

人脸识别 JAVA 离线 SDK

用户接入文档

文档名称	人脸识别 JAVA 离线 SDK 用户接入文档
所属平台	Windows java
文档提交日期	2023.11.21

百度在线网络技术（北京）有限公司

(版权所有,翻版必究)

修改记录

No	版本号	修改内容简介	修改日期	修改人
1	V1.1	初版	2018-09-28	
2	V1.2	添加奥比中光 mini 活体及华杰艾米镜头活体等	2019-01-19	
3	
4	V6.1	6.1 版、业务层封装、添加人脸库等	2021-07-20	
5	V6.3	1、依赖库拆减、使 sdk 主库不依赖镜头模组库，镜头模组库需要时候单独添加 2、sdk 自动授权激活 3、新增设备指纹接口 4、人脸库优化，支持中文路径、用户信息支持中文参数，人脸图片入库等 5、模型可根据需要删减、模型能力可定制化设置 6、新增特征值多端对齐示例代码	2021-12-08	
6	V8.1	1、8.0 新模型，新算法 2、rgb 静默活体采用多因子策略，提升防攻击能力 3、证件照和生活照合并，统一采用生活照模式 4、多线程支持 5、默认检测人脸数调整为 1，若需检测多人脸，可修改配置文件	2022-06-06	
7	V8.2	1、去除 FaceBox 中的无用 angle 2、完善多线程使用示例 3、多项文档优化	2022-12-08	
8	V8.3	1、多实例多线程方案 2、禁止运行在 windows-server	2023-06-08	

		3、人脸库支持自定义路径 4、多项文档优化		
9	V8.4	1、优化人脸识别结果（可根据配置返回多个） 2、优化人脸抠图 3、修复人脸库更新等问题 4、多项文档优化	2023-11-21	

目 录

人脸识别 JAVA 离线 SDK	1
用户接入文档	1
1 设计背景	1
2 名词解释	1
3 SDK 简介	2
3.1 功能架构	2
3.2 人脸技术流程	2
3.3 版本及兼容性	3
3.4 依赖库及运行环境	4
4 SDK 包结构	4
5 授权激活	5
5.1 SDK 自动激活	5
5.2 激活工具激活(LICENSETOOL.EXE)	6
5.3 官网离线激活	6
5.4 批量激活	7
6 SDK 的编译运行	7
6.1 ECLIPSE 上编译运行	7
6.1.1 环境安装	7
6.1.2 导入工程	7
6.2 IDEA 上编译运行	11
6.2.1 环境安装	11
6.2.2 导入工程	11
7 SDK 及 DEMO 示例工程	17
7.1 DEMO 示例说明	18
7.2 库文件说明	20
8 模型能力加载及模型说明	20
8.1 模型删减说明	21
8.2 模型路径的定制化	22
8.3 SDK 的配置文件	22
8.3.1 修改配置文件输出日志	23
8.3.2 修改配置文件人脸库路径	23
8.4 能力定制化说明	23

8.4.1 detect.json (人脸检测配置)	23
8.4.2 track.json (人脸追踪配置)	24
8.4.3 action_live.json (动作活体配置)	24
9 功能接口	25
9.1 人脸检测 DETECT 接口	25
9.2 人脸跟踪 TRACK 接口	26
9.3 清除人脸跟踪历史	26
9.4 人脸关键点接口	27
9.5 注意力检测接口	27
9.6 人脸属性检测接口	27
9.7 人脸扣图接口	28
9.8 暗光恢复	28
9.9 眼部状态检测接口	28
9.10 嘴巴闭合检测接口	29
9.11 口罩佩戴检测接口	29
9.12 人脸质量	29
9.12.1 人脸姿态角接口	30
9.12.2 人脸光照检测接口	30
9.12.3 人脸遮挡检测接口	30
9.12.4 人脸模糊度检测接口	30
9.12.5 最优人脸检测接口	31
9.13 特征值及人脸比对 (1:1)	31
9.13.1 人脸特征值接口	32
9.13.2 特征值比对接口	32
9.13.3 人脸 1:1 比对接口	33
9.14 动作活体和静默活体	33
9.14.1 动作活体接口	33
9.14.2 清除动作活体历史接口	34
9.14.3 rgb 静默活体接口	34
9.14.4 nir 静默活体接口	35
9.14.5 rgb+depth 双目静默活体接口	35
9.15 人脸库管理	36
9.15.1 人脸注册接口(通过传入图片帧)	37
9.15.2 人脸注册接口(通过传入特征值)	38
9.15.3 人脸更新接口	38
9.15.4 用户删除接口	39
9.15.5 创建用户组接口	40
9.15.6 用户组删除接口	40
9.15.7 用户信息查询接口	40
9.15.8 用户人脸图片查询接口	41
9.15.9 用户组列表查询接口	42
9.15.10 群组列表查询接口	42
9.15.11 人脸库人脸数量查询	43

9.15.12 人脸识别接口	43
9.16 SDK 系统信息接口	46
9.16.1 获取 SDK 版本号接口	46
9.16.2 获取设备指纹接口	46
10 功能接口对应结构体描述	46
10.1 人脸跟踪信息结构体	46
10.2 人脸框信息结构体	47
10.3 人脸关键点信息结构体	47
10.4 人脸特征值结构体	48
10.5 人脸特征值&人脸框&活体结构体	48
10.6 人脸姿态角结构体	49
10.7 人脸属性信息结构体	49
10.8 人脸遮挡信息结构体	50
10.9 人眼闭合状态结构体	51
10.10 注意力结构体	51
10.11 动作活体配置结构体	52
11 多端特征值对齐	52
12 适配的深度双目摄像头特别说明	52
13 SDK 多线程使用说明	53
14 SDK 的版本升级	53
14.1 SDK 的小版本升级	54
14.2 SDK 的大版本升级	54
15 错误码及错误信息	54
16 常见问题:	55
16.1 SDK 推荐使用 ECLIPSE 或 IDEA 进行开发、也支持 MYECLIPSE 等工具，官方推荐采用 ECLIPSE 工具。根据 32 位或 64 位，请分别选择安装对应位数的 JDK 和 ECLIPSE。	55
16.2 模型文件可定制化：在 MAIN 方法入口，可在 SDKINIT 方法中传入模型文件夹的绝对路径，达到模型文件定制化的目的。若不定制化路径，SDKINIT 中传入空即可，默认模型文件在 SDK 的 MODELS 文件夹里面，无需更改。	56
16.3 激活后是否可以激活文件 LICENSE.INI 和 LICENSE.KEY 拷贝到其他设备运行？	56
16.4 是否支持 DEBUG 模式？只支持 RELEASE 模式，不支持 DEBUG 模式	56
16.5 SDK 支持 x86 和 x64 模式，推荐使用 x64 模式。支持运行在 WIN7 或 WIN10 的 32 位或 64 位系统中。	56
16.6 人脸库不支持中文参数？目前用户信息支持中文，其他参数暂只支持英文、数字等模式，建议采用英文小写字母或汉语拼音替代。	56
16.7 多端人脸特征值对齐的问题？WINDOWS JAVA SDK6.0 和安卓 SDK6.0 的人脸特征值是对齐的，RGB 生活照的特征值均为 512 个 BYTE 字节。	56
16.8 因为 WIN 端 JAVA 人脸 SDK 是采用 JNI 及类反射技术实现，所以 SDK 中的 COM.JNI.STRUCT 中定义的 JAVA 结构体请勿修改结构，FACE.JAVA 中的 API 方法，COM.JNI.FACE 也别修改结构，否则可能会引发调用错误。	56

16.9 在 32 位环境下，ECLIPSE 运行诸如 FACE_SCENE 中的示例时候，有的提示 JAVA. LANG. OUTFOMEMORYERROR 的问题.....	56
16.10 生活照、证件照模式是否可以切换使用？8.0 统一调整为生活照模式，1 寸，2 寸的证 据照也可以理解为生活照模式。	56
16.11 如何排查内存泄漏？频繁调用 SDK 的接口，若发现内存持续增长，则可能存在内存泄 漏，可打开 WINDOWS 自带的资源监视器，勾选运行 SDK 的进程观察。SDK 的所有接口均做了内存 泄漏测试，开发者需要注意示例 DEMO 代码或自己写的代码中是否存在内存泄漏的可能，如 OPENCV 的 MAT 需要 RELEASE 等。	56
16.12 SDK 支持一次初始化，处处运行，在程序生命周期开始时候初始化 SDK，生命周期结 束时候再销毁 SDK。切忌毋需多次初始化 SDK，这样不但耗时，也是不正确的使用方式。	57
16.13 什么情况下会发生设备指纹变更？对设备硬件进行操作，如增删网卡，修改硬件设 备，修改网络 MAC 地址，更换硬盘等操作会改变硬件的设备指纹，导致设备授权失败，如无 必要，请勿做如上操作。	57
16.14 为什么 SDK 只能检测出 1 个人脸？SDK 默认检测 1 个人脸，若需要检测多个人脸，可 通过修改如前述 8.4 的 DETECT. JSON 配置文件实现。	57

1 设计背景

- 场景特点：

- **网络**：对于无网、局域网等情况，无法连接公网，API 方式无法运作。如政府单位、金融保险、教育机构等，其中内网情况最为常见，私有化部署是项目开展的前提条件。

- **安全**：即使可以连接外网，因为人脸数据的敏感性，许多客户不希望将人脸数据传入百度服务器，如大学学生照片、部分企业员工数据等，API 形式也往往不被接受。

- **速度**：由于各地网络线路、机房部署、图片采集方式等诸多原因，API 形式往往耗时较高，容易存在部分请求耗时过长的情况，容易影响业务正常运转。

- **稳定**：API 形式容易受网络抖动、机房故障、线上连带 bug 等影响，存在一定的不稳定因素，可用性保障，往往成为在线调用最容易出现问题的地方。

- 客户特点：

- **1：N-小型人脸库检索**：多为通道通行、固定区域人群验证等需求，如写字楼闸机门禁、企业考勤打卡等，人脸库范围较小，且不易经常变动。

- **1：1-自有数据源对比**：将当前采集的人脸，与其他数据源中的人脸进行对比，如身份证芯片照、教务系统图片、档案图片等，进行快速的 1：1 对比验证。

- 核心需求：

- **基础的人脸采集**：包含人脸检测、跟踪、捕获、质量校验等基础功能，获取符合识别条件的人脸。为之前的客户端 SDK 的标准功能，离线版本 SDK 保留以上所有能力。

- **本地特征抽取**：所有在 SDK 中运行的人脸图片，都可以完成本地特征抽取，以便进行对比或识别操作。

- **1：1 对比**：支持两张图片的相似度对比，可直接传入图片，也可调用本地某个人脸特征；

- **1：N 搜索**：支持一定库大小的人脸查找，在指定的人脸集合中查找最相似的人脸，并返回相似度分值；

2 名词解释

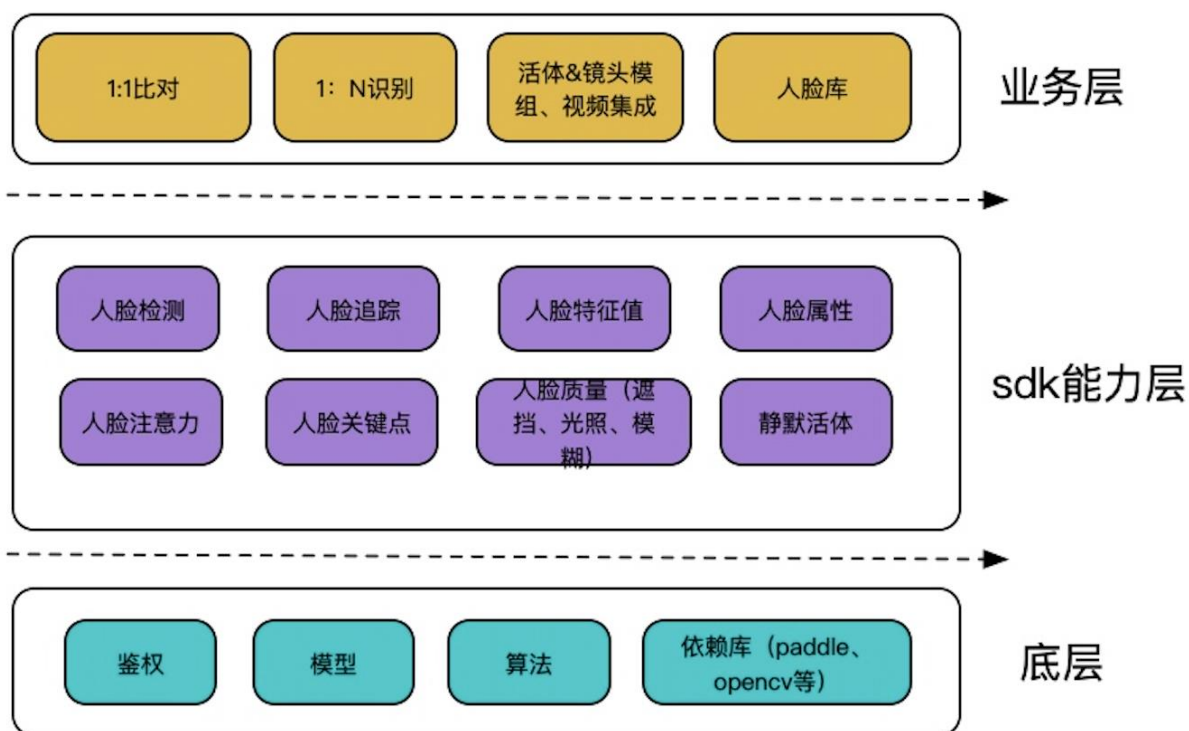
名词	定义
sdk	java 离线人脸识别 sdk(windows 平台)
Eclipse	Java 开发工具，可去 Eclipse 官网下载,同时支持 idea
license	人脸识别激活所需要的激活文件,可利用激活工具生成
key	人脸激活所需的序列号，可从百度 AI 官网申请（ai.baidu.com）
feature	人脸特征值，用来表示人脸特征的 512 个 byte 值
landmark	人脸关键点（72 个关键点）
face_token	对应人脸图片的唯一编码，若一个人上传了 2 张不同图片，则可能有 2 个不同的 face_token,它和图片一一对应

3 SDK 简介

本 SDK 适应于 Windows 平台下的人脸识别系统, 为在 C++ 版本基础上进行 JAVA 接口支持的 SDK, 开发者可在 Eclipse 下面进行开发（推荐使用 Eclipse 官网下载最新支持 java 开发的 Eclipse, 同时也支持使用 idea, 其他工具如 MyEclipse 等也可以使用, 客户可自行琢磨如何运行起来）。SDK 采用 jni 技术调用 C++ 的动态库 DLL 的方式, 所以有少许 DLL 库文件, 这些是 SDK 运行的依赖文件（主要为人脸识别的底层算法库文件等）。

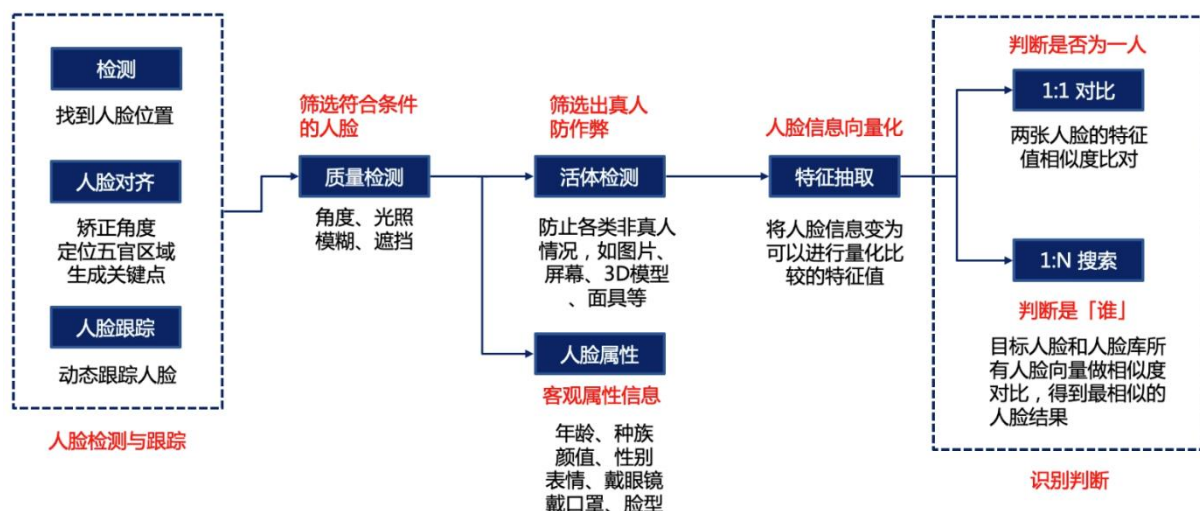
3.1 功能架构

SDK 具有人脸检测、追踪、特征值、静默活体、人脸库、镜头模组集成等诸多功能。架构如下图所示：



3.2 人脸技术流程

SDK 的人脸识别技术流程可如下图所示，通过人脸检测进行人脸特征值的提取，同时，在人脸检测或特征值提取之前可通过人脸质量判断和活体检测进行按需过滤。人脸信息变成可量化的特征值后，可实现人脸特征值的 1:1 比对和人脸的 1: N 识别（N 个人脸特征信息可保存在人脸库中）

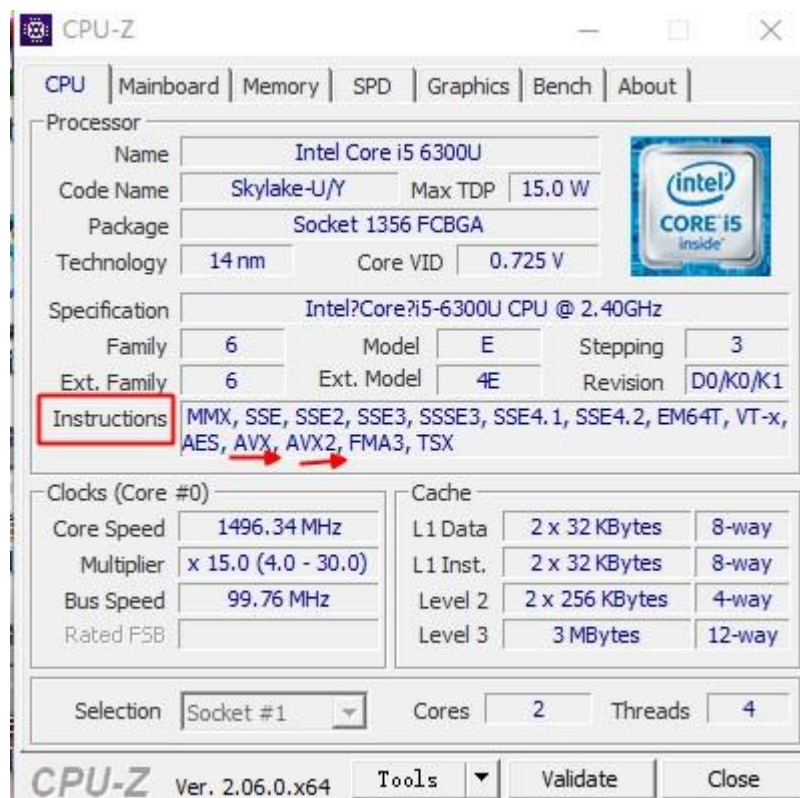


3.3 版本及兼容性

本 SDK 支持 x86、x64 两个平台的版本。

支持 Win7、Win10 操作系统。（Windows server 系列禁用，不支持用作服务端开发）

针对赛扬、奔腾等 cpu 我们有 noavx 版本（通用版不适应这类 cpu, 请使用 SDK 的时候分清楚 cpu 类型是 avx 还是 noavx，若是奔腾、赛扬等 noavx 类型的 cpu，请使用 noavx 版本的 SDK, 查看 cpu 指令集可以使用 SDK 文件夹 tools 中的 cpu-z 文件夹的 exe 安装后查看，若如下图显示 avx 或 avx2 表示是 avx 指令集的 cpu。



推荐使用 Eclipse、idea 等开发工具。

3.4 依赖库及运行环境

本 SDK 在 Eclipse（或 idea）环境下编译。依赖的其他 DLL 库文件包括如 opencv 等都在 SDK 目录，请勿删除。若编译执行提示找不到依赖 DLL 库文件，可根据您的版本在 vc_redist 目录执行 vc_redist.x64.exe 或 vc_redist.x86.exe 进行安装（这些依赖为 vs2015 的运行环境文件，因为底层算法库 DLL 采用 vs2015 编译运行的）。

4 SDK 包结构

SDK 工程包结构如下图所示：

FaceOfflineSdk	#sdk 及示例工程
----.idea	#idea临时文件文件
----db	#人脸库文件夹
----bin	#编译后class文件生成地
----doc	#sdk用户说明文档目录
----conf	#能力定制化配置目录
----images	#示例测试图片目录
----license	#授权文件目录
----models	#sdk模型文件目录
----opencv-jar	#java的opencv jar目录
----src	#sdk的java文件目录
---- --- com.jni.face	#人脸功能及示例包
---- --- --- Face.java	#sdk接口及入口java文件
---- --- --- FaceDemo.java	#sdk示例入口文件
---- ---com.jni.struct	#sdk人脸结构体包
----tools	#sdk工具文件夹
---- ---camera_driver	#镜头模组驱动及库文件夹
---- ---license_tool	#授权工具目录
---- ---vc_redist	#vs依赖目录
----*.dll	#sdk的依赖库文件

5 授权激活

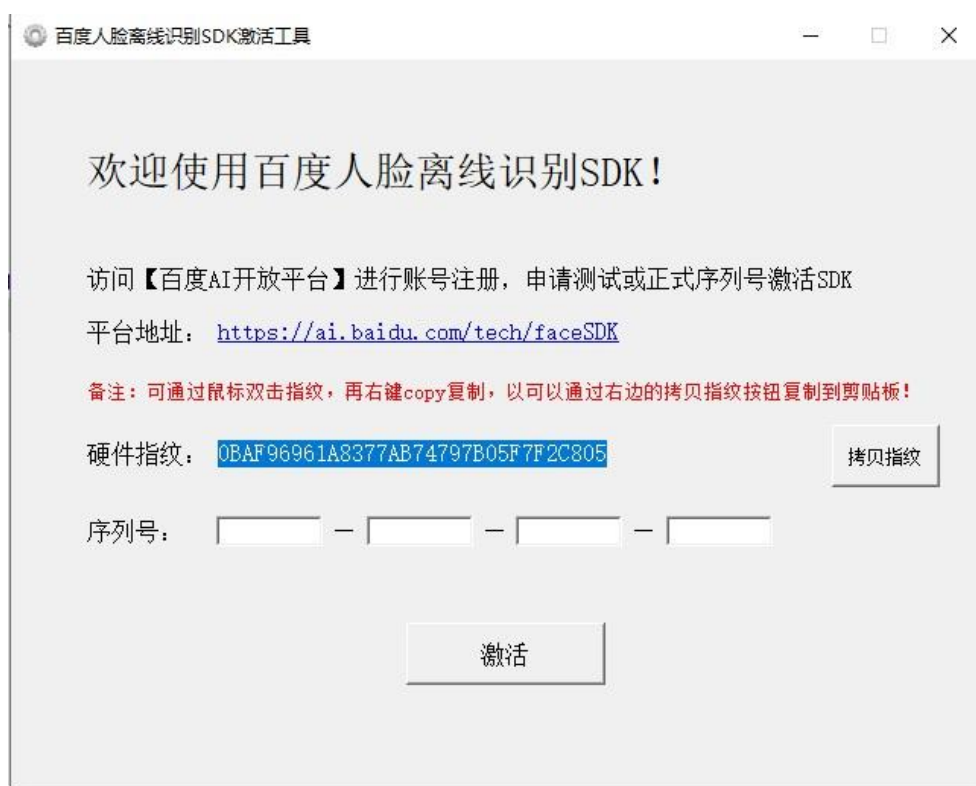
SDK 需要授权激活后才能正常使用，在 SDK 初始化后，若报错误码-13（错误码参考文档最后定义），一般为没有通过授权。通常，SDK 分按设备授权和按应用授权两种方式，大部分采用按设备授权的方式，按应用授权可针对批量大规模客户使用（文档中先只介绍按设备授权，按应用授权可工单或联系百度商务我们提供另外的文档或技术支持）。按设备授权的方式中，SDK 自动激活和激活工具激活需要设备能联网，若设备不能连外网，可采用官网离线激活的方式。

5.1 SDK 自动激活

在 SDK 的目录中有 license 文件夹，里面存放了 2 个文件，license.key 和 license.ini，分别是授权 key 和授权文件，若在百度官网申请了授权系列号 key（16 位），可按 sdk 中的 license 文件夹中 license.key 原格式样子覆盖填写您申请的 key。在设备能联网的情况下，编译运行 SDK 会自动授权激活并拉取新的授权文件 license.ini 覆盖 SDK 中的旧文件。

5.2 激活工具激活(LicenseTool.exe)

在百度官网申请系列号 key（16 位）后，在 FaceOfflineSdk 的 tools->license_tool 目录夹下，双击运行 LicenseTool.exe 激活工具，可见如下所示界面。在输入框中输入 16 位的系列号后，点击激活按钮，稍等片刻，即可见弹出的激活成功对话框，若激活失败，请参考对话框中弹出的 message 信息查找原因。激活成功后会生成 license.zip 文件，解压后获取 license.ini 和 license.key 文件，拿这 2 个文件拷贝覆盖到 sdk 中的 license 文件夹，覆盖对应旧文件，重启 SDK，即可完成授权激活。

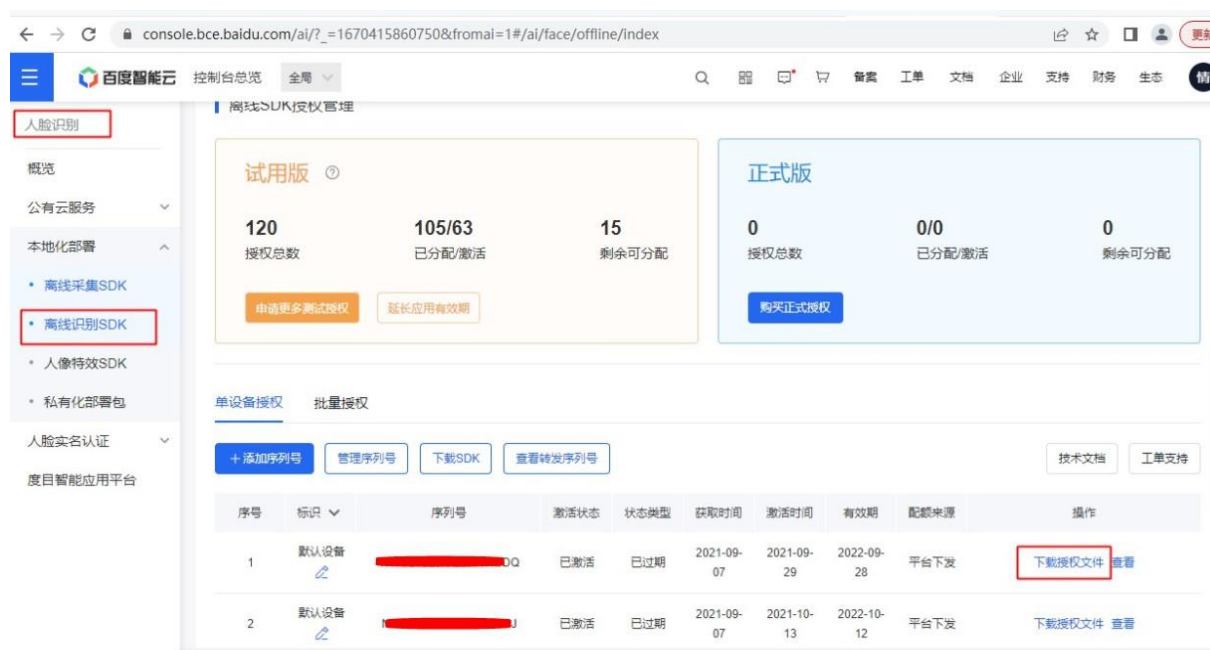


5.3 官网离线激活

若设备不能连外网、通过授权还有另外一种方法，即通过如上 5.2 的激活工具，获取到设备硬件指纹信息，通过百度官网填入指纹信息和申请到的系列号 key，可完成激活并下载获取到 license.ini 文件和 license.key 文件，把这 2 个文件拷贝到 FaceOfflineSdk 的 license 目录下，重新运行 SDK 亦可通过授权激活。

官网申请 key 以及离线激活地址如下：

https://console.bce.baidu.com/ai/?_from=1670415860750&fromai=1#/ai/face/offline

[/index](#)

5.4 批量激活

前述 3 种方式都是按单设备进行授权，SDK 还支持批量进行授权激活，该方式可使用同一个授权 key，对不同设备批量进行激活，适用于激活设备量比较大的客户，具体激活方式可咨询 baidu，单独提供激活文档支持。

6 SDK 的编译运行

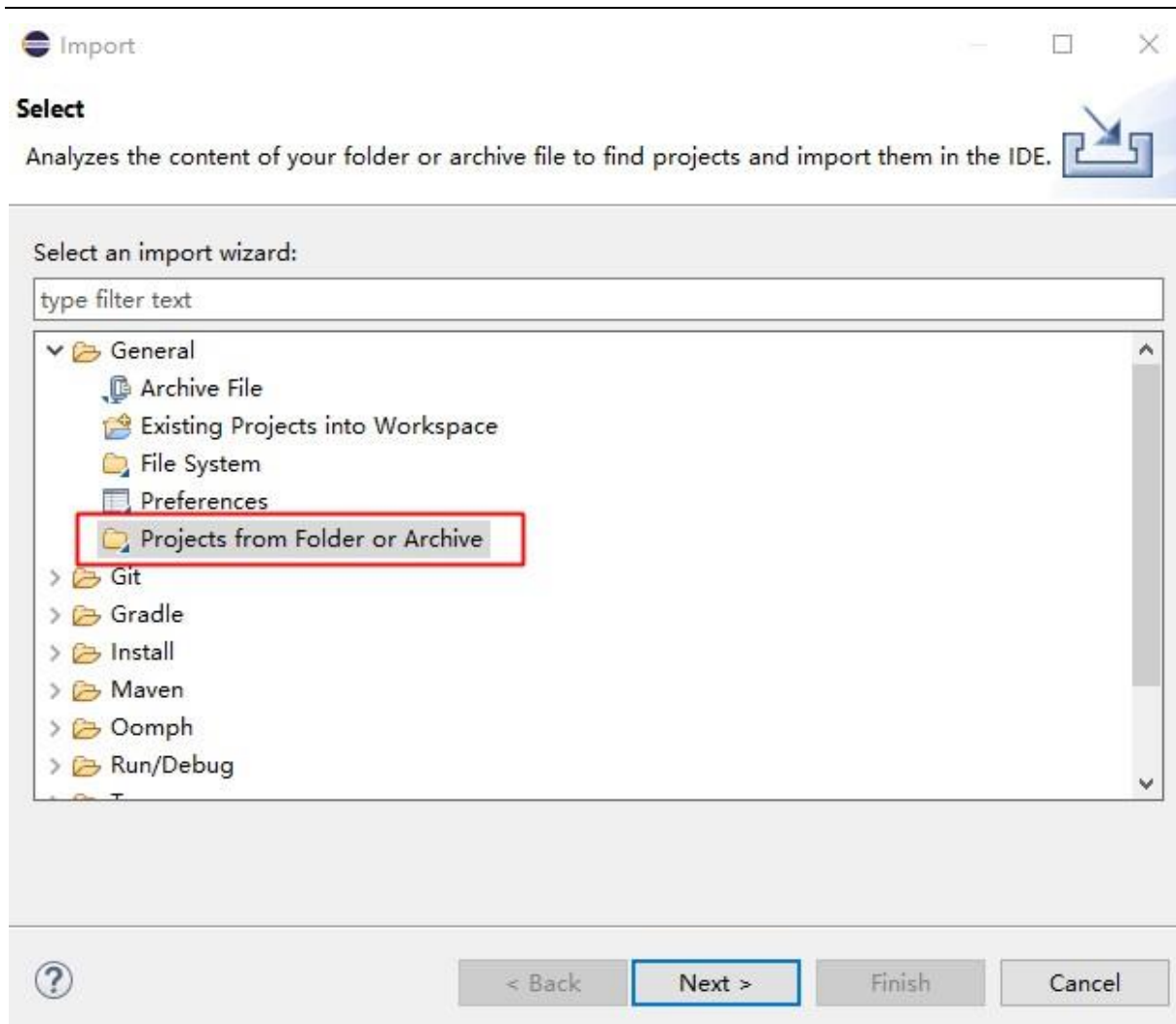
6.1 Eclipse 上编译运行

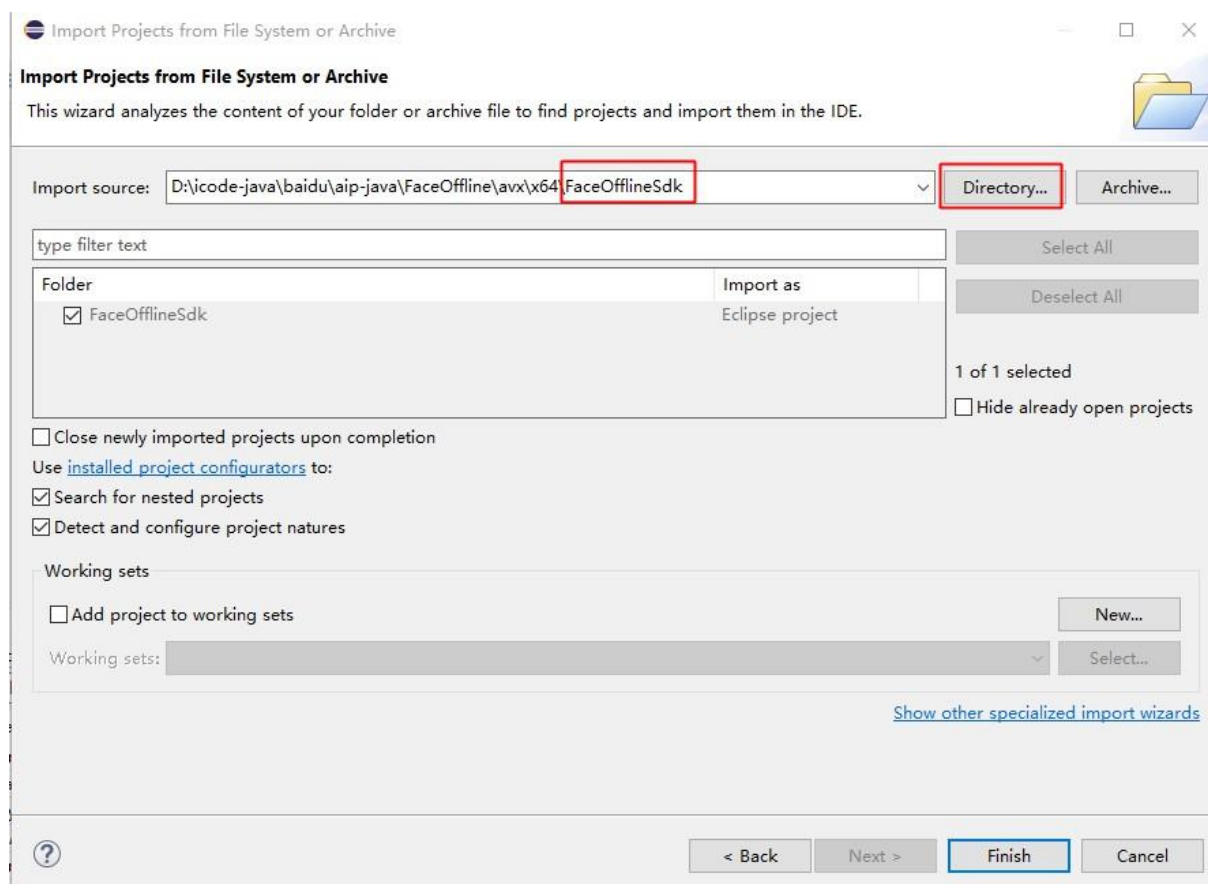
6.1.1 环境安装

为了使 SDK 能在 Eclipse 上正确编译运行，先要安装 Eclipse 环境。首先、安装 jdk，推荐使用 jdk1.8 以上。然后安装 Eclipse(推荐安装 Eclipse 官网的最新能够编译运行 java 的版本)（32 位的请安装 32 位的 jdk 和 Eclipse、64 位的请安装 64 位的 jdk 和 Eclipse）。

6.1.2 导入工程

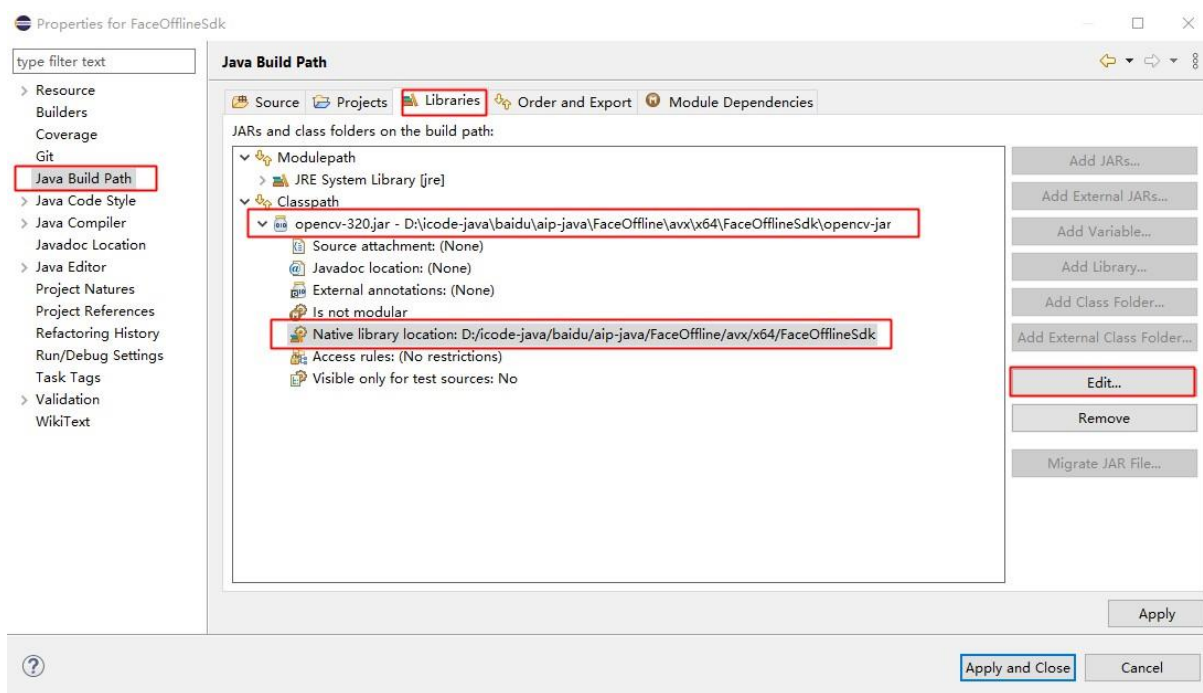
打开 Eclipse，文件->导入，如下图，选择通用->从文件夹导入



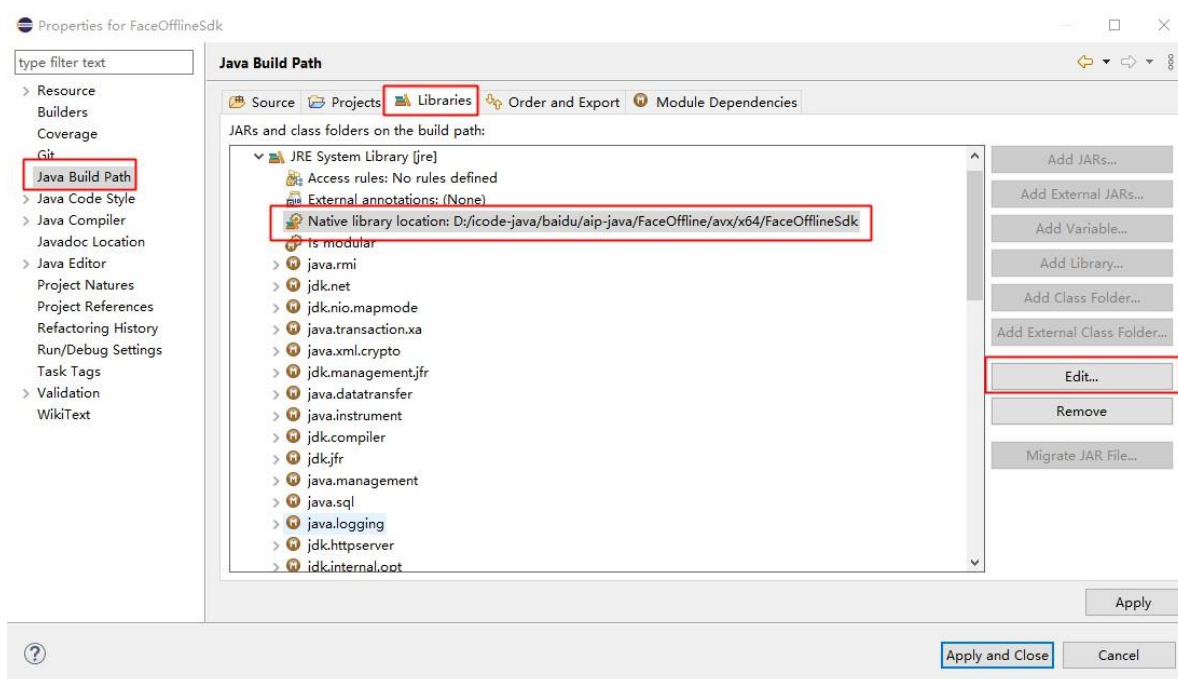


选择下一步，然后在弹出的页面，选路径，填上 SDK 的 FaceOfflineSdk 文件夹，然后点击完成。

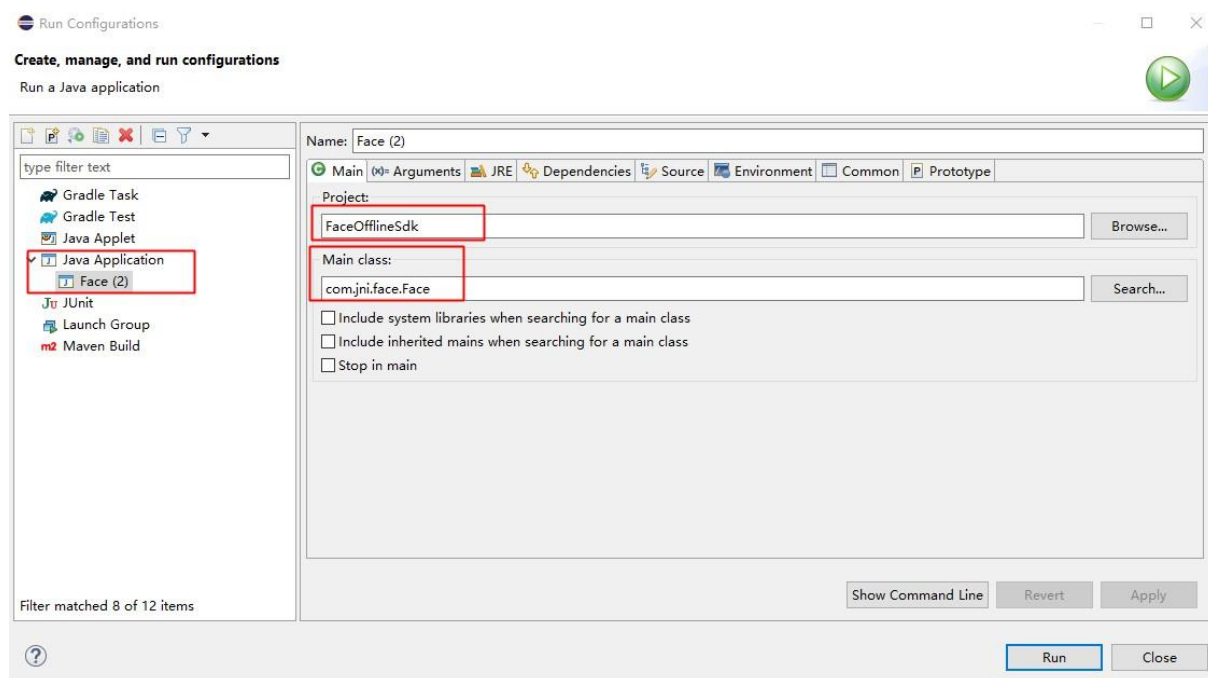
工程导入后，因为没有加载 opencv 库和 face 人脸库，可能会报错，需要另外勾选和加载这些库。右键工程->Properties, Java Build Path->Libraries 后，在如下图三中，通过 Add Jars 加载 opencv 库，选择 opencv-320.jar 在 opencv-jar 目录，后在 jar 中填入 Native library location(对应：opencv_java320.dll)。加载后如图所示：



加载人脸库 BaiduFaceApi.dll。同上，通过 Add JARS, 在 Native library location 选 Edit 定位到 FaceOfflineSdk 目录，这样可确保 java 能找到 c++ 的库文件 BaiduFaceApi.dll 等。如下图所示：



至此，若 java 工程可编译无错误，则表示工程配置正确，接下来就可打开 demo 示例运行 SDK（可参考示例代码，入口文件在 face.java）。可配置 face.java 为 Run 入口方法，配置完毕后即可通过 Run 菜单运行 SDK 工程，配置 Debug Configurations 如下图六所示：



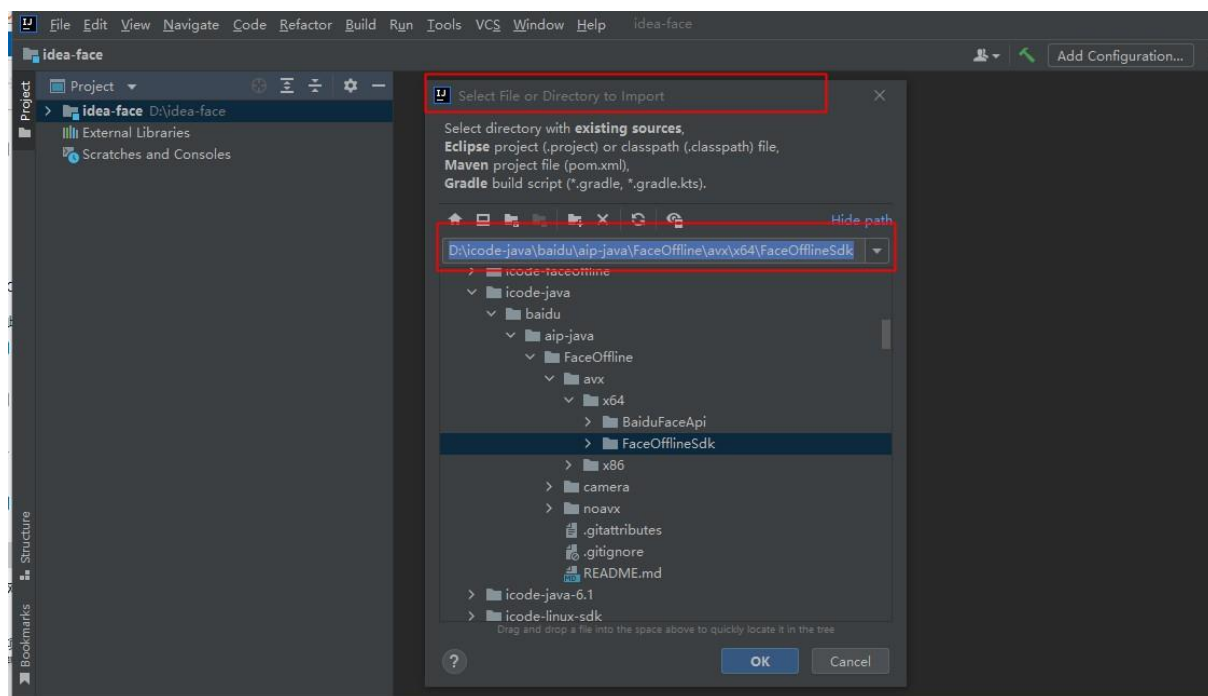
6.2 idea 上编译运行

6.2.1 环境安装

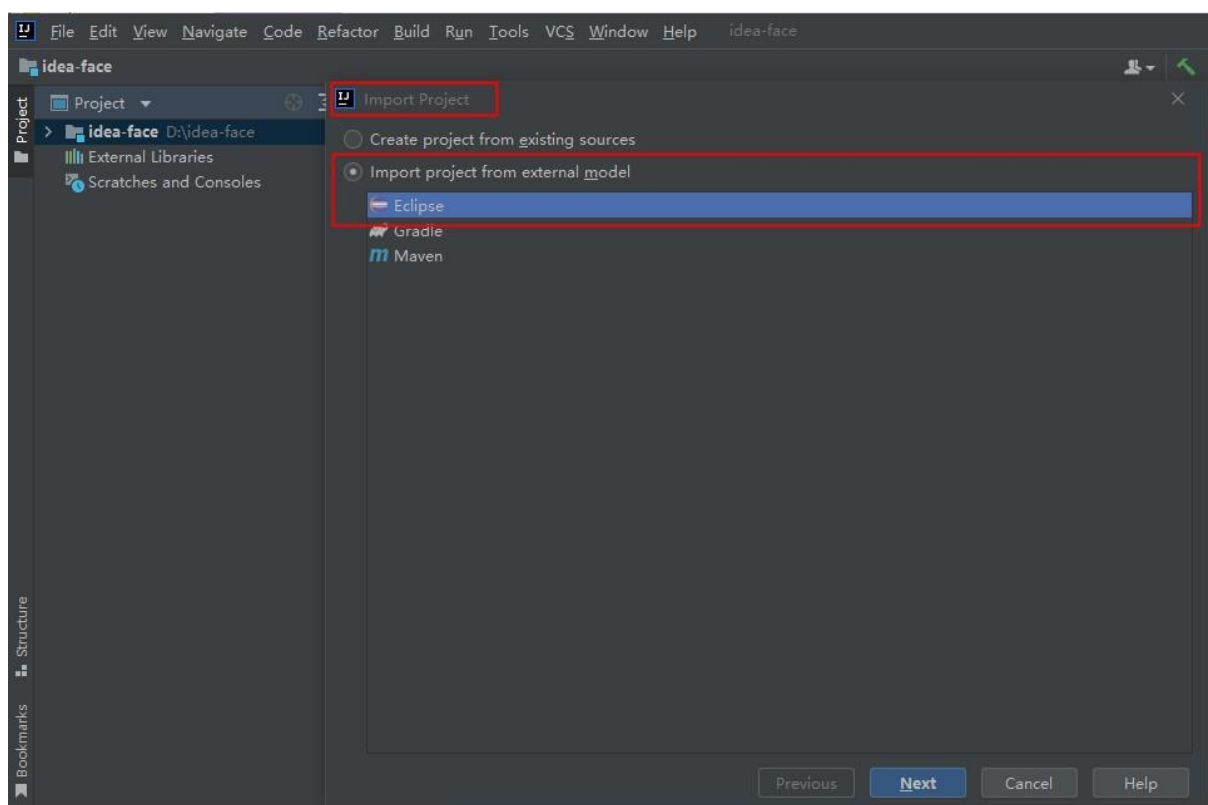
为了使 SDK 能在 idea 上正确编译运行，首先、安装 jdk，推荐使用 jdk1.8 以上。然后安装 idea (推荐安装官网的最新版本)（32 位的请安装 32 位的 jdk 和 idea、64 位的请安装 64 位的 jdk 和 idea）。

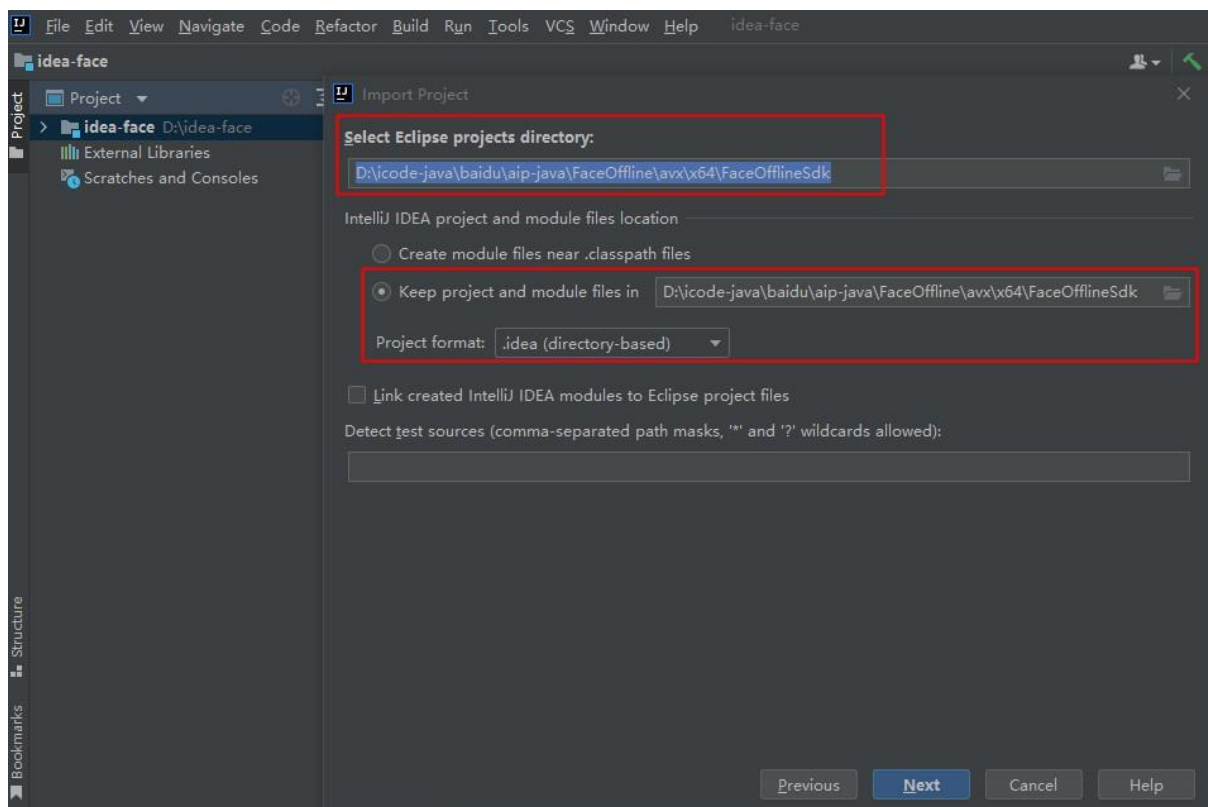
6.2.2 导入工程

首先打开 idea，创建 idea 工作空间，如 idea-face。然后 File->New->Project from existing sources 弹出对话框，选择 FaceOfflineSdk 工程

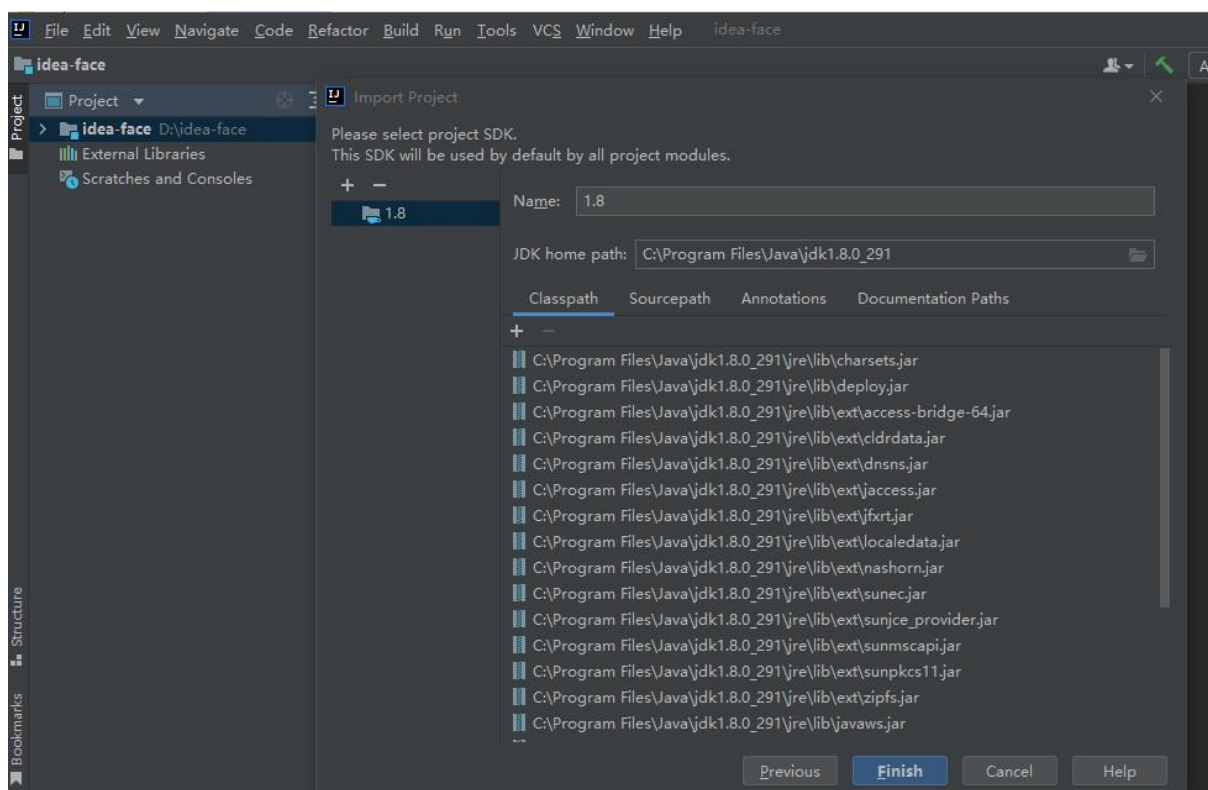


选择 ok，然后：

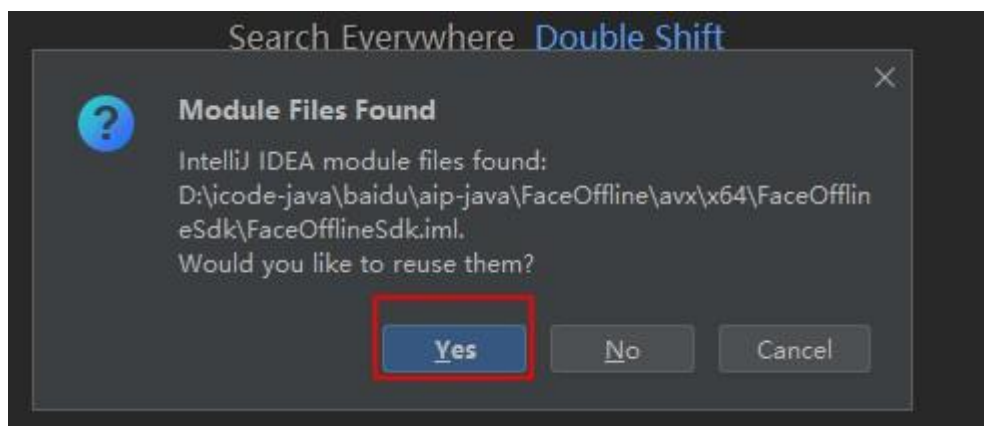




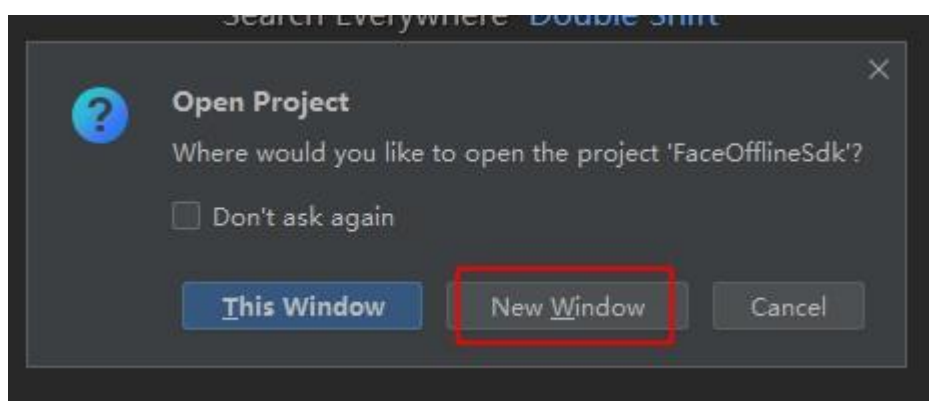
一路 next 到:



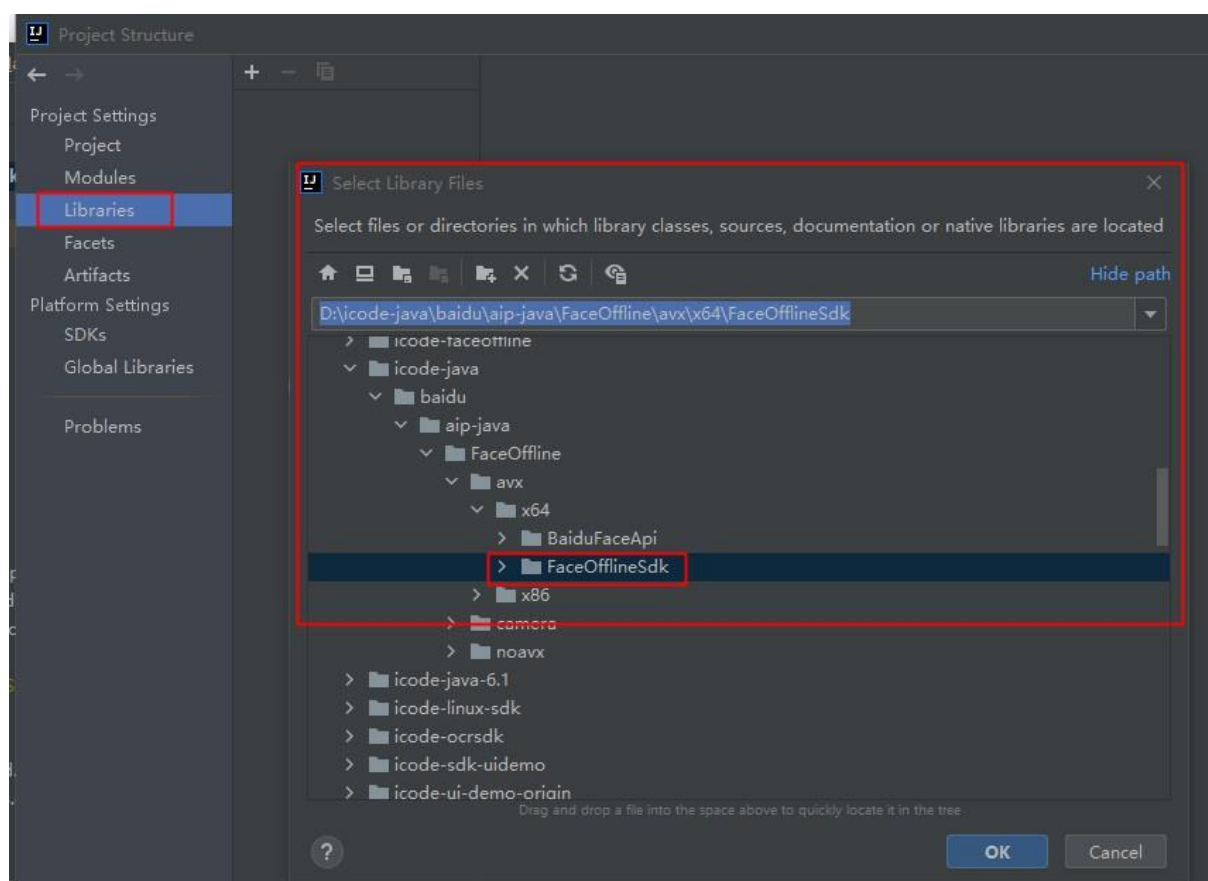
然后 Finish，在弹出的询问是否使用 FaceOfflineSdk.iml，选择 yes



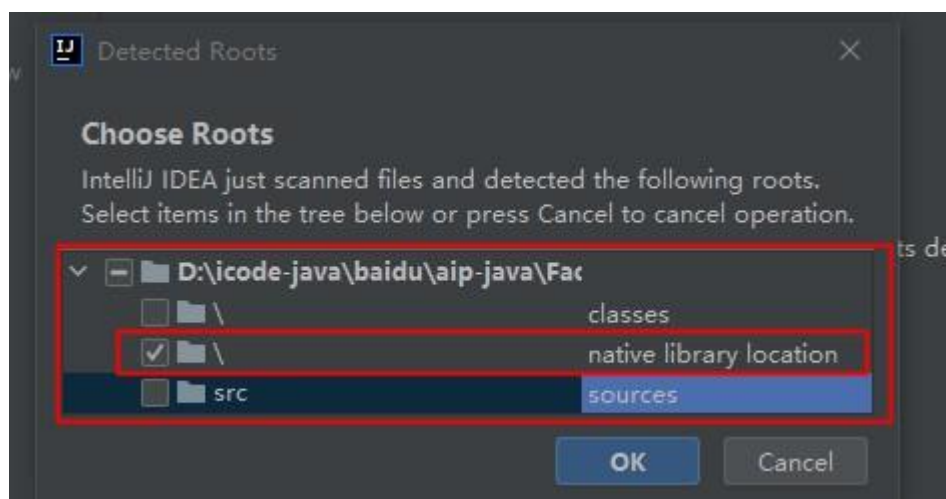
在 Open Project 对话框，选择 New Window。



工程打开后，在 File 选项，选 Project Structure 弹出的对话框中选择 Libraries:

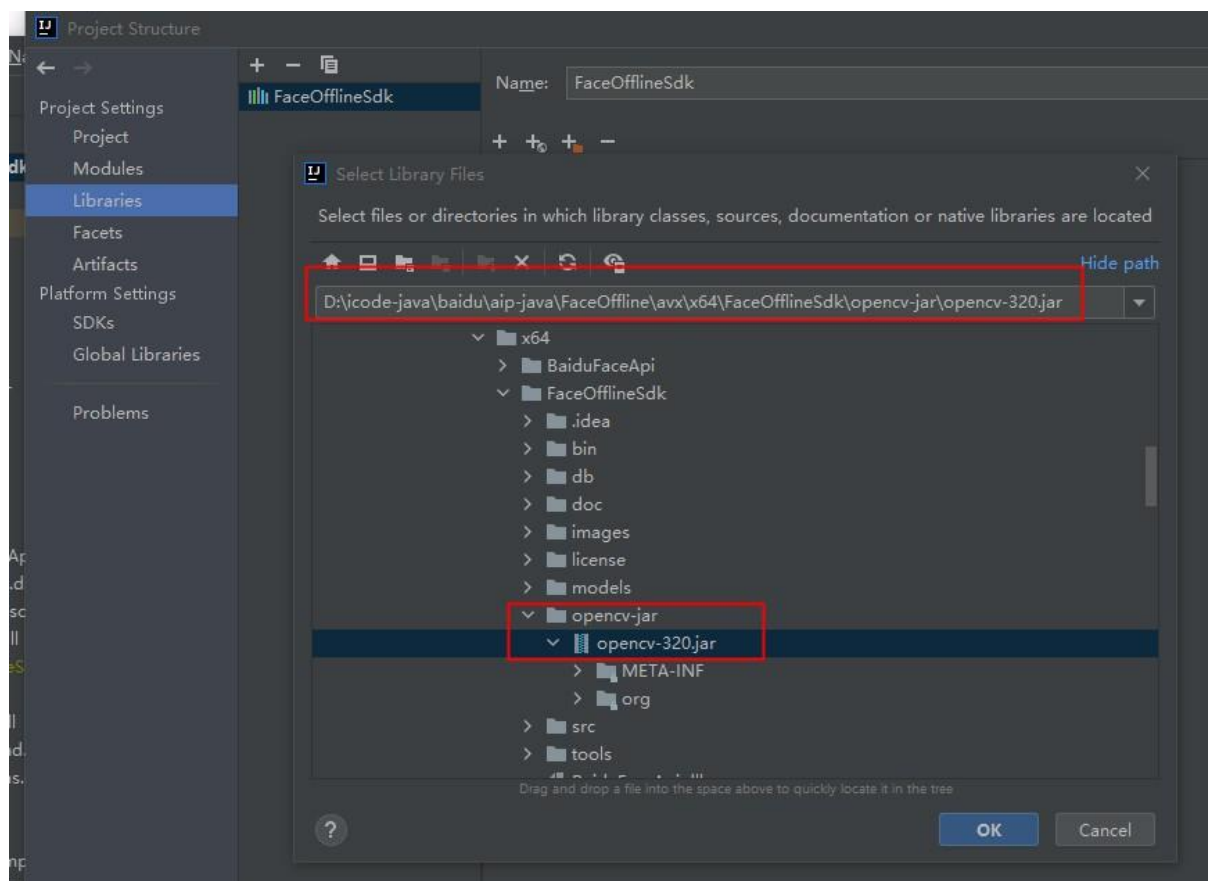


选择 FaceOfflineSdk 根目录，点击 ok 后勾选如下：

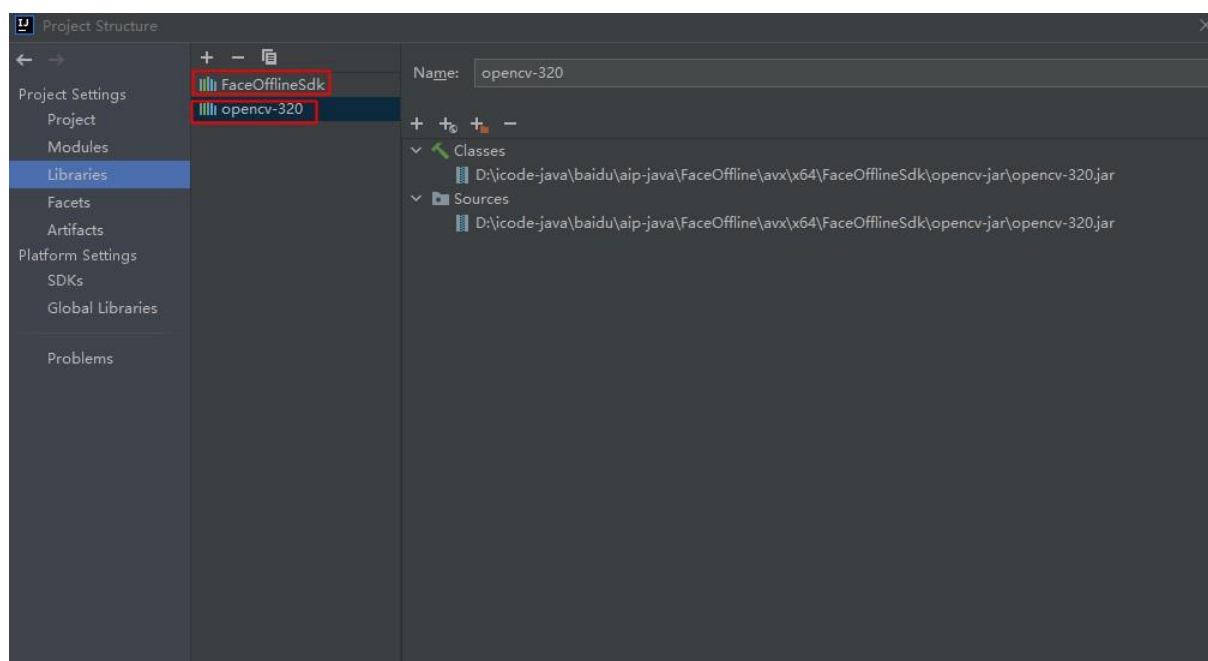


这样，便把 FaceOfflineSdk 根目录的 dll 等加入了依赖库。同样，下一步选择把 opencv-jar 里面的 jar 加入依赖库。

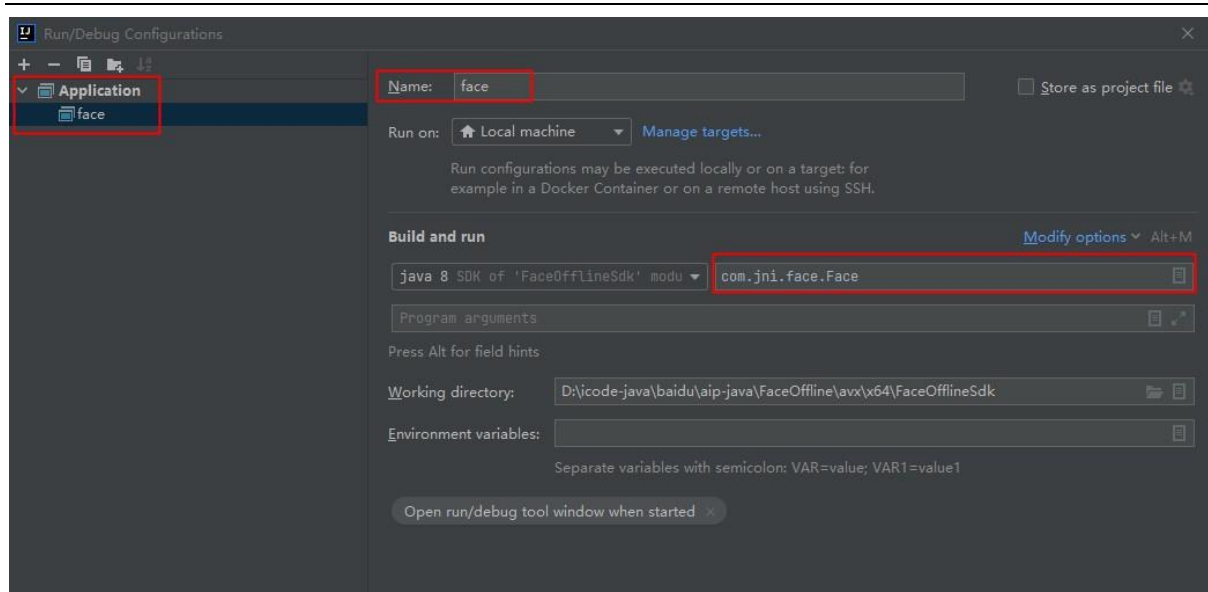
选择 Libraries 右边的+号，勾选 java，点击弹出后定位到 opencv-320.jar 文件如下图：



点击 ok 后，便把 opencv-320 的 jar 加入了库中。如下图：



点击 ok 后，在 run 中进行 Edit Configuration，弹出的对话框中设置如下：



设置完毕这些后，即可 build 选项进行 rebuild 工程，编译无错误后可执行 run。SDK 即正确运行。（若编译或 run 时候提示无法找到或无法加载主类 `com.jni.face`，请检查 `File->Invalidate Caches`，在弹出的对话框中点击 `Invalidate and Restart`，编译 SDK 会生成 `bin` 文件夹，里面有编译出的 `java class` 文件）

7 SDK 及 demo 示例工程

demo 示例工程 `FaceOfflineSdk` 展示了如何集成百度人脸识别离线 SDK。即通过 `jni` 技术调用 `c++` 库文件 `BaiduFaceApi.dll`，引入 SDK 的方法，SDK 的方法定义在 `com.jni.face.Face.java` 中（该类定义了 SDK 的 `api` 方法，挪到你自己的工程也不能改变 `com.jni.face.Face` 以及 `com.jni.struct` 包结构）。SDK 使用如下图所示：


```
public static void main(String[] args) {
    /** sdk初始化 */
    Face api = new Face();
    /** model_path为模型文件夹路径，即models文件夹（里面存的是人脸识别的模型文件）
    /** 传空为采用默认路径，若想定制化路径，请填写全局路径如：d:\\face（models模型文件夹目录放置后为d:\\face\\models）
    /** 若模型文件夹采用定制化路径，则激活文件(license.ini, license.key)也可采用定制化路径放置到该目录如d:\\face\\license
    /** 亦可在激活文件默认生成的路径
    System.out.println("0000");
    String modelPath = "";
    int res = api.sdkInit(modelPath);
    if (res != 0) {
        System.out.printf("sdk-init-fail-and-error=%d\\n", res);
        return;
    }
    System.out.println("1111");
    /** 获取设备指纹
    String deviceId = Face.getDeviceId();
    System.out.printf("device-id:" + deviceId);
    /** 获取版本号
    String ver = Face.sdkVersion();
    System.out.printf("sdk-version:" + ver);
    System.out.printf("2222");
    /** 人脸示例demo
    long begin = TimeUtil.getTimeStamp();
    FaceDemo demo = new FaceDemo();
    demo.faceSample();
    long end = TimeUtil.getTimeStamp();
    System.out.println("time-cost-is:" + (end - begin));
    /** sdk销毁，释放内存防内存泄漏
    api.sdkDestroy();
    System.out.printf("end-sdk");
}
```

在 FaceOfflineSdk 中的 Face.java 中，有定义 SDK 的各个 api 接口方法。

Main 方法中，示例了如何接入 sdk，及其简单，分 2 步如下：

api.sdkInit(model_path); // 第一步：初始化 SDK，可传入定制化模型路径，也可采用默认，默认传""即可，参考示例，sdkInit 会返回错误码，见后续文档定义。

demo.facesample(); // 第二步：执行人脸能力示例

如上，即为调用 sdk 功能的最简单 2 步 2 行代码，当然调用 SDK 在程序退出后，需要释放 SDK 防止内存泄漏，进行 sdk 销毁：api.sdkDestroy();

另外，可通过 is_auth() 方法查看是否通过了授权，通常，需要通过授权，才可以使用 SDK 的各种能力。

7.1 demo 示例说明

示例工程中：分别有以下几个示例 java 文件对应几个常用 SDK 的调用 demo。
解析如下：

文件名	文件说明
Face.java	sdk 总入口及 api 接口，初始化，模型加载，销毁等
FaceDemo.java	Demo 总入口，可打开注释运行各类 demo 示例

FaceDetect. java	人脸检测接口及示例 demo
FaceTrack. java	人脸跟踪接口及示例 demo
FaceFeature. java	人脸特征值接口及示例 demo
FaceCompare. java	人脸 1:1 1:N 比对，特征值比较等接口及示例 demo
FaceLiveness. java	可见光 RGB、红外 IR 活体检测、RGB&IR 双目摄像头静默活体检测，RGB&DEPTH 双目摄像头静默活体检测等
FaceManager. java	人脸库管理类，包括人脸注册、删除，更新、组管理，人脸信息查询等
FaceAttr. java	人脸属性(年龄、性别、种族等) 接口及示例
FaceHeadPose. java	人脸姿态角接口及示例
Faceillumination. java	人脸光照检测接口及示例
FaceOcclusion. java	人脸遮挡度检测接口及示例
FaceBlur. java	人脸模糊度检测接口及示例
FaceActionLive. java	动作活体检测接口及示例
FaceBest. java	最优人脸接口及示例
FaceCrop. java	人脸扣图接口及示例
FaceGaze. java	双眼注意力检测接口及示例
FaceEyeClose. java	眼睛闭合检测接口及示例
FaceMouthClose. java	嘴巴闭合检测接口及示例
FaceMouthMask. java	是否佩戴口罩检测接口及示例
FaceCrop. java	人脸扣图接口及示例
MultiThread. java	多线程运行人脸能力示例
Orbe. java	奥比中光双目摄像头接口(示例请参考 FaceLiveness. java 中的奥比部分)

Aimi.java	华捷艾米双目摄像头接口(示例请参考 FaceLiveness.java 中的华捷部分)
-----------	---

7.2 库文件说明

SDK 根目录有很多 DLL 库文件，这是底层人脸算法库文件以及第三方库文件，需要集成到 SDK 中(集成方法详见前述 Eclipse 或 idea 部分)。

以下对库文件做简要说明：

dll 库文件名称	库文件说明	是否可删除
BaiduFaceApi.dll	百度人脸 sdk 库文件	否
face_sdk.dll	百度人脸 sdk 库文件	否
bd_license.dll	sdk 授权库文件	否
libcurl.dll	第三方 curl 联网库文件	否
libeay32.dll ssleay32.dll	第三方 openssl 库文件	否
libiomp5md.dll paddle_fluid.dll mkldnn.dll mklml.dll	sdk 算法库文件	否
opencv_ffmpeg320_64.dll opencv_java320.dll opencv_world320.dll	第三方 opencv 库文件（可用来打开 usb 视频及显示图片等）	否

除上述 DLL 库文件外，还有 opencv-jar 目录的 opencv-320.jar 文件也需要集成到 SDK 中。

8 模型能力加载及模型说明

8.1 模型删减说明

SDK 支持按需配置模型和加载能力，若 SDK 有某部分功能不需要使用，可尝试删除一些模型，删除后模型既不会加载也不会占用内存。SDK 中 models 文件夹里的模型及是否可删减说明如下表：

模型文件夹名称	说明	是否可删减	删减说明
detect	人脸检测模型	是	若不使用 nir 近红外功能，可删除 detect_nir 开头的模型文件
align	人脸关键点模型	是	若不使用 nir 近红外功能，可删除 align_nir 开头的模型文件
attribute	人脸属性	是	若不使用人脸属性检测功能，该文件夹可删除
best_image	最佳人脸	是	若不使用最佳人脸检测功能，该文件夹可删除
blur	人脸质量模糊度检测	是	若不使用该功能，该文件夹可删除
dark_enhance	暗光恢复	是	若不使用该功能，该文件夹可删除
eye_close	眼睛闭合	是	若不使用眼睛闭合及动作活体功能，该文件夹可删除
feature	人脸特征值	是	若仅使用 rgb 可见光人脸特征值，feature_nir 文件可删除（nir 特征值）
gaze	人脸注意力检测	是	若不使用该功能，该文件夹可删除
mouth_close	嘴巴闭合检测	是	若不使用嘴巴闭合及动作活体功能，该文件夹可删除
mouth_mask	口罩佩戴	是	若不使用该功能，该文件夹可删除

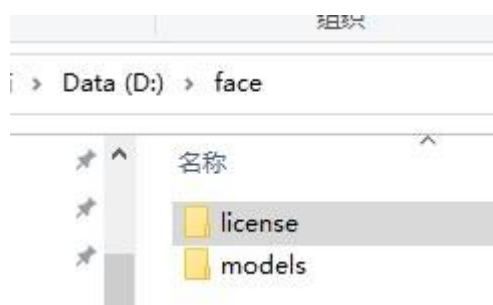
	检测		
occlusion	人脸遮挡 检测	是	若不使用该功能，该文件夹可删除
silent_live	静默活体 检测	是	若只使用 rgb 可见光单目静默活体，则 liveness_nir 和 live_depth 文件可删除

8.2 模型路径的定制化

SDK 支持模型文件夹 models 的路径自定义，当 SDK 初始化 `api.sdk_init(“”)` 传空的时候，为 SDK 支持模型文件夹路径在默认路径，即 models 在 SDK 现有位置。同时也支持 models 通过 `sdk_init` 中传入绝对路径。若把 models 文件夹拷贝到 d 盘的 face 文件夹下面，则可定义：

```
api.sdk_init(“d:\\face”);
```

此时，授权文件夹 license 也需要随之变为 d 盘的 face 文件夹下面。否则会出现授权不通过的问题。即 d 盘 face 文件夹里面同时有 models 文件夹和 license 文件夹。如下图所示：



另外，若使用了能力自定义的 config 文件夹，也需要拷贝到 d 盘的 face 文件夹下面。否则能力定制化也不会生效而是使用的系统默认。

8.3 SDK 的配置文件

SDK 提供配置文件 `face.ini`，通过修改配置文件，可输出文件日志，定制化数据库文件 db 路径等。

同上 8.2，当 `sdk_init` 传 `nullptr` 的时候，`face.ini` 默认在 SDK 的可执行文件 exe 同目录夹中。若 `sdk_init` 传入了绝对路径，则 `face.ini` 文件需要放置在传入的绝对路径文件夹中，同 models 文件夹在同级目录中。

8.3.1 修改配置文件输出日志

为方便排查授权等问题及定位原因，SDK 提供输出文件日志的方式可供查看底层授权等原因，SDK 的配置文件 face.ini 中有配置：

LogType = 1 // 表示输出文件日志，请注意，输出日志会拖慢 SDK 执行速度

LogType = 0 // 表示关闭输出文件日志，项目发布时候请关闭日志。

8.3.2 修改配置文件人脸库路径

人脸库路径 db 文件夹，也可以定制化，同样是通过修改 face.ini 文件。

若 face.ini 定义字段：DbDir = d:\face，则表示人脸库路径会生成在 d 盘的 face 文件夹中，若无需定制，则 face.ini 中删除 DbDir 字段即可。

ini 配置文件中，配置字段前加 #可表示注释。

8.4 能力定制化说明

SDK 支持能力自定义、通过读取配置文件的方式进行能力定制化。SDK 默认能力加载无需定制化，若需要定制化，请把 SDK 根目录里面的 conf 文件夹重命名为 config 文件夹。并且在 SDK 中，把里面的 json 配置按如下说明做修改，可起到定制化能力加载的效果，配置文件简要说明如下（若需定制化修改，请参考示例 json，修改 json 字段的值来达到定制化的目的）

8.4.1 detect.json (人脸检测配置)

配置文件名	detect.json	
说明	人脸检测自定义能力配置文件 <pre>{ "max_detect_num":5, "min_face_size":0, "scale_ratio":-1, "not_face_thr":0.5 }</pre>	
参数	类型	说明
max_detect_num	int	最大检测的人脸数量，最多支持 50，最小 1,默认 1

min_face_size	int	默认 0，可用来设置检测的最小人脸，比如可设为 10，则表示小于 10*10 的人脸，sdk 检测不到
scale_ratio	int	默认-1，表示进行人脸检测时候的图片缩放比率。建议用-1 表示传入原图 sdk 自己缩放（为保证检测效果，该参数建议用默认）
not_face_thr	float	默认 0.5，表示非人脸的阈值，取值范围 0-1

8.4.2 track.json （人脸追踪配置）

配置文件名	track.json	
说明	人脸追踪自定义能力配置文件 <pre>{ "detect_intv_before_track":0.02, "detect_intv_during_track":0.02 }</pre>	
参数	类型	说明
detect_intv_before_track	float	表示人脸追踪前进行人脸检测的时间间隔（单位毫秒）
detect_intv_during_track	float	表示人脸追踪时候进行人脸检测的时间间隔（单位毫秒）

8.4.3 action_live.json (动作活体配置)

配置文件名	action_live.json	
说明	人脸动作活体自定义能力配置文件 <pre>{ "eye_open_threshold":0.5, "eye_close_threshold":0.5, "mouth_open_threshold":0.5, "mouth_close_threshold":0.5, "look_up_threshold":0.5, "look_down_threshold":0.5, "turn_left_threshold":0.5, "turn_right_threshold":0.5, }</pre>	

	<pre> "nod_threshold":0.5, "shake_threshold":0.5, "max_cache_num":1 } </pre>	
参数	类型	说明
eye_open_threshold	float	表示人脸动作活体眼睛睁开的置信度阈值
eye_close_threshold	float	表示人脸动作活体眼睛闭合的置信度阈值
mouth_open_threshold	float	表示人脸动作活体嘴巴张开的置信度阈值
mouth_close_threshold	float	表示人脸动作活体嘴巴闭合的置信度阈值
look_up_threshold	float	表示人脸动作活体抬头的置信度阈值
look_down_threshold	float	表示人脸动作活体低头的置信度阈值
turn_left_threshold	float	表示人脸动作活体向左转头的置信度阈值
turn_right_threshold	float	表示人脸动作活体向右转头的置信度阈值
nod_threshold	float	表示人脸动作活体点头的置信度阈值
shake_threshold	float	表示人脸动作活体摇头的置信度阈值
max_cache_num	float	最大缓存数、推荐为 1 不做修改

9 功能接口

SDK 的各功能接口定义在 Face.java 中。

功能接口的使用示例可参考各示例 java 文件，文件类定义后如 FaceDetect.java 类定义有接口 detect(人脸检测)如(其他接口等分别参考各示例 java 文件中的定义)：

SDK 实现的主要功能有人脸实时跟踪检测、人脸特征值提取、动作活体、RGB&IR 静默活体检测、RGB&DEPTH 静默活体检测、人脸注册、人脸更新、组管理、用户管理以及 1:1 人脸对比, 1:N 人脸识别、特征值的比对和通过 usb 或笔记本自带摄像头检测视频帧，返回识别出的人脸信息、人脸属性等，另外支持对人脸检测进行能力加载设置，达到根据设置进行识别的目的。

各接口功能及传入参数和返回结果等定义如下：

9.1 人脸检测 detect 接口

方法名	detect
-----	--------

说明	人脸检测，返回人脸框结构体数组			
函数	FaceBox[] detect(long matAddr, int type)			
请求参数	说明	必须	类型	示例描述
matAddr	java 的 opencv 图片帧地址	是	long	参考示例
type	检测类型（传 0 表示 rgb 可见光人脸检测，1 表示 nir）	否	int	0 或 1，不传默认为 0
返回信息	函数的返回	是	FaceBox[]	人脸框结构体数组，见后续定义

9.2 人脸跟踪 track 接口

方法名	track			
说明	人脸跟踪，返回人脸信息			
函数	TrackFaceInfo[] track(long matAddr, int type)			
请求参数	说明	必须	类型	示例描述
matAddr	java 的 opencv 图片帧地址	是	long	请参考示例
type	检测类型（传 0 表示 rgb 可见光人脸检测，1 表示 nir）	否	int	0 或 1，不传默认为 0
返回信息	函数的返回	是	TrackFaceInfo[]	人脸信息结构体数组，见后续定义

9.3 清除人脸跟踪历史

方法名	clearTrackHistory			
说明	清除人脸跟踪历史信息			
函数	int clearTrackHistory()			
请求参数	说明	必须	类型	示例描述

返回信息	函数的返回	是	int	0 为正常，其他为错误码见后续定义
------	-------	---	-----	-------------------

9.4 人脸关键点接口

方法名	faceLandmark			
说明	人脸关键点接口			
函数	Landmarks[] faceLandmark(long matAddr, int type)			
请求参数	说明	必须	类型	示例描述
matAddr	java 的 opencv 图片帧地址	是	long	请参考示例
type	0: accurate 模型的 rgb 人脸关键点 1: fast 模型的 rgb 人脸关键点 2: nir 近红外人脸关键点	是	int	
返回信息	函数的返回	是	Landmarks[]	人脸关键点结构体数组，见后续定义

9.5 注意力检测接口

方法名	faceGaze			
说明	注意力检测			
函数	GazeInfo[] faceGaze(long matAddr)			
请求参数	说明	必须	类型	示例描述
matAddr	java 的 opencv 图片帧地址	是	long	请参考示例
返回信息	函数的返回	是	GazeInfo[]	返回注意力结构体数组，见后续定义

9.6 人脸属性检测接口

方法名	faceAttr			
说明	人脸属性检测			

函数	Attribute[] faceAttr(long matAddr)			
请求参数	说明	必须	类型	示例描述
matAddr	java 的 opencv 图片帧地址	是	long	请参考示例
返回信息	函数的返回	是	Attribute[]	返回人脸属性结构体数组，见后续定义

9.7 人脸扣图接口

方法名	faceCrop			
说明	人脸扣图，回传人脸扣图（仅支持单人脸抠图）			
函数	int faceCrop(long matAddr, long outAddr)			
请求参数	说明	必须	类型	示例描述
matAddr	java 的 opencv 图片帧地址	是	long	请参考示例
outAddr	java 的 opencv 图片帧地址	是	long	请参考示例
返回信息	函数的返回: 0 成功，其他错误码	是	int	请参考后面错误码定义

9.8 暗光恢复

方法名	darkEnhance			
说明	暗光恢复（仅支持单人脸图）			
函数	int darkEnhance(long matAddr, outAddr)			
请求参数	说明	必须	类型	示例描述
matAddr	java 的 opencv 图片帧地址	是	long	请参考示例
outAddr	java 的 opencv 图片帧地址	是	long	请参考示例
返回信息	函数的返回: 0 成功，其他错误码	是	int	请参考后面错误码定义

9.9 眼部状态检测接口

方法名	faceEyeClose			
说明	眼部状态检测			
函数	EyeClose[] faceEyeClose(long matAddr)			
请求参数	说明	必须	类型	示例描述
matAddr	java 的 opencv 图片帧地址	是	long	请参考示例
返回信息	函数的返回	是	EyeClose[]	返回眼睛闭合结构体数组，请参考后续定义

9.10 嘴巴闭合检测接口

方法名	faceMouthClose			
说明	嘴巴闭合检测			
函数	float[] faceMouthClose(long matAddr)			
请求参数	说明	必须	类型	示例描述
matAddr	java 的 opencv 图片帧地址	是	long	请参考示例
返回信息	函数的返回	是	float[]	嘴巴是否闭合的置信度分值数组

9.11 口罩佩戴检测接口

方法名	faceMouthMask			
说明	口罩佩戴检测			
函数	float[] faceMouthMask(long matAddr)			
请求参数	说明	必须	类型	示例描述
matAddr	java 的 opencv 图片帧地址	是	long	请参考示例
返回信息	函数的返回	是	float[]	口罩是否佩戴的置信度分值数组

9.12 人脸质量

人脸质量判断可由以下几个因素自由组合综合判断，如姿态角、光照、遮挡、模糊以及最佳人脸。

9.12.1 人脸姿态角接口

方法名	faceHeadPose			
说明	人脸姿态角检测（yaw：左右边转角 roll：屏幕旋转角 pitch：上下偏转角 姿态角推荐范围 -15~15，越大角度越不正不合格）			
函数	HeadPose[] faceHeadPose(long matAddr)			
请求参数	说明	必须	类型	示例描述
matAddr	java 的 opencv 图片帧地址	是	long	请参考示例
返回信息	函数的返回	是	HeadPose[]	人脸姿态角结构体数组，见后续定义

9.12.2 人脸光照检测接口

方法名	faceIllumination			
说明	人脸光照检测（光照分值 0-1、分值越大光照越强，推荐阈值 0.6）			
函数	float[] faceIllumination(long matAddr)			
请求参数	说明	必须	类型	示例描述
matAddr	java 的 opencv 图片帧地址	是	long	请参考示例
返回信息	函数的返回	是	float[]	float 数组，光照值

9.12.3 人脸遮挡检测接口

方法名	faceOcclusion			
说明	人脸遮挡检测（遮挡分值 0-1，分值越大遮挡度越高，推荐阈值 0.6）			
函数	Occlusion[] faceOcclusion(long matAddr)			
请求参数	说明	必须	类型	示例描述
matAddr	java 的 opencv 图片帧地址	是	long	请参考示例
返回信息	函数的返回	是	Occlusion[]	返回遮挡结构体数组，见后续定义

9.12.4 人脸模糊度检测接口

方法名	faceBlur			
说明	人脸模糊检测(模糊度分值 0-1，分值越大模糊度越高，推荐阈值 0.6)			
函数	float[] faceBlur(long matAddr)			
请求参数	说明	必须	类型	示例描述
matAddr	java 的 opencv 图片帧地址	是	long	请参考示例
返回信息	函数的返回	是	float[]	返回模糊度数值 float 数组

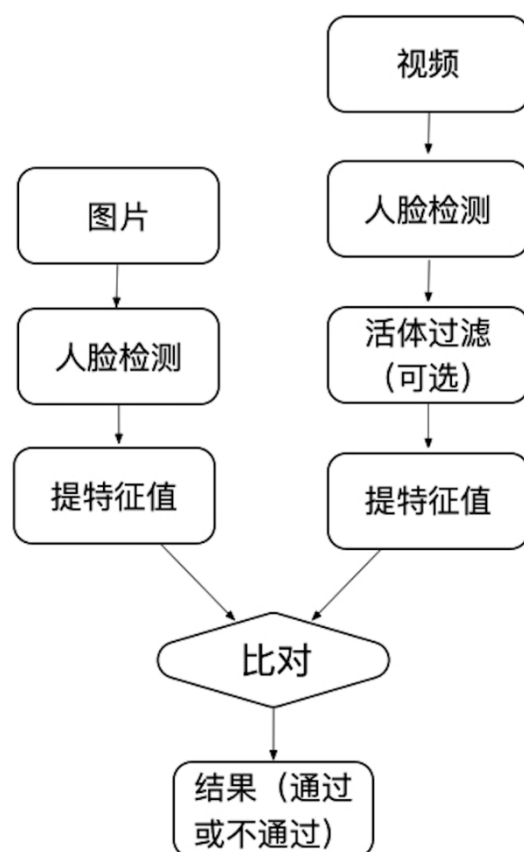
9.12.5 最优人脸检测接口

方法名	faceBest			
说明	最佳人脸检测(最优人脸分值 0-1，分值越高，最佳人脸图片得分越高，推荐阈值 0.5)			
函数	float[] faceBest(long matAddr)			
请求参数	说明	必须	类型	示例描述
matAddr	java 的 opencv 图片帧地址	是	long	请参考示例
返回信息	函数的返回	是	float[]	返回最优人脸分值数组

9.13 特征值及人脸比对（1:1）

人脸比对的原理实际是特征值比对，通过提取图片中的人脸特征值，根据特征值调用 compare_feature 接口进行比对, 推荐比对分值超过 80 分为同一人，可根据实际检测比对情况动态调整。

人脸 1:1 比对流程可如下图，实现如人证比对功能。（人的照片和实时视频比对，可根据使用情况选择是否启用质量检测）



9.13.1 人脸特征值接口

方法名	faceFeature			
说明	人脸特征值检测，回传人脸信息			
函数	FeatureInfo[] faceFeature(long matAddr, int type)			
请求参数	说明	必须	类型	示例描述
matAddr	java 的 opencv 图片帧地址	是	long	请参考示例
type	检测类型（传 0 表示生活照,包含 1 寸证件照，1 表示近红外）	否	int	0 或 1, 未传默认为 0
返回信息	函数的返回	是	FeatureInfo[]	返回人脸特征值信息结构体数组，见后续定义

9.13.2 特征值比对接口

方法名	compareFeature			
说明	人脸跟踪，返回人脸信息			
函数	float compareFeature(Feature f1, Feature f2, int type)			
请求参数	说明	必须	类型	示例描述
f1	人脸特征值结构体	是	Feature	人脸特征值结构体信息 (仅支持单个结构体的比对)
f2	人脸特征值结构体	是	Feature	人脸特征值结构体信息 (仅支持单个结构体的比对)
type	特征值类型 (0,1)	否	int	特征值类型 (传 0 表示生活照,包含 1 寸证件照, 1 表示近红外) 未传默认为 0
返回信息	函数的返回	是	float	特征值比对分值

9.13.3 人脸 1:1 比对接口

方法名	match			
说明	人脸比对，返回人脸比对分值			
函数	int match(long matAddr1, long matAddr2, int type)			
请求参数	说明	必须	类型	示例描述
matAddr1	传入 opencv 图片帧地址	是	long	
matAddr2	传入 opencv 图片帧地址	是	long	
type	比对类型 (0,1)	否	int	0: 生活照 1: 近红外特征值 未传默认为 0(1 寸或 2 寸的证据照也推荐当作生活照模式)
返回信息	函数的返回	是	int	人脸比对分值 (同特征值比对分值结果应该是 float, 这里返回 int 是做了四舍五入)

9.14 动作活体和静默活体

9.14.1 动作活体接口

方法名	faceActionLive			
说明	动作活体（眨眨眼、张张嘴等动作校验活体）			
函数	FaceBox[] faceActionLive(long matAddr, int actionType, int[] actionResult)			
参数	说明	必须	类型	示例描述
matAddr	java 的 opencv 图片帧地址	是	long	请参考示例
actionType	传入的活体动作	是	int	动作活体类型，如 0 表示眨眨眼，1 表示张嘴等
actionResult	传出的动作结构	是	int[]	返回结果为 1 表示存在该动作活体，数组为 1 维数组
返回字段描述				
返回字段	FaceBox[] 人脸框结构体数组，请参考后续结构体定义			
返回示例				

9.14.2 清除动作活体历史接口

方法名	clearActionLiveHistory			
说明	清除动作活体历史			
函数	int clearActionLiveHistory()			
参数	说明	必须	类型	示例描述
返回字段描述				
返回字段	0: 表示成功， 非 0: 其他错误码			
返回示例				

9.14.3 rgb 静默活体接口

rgb 静默活体采用了多因子策略防攻击，效果更佳，但耗时相对来说会稍大点，若希望减少活体耗时，可取消多因子策略。可把 SDK 中 conf 文件夹里面的 no_factor.conf 文件拷贝一份放入 exe 同目录。则会不启用多因子，减少静默活体的检测耗时。通常，静默活体的分值大于 0.8 为推荐阈值，低于 0.8 可认为非活体。

方法名	rgbLiveness			
说明	rgb 可见光单目静默活体（推荐 rgb+nir 或 rgb+depth 进行双目活体校验）			
函数	LivenessInfo[] rgbLiveness(long matAddr)			
参数	说明	必须	类型	示例描述
matAddr	java 的 opencv 图 片帧地址	是	long	请参考示例
返回字段描述				
返回字段	LivenessInfo[], 人脸活体结构体数组			
返回示例				

9.14.4 nir 静默活体接口

方法名	nirLiveness			
说明	nir 近红外单目静默活体（推荐 rgb+nir 或 rgb+depth 进行双目活体校验）			
函数	int nirLiveness(long matAddr)			
参数	说明	必须	类型	示例描述
matAddr	java 的 opencv 图 片帧地址	是	long	请参考示例
返回字段描述				
返回字段	LivenessInfo[], 人脸活体结构体数组			
返回示例				

9.14.5 rgb+depth 双目静默活体接口

方法名	rgbAndDepthLiveness			
说明	rgb+depth 双目静默活体			
函数	RgbDepthInfo[] rgbAndDepthLiveness(long rgbAddr, long depthAddr)			
参数	说明	必须	类型	示例描述

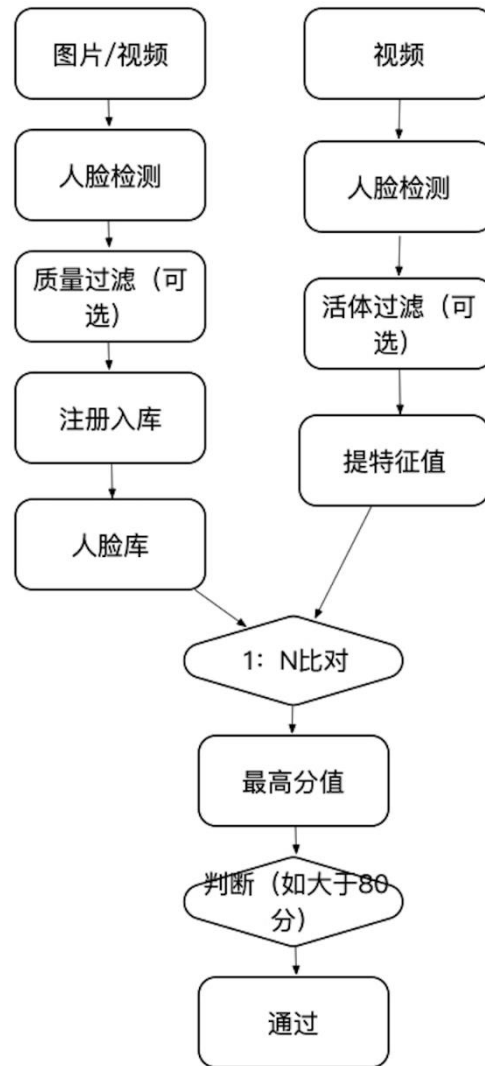
rgbAddr	java 的 opencv 图 片帧地址	是	long	请参考示例
depthAddr	java 的 opencv 图 片帧地址	是	long	请参考示例
返回字段描述				
返回字段	RgbDepthInfo[],人脸 rgb 和深度结构体数组			
返回示例				

9.15 人脸库管理

SDK 提供支持 5 万以下的人脸库管理，采用的是 sqlite 数据库，SDK 启动后会自动生成 db 文件夹和 face.db 文件（人脸库数据文件）、db 文件夹可手动删除，删除后人脸数据库即被整体删除，sdk 重启后会自动重新创建新库。人脸库数据结构可采用 sqliteExpert 等可视化工具查看人脸库表结构。人脸库创建后有三张表，feature 表（用来保存人脸特征值），user 表（用来保存人脸用户信息，如 userid，groupid 以及人脸图片信息，用户信息等）以及 user_group 表（用户组表）。

人脸库可按组(group_id)划分,组就好比一个集团的子公司，人脸注册或查找既可以按整个库查找，也可以按组（子公司）查找，按组查找速度更快（范围小）。用户 id（user_id）是用来标识人脸用户的唯一 id，组 id（group_id）是用来标识组（子公司）的唯一 id。用户信息（user_info）可作为用户 id 的说明，如标识用户名称、住址等信息，也可不填写。人脸的比对或识别、归根结底是人脸特征值的比对。人脸库的保存实际上是保存了对应用户 user_id 的特征值在人脸库上，同时保存了用户 user_id 和 group_id 及其对应关系（一个用户对应一张人脸、一个特征值数据）。除 user_info 字段（用户信息）支持中文外，user_id 和 group_id 仅支持英文、数字和下划线的参数组合。

人脸库 1：N 识别可如如下图所示流程：



9.15.1 人脸注册接口(通过传入图片帧)

方法名	userAddByMat			
说明	用户注册(传入图片帧, 人脸图片缩略图入库, 该接口通常指生活照模式, 1 寸或 2 寸的证据照也可采用该接口)			
函数	String userAddByMat(long matAddr, string userId, string groupId, string userInfo="")			
参数	说明	必须	类型	示例描述
matAddr	人脸图片帧地址	是	long	
userId	用户 id	是	string	用户 id (由数字、字母、下划线组成), 长度限制 128B
groupId	组 id	是	string	用户组 id, 标识一组用户 (由数字、字母、

				下划线组成), 长度限制 128B
userInfo	用户资料	否	string	256 个字符以内(支持中文, 注意一个中文是至少 2 个字符)
返回字段描述(json)				
errno 及 msg 映射	errno	Msg		
	0	Success		
	<0	失败的原因		
返回示例				

9.15.2 人脸注册接口(通过传入特征值)

方法名	userAdd			
说明	用户注册（传入特征值，特征值入库，图片不入库）			
函数	String userAdd(Feature f1, string userId, string groupId, string userInfo="")			
参数	说明	必须	类型	示例描述
f1	特征值 结构体	是	Feature	人脸特征值结构体
userId	用户 id	是	string	用户 id（由数字、字母、下划线组成），长度限制 128B
groupId	组 id	是	string	用户组 id，标识一组用户（由数字、字母、下划线组成），长度限制 128B
userInfo	用户资料	否	string	256 个字符以内(支持中文，注意一个中文是至少 2 个字符)
返回字段描述(json)				
errno 及 msg 映射	errno	Msg		
	0	Success		
	<0	失败的原因		
返回示例				

9.15.3 人脸更新接口

方法名	userUpdate
说明	人脸更新(进行更新后，人脸图片会入库)

函数	String userUpdate(long matAddr, string userId, string groupId, string userInfo=’’)			
参数	说明	必须	类型	示例描述
matAddr	人脸图 片帧地 址	是	long	
userId	用户 id	是	string	用户 id（由数字、字母、下划线组成），长度限制 128B
groupId	组 id	是	string	用户组 id，标识一组用户（由数字、字母、下划线组成），长度限制 128B
userInfo	用户信 息	否	string	256 个字符以内(支持中文，注意一个中文是至少 2 个字符)
返回字段描述(json)				
errno 及 message 映射	errno		Msg	
	0		Success	
	<0		失败的原因	
返回示例				

9.15.4 用户删除接口

方法名	userDelete			
说明	用户删除（用户删除后，人脸特征值、人脸图片也被删除）			
函数	String userDelete(string userId, string groupId)			
参数	说明	必须	类型	示例描述
userId	用户 id	是	string	用户 id（由数字、字母、下划线组成），长度限制 128B
groupId	组 id	是	string	用户组 id，标识一组用户（由数字、字母、下划线组成），长度限制 128B
返回字段描述(json)				
errno 及 msg 映射	errno		Msg	
	0		Success	
	<0		失败的原因	
data 字段	log_id	string		请求日志标识

返回示例	
------	--

9.15.5 创建用户组接口

方法名	groupAdd			
说明	创建用户组			
函数	String groupAdd(string gourpId)			
参数	说明	必须	类型	示例描述
groupId	用户组 id	是	string	用户组 id，标识一组用户（由数字、字母、下划线组成），长度限制 128B
返回字段描述(json)				
errno 及 msg 映射	errno	msg		
	0	Success		
	<0	失败的原因		
data 字段	log_id	string	请求日志标识	
返回示例				

9.15.6 用户组删除接口

方法名	groupDelete			
说明	用户组删除			
函数	String groupDelete(string groupId)			
参数	说明	必须	类型	示例描述
groupId	组 id	是	string	用户组 id，标识一组用户（由数字、字母、下划线组成），长度限制 128B
返回字段描述(json)				
errno 及 msg 映射	Errno	Msg		
	0	Success		
	<0	失败的原因		
data 字段	log_id	string		请求日志标识
返回示例				

9.15.7 用户信息查询接口

方法名	getUserInfo			
说明	用户信息查询接口			
函数	String getUserInfo(string useId, string groupId)			
参数	说明	必须	类型	示例描述
userId	用户 id	是	string	用户 id（由数字、字母、下划线组成），长度限制 128B
groupId	组 id	是	string	用户组 id，标识一组用户（由数字、字母、下划线组成），长度限制 128B
返回字段描述(json)				
errno 及 msg 映射	errno	Msg		
	0	Success		
	<0	失败的原因		
data 字段	log_id	string		请求日志标识
	result	array		识别结果列表
	group_id	string		组 id
	face_token	string		人脸特征的唯一标识
	user_info	string		用户信息
	create_time	string		人脸首次注册时间
返回示例				

9.15.8 用户人脸图片查询接口

方法名	getUserImage			
说明	用户人脸图片查询接口			
函数	int getUserImage(long outAddr, String userId, String groupId)			
参数	说明	必须	类型	示例描述
outAddr	通过引用返回	是	long	

	的图片 帧地址			
userId	用户 id	是	String	用户 id（由数字、字母、下划线组成），长度限制 128B
groupId	组 id	是	String	用户组 id，标识一组用户（由数字、字母、下划线组成），长度限制 128B
返回字段描述: 返回 int, 0 为成功, 其他为错误码				

9.15.9 用户组列表查询接口

方法名	getUserLlist			
说明	用户组列表查询接口			
函数	String getUserList(string groupId, int start = 0, int length = 100)			
参数	说明	必须	类型	示例描述
groupId	用户组 id	是	string	用户组 id
start	查询起始序号	否	int	默认值为 0
length	返回数量	否	int	默认值 100，最大值 1000
返回字段描述(json)				
errno 及 msg 映射	Errno	Msg		
	0	Success		
	<0	失败的原因		
data 字段	log_id	string	请求日志标识	
	user_id_list	array	user_id 列表数组	
返回示例				

9.15.10 群组列表查询接口

方法名	getGroupList			
说明	组列表查询			
函数	String getGroupList(int start = 0, int length = 100)			
参数	说明	必须	类型	示例描述
start	起始序号	否	Int	默认值为 0, 从 0 开始

length	返回数量	否	Int	默认值为 100，最大为 1000
返回字段描述(json)				
errno 及 msg 映射	errno	Msg		
	0	Success		
	<0	失败的原因		
data 字段	log_id	string	请求日志标识	
	group_id_list	array	group_id 列表数组	
返回示例				

9.15.11 人脸库人脸数量查询

方法名	dbFaceCount			
说明	查询数据库人脸数量 (传入组 id 表示查询该组都人脸数量, “” 空字符串表示查整个库的人脸数量)			
函数	int dbFaceCount(String groupId)			
参数	说明	必须	类型	示例描述
groupId	组 id	是	String	人脸分组的组 id(传 “” 空字符串表示查询整个库的数量)
返回示例	int >=0 (返回的数量)			

9.15.12 人脸识别接口

9.15.12.1 人脸识别结果配置

人脸识别结果，通过传入特征值或图片进行人脸的 1:N 识别，即传入的 1 和人脸库的 N 进行识别（人脸库自动创建，需要通过 user_add 等把人脸注册入库），人脸识别返回默认大于 80 分的识别结果。可通过配置文件 face.ini 的字段 IdentifyScore 的分值控制识别返回的结果。若该字段分值设为 101，则默认返回 1 个识别结果（1:N 的得分最高的结果）。

9.15.12.2 人脸识别耗时（参考值）

通常在 1w 左右的人脸库中，进行 1:N 人脸识别，耗时 500 毫秒左右（测试机器为 Thinkpad x270，其中包括单张图片的特征值提取耗时 100ms）。2.5w 左右的人脸库，进行 1:N 人脸识别，耗时 800 毫秒。若人脸库更大，建议可按 group_id 分组注册入库，

识别时候根据传入的 `group_id` 查找（根据不同的组 `group_id` 查找，可避免整个库大范围查找，会节约查找耗时），这样能节约识别耗时。

9.15.12.3 人脸识别接口(1:N) (传入 opencv 图片帧)

方法名	identifyByMat			
说明	人脸识别（1：N），可传入人脸组，比对某个组的所有人脸（人脸库可按单位分组，缩小组范围，减少识别比对时间）			
函数	String identify(long matAddr, string groupIdList, string userId, int type = 0)			
参数	说明	必须	类型	示例描述
matAddr	图片帧地址	是	long	
groupIdList	组列表	是	string	组列表，可单传组名
userId	用户 id	否	string	用户 id
type	特征值类型	否	int	特征值类型（0、是生活照、1 寸的证据照也推荐用生活照模式、1、证件照 2、近红外，参考特征值提取），未传默认为 0
返回字段描述				
返回字段	比对分值	float		
返回示例	返回 json 字符串，同上人脸库			

9.15.12.4 人脸识别接口(1:N)(传入特征值)

方法名	identify			
说明	人脸识别（1：N），可传入人脸组，比对某个组的所有人脸（人脸库可按单位分组，缩小组范围，减少识别比对时间）			
函数	String identify(Feature f1, string groupIdList, string userId, int type = 0)			
参数	说明	必须	类型	示例描述
f1	特征值	是	Feature	特征值结构体
groupIdList	组列表	是	string	组列表，可单传组名
userId	用户 id	否	string	用户 id

type	特征值类型	否	int	特征值类型（0、是生活照、1、证件照 2、近红外，参考特征值提取），未传默认为 0
返回字段描述				
返回字段	比对分值	float		
返回示例	返回 json 字符串，同上人脸库			

9.15.12.5 人脸识别接口(1:N)(传图片帧)

方法名	identifyWithAllByMat			
说明	人脸识别（1：N），和整个人脸库比对			
函数	String identifyWithAll(long matAddr, int type = 0)			
参数	说明	必须	类型	示例描述
matAddr	图片帧地址	是	long	
type	特征值类型	否	int	特征值类型（0、是生活照、1、证件照 2、近红外，参考特征值提取），未传默认为 0
返回字段描述				
返回字段	比对分值	float		
返回示例	返回 json 字符串，同上人脸库			

9.15.12.6 人脸识别接口(1:N)（传特征值）

方法名	identifyWithAll			
说明	人脸识别（1：N），和整个人脸库比对			
函数	String identifyWithAll(Feature fl, int type = 0)			
参数	说明	必须	类型	示例描述
fl	特征值	是	Feature	特征值结构体
type	特征值类型	否	int	特征值类型（0、是生活照、1、证件照 2、近红外，参考特征值提取），未传默认为 0

返回字段描述			
返回字段	比对分值	float	
返回示例	返回 json 字符串, 同上人脸库		

9.16 SDK 系统信息接口

9.16.1 获取 SDK 版本号接口

方法名	sdkVersion
说明	返回 sdk 版本号
函数	String sdkVersion()
返回字段描述: String ,如： 6.3.3	

9.16.2 获取设备指纹接口

方法名	getDeviceId
说明	返回设备指纹
函数	String getDeviceId
返回字段描述: String ,如： 0BAF96961A8377AB74797B05F7F2C805	

10 功能接口对应结构体描述

10.1 人脸跟踪信息结构体

结构体名	TrackFaceInfo		
说明	人脸跟踪信息结构体 <pre>public class TrackFaceInfo { public int faceId; public BDFaceBBox box; public Landmarks landmark; }</pre>		
参数	类型	说明	
faceId	int	人脸 id	

box	FaceBBox	人脸框结构体
landmark	Landmarks	人脸关键点结构体

10.2 人脸框信息结构体

结构体名	FaceBox	
说明	人脸框信息结构体 <pre>public class FaceBox { public int index; public float centerx; public float centery; public float width; public float height; public float score; }</pre>	
参数	类型	说明
index	int	人脸索引值
centerx	float	人脸中心点 x 坐标
centery	float	人脸中心点 y 坐标
width	float	人脸宽度
height	float	人脸高度
score	float	人脸置信度

10.3 人脸关键点信息结构体

结构体名	Landmarks	
说明	人脸关键点信息结构体 <pre>public class Landmarks { public int index; public int size; public float[] data; public float score; }</pre>	

	}	
参数	类型	说明
index	int	人脸关键点索引
size	int	人脸关键点数量（size 的值通常为 144,72 个 x, y 坐标）
data	float[]	人脸关键点坐标数组（144）（72 个 x, y 坐标）
score	float	人脸关键点置信度

10.4 人脸特征值结构体

结构体名	Feature	
说明	人脸特征值结构体 <pre>public class Feature { public int size; public byte[] data; }</pre>	
参数	类型	说明
size	int	特征值大小（通常为 512 个 byte 值数据）
data	byte[]	512 个 byte 数组

10.5 人脸特征值&人脸框&活体结构体

结构体名	LiveFeatureInfo	
说明	人脸特征值&人脸框&活体结构体 <pre>public class LiveFeatureInfo { public Feature feature; public FaceBox box; public float score; }</pre>	
参数	类型	说明
feature	Feature	人脸特征值
box	FaceBox	人脸框

score	float	活体置信度分值
-------	-------	---------

10.6 人脸姿态角结构体

结构体名	HeadPose	
说明	人脸姿态角结构体 <pre>public class HeadPose { public float yaw; public float roll; public float pitch; }</pre>	
参数	类型	说明
yaw	float	左右偏转角
roll	float	与人脸平行平面内的头部旋转角
pitch	float	上下偏转角

10.7 人脸属性信息结构体

结构体名	Attribute	
说明	人脸属性信息结构体 <pre>public class Attribute { public int age; public int race; public int emotion; public int glasses; public int gender; }</pre>	
参数	类型	说明
age	int	年龄
race	int	种族 0,1,2,3 值如下： BDFace_RACE_YELLOW = 0, // 黄种人 BDFace_RACE_WHITE = 1, // 白种人

		BDFace_RACE_BLACK = 2, // 黑种人 BDFace_RACE_INDIAN = 3, // 印第安人
emotion	int	表情 0,1,2 值如下： BDFACE_ATTRIBUTE_EMOTION_FROWN = 0, // 皱眉 BDFACE_ATTRIBUTE_EMOTION_SMILE = 1, // 笑 BDFACE_ATTRIBUTE_EMOTION_CALM = 2, // 平静
glasses	int	戴眼镜状态 0,1,2 值如下： BDFACE_NO_GLASSES = 0, // 无眼镜 BDFACE_GLASSES = 1, // 有眼镜 BDFACE_SUN_GLASSES = 2 // 墨镜
gender	int	性别 0,1 值如下： BDFACE_GENDER_FEMILE = 0, // 女性 BDFACE_GENDER_MALE = 1, // 男性

10.8 人脸遮挡信息结构体

结构体名	Occlusion	
说明	人脸遮挡置信度返回 int 数组 <pre> public class Occlusion { public float leftEye; public float rightEye; public float nose; public float mouth; public float leftCheek; public float rightCheek; public float chin; } </pre>	
参数	类型	说明
leftEye	float	左眼遮挡置信度
rightEye	float	右眼遮挡置信度
nose	float	鼻子遮挡置信度

mouth	float	嘴巴遮挡置信度
leftCheek	float	左脸遮挡置信度
rightCheek	float	右脸遮挡置信度
chin	float	下巴遮挡置信度

10.9 人眼闭合状态结构体

结构体名	EyeClose	
说明	<p>人眼闭合状态结构体</p> <pre>public class EyeClose { public float leftEyeCloseConf; public float rightEyeCloseConf; }</pre>	
参数	类型	说明
leftEyeCloseConf	float	置信度分值
rightEyeCloseConf	float	置信度分值

10.10 注意力结构体

结构体名	GazeInfo
说明	<p>注意力结构体</p> <pre>public class GazeInfo { public int leftEyeDirection; //左眼凝视方向 public int leftEyeConf; // 左眼置信度 public int rightEyeDirection; //右眼凝视方向 public int rightEyeConf; // 右眼置信度 }</pre> <p>凝视方向 int 值描述如下：</p> <pre>BDFACE_GAZE_DIRECTION_UP = 0, // 向上看 BDFACE_GAZE_DIRECTION_DOWN = 1, // 向下看 BDFACE_GAZE_DIRECTION_RIGHT = 2, // 向右看 BDFACE_GAZE_DIRECTION_LEFT = 3, // 向左看</pre>

	BDFACE_GAZE_DIRECTION_FRONT = 4, // 向前看 BDFACE_GAZE_DIRECTION_EYE_CLOSE = 5, // 闭眼
--	---

10.11 动作活体配置结构体

结构体名	ActionLiveConf
说明	活体信度结构体 <pre> public class ActionLiveConf { public float eyeOpenThreshold; //睁眼的阈值 public float eyeCloseThreshold; //闭眼的阈值 public float mouthOpenThreshold; //张嘴的阈值 public float mouthCloseThreshold; //闭嘴的阈值 public float lookUpThreshold; //抬头的阈值 public float lookDownThreshold; // 低头的阈值 public float turnLeftThreshold; // 向左转头的阈值 public float turnRightThreshold; //向右转头的阈值 public float nodThreshold; // 点头的角度阈值 public float shakeThreshold; // 摇头的角度阈值 public int maxCacheNum; // 缓存的帧数 } </pre>

11 多端特征值对齐

对于 Windows SDK 来说，支持和安卓特征值对齐，Windows sdk6.x 系列，特征值对齐安卓 6.x 系列，但和如安卓 7.x 系列或 5.x 系列，则不对齐，且只支持 rgb 可见光特征值对齐。

在 Windows JAVA SDK 中，可通过提取的特征值 512 个 byte，保存成字节流数据，即和安卓中的 512 个 byte 保存成字节流的数据对齐。

12 适配的深度双目摄像头特别说明

SDK 自带了奥比中光以及华捷艾米的双目摄像头，支持 rgb+depth，同时也支持迪

威泰或视派尔的 rgb+nir 双目摄像头，奥比中光或华捷艾米的双目深度 rgb+depth 摄像头，奥比或华捷的摄像头需要安装驱动，在 SDK 目录的 tools->camera_driver 里面，分别双击安装即可（奥比的驱动对应 orbe 目录，华捷艾米的对应 hjimi，注意双击安装后可能需要重启设备生效），安装完毕后，可参考示例 demo 接入奥比或华捷的双目摄像头进行活体检测。因为 SDK 需要用动态库 DLL，所以需要根据您的 x64 或 x86 平台，分别拷贝奥比或华捷的 DLL 里面的所有文件到 SDK 的 FaceOfflineSDK 根目录所在文件夹里面，才能保证镜头模组能够正常运行（若未拷贝，运行 SDK 会提示找不到对应的如奥比或华捷的动态库 DLL 文件）。若运行镜头模组的时候还提示找不到依赖，可选择奥比的镜头时候安装一下 vc_redist.exe（vs2013 的依赖库，可微软官网查找下载）。华捷艾米的则可能需要安装 vc_redist.exe（vs2010 的依赖库，可微软官网查找下载）

13 SDK 多线程使用说明

SDK 支持多实例多线程使用，单实例各接口不支持在多线程中调用。SDK 多实例多线程使用可参考示例代码 MultiThread.java。SDK 最多支持 3 个实例的多线程。每个实例需要执行 sdkInit 和 sdkDestroy 操作。如第二个实例的初始化方法为 sdkInit2，销毁为 sdkDestroy2，第三个带编号 3，其他方法以此类推。如示例所示：若需要 3 个线程，分别执行人脸检测 detect，特征值提取 faceFeature，人脸 rgb 活体 rgbLiveness。则建议需要执行 3 次 SDK 初始化，3 次 sdk 销毁。检测的方法分别用 detect（表示第一个实例的方法），faceFeature2（表示第二个实例的特征值提取），rgbLiveness3（表示第三个实例的 rgb 活体）。

另外：人脸库管理各接口不支持在多线程中使用，人脸库注册可在单线程中注册。如需一边提取特征值一边进行人脸识别，可特征值提取为一个线程，利用特征值识别的 identify（传递特征值的 identify 接口）接口为一个线程。

14 SDK 的版本升级

对 SDK 进行版本升级，分 2 种情况，第一种是大版本号不变，如 8.1 升级到 8.3 或 8.4，第二种是大版本号变化了的升级，如 6.1 升级到 8.1，第一种叫小版本升级，第二种叫大版本升级，以下分别说明。

14.1 SDK 的小版本升级

SDK 的小版本升级，是指如 8.1 升级到 8.4（首位版本号不变），这种级别的版本升级，SDK 的模型，依赖库等都维持不变，提取的人脸特征值也不会发生变化，上个版本的特征值直接可平移到新版本。需要变更升级的文件是：BaiduFaceApi.dll，BaiduFaceApi.lib 以及接口头文件 baidu_face_api.h（大部分情况下接口头文件也不会变化）。

14.2 SDK 的大版本升级

SDK 的大版本升级，是指如 6.1 升级到 8.4（首位版本号发生变化），这种级别的版本升级，SDK 到模型，依赖库 DLL 文件都会发生变化，提取的人脸特征值也会发生变化，上个版本的特征值和新版本的特征值不能通用。需要变更升级到文件：模型文件夹，依赖库 DLL（所有），BaiduFaceApi.dll, BaiduFaceApi.lib 以及接口头文件 baidu_face_api.h 等。

15 错误码及错误信息

各接口返回结果 error_code 及 msg 信息如下：

错误码	错误内容	错误描述
0	SUCCESS	成功
-1	ILLEGAL_PARAMS	失败或非法参数
-2	MEMORY_ALLOCATION_FAILED	内存分配失败
-3	INSTANCE_IS_EMPTY	实例对象为空
-4	MODEL_IS_EMPTY	模型内容为空
-5	UNSUPPORT_ABILITY_TYPE	不支持的能力类型
-6	UNSUPPORT_INFER_TYPE	不支持的预测库类型
-7	NN_CREATE_FAILED	预测库对象创建失败
-8	NN_INIT_FAILED	预测库对象初始化失败
-9	IMAGE_IS_EMPTY	图像数据为空
-10	ABILITY_INIT_FAILED	人脸能力初始化失败
-11	ABILITY_UNLOAD	人脸能力未加载
-12	ABILITY_ALREADY_LOADED	人脸能力已加载

-13	NOT_AUTHORIZED	未授权
-14	ABILITY_RUN_EXCEPTION	人脸能力运行异常
-15	UNSUPPORT_IMAGE_TYPE	不支持的图像类型
-16	IMAGE_TRANSFORM_FAILED	图像转换失败
-1001	SYSTEM_ERROR	系统错误
-1002	PARAM_ERROR	参数错误
-1003	DB_OP_FAILED	数据库操作失败
-1004	NO_DATA	没有数据
-1005	RECORD_UNEXIST	记录不存在
-1006	RECORD_ALREADY_EXIST	记录已经存在
-1007	FILE_NOT_EXIST	文件不存在
-1008	GET_FEATURE_FAIL	提取特征值失败
-1009	FILE_TOO_BIG	文件太大
-1010	FACE_RESOURCE_NOT_EXIST	人脸资源文件不存在
-1011	FEATURE_LEN_ERROR	特征值长度错误
-1012	DETECT_NO_FACE	未检测到人脸
-1013	CAMERA_ERROR	摄像头错误或不存在
-1014	FACE_INSTANCE_ERROR	人脸引擎初始化错误
-1015	LICENSE_FILE_NOT_EXIST	授权文件不存在
-1016	LICENSE_KEY_EMPTY	授权序列号为空
-1017	LICENSE_KEY_INVALID	授权序列号无效
-1018	LICENSE_KEY_EXPIRE	授权序列号过期
-1019	LICENSE_ALREADY_USED	授权序列号已被使用
-1020	DEVICE_ID_EMPTY	设备指纹为空
-1021	NETWORK_TIMEOUT	网络超时
-1022	NETWORK_ERROR	网络错误
-1023	CONF_INI_UNEXIST	配置 ini 文件不存在
-1024	WINDOWS_SERVER_ERROR	禁用在 Windows Server

16 常见问题：

16.1 SDK 推荐使用 Eclipse 或 idea 进行开发、也支持 MyEclipse 等工具，官方推荐采

用 Eclipse 工具。根据 32 位或 64 位，请分别选择安装对应位数的 jdk 和 eclipse。

16.2 模型文件可定制化：在 main 方法入口，可在 sdkInit 方法中传入模型文件夹的绝对路径，达到模型文件定制化的目的。若不定制化路径，sdkInit 中传入空即可，默认模型文件在 SDK 的 models 文件夹里面，无需更改。

16.3 激活后是否可以把激活文件 license.ini 和 license.key 拷贝到其他设备运行？

不能，离线 SDK 和设备绑定，每个设备对应一个 key 和一个 license 文件，换设备无法运行。但对同一台设备，可把 Release 下的 license.ini 和 license.key 拷贝到本电脑的另外 sdk，该设备也等同于激活，可以使用。

16.4 是否支持 debug 模式？只支持 Release 模式，不支持 debug 模式

16.5 SDK 支持 x86 和 x64 模式，推荐使用 x64 模式。支持运行在 win7 或 win10 的 32 位或 64 位系统中。

16.6 人脸库不支持中文参数？目前用户信息支持中文，其他参数暂只支持英文、数字等模式，建议采用英文小写字母或汉语拼音替代。

16.7 多端人脸特征值对齐的问题？windows java sdk6.0 和安卓 sdk6.0 的人脸特征值是对齐的，rgb 生活照的特征值均为 512 个 byte 字节。

16.8 因为 win 端 java 人脸 sdk 是采用 jni 及类反射技术实现，所以 sdk 中的 com.jni.struct 中定义的 java 结构体请勿修改结构，Face.java 中的 api 方法，com.jni.face 也别修改结构，否则可能会引发调用错误。

16.9 在 32 位环境下，eclipse 运行诸如 face_scene 中的示例时候，有的提示 java.lang.outofmemoryError 的问题

这是因为 eclipse 的可用内存有限制导致，可在代码上右键，依次点击“Run As”->“Run Configurations”，在 Arguments 参数中的“VM arguments:”中填入 -Xms64m -Xmx128m 值即可。

16.10 生活照、证件照模式是否可以切换使用？8.0 统一调整为生活照模式，1 寸，2 寸的证据照也可以理解为生活照模式。

14.12 是否支持多线程？sdk 提供了 MultiThread.java 多线程示例，可参考使用，另外请注意 opencv 的 mat 读取图片可能不是线程安全的。

16.11 如何排查内存泄漏？频繁调用 sdk 的接口，若发现内存持续增长，则可能存在

内存泄漏，可打开 windows 自带的资源监视器，勾选运行 sdk 的进程观察。sdk 的所有接口均做了内存泄漏测试，开发者需要注意示例 demo 代码或自己写的代码中是否存在内存泄漏的可能，如 opencv 的 mat 需要 release 等。

16.12 SDK 支持一次初始化，处处运行，在程序生命周期开始时候初始化 SDK，生命周期结束时候再销毁 sdk。切忌毋需多次初始化 sdk，这样不但耗时，也是不正确的使用方式。

16.13 什么情况下会发生设备指纹变更？对设备硬件进行操作，如增删网卡，修改硬件设备，修改网络 mac 地址，更换硬盘等操作会改变硬件的设备指纹，导致设备授权失败，如无必要，请勿做如上操作。

16.14 为什么 SDK 只能检测出 1 个人脸？SDK 默认检测 1 个人脸，若需要检测多个人脸，可通过修改如前述 8.4 的 detect.json 配置文件实现。