

# 接口文档

## high-level接口

### 初始化接口

```
/**
 * 初始化操作，如算法鉴权、加载模型等，完成后回调初始化结果；
 *
 * 使用场景：在应用启动时调用（大多数场景只做一次鉴权、模型加载，无需每次检测时重新初始化），希望鉴权失败或模型加载失败等
 * 告知用户
 * 超时时间：15s
 *
 * @param context
 * @param initOption
 * @param initCallback
 */
public void init(final Context context, InitOption initOption, final InitCallback initCallback);
```

# 人脸活体检测接口

```
/**
 * 人脸活体检测接口
 *
 * 开启检测流程，如人人脸检测->跟踪->活检，筛选出符合活检要求的主体进行行活体检测，
 * 拿到通过活检的人人脸相关信息与不符合活检主体要求时的错误提示信息
 * 使用场景：在预览成功的情况下开启检测，需要即时响应用户的错误状态。
 * <p>
 * 注：暂无人脸跟踪能力时，调用方可关闭人脸跟踪开关，只做活体检测，无需实现跟踪Callback；若支持跟踪，需支持只做跟踪
 * 不做活检的情况，调用方通过开关控制。
 *
 * @param rgbFrame
 * @param nirFrame
 * @param depthFrame
 * @param livenessDetectionOption
 * @param faceTrackCallback
 * @param livenessCallback
 */
public void startLivenessDetectionDetection(final ImageFrame rgbFrame,
                                           final ImageFrame nirFrame,
                                           final LivenessDetectionOption livenessDetectionOption,
                                           final FaceTrackCallback faceTrackCallback,
                                           final LivenessCallback livenessCallback);
```

# 人脸特征抽取接口

```
/**
 * 人脸特征抽取接口
 *
 * 传入图像，获取人脸特征
 * <p>
 *
 * @param rgbFrame
 * @param nirFrame
 * @param livenessDetectionOption
 * @param faceTrackCallback
 * @param livenessCallback
 */
public void startFeature(final ImageFrame rgbFrame,
                        final ImageFrame nirFrame,
                        final LivenessDetectionOption livenessDetectionOption,
                        final FaceTrackCallback faceTrackCallback,
                        final LivenessCallback livenessCallback);
```

## 人脸属性分析接口

[illegible]

# 驾驶员行为分析接口

```
/**
 * 驾驶员行为分析接口
 *
 * 使用场景:获取图像中人脸的驾驶员相关属性:抽烟、进食、饮水、使用手机、注意力、安全带等
 *
 * @param rgbFrame
 * @param nirFrame
 * @param driveOption
 * @param driveCallback
 */
public void startDrive(final ImageFrame rgbFrame,
                      final ImageFrame nirFrame,
                      DriveOption driveOption,
                      FaceTrackCallback faceTrackCallback,
                      DriveCallback driveCallback);
```

# M:N人脸搜索接口

```
/**
 * 识别接口 M:N
 * 判断是否在注册库中 , 若在则返回具体的身份信息
 * 开启检测流程, 如人人脸检测->跟踪->活检->识别->检索
 * <p>
 * 注:暂无人脸跟踪能力时, 调用方则可关闭人脸跟踪开关, 只做识别检测, 无需实现跟踪Callback; 若支持跟踪, 跟踪
 * 活检的情况, 调用方通过开关控制。
 *
 * @param rgbFrame
 * @param nirFrame
 * @param depthFrame
 * @param livenessDetectionOption
 * @param faceTrackCallback
 * @param livenessCallback
 */
public synchronized void startIdentification(final ImageFrame rgbFrame,
                                             final ImageFrame nirFrame,
                                             final ImageFrame depthFrame,
                                             final LivenessDetectionOption livenessDetectionOption,
                                             final FaceTrackCallback faceTrackCallback,
                                             final LivenessMultiCallback livenessCallback);
```