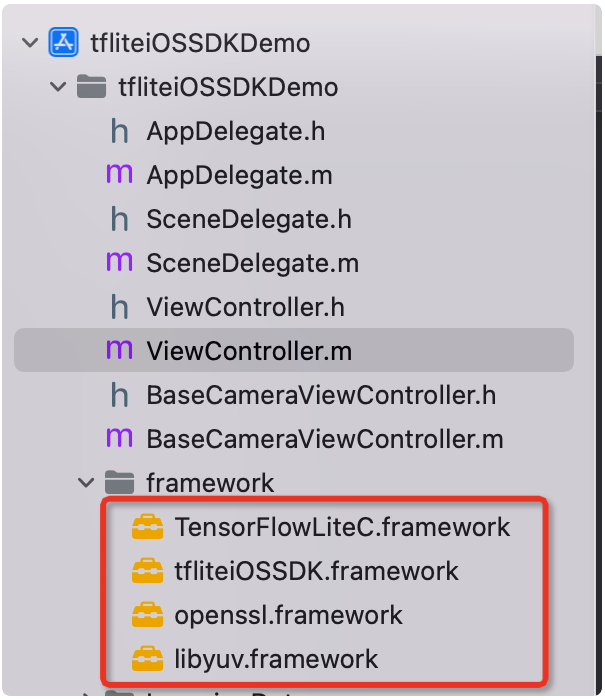
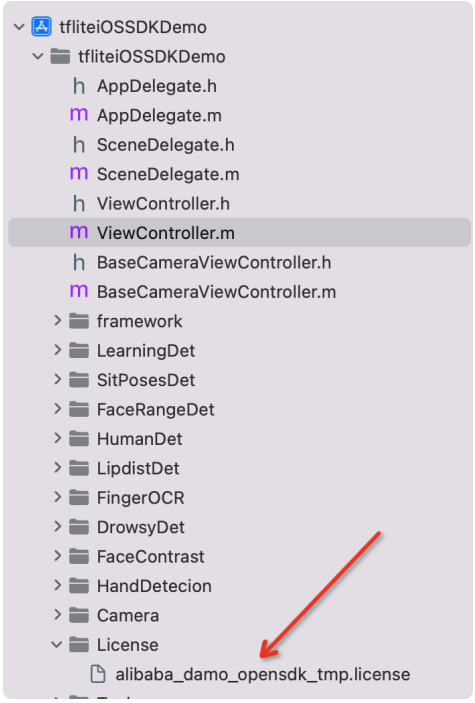
iOS端tfliteiOSSDK人脸比对集成说明文档

一、Xcode配置工程

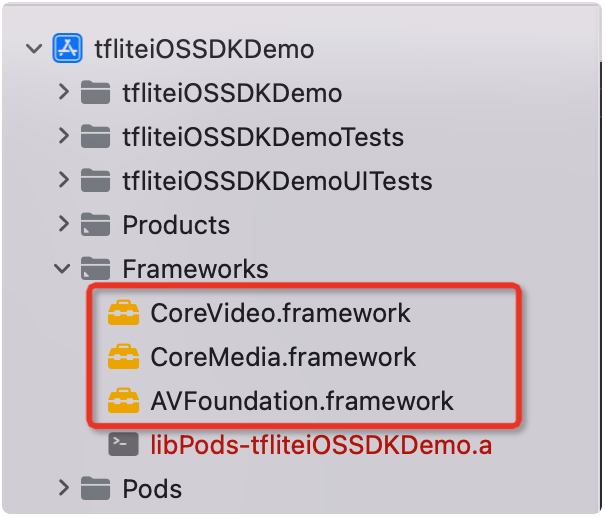
1、获取相关资源压缩包（由阿里云相关人员提供下载链接）后，解压压缩包，可看到如下资源文件framework包及支持相关能力的license文件。如下图：





注意：临时tmp license，不能改名字，正式license可以改名字，但是不能与tmp license重名。

2、添加系统依赖库



3、需要配置相机的权限，项目下的Info.plist文件并在target 中添加如下flag，如下图：

Key	Type	Value
Information Property List	Dictionary	(20 items)
Localization native development region	String	$(DEVELOPMENT_LANGUAGE)
Executable file	String	$(EXECUTABLE_NAME)
Bundle identifier	String	$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	$(PRODUCT_NAME)
Bundle OS Type code	String	$(PRODUCT_BUNDLE_PACKAGE_TYPE)
Bundle version string (short)	String	1.0
Bundle version	String	1
Application requires iPhone environment	Boolean	YES
Privacy - Camera Usage Description	String	是否允许访问您的相机?
Privacy - Location When In Use Usage Description	String	是否允许使用定位
Privacy - Microphone Usage Description	String	是否允许使用麦克风
Privacy - Photo Library Usage Description	String	是否允许访问相册?
Application Scene Manifest	Dictionary	(2 items)
Application supports indirect input events	Boolean	YES
Launch screen interface file base name	String	LaunchScreen
Main storyboard file base name	String	Main
Required device capabilities	Array	(1 item)
Supported interface orientations	Array	(3 items)
Supported interface orientations (iPad)	Array	(4 items)

二、功能实现：

具体代码示例如下：

1、引入头文件 #import <tfliteiOSSDK/OVFaceRecognition.h>

2、初始化功能对象

Plain Text 复制代码

```
- (void) initFaceRecognition{

    self.faceRecognition =[[OVFaceRecognition alloc] init];
    self.faceRecognition.delegate =self;
   NSString *licensePath = [[[NSBundle mainBundle]bundlePath]stringByAppendingPathComponent:@"alibaba_damo_opensdk_tmp.license"];
   int ret =[_faceRecognition faceRecogintionCheckLicensePath:licensePath];
   NSString *deta =[_faceRecognition faceRecogintionLicenseExpirTime];
   NSLog(@"license:%d-%@",ret,deta);

   NSString *detModelPath = [[NSBundle mainBundle] pathForResource:@"face_detection" ofType:@"model"];
   NSString *landModelPath = [[NSBundle mainBundle] pathForResource:@"face_landmarks" ofType:@"model"];
   NSString *valModelPath = [[NSBundle mainBundle] pathForResource:@"face_validate" ofType:@"model"];
   NSString *idModelPath = [[NSBundle mainBundle] pathForResource:@"face_id" ofType:@"model"];
   NSString *qualityModelPath = [[NSBundle mainBundle] pathForResource:@"face_iqa_big" ofType:@"model"];

   int state = [_faceRecognition createFaceRecogintionWithDetModelPath:detModelPath andWithlandmarksModelPath:landModelPath andWithValModelPath:valModelPath andWithidModelPath:idModelPath andWithQualityModelPath:qualityModelPath];
   NSLog(@"createFaceRecogintionWithDetModelPath:%d",state);
}
```

3、点击检测按钮调用相应方法

Plain Text 复制代码

```
- (void)detBtnClicked
{
    if (self.image) {
        CVPixelBufferRef pixbuffer = [UIImage pixelBufferFromImage:self.image];
        [self.glView displayPixelBuffer:pixbuffer];
        //开始检测
        [self.faceRecognition startFaceDetFromBuffer:pixbuffer orientation:UIDeviceOrientationFaceUp];
        [self.glView displayRenderingResults];
        CVPixelBufferRelease(pixbuffer);
    }
}
```

4、人脸比对方法调用

Plain Text 复制代码

```
- (void)contrastBtnClicked
{
    if (_pixbufferL ==nil ||_pixbufferR ==nil) {

        [ViProgressHub showMessage:@"必须选择两张图片" inView:self.view];
        return;
    }
    [self.faceRecognition startFaceMatch1v1FromBuffer:_pixbufferL withPixelBufferR:_pixbufferR];
}
```

5、回调方法

Plain Text 复制代码

```
//人脸检测回调
- (void)faceDetDidTrackObjects:(NSArray <FaceDetInfo *> *)faceDrowsy{
    if (faceDrowsy.count >0) {
        FaceDetInfo *info =faceDrowsy[0];
        self.faceNumLab.text =[NSString stringWithFormat:@"人脸个数: %ld",faceDrowsy.count];
        self.qulityLab.text =[NSString stringWithFormat:@"qulity: %.5f",info.quality];
        self.numLab.text =[NSString stringWithFormat:@"score: %.5f",info.score];
    }else{
        self.faceNumLab.text =[NSString stringWithFormat:@"人脸个数: 0"];
        self.qulityLab.text =[NSString stringWithFormat:@"qulity: 0"];
        self.numLab.text =[NSString stringWithFormat:@"score: 0"];
    }
    for (int i=0; i< faceDrowsy.count; i ++) {

        vi_rgba_color vi_COLOR_RED = {0.0, 0.0, 1.0, 1.0};
        FaceDetInfo *model = faceDrowsy[i];
        FaceModuleRect *faceRect =[[FaceModuleRect alloc]init];

        faceRect.top_left_x = model.rect.top_left_x;
        faceRect.top_left_y = model.rect.top_left_y;
        faceRect.height = model.rect.height;
        faceRect.width = model.rect.width;
//        NSLog(@"faceRect.top_left_x==%f   faceRect.top_left_y====%f   faceRect.height==%f  faceRect.width===%f",faceRect.top_left_x,faceRect.top_left_y,faceRect.height,faceRect.width);
        [self.glView drawRect:faceRect withColor:vi_COLOR_RED lineWidth:4];
    }

}
//人脸比对结果回调
- (void)faceMatch1v1Result:(CGFloat)confidence withMatchFace1Info:(FaceDetInfo *)face1 withMatchFace2Info:(FaceDetInfo *)face2{

    self.qulityLab.text =[NSString stringWithFormat:@"qulity: %.5f",face1.quality];
    self.numLab.text =[NSString stringWithFormat:@"confidence: %.5f",confidence];
    if (confidence >0.55) {
        self.faceNumLab.text =@"比对成功";
    }else{
        self.faceNumLab.text =@"比对失败";
```

```
    }
    vi_rgba_color vi_COLOR_RED = {1.0, 0.0, 0.0, 1.0};
    FaceModuleRect *faceRect1 =[[FaceModuleRect alloc]init];
    faceRect1.top_left_x = face1.rect.top_left_x;
    faceRect1.top_left_y = face1.rect.top_left_y;
    faceRect1.height = face1.rect.height;
    faceRect1.width = face1.rect.width;
    [self.glView clearRenders];
    [self.glView displayPixelBuffer:_pixbufferL];
    [self.glView drawRect:faceRect1 withColor:vi_COLOR_RED lineWidth:3];
    [self.glView displayRenderingResults];


    FaceModuleRect *faceRect2 =[[FaceModuleRect alloc]init];
    faceRect2.top_left_x = face2.rect.top_left_x;
    faceRect2.top_left_y = face2.rect.top_left_y;
    faceRect2.height = face2.rect.height;
    faceRect2.width = face2.rect.width;
    [self.glViewR clearRenders];
    [self.glViewR displayPixelBuffer:_pixbufferR];
    [self.glViewR drawRect:faceRect2 withColor:vi_COLOR_RED lineWidth:3];
    [self.glViewR displayRenderingResults];
}
```

三、其他注意

license鉴权报错码定义

-20011 license没有初始化，直接调用API接口。

-20012 当前的license与调用app不是绑定关系，license用在其他app中使用。

-20013 license无效。

-20014 license授权时间过期。

-20015 此license中不包含调用的算法能力（未购买此能力）。

-20016 bundle ID获取失败。

-20017 临时licese时间校验失败。