



云从科技安卓人脸 SDK 开发文档

Copyright© 2016 CloudWalk Technology Co., Ltd.

First printing, July 2016



前言

作为国内最先进的人脸识别技术服务提供商，我们始终坚持聚焦战略，对图像识别基础算法、云平台和智能终端设备等持续进行研发投入，以尖端技术创新和客户需求为驱动，使公司始终处于国际领先水平，引领行业的发展，是我们一直努力的方向。

开发文档会为您提供 SDK 基本的开发指南和技术背景介绍，帮助您更好地使用我们的技术服务。

衷心感谢您对我们技术与产品的信任和支持！

修改记录

版本号	拟制/修改日期	主要更改内容
5.0.0	2018.09.09	新版本 SDK 首次撰写
5.1.0	2018.10.23	属性接口封装
5.2.0	2018.11.19	红外双目活体接口封装

目录

1. SDK 使用说明.....	1
1.1. 头文件.....	1
1.2. 库文件.....	1
1.3. 模型文件.....	1
1.4. 示例程序 DEMO.....	2
1.5. 开发建议.....	2
2. SDK 授权说明.....	3
2.1. 授权.....	3
2.2. 并发控制与性能控制.....	3
3. SDK 开发文档.....	4
3.1. 数据结构.....	4
3.1.1. 输入图像.....	4
3.1.2. 人脸检测功能参数.....	6
3.1.3. 人脸综合信息.....	6
3.1.4. 错误码说明.....	8
3.1.5. 人脸预处理功能开关.....	11
3.2. 人脸预处理接口.....	13
3.2.1. 创建检测器句柄.....	13
3.2.2. 销毁检测器句柄.....	13
3.2.3. 获取人脸功能参数.....	13
3.2.4. 设置人脸功能参数.....	14
3.2.5. 人脸功能操作.....	14
3.2.6. 清除检测跟踪状态信息.....	15
3.3. 特征提取和比对接口.....	16
3.3.1. 创建识别句柄.....	16
3.3.2. 销毁识别句柄.....	16
3.3.3. 获取特征长度.....	16
3.3.4. 提取用于注册的人脸特征.....	17
3.3.5. 人脸特征比对.....	18
3.4. 人脸属性接口.....	19
3.4.1. 创建年龄段、性别、人种句柄.....	19
3.4.2. 销毁识别句柄.....	19
3.4.3. 获取年龄段估计.....	19
3.4.4. 获取性别估计.....	20
3.4.5. 获取人种估计.....	21
3.5. SDK 版本信息接口.....	24
3.5.1. 获取 SDK 当前版本信息.....	24



3.5.2.	获取设备唯一码.....	24
3.5.3.	获取移动端授权码.....	24

1. SDK 使用说明

安卓 SDK 中主要包括底层头文件，库文件，jni 层库文件，java 层接口，模型文件以及 Demo 示例。需要注意的是，Demo 中加载模型的路径为 sdcard 根目录，想要实现 Demo 中功能，需要将“CWModels”整个文件夹复制到 sdcard 根目录。

1.1. 头文件

底层接口为 C 标准接口，位于“CWFaceSDK”文件夹中，4 个头文件分别对应人脸检测“CWFaceDetection.h”，人脸识别“CWFaceRecognition.h”，数据结构配置文件“CWFaceConfig.h”，版本信息文件“CWFaceVersion.h”。使用 C++ 进行 NDK 开发，或者进行 JNI 封装，需要用到这些底层 C 接口。

1.2. 库文件

库文件位于“libs”文件夹中，支持“armeabi-v7a”和“arm64-v8a”环境，主要包含一下三项库文件：

底层核心算法库：三个，分别是“libCWFaceDetTrack.so”，“libCloudWalkRecog.so”和“libCWFaceSDK.so”，其中，对外接口均封装在“libCWFaceSDK.so”中；

JNI 层封装库：“libCWFaceSDKJni.so”，封装底层算法库，提供 java 能直接调用的 JNI 接口；

安卓层 jar 包：“CWFaceSDK.jar”，对接 JNI 层库，提供安卓可以直接调用的 java 接口，该 jar 包通过 Demo 中“package cn. face.sdk”导出。用户进行安卓开发，可选择调用该 jar 包，也可以直接使用“package cn. face.sdk”中的 java 文件。

1.3. 模型文件

模型文件位于“CWModels”文件夹中，包括检测模型，比对识别。人脸检测加载“_configs_frontend_x86_arm.xml”，人脸比对加载“CWR_Config_1_1.xml”。建议将模型文件夹放入 sdcard 根目录，创建句柄时传入要加载的 xml 路径即可。

1.4. 示例程序 Demo

Android 平台提供了两个 Demo:

一个 NDK 开发，目录为“Demo_NDK”，使用 C++编写，直接用 NDK 编译，将编译好的可执行程序连同 so 库，模型文件一起拷贝到手机中即可运行。该 demo 包含四个功能：人脸检测，人脸质量，人脸比对，人脸识别。

一个 APP 开发，目录为“Demo_APP”，该 Demo 主要包含两个包“cn.face.sdk”和“cn.face.localsdkdemo”。“cn.face.sdk”对接 jni 层 so 库，提供 java 可调用的 native 接口，以及 sdk 中需使用的数据结构。“cn.face.localsdkdemo”在 sdk 包的基础上做了便于调用的封装，开发了一个功能展示程序：包括人脸检测，关键点，比对，识别，属性以及实时人脸分析。该 Demo 中的授权码已过时，需要向云从科技申请试用或者正式的授权码。

1.5. 开发建议

用户开发可以直接使用 1.4 章节里“Demo_APP”中的 sdk 包“cn.face.sdk”，也可使用 1.2 章节里讲到的“libs”文件夹中的 jar 包，本质上是一样的。

关于线程和句柄的使用：**建议程序启动时创建好句柄和线程，一个线程对应一个句柄，程序退出时销毁句柄和线程。**不要程序里不停的创建和销毁句柄，会造成大量内存和时间的耗用。

模型文件夹建议放到 sdcard 根目录，创建句柄时输入路径加载即可。

2. SDK 授权说明

2.1. 授权

本套 SDK 需要授权码才能成功创建检测识别句柄，该授权码可通过网络授权接口“cwGetLicence”获取，也可通过云从授权工具离线获取。

2.2. 并发控制与性能控制

本套 SDK 的所有函数接口**都不是**线程安全的。也即是说，使用同一检测器句柄，在多线程环境下同时进行人脸检测、跟踪、关键点提取、质量评估，会造成未定义错误甚至应用程序崩溃。为了达到并发的目的，您应该为每个线程创建独立的句柄以实现本 SDK 提供的功能，因此您能够创建的句柄总数由授权的并发数决定。

并发数是通过授权码来进行控制，授权的最大并发数由云从科技提供。

性能控制主要包括底库数量和每秒比对的次数，这两个参数都是通过授权码来控制。

3. SDK 开发文档

3.1. 数据结构

3.1.1. 输入图像

3.1.1.1. 图像格式

名称: cw_img_form_t

功能: 定义图像所采取的编码格式

声明:

```
interface cw_img_form_t {  
    int CW_IMAGE_GRAY8 = 0;  
    int CW_IMAGE_BGR888 = 1;  
    int CW_IMAGE_BGRA8888 = 2;  
    int CW_IMAGE_RGB888 = 3;  
    int CW_IMAGE_RGBA8888 = 4;  
    int CW_IMAGE_YUV420P = 5;  
    int CW_IMAGE_YV12 = 6;  
    int CW_IMAGE_NV12 = 7;  
    int CW_IMAGE_NV21 = 8;  
    int CW_IMAGE_BINARY = 9;  
}
```

描述:

- CW_IMAGE_GRAY8:** 单通道灰度图 (1 byte / pixel)
- CW_IMAGE_BGR888:** 三通道 BGR 图 (1 byte / pixel / channel)
- CW_IMAGE_BGRA8888:** 四通道 BGRA 图 (1 byte / pixel / channel)
- CW_IMG_RGB888:** 三通道 RGB 图 (1 byte / pixel / channel)
- CW_IMG_RGBA8888:** 四通道 RGB A 图 (1 byte / pixel / channel)
- CW_IMAGE_YUV420P:** YUV 4:2:0 12bpp (双通道, 一个是连续亮度通道, 另一个是 U/V 分离交错通道)
- CW_IMAGE_YV12:** YVU 4:2:0 12bpp (三通道, 一个是连续亮度通道, 另两个是连续 U 分量和 V 分量通道)
- CW_IMAGE_NV12:** YUV 4:2:0 12bpp (双通道, 一个是连续亮度通道, 另一个是 U/V 分离交错通道)
- CW_IMAGE_NV21:** YUV 4:2:0 12bpp (双通道, 一个是连续亮度通

道，另一个是 V/U 分离交错通道)

CW_IMAGE_BINARY: 图像文件的二进制流 (jpg、bmp、png 等)

3.1.1.2. 图像旋转

名称: cw_img_angle_t

功能: 定义图像需要旋转的角度, 逆时针方向

声明:

```
interface cw_img_angle_t {  
    int CW_IMAGE_ANGLE_0 = 0;  
    int CW_IMAGE_ANGLE_90 = 1;  
    int CW_IMAGE_ANGLE_180 = 2;  
    int CW_IMAGE_ANGLE_270 = 3;  
}
```

描述:

CW_IMAGE_ANGLE_0: 不旋转

CW_IMAGE_ANGLE_90: 逆时针旋转 90 度

CW_IMAGE_ANGLE_180: 逆时针旋转 180 度

CW_IMAGE_ANGLE_270: 逆时针旋转 270 度

3.1.1.3. 图像镜像

名称: cw_img_mirror_t

功能: 定义图像的镜像方式

声明:

```
interface cw_img_mirror_t {  
    int CW_IMAGE_MIRROR_NONE = 0;  
    int CW_IMAGE_MIRROR_HOR = 1;  
    int CW_IMAGE_MIRROR_VER = 2;  
    int CW_IMAGE_MIRROR_HV = 3;  
}
```

描述:

CW_IMAGE_MIRROR_NONE: 不镜像

CW_IMAGE_MIRROR_HOR: 垂直镜像

CW_IMAGE_MIRROR_VER: 水平镜像

CW_IMAGE_MIRROR_HV: 垂直和水平镜像

3.1.2. 人脸检测功能参数

名称: FaceParam

功能: 设置人脸检测的各项参数，影响各个子功能的性能和效果，可以根据实际使用场景进行设置。

声明:

```
public class FaceParam {  
    public int    roiX;  
    public int    roiY;  
    public int    roiWidth;  
    public int    roiHeight;  
  
    public int    minSize;  
    public int    maxSize;  
}
```

描述:

roiX: ROI（感兴趣区域）的左上角 x 坐标。默认为 0

roiY: ROI（感兴趣区域）的左上角 y 坐标。默认为 0

roiWidth: ROI（感兴趣区域）的宽度。默认为 0，表示输入图像的宽度

roiHeight: ROI（感兴趣区域）的高度。默认为 0，表示输入图像的高度

minSize: 检测出的最小人脸尺寸。PC 端默认 48，移动端默认 100，值越大检测速度越快

maxSize: 检测出的最大人脸尺寸。PC 端默认 600，移动端默认 400，大于该尺寸的人脸将被忽略

3.1.3. 人脸综合信息

名称: FaceInfo

功能: 定义人脸综合信息，包括人脸来源、人脸 ID、人脸框、关键点、对

齐人脸、人脸质量等结果。

声明:

```
public class FaceInfo {  
    public int detected;  
    public int trackId;  
  
    public int x;  
    public int y;  
    public int width;  
    public int height;  
  
    public byte[] alignedData;  
    public int alignedW;  
    public int alignedH;  
    public int nChannels;  
  
    public cw_quality_errcode_t errcode;  
    public float[] scores;  
}
```

描述:

detected: 0: 跟踪到的人脸; 1: 检测到的人脸; 2:检测到但不会被进行后续计算(关键点)的人脸; 3: 可能是静态误检框; 4:大角度人脸; 5:关键点错误; 6:不需再处理的人脸; 7:被估计为低质量人脸。
只有该值为 1, 才有对齐人脸数据和质量分。

trackId: 跟踪 ID, 小于 0 时表示尚未进入跟踪

x: 左上角 x 坐标

y: 左上角 y 坐标

width: 人脸宽

height: 人脸高

alignedData: 对齐后的人脸图像数据

alignedW: 对齐后的人脸图像宽度

alignedH: 对齐后的人脸图像高度

nChannels: 对齐后的人脸图像通道

errcode: 质量分析错误码

scores: 质量分分数项，具体含义（根据数据下标顺序）：

- 0: 人脸质量总分，0~1.0 之间，越大人脸质量越好，推荐范围 0.65-1.0
- 1: 清晰度，越大表示越清晰，推荐范围 0.65-1.0
- 2: 亮度，越大表示越亮，推荐范围 0.2-0.8
- 3: 人脸角度，左转为正，右转为负
- 4: 人脸角度，抬头为正，低头为负
- 5: 人脸角度，顺时针为正，逆时针为负
- 6: 肤色接近真人肤色程度，越大表示越真实，推荐范围 0.5-1.0
- 7: 戴黑框眼镜置信度，越大戴黑框眼镜可能性越大，推荐范围 0.0-0.5
- 8: 戴墨镜的置信分，越大表示戴墨镜的可能性越大，推荐范围 0.0-0.5

3.1.4. 错误码说明

3.1.4.1. 函数通用错误码

名称: cw_errcode_t

功能: 接口返回的错误码

声明:

```
interface cw_errcode_t {  
    int CW_SDKLIT_OK = 0;  
    int CW_UNKNOWN_ERR = 20000;  
    int CW_DETECT_INIT_ERR = 20001;  
    int CW_KEYPT_INIT_ERR = 20002;  
    int CW_QUALITY_INIT_ERR = 20003;  
  
    int CW_DET_ERR = 20004;  
    int CW_TRACK_ERR = 20005;  
    int CW_KEYPT_ERR = 20006;  
    int CW_ALIGN_ERR = 20007;  
    int CW_QUALITY_ERR = 20008;  
    int CW_EMPTY_FRAME_ERR = 20009;  
    int CW_UNSUPPORT_FORMAT_ERR = 20010;  
    int CW_ROI_ERR = 20011;  
    int CW_UNINITIALIZED_ERR = 20012;
```

```
int CW_MINMAX_ERR = 20013;
int CW_OUTOF_RANGE_ERR = 20014;
int CW_UNAUTHORIZED_ERR = 20015;
int CW_METHOD_UNAVAILABLE = 20016;
int CW_PARAM_INVALID = 20017;
int CW_BUFFER_EMPTY = 20018;
int CW_FILE_UNAVAILABLE = 20019;
int CW_DEVICE_UNAVAILABLE = 20020;
int CW_DEVICE_ID_UNAVAILABLE = 20021;
int CW_EXCEEDMAXHANDLE_ERR= 20022;

int CW_RECOG_FEATURE_MODEL_ERR = 20023;
int CW_RECOG_ALIGNEDFACE_ERR = 20024;
int CW_RECOG_MALLOCMEMORY_ERR = 20025;

int CW_RECOG_FEATUREDATA_ERR=20026;
int CW_RECOG_EXCEEDMAXFEASPEED = 20027;
int CW_RECOG_EXCEEDMAXCOMSPEED = 20028;

int CW_RECOG_GROUPSIZE_ERR = 20029;
int CW_RECOG_CONVERT_ERR = 20030;
int CW_RECOG_NOFACEDET = 20031;
}
```

描述:

CW_SDKLIT_OK:	成功
CW_DETECT_INIT_ERR:	初始化人脸检测器失败
CW_KEYPT_INIT_ERR,:	初始化关键点检测器失败
CW_QUALITY_INIT_ERR:	初始化跟踪器失败
CW_DET_ERR:	人脸检测失败
CW_TRACK_ERR:	人脸跟踪失败
CW_KEYPT_ERR:	关键点检测失败
CW_ALIGN_ERR:	人脸对齐失败
CW_QUALITY_ERR:	人脸质量评估失败

CW_EMPTY_FRAME_ERR:	空图像
CW_UNUPPORT_FORMAT_ERR:	不支持的图像格式
CW_ROI_ERR:	ROI 设置错误
CW_UNINITIALIZED_ERR:	功能句柄尚未初始化
CW_MINMAX_ERR:	最小/最大人脸尺寸设置错误
CW_UNAUTHORIZED_ERR:	未授权
CW_METHOD_UNAVAILABLE:	方法无效
CW_PARAM_INVALID:	参数无效
CW_BUFFER_EMPTY:	缓冲区空
CW_FILE_UNAVAILABLE:	文件不存在：如模型不存在等
CW_DEVICE_UNAVAILABLE:	设备不存在：如 GPU 等
CW_DEVICE_ID_UNAVAILABLE:	设备 id 不存在：如 GPU id 等
CW_EXCEEDMAXHANDLE_ERR:	超过授权最大数量
CW_RECOG_FEATURE_MODEL_ERR:	加载特征识别模型失败
CW_RECOG_ALIGNEDFACE_ERR:	对齐图片数据错误
CW_RECOG_MALLOCMEMORY_ERR:	预分配特征空间不足
CW_RECOG_EXCEEDMAXFEASPEED,	超过授权最大提特征速度
CW_RECOG_EXCEEDMAXCOMSPEED,	超过授权最大比对速度
CW_RECOG_GROUPSIZE_ERR,	底库 N 超过最大授权数
CW_RECOG_CONVERT_ERR,	特征转换失败
CW_RECOG_NOFACEDET,	未检测到人脸
CW_LICENCE_JSON_CREATE_ERR,	Json 操作失败
CW_LICENCE_DECRYPT_ERR,	加密失败
CW_LICENCE_HTTP_ERROR,	HTTP 请求失败
CW_LICENCE_MALLOCMEMORY_ERR,	内存分配不足
CW_LICENCE_KEY_DEVICE_ERR,	获取设备文件错误

CW_LICENCE_KEY_LICENSE_ERR,	获取授权文件错误
CW_LICENCE_KEY_INSTALL_ERR,	安装授权文件错误
CW_ATTRI_AGE_GENDER_MODEL_ERR,	加载属性模型失败
CW_ATTRI_EVAL_AGE_ERR,	年龄识别失败
CW_ATTRI_EVAL_GENDER_ERR,	性别识别失败
CW_ATTRI_EVAL_RACE_ERR,	种族识别失败

3.1.4.2. 质量分析错误码

名称: cw_quality_errcode_t

功能: 定义人脸质量分析错误码

声明:

```
interface cw_quality_errcode_t {  
    int CW_QUALITY_OK = 0;  
    int CW_QUALITY_NO_DATA = 20150;  
    int CW_QUALITY_ERROR_UNKNOWN = 20151  
}
```

描述:

CW_QUALITY_OK: 质量分数据有效

CW_QUALITY_NO_DATA: 质量分数据无效, 原因: 尚未检测

CW_QUALITY_ERROR_UNKNOWN: 未知错误

3.1.5. 人脸预处理功能开关

名称: cw_op_t

功能: 定义功能选择开关。根据用户需求, 每项功能都可与其他功能进行自由组合。

声明:

```
interface cw_op_t {  
    int CW_OP_DET = 0;  
    int CW_OP_TRACK = 2;  
    int CW_OP_ALIGN = 8;  
    int CW_OP_QUALITY = 16;  
    int CW_OP_ALL = 30;  
}
```

描述:

- CW_OP_DET:** 人脸检测开关。基础功能，若不显示开启，系统内部也会隐式开启。此功能返回人脸框信息
- CW_OP_TRACK:** 人脸跟踪开关。若开启，会赋予人脸一个唯一 ID，并对视频流中的同一人物保持此 ID
- CW_OP_ALIGN:** 人脸对齐开关。若开启，会计算得到对齐后的人脸图像。否则，对应数据项无效
- CW_OP_QUALITY:** 人脸质量分评估开关
- CW_OP_ALL:** 所有开关综合

3.2. 人脸预处理接口

3.2.1. 创建检测器句柄

名称: cwCreateDetHandle

功能: 创建检测器句柄

声明:

```
int cwCreateDetHandle( String pConfigFile,  
                      String pLicence);
```

形参:

pConfigFile: 模型参数配置文件路径 [in]

pLicence: 移动端授权码 in]

返回值:

检测器句柄

3.2.2. 销毁检测器句柄

名称: cwReleaseDetHandle

功能: 销毁功能句柄的函数, 释放所有资源

声明:

```
int cwReleaseDetHandle(int pDetector);
```

形参:

pDetector: 功能句柄 [in]

返回值:

错误码

3.2.3. 获取人脸功能参数

名称: cwGetFaceParam

功能: 获取功能参数

声明:

```
int cwGetFaceParam(int pDetector, FaceParam param);
```

形参:

pDetector: 功能句柄 [in]

param: 功能参数 [out]

返回值:

函数错误码

3.2.4. 设置人脸功能参数

名称: cwSetFaceParam

功能: 设置功能参数

声明:

```
int cwSetFaceParam(int pDetector, FaceParam param, String pConfigFile);
```

形参:

pDetector: 功能句柄 [in]

param: 功能参数 [in]

pConfigFile: 模型参数配置文件路径 [in]

返回值:

函数错误码

3.2.5. 人脸功能操作

名称: cwFaceDetection

功能: 接受一张独立的图像或视频流中的一帧图像，实现人脸检测、人脸跟踪、关键点检测、人脸对齐、质量评估等功能的自由组合

声明:

```
int cwFaceDetection(int pDetector,  
                    byte[] pFrameImg,  
                    int iWidth,  
                    int iHeight,  
                    int iFormat,  
                    int iAngle,  
                    int iMirror,  
                    int iOp  
                    FaceInfo[] pFaceBuffer);
```

形参:

- pDetector: 功能句柄 [in]
pFrameImg: 待检测图像数据 [in]
iWidth: 图像宽度 [in]
iHeight: 图像高度 [in]
iFormat: 图像格式 [in]
iAngle: 图像角度 [in]
iMirror: 图像镜像 [in]
iOp: 功能开关组合 [in]
pFaceBuffer: 返回的人脸综合信息 [out]

返回值:

成功返回检测到的人脸个数，失败返回函数错误码

3.2.6. 清除检测跟踪状态信息

名称: cwResetDetTrackState

功能: 清除检测跟踪状态信息

声明:

```
int cwResetDetTrackState (int pDetector);
```

形参:

pDetector: 功能句柄 [in]

返回值:

函数错误码

3.3. 特征提取和比对接口

3.3.1. 创建识别句柄

名称: cwCreateRecogHandle

功能: 加载模型文件, 创建识别句柄

声明:

```
int cwCreateRecogHandle( String pConfigurePath,  
                        String pLicence,  
                        int emRecogPattern);
```

形参:

pConfigurePath: 配置文件路径 [in]

pLicence: 移动端授权码 [in]

emRecogPattern: 创建的句柄类型 [in]

返回值:

成功返回识别句柄, 失败返回错误码

3.3.2. 销毁识别句柄

名称: cwReleaseRecogHandle

功能: 销毁识别句柄

声明:

```
int cwReleaseRecogHandle(int pRecogHandle);
```

形参:

pRecogHandle: 识别句柄 [in]

返回值:

错误码

3.3.3. 获取特征长度

名称: cwGetFeatureLength

功能: 获取特征长度

声明:

```
int cwGetFeatureLength(int pRecogHandle);
```

形参:

pRecogHandle: 创建的识别句柄 [in]

返回值:

特征长度

3.3.4. 提取用于注册的人脸特征

名称: cwGetFaceFeature

功能: 提取人脸特征, 一次只能提取一个特征

声明:

```
int cwGetFaceFeature( int pRecogandle ,  
                      byte[] dataAlign,  
                      int iWidth,  
                      int iHeight,  
                      int iChannels,  
                      byte[] pFeatueData);
```

形参:

pRecogHandle: 识别句柄 [in]

dateAlign: 对齐图片数据 [in]

iWidth: 对齐图片宽度 [in]

iHeight: 对齐图片高度 [in]

iChannels: 对齐图片通道 [in]

pFeatueData: 返回的人脸特征, 需要分配空间 [out]

返回值:

错误码

3.3.5. 人脸特征比对

名称: cwComputeMatchScore

功能: 人脸特征比对，计算 1 个特征与 N 个特征的相似度。返回的相似度 scores 的个数为 N

声明:

```
int cwComputeMatchScore (int    pRecogHandle ,  
                        byte[] pFeaProbe,  
                        byte[] pFeaFiled,  
                        int    iFeaFiledNum,  
                        float[] pScores);
```

形参:

pRecogHandle: 识别句柄 [in]

pFeaProbe: 用于检索的人脸特征[in]

pFeaFiled: 用于注册的特征 [in]

iFeaFiledNum: 用于注册的特征数量 [in]

pScores: 返回的相似度分数数组，长度为 iFeaProbeNum [out]

返回值:

错误码

3.4. 人脸属性接口

3.4.1. 创建年龄段、性别、人种句柄

名称: cwCreateAttributeHandle

功能: 加载模型文件, 创建年龄、性别、人种属性句柄

声明:

```
int cwCreateAttributeHandle(String pConfigurePath, String pLicence);
```

形参:

pConfigurePath: 配置文件路径 [in]

pLicence: 移动端授权码, PC 端传 0 即可 [in]

返回值:

创建的识别句柄

3.4.2. 销毁识别句柄

名称: cwReleaseAttributeHandle

功能: 释放年龄段、性别和人种属性句柄

声明:

```
int cwReleaseAttributeHandle(int pAttributeHandle);
```

形参:

pAttributeHandle: 年龄段、性别或人种属性句柄 [in]

返回值:

无

3.4.3. 获取年龄段估计

名称: cwGetAgeEval

功能: 获取年龄段估计

声明:

```
int cwGetAgeEval(int pAttributeHandle, byte[] dataAlign,  
                int iWidth, int iHeight,  
                int iChannels, FaceAttrRet attr);
```

形参:

pAttributeHandle: 年龄段属性句柄 [in]
dateAlign: 对齐图片数据 [in]
iWidth: 对齐图片宽度 [in]
iHeight: 对齐图片高度 [in]
iChannels: 对齐图片通道 [in]
attr: 返回的年龄段估计和置信度

返回值:

错误码

3.4.4. 获取性别估计

名称: cwGetGenderEval

功能: 获取性别估计

声明:

```
int cwGetGenderEval(int pAttributeHandle, byte[] dataAlign,  
                   int iWidth, int iHeight,  
                   int iChannels, FaceAttrRet attr);
```

形参:

pAttributeHandle: 性别属性句柄 [in]
dateAlign: 对齐图片数据 [in]
iWidth: 对齐图片宽度 [in]
iHeight: 对齐图片高度 [in]
iChannels: 对齐图片通道 [in]
attr: 返回的性别估计和置信度

返回值:

错误码

3.4.5. 获取人种估计

名称: cwGetRaceEval

功能: 获取性别估计

声明:

```
int cwGetRaceEval(int pAttributeHandle, byte[] dataAlign,  
                 int iWidth, int iHeight,  
                 int iChannels, FaceAttrRet attr);
```

形参:

pAttributeHandle:	人种属性句柄 [in]
dataAlign:	对齐图片数据 [in]
iWidth:	对齐图片宽度 [in]
iHeight:	对齐图片高度 [in]
iChannels:	对齐图片通道 [in]
attr:	返回的人种估计和置信度

返回值:

错误码

3.5. 红外活体接口

3.5.1. 创建红外活体检测器

名称: cwCreateNirLivenessHandle

功能: 创建红外活体检测器

声明:

```
int cwCreateNirLivenessHandle(String pNirModelPath,  
                               String pRecogModelPath,  
                               String pPairFilePath,  
                               String pLogPath,  
                               float fSkinThresh,  
                               String pLicence);
```

形参:

pNirModelPath: 红外活体检测器模型文件 [in]

pRecogModelPath: 红外活体识别模型文件[in]

pPairFilePath: 匹配文件路径,只支持 640*480 以及 480*640 分辨率 [in]

pLogPath: 要保存 log 目录[in]

fSkinThresh: 肤色阈值,默认 0.35 [in]

pLicence: 授权码(仅用于安卓平台,PC端传 NULL 即可) [in]

返回值:

红外活体句柄 成功返回句柄,失败返回 0

3.5.2. 释放红外活体句柄

名称: cwReleaseNirLivenessHandle

功能: 释放红外活体句柄

声明:

```
int cwReleaseNirLivenessHandle(int pHandle);
```

形参:

pHandle: 红外活体句柄 [in]

返回值:

无

3.5.3. 红外活体检测器接口

名称: cwFaceNirLivenessDet

功能: 红外活体检测器

声明:

```
int cwFaceNirLivenessDet(int pHandle,
                          FaceNisLiveParam pLivenessDetInfo,
                          byte[] pVisData,
                          float visSkinScore, float visKeyPtScore,
                          float[] visKeypt_x, float[] visKeypt_y,
                          byte[] pNirData,
                          float nirSkinScore, float nirKeyPtScore,
                          float[] nirKeypt_x, float[] nirKeypt_y,
                          FaceNisLiveResInfo[] pNirLivenessRes);
```

形参:

pHandle:	红外活体句柄 [in]
pLivenessDetInfo:	输入的红外和可见光图片及关键点信息等[in]
pVisData:	可见光图片数据[in]
visSkinScore:	可见光图片肤色分[in]
visKeyPtScore	可见光图片关键点得分 [in]
visKeypt_x:	可见光图片关键点横坐标数组[in]
visKeypt_y:	可见光图片关键点纵坐标数组[in]
pNirData:	红外图片数据[in]
nirSkinScore:	红外图片肤色分[in]
nirKeyPtScore	红外图片关键点得分 [in]
nirKeypt_x:	红外图片关键点横坐标数组[in]
nirKeypt_y:	红外图片关键点纵坐标数组[in]
pNirLivenessRes	红外活体检测结果[out]

返回值:

错误码

3.6. SDK 版本信息接口

3.6.1. 获取 SDK 当前版本信息

名称: cwGetSDKVersion

功能: 获取 SDK 版本信息

声明:

```
String cwGetSDKVersion();
```

形参:

无

返回值:

SDK 版本信息

3.6.2. 获取设备唯一码

名称: cwGetDeviceInfo

功能: 获取设备唯一码

声明:

```
String cwGetDeviceInfo();
```

形参:

无

返回值:

设备唯一码

3.6.3. 获取移动端授权码

名称: cwGetLicence

功能: 获取移动端授权码 licence

声明:



```
String cwGetLicence(String sAppKey, String sAppSecret, String sProductId);
```

形参:

sAppKey: 授权 AppKey, 需要从云从科技获取 [in]

sAppSecret: 授权 AppSecret, 需从云从科技获取

sProductId: 产品授权 Id, 需从云从科技获取 [in]

返回值:

授权码信息