼
OCFT_RANDOM	203	单一随机
OCFT_ALLRANDOM	204	全部随机

2) delegate是遵循OCFTFaceDetectProtocol的OC对象，用于接收活体动作检测过程中的事件

调用示例

```
self.livenessDetector = [OCFTFaceDetector getDetectorWithDetectType:OCFT_ALLRANDOM  
delegate:self];
```

2.获取视频预览界面：

```
-(AVCaptureVideoPreviewLayer *)videoPreview;
```

调用示例

```
//注解：self.livenessDetector为OCFTFaceDetector对象  
CALayer * viewLayer = [self.view layer];  
[viewLayer insertSublayer:[self.livenessDetector videoPreview] below:[viewLayer sublayers] objectAtIndex:0];
```

3.获取SDK信息：

```
+(OCFTSDKInfo *)getSDKInfo;
```

返回值

```
@interface OCFTSDKInfo : NSObject  
    @property (readonly) NSString *version;                /** SDK版本号 **/  
@end
```

调用示例

```
NSString *version = [[OCFTFaceDetector getSDKInfo] version];
NSLog(@"%-----@", version);
```

4.销毁SDK

```
- (void)destroySDK;
```

调用示例

```
//注解: self.livenessDetector为OCFTFaceDetector对象
[self.livenessDetector destroySDK];
```

5.OCFTFaceDetectProtocol

OCFTFaceDetectProtocol规范了接收活体动作检测事件的对象，必须实现的接口约定。目前有四个接口，用于接收“检测失败事件”、“检测成功事件”、“提示用户做活体动作事件”、“提示改善拍摄条件事件”。

5.1检测成功事件回调接口

该接口在活体图片采集成功时被调用，用于返回活体图片。

```
-(void)onDetectionSuccess:(OCFTFaceDetectionFrame *)faceInfo;
```

返回参数详解:

返回值是一个OCFTFaceDetectionFrame类型数据，其定义如下，其包裹的对象OCFTFaceImageInfo为每一个动作检测所包含的数据：

```
@property (readonly) OCFTFaceImageInfo *faceImage;  /** 正脸图片 */
// 动作图片
@property (readonly) OCFTFaceImageInfo *moImage;    /** 张嘴图片 */
@property (readonly) OCFTFaceImageInfo *eyeImage;   /** 眨眼图片 */
```

OCFTFaceImageInfo定义如下：

```

@property (readonly) UIImage *targetImage;           /** 检测目标图片 */
@property (readonly) UIImage *targetFaceImage;       /** 检测目标人脸区域图片 */

以下为该检测帧对应的信息
@property (readonly) int result;                     /** 检验结果 */
@property (readonly) int faceNum;                    /** 人脸个数 */
@property (readonly) CGRect face_rect;               /** 人脸位置 */
@property (readonly) CGFloat confidence;             /** 图片综合质量 */
@property (readonly) float pitch;                   /** 上下角度 */
@property (readonly) float yaw;                     /** 左右角度 */
@property (readonly) int roll;                      /** 旋转角度 */
@property (readonly) int brightness;                /** 亮度值 */
@property (readonly) int fuzz;                      /** 模糊值 */
@property (readonly) float mouth_open_value;        /** 张嘴程度 */
@property (readonly) float eye_blink_value_left;    /** 眨眼程度 */
@property (readonly) float eye_blink_value_right;   /** 眨眼程度 */

```

5.2检测失败事件回调接口

该接口在活体图片采集失败时被调用，会返回失败原因。

```
-(void)onDetectionFailed:(OCFTFaceDetectFailedType)failedType;
```

返回参数详解：

返回值是一个OCFTFaceDetectFailedType枚举，其定义如下：

```

typedef enum OCFTFaceDetectFailedType {

    OCFT_DISCONTINUITY_ATTACK = 301,    //非连续性攻击（可理解为用户晃动过大）
    OCFT_CAMERA_AUTH_FAIL,              //相机权限获取失败
    OCFT_SDK_ERROR                      //SDK异常

} OCFTFaceDetectFailedType;

```

枚举	value	说明
OCFT_DISCONTINUITY_ATTACK	301	非连续性攻击（可理解为用户晃动过大）
OCFT_CAMERA_AUTH_FAIL	302	OCFTCAMERAAUTH_FAIL
OCFT_SDK_ERROR	303	SDK异常

5.3提示用户做动作回调接口

该接口在提示用户做动作时被调用，会返回动作类型，目前支持张嘴、眨眼两种动作。

```
-(void)onDetectionChangeAnimation:(OCFTFaceDetectActionType)type options:(NSDictionary*)options;
```

返回参数详解：

返回值是一个OCFTFaceDetectActionType枚举，其定义如下：

```
typedef enum OCFTFaceDetectActionType {  
  
    OCFT_COLLECTFACE = 200,           //无动作  
    OCFT_MOUTH = 201,                 //张嘴提示  
    OCFT_EYEBLINK = 202,              //眨眼  
    OCFT_RANDOM = 203,                //单一随机  
    OCFT_ALLRANDOM = 204               //全部随机  
  
}OCFTFaceDetectActionType;
```

该处只会返回OCFTMOUTH/OCFTEYEBLINK两种具体的动作提示

使用示例

```
switch (type) {  
    case OCFT_MOUTH:  
        NSLog(@"Animation Open Mouth");  
        break;  
    case OCFT_EYEBLINK:  
        NSLog(@"Animation EyeBlink");  
        break;  
    default:  
        break;  
}
```

5.4提示改善拍摄条件回调接口

该接口在提示改善拍摄条件时被调用，会返回提醒事项（比如当前光线过亮 / 暗的提示），用于提示用户如何做以便提高拍摄质量。

```
-(void)onSuggestingOptimization:(OCFTFaceDetectOptimizationType)type;
```

返回参数详解：

返回值是一个OCFTFaceDetectOptimizationType枚举，其定义如下：

```
typedef enum OCFTFaceDetectOptimizationType {

    OCFT_DETECT_NORMAL = 101,                // 正常
    OCFT_DETECT_STAYSTILL,                    // 请保持相对静止

    OCFT_DETECT_ERROR_DARK,                   // 过于灰暗
    OCFT_DETECT_ERROR_BRIGHT,                // 过于明亮
    OCFT_DETECT_ERROR_CLOSE,                 // 过于靠近
    OCFT_DETECT_ERROR_FUZZY,                 // 过于模糊

    OCFT_DETECT_ERROR_MO,                    // 正脸不合格,正脸不能张着嘴
    OCFT_DETECT_ERROR_EYE,                   // 正脸不合格,眼睛不能闭着
    OCFT_DETECT_ERROR_YL,                    // 正脸不合格,角度过于偏左
    OCFT_DETECT_ERROR_YR,                    // 正脸不合格,角度过于偏右
    OCFT_DETECT_ERROR_PU,                    // 正脸不合格,角度过于仰头
    OCFT_DETECT_ERROR_PD,                    // 正脸不合格,角度过于低头
    OCFT_DETECT_MULTIFACE,                   // 多人存在
    OCFT_DETECT_REFLECTIVE,                  // 反光严重

    OCFT_NO_FACE,                            // 没有人脸

} OCFTFaceDetectOptimizationType;
```


枚举	value	说明
OCFT_DETECT_NORMAL	101	正常
OCFT_DETECT_STAYSTILL	102	请保持相对静止
OCFT_DETECT_ERROR_DARK	103	过于灰暗
OCFT_DETECT_ERROR_BRIGHT	104	过于明亮
OCFT_DETECT_ERROR_CLOSE	105	过于靠近
OCFT_DETECT_ERROR_FUZZY	106	过于模糊
OCFT_DETECT_ERROR_MO	107	正脸不合格,正脸不能张着嘴
OCFT_DETECT_ERROR_EYE	108	正脸不合格,眼睛不能闭着
OCFT_DETECT_ERROR_YL	109	正脸不合格,角度过于偏左
OCFT_DETECT_ERROR_YR	110	正脸不合格,角度过于偏右
OCFT_DETECT_ERROR_PU	111	正脸不合格,角度过于仰头
OCFT_DETECT_ERROR_PD	112	正脸不合格,角度过于低头
OCFT_DETECT_MULTIFACE	113	多人存在
OCFT_DETECT_REFLECTIVE	114	反光严重
OCFT_NO_FACE	115	没有人脸

5.5提示SDK开始检测的接口

该接口为可选接口(optional)表示SDK已经开始检测，通常情况下当你调用videoPreview将相机界面展示时，SDK已经开始检测。但是，如果是第一次使用，可能有相机权限的获取提示（系统alert），此时SDK需要等待用户授权后才能开始检测。

```
-(void)onStartDetection:(NSDictionary *)info;
```

返回参数详解：

返回值info为一个字典，为预留字段，目前该值为nil，请勿使用。