# 合肥钛灵信息科技有限公司

版权所有 侵权必究

## 智慧相册 SDK接口文档

（**v2.1.41**）

|  |  |
| --- | --- |
| 系统名称 | Linux ARM |
| 负责人 | 李晨 |
| 文档发布日期 | 2022/06/14 |

## 一、SDK概述

### 1.1、SDK功能简介

1. 人脸检测

2. 人脸对齐

3. 人脸识别（特征提取）

4. 年龄识别、性别识别

5. 生成影集

### 1.2、使用方法

1）鉴权（InitParams）

在调用人脸、分类算法前，先调用鉴权接口。返回值为0表示鉴权成功。

2）人脸识别、年龄识别、性别识别

先做人脸检测和人脸对齐，把人脸对齐的图片作为识别算法的输入图片。

### 1.3、算法支持的视频/图象格式

支持主流的图片格式

BMP DIB JPEG JPG JPE PNG PBM PGM PPM WEBP RAS TIFF TIF EXR HEIC HEIF

### 1.4、算法与模型对照表

|  |  |
| --- | --- |
| 人脸检测 | SCRFD320v1.tim |
| 人脸对齐 | PLandmark106v1.tim |
| 人脸识别 | Glint360KR34v1.tim（大模型）  MobileFacev1.tim（小模型） |
| 年龄识别 | HLFaceAge.tim |
| 性别识别 | HLGender.tim |
| 生成影集 | PhotographAlbum.tim |

## 二、C 接口参数说明

**2.1 人脸检测**

### 2.1.1 GetHandle

函数声明：

struct tiorb\_FaceCheck\_info {

int rotation = 0;

int num\_faces = 0; // 人脸数量

int\* location = nullptr; // 人脸坐标

int\* headlocate=nullptr; // 人头坐标

};

struct Handle\* GetHandle();

函数功能：

获取句柄，用于多线程场景。

### 2.1.2 FaceFastInit

函数声明

int FaceFastInit(Handle\* handle, const char\* ModelPath, int ForwardType = 0);

参数列表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***类别*** | ***参数名称*** | ***是否必选*** | ***类型*** | ***默认值*** | ***描述*** |
| *入参1* | *handle* | *是* | *Handle\** | *无* | *设置句柄* |
| *入参2* | *ModelPath* | *是* | *const char\** | *无* | *设置模型路径(不包括模型名称)* |
| *入参3* | *ForwardType* | *是* | *int* | *无* | *设置0(CPU)或者1(GPU)* |
| *出参* | *返回值* |  | *int* |  | *状态码* |

函数功能

初始化，设置模型路径、CPU或者GPU、模型类型

### 2.2.3 FaceCheckOut

函数声明

int FaceCheckOut(Handle\* handle, const unsigned char\* src, int size, tiorb\_FaceCheck\_info\* face\_info, int rotation=0);

函数参数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***类别*** | ***参数名称*** | ***是否必选*** | ***类型*** | ***默认值*** | ***描述*** |
| *入参1* | *handle* | *是* | *Handle\** | *无* | *设置句柄* |
| *入参2* | *src* | *是* | *const char\** | *无* | *二进制图片* |
| *入参3* | *size* | *是* | *const char\** | *无* | *图片大小* |
| *入参4* | *feats\_info* | *是* | *结构体指针* | *无* | *返回结果* |
| *入参5* | *scale* | *否* | *float* | *有* | *设置人脸比例，默认值1.5f* |
| *入参6* | *outsize* | *否* | *int* | *有* | *返回的人脸尺寸，默认值112* |
| *出参* | *返回值* |  | *int* |  | *状态码* |

函数功能

检测图片中的人脸，返回坐标框。

### 2.2.4 FaceDestructStruct

函数声明

int FaceDestructStruct(tiorb\_FaceCheck\_info\* face\_info);

函数参数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***类别*** | ***参数名称*** | ***是否必选*** | ***类型*** | ***默认值*** | ***描述*** |
| *入参1* | face\_info | *是* | *结构体指针* | *无* | *释放结构体指针* |
| *出参* | *返回值* |  | *int* |  | *状态码* |

函数功能

释放结构体所占内存

### 2.2.5 FaceDestructModel

函数声明

int FaceDestructModel(Handle\* handle);

函数参数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***类别*** | ***参数名称*** | ***是否必选*** | ***类型*** | ***默认值*** | ***描述*** |
| *入参1* | handle | *是* | *指针* | *无* | *释放Handle指针* |
| *出参* | *返回值* |  | *int* |  | *状态码* |

函数功能

释放模型所占内存

**2.2 人脸对齐**

### 2.2.1 GetAlignHandle

struct tiorb\_face\_landermark\_info {

int angle = 0; // 正值表示顺时针旋转, 负值表示逆时针旋转

unsigned char\* img = nullptr; // 对齐后的人脸图片

int img\_size = 0; unsigned char\* dst = nullptr; // 人头图片

int dst\_size = 0;

};

struct Handle\* GetAlignHandle();

### 2.2.2 LandmarkInit

函数声明

int LandmarkInit(Handle\* handle, const char\* ModelPath, int ForwardType);

参数列表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***类别*** | ***参数名称*** | ***是否必选*** | ***类型*** | ***默认值*** | ***描述*** |
| *入参1* | *handle* | *是* | *Handle\** | *无* | *设置句柄* |
| *入参2* | *ModelPath* | *是* | *const char\** | *无* | *设置模型路径(不包括模型名称)* |
| *入参3* | *ForwardType* | *是* | *int* | *无* | *设置0(CPU)或者1(GPU)* |
| *出参* | *返回值* |  | *int* |  | *状态码* |

函数功能

初始化，设置模型路径、CPU或者GPU、模型类型

### 2.2.3 ExtractKeypoints

函数声明

int ExtractKeypoints(Handle\* handle, const unsigned char\* src, int size, int\* face\_info, tiorb\_face\_landermark\_info\* landermark\_info, float scale=1.5f, int rotation=0);

函数参数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***类别*** | ***参数名称*** | ***是否必选*** | ***类型*** | ***默认值*** | ***描述*** |
| *入参1* | *handle* | *是* | *Handle\** | *无* | *设置句柄* |
| *入参2* | *src* | *是* | *const char\** | *无* | *二进制图片* |
| *入参3* | *size* | *是* | *int* | *无* | *图片大小* |
| *入参4* | face\_info | *是* | int\* | *无* | *人脸坐标* |
| *入参5* | landermark\_info | *是* | 结构体指针 | *无* | 返回结果 |
| *入参56* | scale | *否* | float | *有* | 设置人脸比例，默认1.5 |
| *入参5* | rotation | *否* | *int* | *有* | *是否旋转人脸，默认0* |

函数功能

检测图片中的人脸，返回坐标框。

### 2.2.4 LandmarkDestroyStruct

函数声明

int LandmarkDestroyStruct(tiorb\_face\_landermark\_info\* landermark\_info);

函数参数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***类别*** | ***参数名称*** | ***是否必选*** | ***类型*** | ***默认值*** | ***描述*** |
| *入参1* | landermark\_info | *是* | *结构体指针* | *无* | *释放结构体指针* |
| *出参* | *返回值* |  | *int* |  | *状态码* |

函数功能

释放结构体所占内存

### 2.2.5 LandmarkDestroyModel

函数声明

int LandmarkDestroyModel(Handle\* handle);

函数参数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***类别*** | ***参数名称*** | ***是否必选*** | ***类型*** | ***默认值*** | ***描述*** |
| *入参1* | handle | *是* | *指针* | *无* | *释放Handle指针* |
| *出参* | *返回值* |  | *int* |  | *状态码* |

函数功能

释放模型所占内存

**2.3 人脸识别**

### 2.3.1 GetFeatureHandle

函数声明：

struct tiorb\_face\_feature\_info {

float face\_quality = 0.f;

float\* FaceFeature = nullptr;

};

struct Handle\* GetFeatureHandle();

函数功能：

获取句柄，用于多线程场景。

### 2.3.2 FeatureInit

函数声明

int FeatureInit(Handle\* handle, const char\* ModelPath, int ForwardType, int ModelType);

参数列表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***类别*** | ***参数名称*** | ***是否必选*** | ***类型*** | ***默认值*** | ***描述*** |
| *入参1* | *handle* | *是* | *Handle\** | *无* | *设置句柄* |
| *入参2* | *ModelPath* | *是* | *const char\** | *无* | *设置模型路径(不包括模型名称)* |
| *入参3* | *ForwardType* | *是* | *int* | *无* | *设置0(CPU)或者1(GPU)* |
| 入参4 | ModelType | 是 | int | 无 | 设置0(CPU)或者1(GPU) |
| 出参 | 返回值 |  | int |  | 状态码 |

函数功能

初始化，设置模型路径、CPU或者GPU、模型类型

### 2.3.3 ExtractFeature

函数声明

int ExtractFeature(Handle\* handle, const unsigned char\* src, int size, tiorb\_face\_feature\_info\* feature\_info);

函数参数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***类别*** | ***参数名称*** | ***是否必选*** | ***类型*** | ***默认值*** | ***描述*** |
| *入参1* | *handle* | *是* | *Handle\** | *无* | *设置句柄* |
| *入参2* | *src* | *是* | *const char\** | *无* | *二进制图片* |
| *入参3* | *size* | *是* | *const char\** | *无* | *图片大小* |
| *入参4* | *feats\_info* | *是* | *结构体指针* | *无* | *返回结果* |
| *入参5* | *scale* | *否* | *float* | *有* | *设置人脸比例，默认值1.5f* |
| *入参6* | *outsize* | *否* | *int* | *有* | *返回的人脸尺寸，默认值112* |
| *出参* | *返回值* |  | *int* |  | *状态码* |

函数功能

检测图片中的人脸，返回坐标框。

### 2.3.4 FeatureDestroyStruct

函数声明

int FeatureDestroyStruct(tiorb\_face\_feature\_info\* feature\_info);

函数参数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***类别*** | ***参数名称*** | ***是否必选*** | ***类型*** | ***默认值*** | ***描述*** |
| *入参1* | *feature\_info* | *是* | *结构体指针* | *无* | *释放结构体指针* |
| *出参* | *返回值* |  | *int* |  | *状态码* |

函数功能

释放结构体所占内存

### 2.3.5 FeatureDestroyModel

函数声明

int FeatureDestroyModel(Handle\* handle);

函数参数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***类别*** | ***参数名称*** | ***是否必选*** | ***类型*** | ***默认值*** | ***描述*** |
| *入参1* | handle | *是* | *指针* | *无* | *释放Handle指针* |
| *出参* | *返回值* |  | *int* |  | *状态码* |

函数功能

释放模型所占内存

**2.4 性别、年龄识别**

### 2.4.1 GetEmotionHandle

函数声明：

struct tiorb\_emotion\_detect\_info {

int num\_faces = 0; // 人脸数量

int\* label = nullptr; // gender age

int\* location = nullptr;// 人脸坐标

};

struct Handle\* GetEmotionHandle();

函数功能：

获取句柄，用于多线程场景。

### 2.4.2 EmotionInit

函数声明

int EmotionInit(Handle\* handle, const char\* ModelPath, int ForwardType, int detect);

参数列表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***类别*** | ***参数名称*** | ***是否必选*** | ***类型*** | ***默认值*** | ***描述*** |
| *入参1* | *handle* | *是* | *Handle\** | *无* | *设置句柄* |
| *入参2* | *ModelPath* | *是* | *const char\** | *无* | *设置模型路径(不包括模型名称)* |
| *入参3* | *ForwardType* | *是* | *int* | *无* | *设置0(CPU)或者1(GPU)* |
| 入参4 | detect | 是 | int | 无 | 设置0(不检测人脸)或者1(检测人脸) |
| *出参* | *返回值* |  | *int* |  | *状态码* |

函数功能

初始化，设置模型路径、CPU或者GPU、模型类型

### 2.4.3 EmotionDetect

函数声明

int EmotionDetect(Handle\* handle, const unsigned char\* src, int size, tiorb\_emotion\_detect\_info\* emotion\_info);

函数参数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***类别*** | ***参数名称*** | ***是否必选*** | ***类型*** | ***默认值*** | ***描述*** |
| *入参1* | *handle* | *是* | *Handle\** | *无* | *设置句柄* |
| *入参2* | *src* | *是* | *const char\** | *无* | *二进制图片* |
| *入参3* | *size* | *是* | *const char\** | *无* | *图片大小* |
| *入参4* | emotion\_info | *是* | *结构体指针* | *无* | *返回结果* |
| *出参* | *返回值* |  | *int* |  | *状态码* |

函数功能

检测图片中的人脸，返回坐标框。

### 2.4.4 EmotionDestroyStruct

函数声明

int EmotionDestroyStruct(tiorb\_emotion\_detect\_info\* emotion\_info);

函数参数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***类别*** | ***参数名称*** | ***是否必选*** | ***类型*** | ***默认值*** | ***描述*** |
| 入参1 | *emotion\_info* | *是* | *结构体指针* | *无* | *释放结构体指针* |
| *出参* | *返回值* |  | *int* |  | *状态码* |

函数功能

释放结构体所占内存

### 2.4.5 EmotionDestroyModel

函数声明

int EmotionDestroyModel(Handle\* handle);

函数参数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***类别*** | ***参数名称*** | ***是否必选*** | ***类型*** | ***默认值*** | ***描述*** |
| *入参1* | handle | *是* | *指针* | *无* | *释放Handle指针* |
| *出参* | *返回值* |  | *int* |  | *状态码* |

函数功能

释放模型所占内存

**2.5 生成影集**

### 2.5.1 GetAlbumHandle

函数声明：

struct tiorb\_scene\_detect\_info {

int num\_scenes = 0;

int \*index = nullptr;

float \*accuracy = nullptr;

};

struct Handle \*GetAlbumHandle();

函数功能：

获取句柄，用于多线程场景。

### 2.5.2 AlbumInit

函数声明

int AlbumInit(Handle \*handle, const char \*model\_path, int forward\_type);

参数列表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***类别*** | ***参数名称*** | ***是否必选*** | ***类型*** | ***默认值*** | ***描述*** |
| *入参1* | *handle* | *是* | *Handle\** | *无* | *设置句柄* |
| *入参2* | *ModelPath* | *是* | *const char\** | *无* | *设置模型路径(不包括模型名称)* |
| *入参3* | *ForwardType* | *是* | *int* | *无* | *设置0(CPU)或者1(GPU)* |
| *出参* | *返回值* |  | *int* |  | *状态码* |

函数功能

初始化，设置模型路径、CPU或者GPU

### 2.5.3 AlbumDetect

函数声明

int AlbumDetect(Handle \*handle, const unsigned char \*src, int size, tiorb\_scene\_detect\_info \*scene\_info);

函数参数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***类别*** | ***参数名称*** | ***是否必选*** | ***类型*** | ***默认值*** | ***描述*** |
| *入参1* | *handle* | *是* | *Handle\** | *无* | *设置句柄* |
| *入参2* | *src* | *是* | *const char\** | *无* | *二进制图片* |
| *入参3* | *size* | *是* | *const char\** | *无* | *图片大小* |
| *入参4* | scene\_info | *是* | *结构体指针* | *无* | *返回结果* |
| *出参* | *返回值* |  | *int* |  | *状态码* |

函数功能

检测图片中的人脸，返回坐标框。

### 2.5.4 AlbumDestroyStruct

函数声明

int AlbumDestroyStruct(tiorb\_scene\_detect\_info \*scene\_info);

函数参数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***类别*** | ***参数名称*** | ***是否必选*** | ***类型*** | ***默认值*** | ***描述*** |
| 入参1 | scene\_info | *是* | *结构体指针* | *无* | *释放结构体指针* |
| *出参* | *返回值* |  | *int* |  | *状态码* |

函数功能

释放结构体所占内存

### 2.5.5 AlbumDestroyModel

函数声明

int AlbumDestroyModel(Handle \*handle);

函数参数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***类别*** | ***参数名称*** | ***是否必选*** | ***类型*** | ***默认值*** | ***描述*** |
| *入参1* | handle | *是* | *指针* | *无* | *释放Handle指针* |
| *出参* | *返回值* |  | *int* |  | *状态码* |

函数功能

释放模型所占内存

## 三、技术支持与联系方式

地    址：合肥钛灵信息科技有限公司

联系邮箱：alineye@gravitylink.com