

iOS_人脸检测_联网授权V3_集成文档

- [iOS_人脸检测_联网授权V3_集成文档](#)
 - [SDK使用大致流程](#)
 - [集成准备工作](#)
 - [集成到自己的项目](#)
 - [导入人脸检测SDK和模型文件](#)
 - [导入授权文件](#)
 - [调用代码](#)
 - [调用联网授权代码进行授权](#)
 - [初始化人脸检测器](#)
 - [检测视频流/视频帧](#)
 - [检测图片](#)
 - [获取年龄,性别,眼睛状态,嘴巴状态](#)
 - [关键点坐标对应的人脸位置](#)
- [技术支持](#)

SDK使用大致流程

这个SDK是用来检测实时视频流和图片里的人脸，并获取人脸关键点坐标信息。大致流程如下：

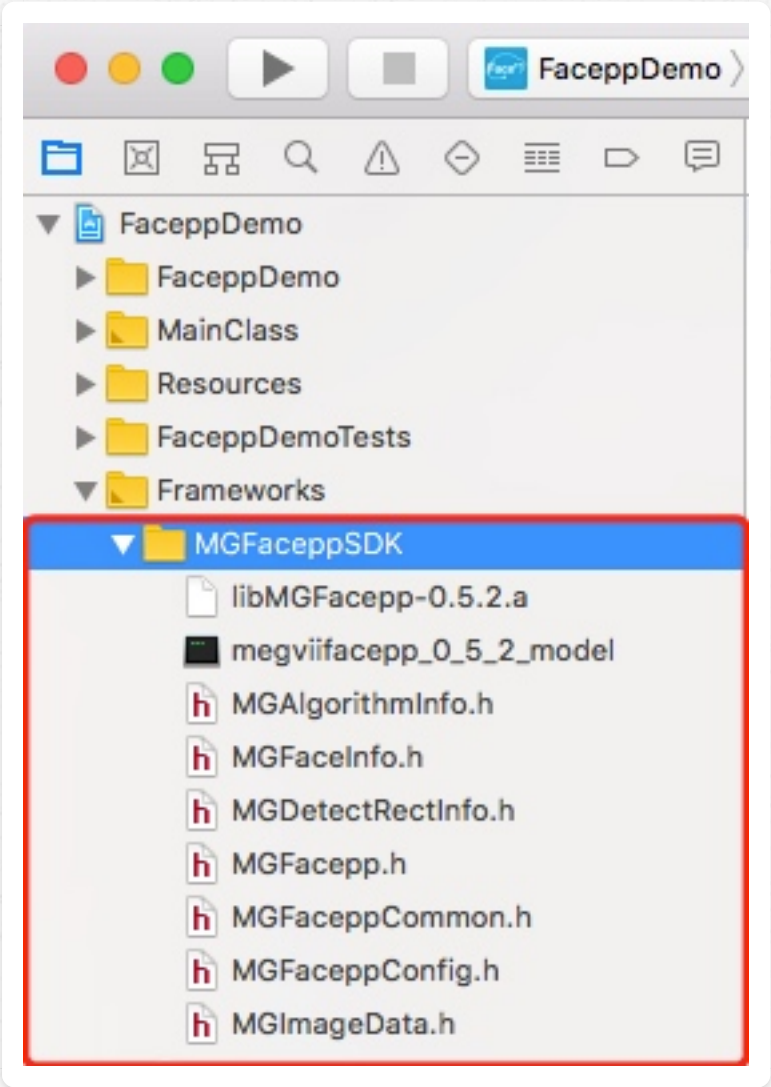
1. 授权SDK,确保SDK能够正常工作（需要联网）
2. 初始化人脸检测器
3. 打开摄像头,获取视频帧或者 获取图片
4. 检测视频帧/图片,获取关键点

集成准备工作

先运行官方Demo成功,然后再执行下面的步骤。Demo运行成功的标志是能够显示人脸上的关键点。如果还没有运行官方Demo,请看目录下的文档 [iOS_人脸检测_联网授权V3版_运行Demo](#)

集成到自己的项目

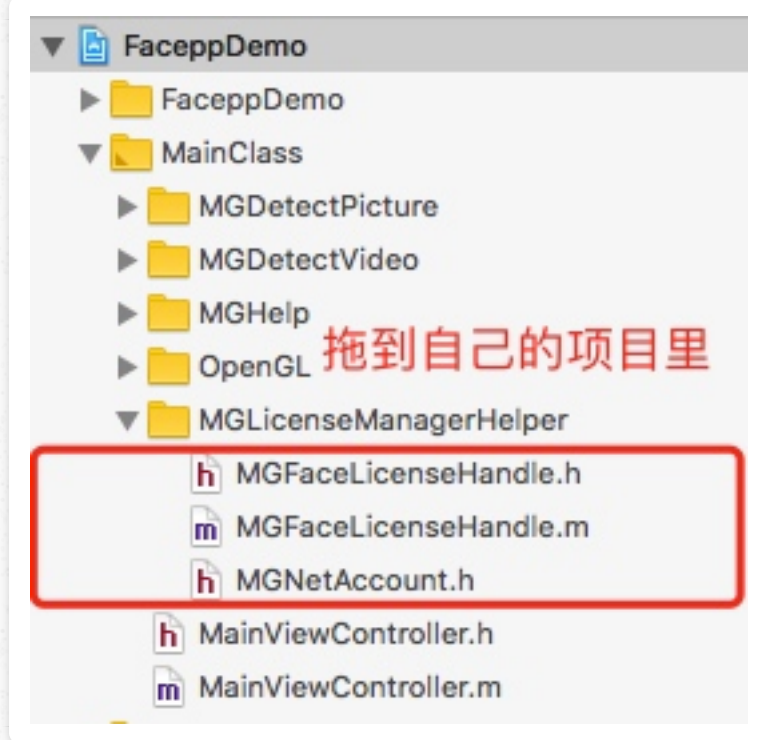
导入人脸检测SDK和模型文件



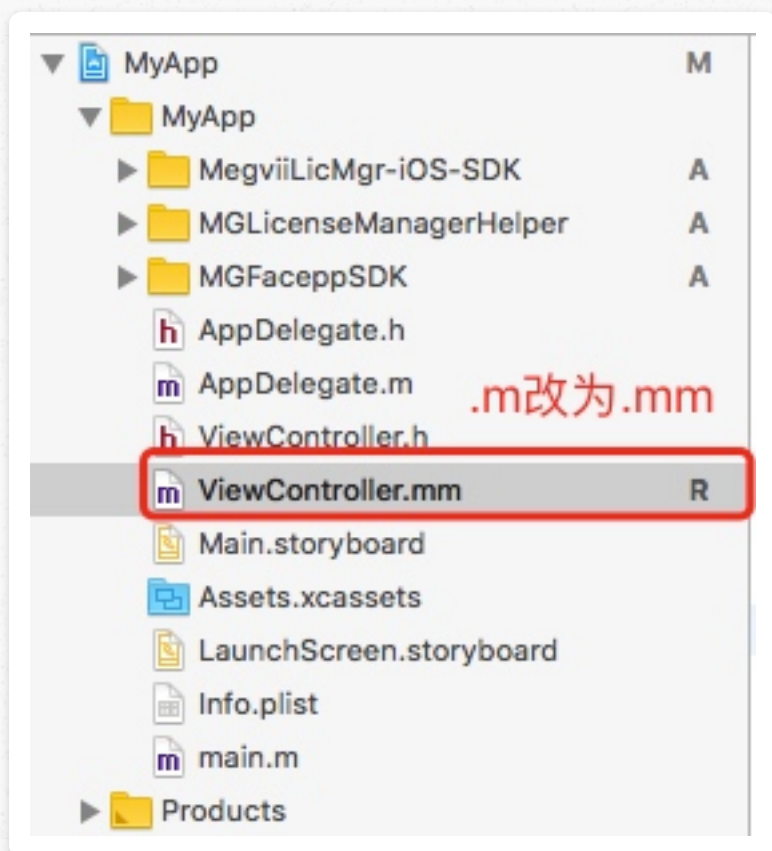
导入授权文件

将授权文件导入到自己的项目里





修改任意 `.m` 文件为 `.mm` 文件



修改 `MGFaceLicenseHandle` 引用的头文件,修改为 `#import "MGLicenseManager.h"`

```

1  //
2  //  MGLicenseHandle.h
3  //  MGSDKV2Test
4  //
5  //  Created by 张英堂 on 16/9/7.
6  //  Copyright © 2016年 megvii. All rights reserved.
7  //
8
9  #import <Foundation/Foundation.h>
10 #import <UIKit/UIKit.h>
11 #import "MGLicenseManager.h"  此处修改为这样
12 #import "MGNetAccount.h"
13
14
15
16 @interface MGFaceLicenseHandle : NSObject
17

```

至此,相关文件导入文件,然后编译查看是否报错

调用代码

调用联网授权代码进行授权

建议尽早调用该方法, 确保下一步的初始化检测器成功。

该方法内部会判断是否有必要发起联网授权请求,如果本地存在缓存,并且缓存在有效期内,则不会发起联网授权请求,否则会发起联网授权请求,成功后在本地做缓存

导入头文件 `#import "MGFaceLicenseHandle.h"`

```

[MGFaceLicenseHandle licenseForNetwokrFinish:^(bool License, NSDate *sdkDate) {
    if (!License) {
        NSLog(@"联网授权失败 !!!");
    } else {
        NSLog(@"联网授权成功");
    }
}];

```

初始化人脸检测器

导入头文件 `#import "MGFacepp.h"`


```

NSString *modelPath = [[NSBundle mainBundle] pathForResource:KMGFACE_MODELNAME
NSData *modelData = [NSData dataWithContentsOfFile:modelPath];
self.markManager = [[MGFacepp alloc] initWithModel:modelData
                                faceppSetting:^(MGFaceppConfig *config)
                                    //配置相关参数
                                    config.orientation = 90;
                                    config.pixelFormatType = PixelFormatType
}];

```

config的相关参数介绍如下:

```

/** 人脸大小 默认 100, 低于 100*100像素的人脸将不会被检测到*/
@property (nonatomic, assign) int minFaceSize;

/** 重新全局检测间隔, 已废弃 */
@property (nonatomic, assign) int interval;

/** 旋转角度 default 0, [0,90,180,270,360], 和视频帧的方向保持一致*/
@property (nonatomic, assign) int orientation;

/**
 * MGFppDetectionModeDetect = 0, //检测图片
 * MGFppDetectionModeTrackingFast = 3, //检测视频流, 速度较快
 * MGFppDetectionModeTrackingRobust = 4, //检测视频流, 精度较高, 推荐
 */
@property (nonatomic, assign) MGFppDetectionMode detectionMode;

/** 废弃参数 */
@property (nonatomic, assign) MGDetectROI detectROI;

/** 设置视频流格式, 默认PixelFormatTypeRGBA, 注意要和你的视频流格式保持一致*/
@property (nonatomic, assign) MGPixelFormatType pixelFormatType;

/** 置信度低于该值的人脸将会被忽略, 防止误检*/
@property (nonatomic, assign) float faceConfidenceFilter;

```

检测视频流/视频帧

检测视频帧, 并打印出关键点坐标, 坐标系为视频帧真实方向的坐标系, 例如视频帧默认是横向的, 那么坐标系就是横向状态下的左上角为(0,0)点, 单位是像素.

```

- (void)detectSampleBuffer:(CMSampleBufferRef)sampleBuffer{

    MGIImageData *imageData = [[MGIImageData alloc] initWithSampleBuffer:sampleBuff

    [self.markManager beginDetectionFrame];

```

```

NSArray *tempArray = [self.markManager detectWithImageData:imageData];
NSUInteger faceCount = tempArray.count;
NSLog(@"face Count : %zd",faceCount);

for (MGFaceInfo *faceInfo in tempArray) {
    [self.markManager GetGetLandmark:faceInfo isSmooth:YES pointsNumber:106];
    NSLog(@"landmark - %@",faceInfo.points);
}

[self.markManager endDetectionFrame];
}

```

检测图片

- 初始化检测器时,设定检测模式为 `MGFppDetectionModeDetect` 模式

```

NSString *modelPath = [[NSBundle mainBundle] pathForResource:KMGFACEMODELNAME
NSData *modelData = [NSData dataWithContentsOfFile:modelPath];
self.markManager = [[MGFacepp alloc] initWithModel:modelData
                                faceppSetting:^(MGFaceppConfig *config)
                                    config.detectionMode = MGFppDetectionModeDetect
                                }];

```

- 检测图片

```

- (void)detectImage:(UIImage *)image{
    MGIImageData *imageData = [[MGIImageData alloc] initWithImage:image];

    [self.markManager beginDetectionFrame];

    NSArray *faceArray = [self.markManager detectWithImageData:imageData];

    if (faceArray.count > 0) {
        NSLog(@"face count : %ld", faceArray.count);

        MGFaceInfo *faceInfo = faceArray[0];
        [self.markManager GetGetLandmark:faceInfo isSmooth:YES pointsNumber:106];
        [self.markManager GetAttribute3D:faceInfo];

        NSLog(@"landmark - %@",faceInfo.points);
    }else{
        NSLog(@"no face detected");
    }
    [self.markManager endDetectionFrame];
}

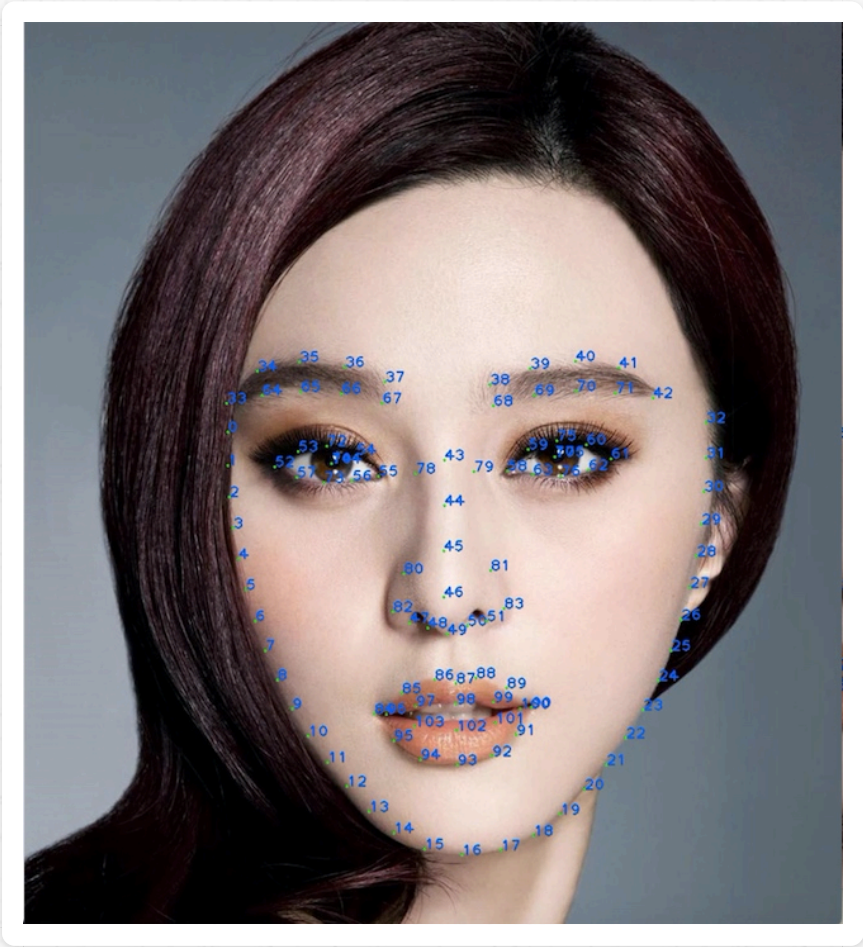
```


获取年龄,性别,眼睛状态,嘴巴状态

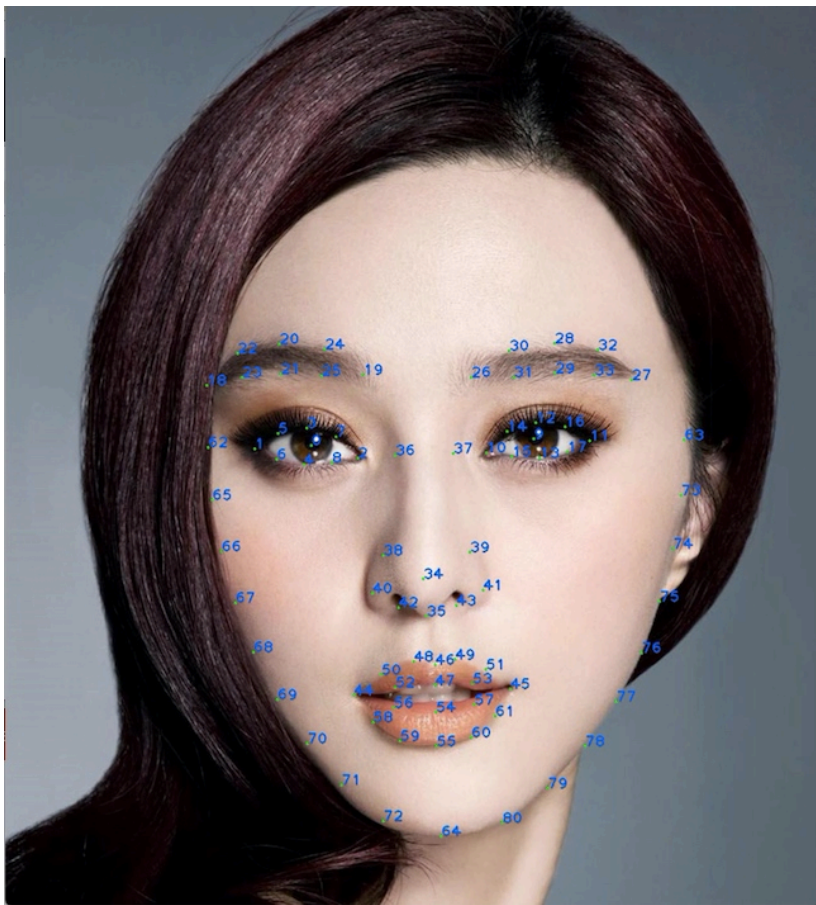
该SDK不包含以上功能,如果需要以上能力的SDK,请提交工单给售前咨询<https://www.faceplusplus.com.cn/contact-us/>并详细描述清楚您的需求以及应用场景.

关键点坐标对应的人脸位置

请查看目录下的 `doc` 文件夹的标注图片.



上面是106点标注图



上面81点标注图

技术支持

请先阅读一下目录下的FAQ文档，如果未能解决您的问题,请提交工单给技术支持: <https://www.faceplusplus.com.cn/contact-us/>