合肥钛灵信息科技有限公司

智慧相册 SDK接口文档

（v2.1.41）

系统名称	Linux ARM
负责人	李晨/金晨曦
文档发布日期	2022/06/16

一、SDK概述

1.1、SDK功能简介

a. 魔法背景
b. 魔法天空
c. 风格迁移
d. 卡通迁移

1.2、编程语言

Linux ARM C/C++

1.3、运行环境

Linux、Windows、ARM 、Android、iOS

1.4、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

1.5、状态码

- **常见状态码**
  - 10008 鉴权失败　　10101 服务端连接失败

- 10104 鉴权时间过期　　10106 lisence文件不存在
- 20001 模型错误　　20002 输入图片为空
- 20003 模型推理错误　　20004 对象指针为空
- 20005 输入通道数错误　　20006 模型文件不存在
- 20008 人脸区域太小　　20010 图片格式不支持
- 20013 图片长宽比例不支持　20014 图片过大

- **魔法天空状态码**
  - 20007 天空替换天空区域太小

1.6、算法与模型对照表

魔法背景(2选1)	MODSegment：MODNetv1.tim RVMSegment： rvmsegmentv1.tim
魔法天空	skysegmentv1.tim
风格迁移	stylecandy.tim stylemosaic.tim stylepointilism.tim stylerainprincess.tim styleudnie.tim
卡通迁移	人脸卡通画 SCRFD320.tim　paint512v2.tim 整图卡通画 cartoonization.tim

二、C 接口参数说明

2.1 魔法背景

2.1.1 MODSegmentHandle、RVMSegmentHandle

- 函数声明

```
struct tiorb_PersonSegment_info {
    unsigned char* img = nullptr;
    int img_size = 0;
};
struct Handle* MODSegmentHandle();
struct tiorb_PersonSegment_info {
    unsigned char* img = nullptr;
    int img_size = 0;
};
struct Handle* RVMSegmentHandle();
```

- 函数功能

获取句柄，用于多线程场景。

2.1.2 ISegmentInit、 VSegmentInit

- 函数声明

```
int ISegmentInit(Handle* handle, const char* ModelPath, int ForwardType=0);
int VSegmentInit(Handle* handle, const char* ModelPath, int ForwardType);
```

- 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	*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

2.1.3 ISegmentInfer、VSegmentInfer

- 函数声明

```
ISegmentInfer(Handle* handle, const unsigned char* src, int size, const unsigned char* bgimg, int length, tiorb_PersonSegment_info* segment_info);
VSegmentInfer(Handle* handle, const unsigned char* src, int size, const unsigned char* bgimg, int length, tiorb_PersonSegment_info* segment_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 unsigned char*	*无*	*二进制图片*
入参3	*size*	*是*	*int*	*无*	*图片大小*
入参4	bgimg	*是*	const unsigned char*		*二进制图片*
入参5	length	*是*	int		*图片大小*
入参4	segment_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

  替换图片中的天空

2.1.4 ISegmentDestroyStruct、VSegmentDestroyStruct

- 函数声明

```
int ISegmentDestroyStruct(tiorb_PersonSegment_info* segment_info);
int VSegmentDestroyStruct(tiorb_PersonSegment_info* segment_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	segment_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
  - 释放结构体所占的内存
  - 每次推理后都需要调用

2.1.5 ISegmentDestroyModel、VSegmentDestroyModel

- 函数声明

```
int ISegmentDestroyModel(Handle* handle);
int VSegmentDestroyModel(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*	*Handle **	*无*	*释放Handle 指针*
出参	*返回值*		*int*		*状态码*

- 函数功能
  - 释放模型所占的内存
  - 释放后无法继续推理

2.1 魔法天空

2.2.1 MODSegmentHandle

- 函数声明

```
struct tiorb_SkySegment_info {
```

```
    unsigned char* img = nullptr;
    int img_size = 0;
};
struct Handle* SKYSegmentHandle();
```

- 函数功能

  获取句柄，用于多线程场景。

2.2.2 SkySegmentInit

- 函数声明

```
int SkySegmentInit(Handle* handle, const char* ModelPath, int ForwardType);
```

- 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	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

2.2.3 SkySegmentInfer

- 函数声明

```
int SkySegmentInfer(Handle* handle, const unsigned char* src, int size, const unsigned char* bgimg, int len
tiorb_SkySegment_info* sky_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 unsigned char*	无	二进制图片
入参3	size	是	int	无	图片大小
入参4	bgimg	是	const unsigned char*		二进制图片
入参5	length	是	int		图片大小

入参4	sky_info	*是*	*结构体指针*	*无*	*返回结果*
出参	*返回值*		*int*		*状态码*

2.2.4 SkySegmentDestroyStruct

- 函数声明

```
int SkySegmentDestroyStruct(tiorb_SkySegment_info* sky_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	sky_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
  - 释放结构体所占的内存
  - 每次推理后都需要调用

2.2.5 SkySegmentDestroyModel

- 函数声明

```
int SkySegmentDestroyModel(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*	*Handle **	*无*	*释放Handle 指针*
出参	*返回值*		*int*		*状态码*

- 函数功能
  - 释放模型所占的内存
  - 释放后无法继续推理

2.3 风格迁移

2.3.1 MODSegmentHandle

- 函数声明

```
struct tiorb_ImageTransfer_info {
    unsigned char* img = nullptr;
```

```
    int img_size = 0;
};
struct Handle* GetTransferHandle();
```

- 函数功能

  获取句柄，用于多线程场景。

2.3.2 TransferInit

- 函数声明

```
int TransferInit(Handle* handle, const char* ModelPath, int style, int ForwardType=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	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

2.3.3 TransferInfer

- 函数声明

```
int TransferInfer(Handle* handle, const unsigned char* src, int size, tiorb_ImageTransfer_info* transfer_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 unsigned char*	无	二进制图片
入参3	size	是	int	无	图片大小
入参4	transfer_info	是	结构体指针	无	返回结果
出参	返回值		int		状态码

2.3.4 TransferDestroyStruct

- 函数声明

```
int TransferDestroyStruct(tiorb_ImageTransfer_info* transfer_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	transfer_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
  - 释放结构体所占的内存
  - 每次推理后都需要调用

2.3.5 TransferDestroyModel

- 函数声明

```
int TransferDestroyModel(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	Handle *	无	释放Handle 指针
出参	返回值		int		状态码

- 函数功能
  - 释放模型所占的内存
  - 释放后无法继续推理

2.4 卡通迁移

2.4.1 MODSegmentHandle、GetANFaceHandle

- 函数声明

```
struct tiorb_ImageCartoon_info {
    unsigned char* img = nullptr;
    int img_size = 0;
};
// 整张图片漫画风
struct Handle* GetCartoonHandle();
// 人脸漫画风
struct Handle* GetANFaceHandle();
```

- 函数功能

  获取句柄，用于多线程场景。

2.4.2 CartoonInit、ANFaceInit

- 函数声明

```
// 整张图片漫画风
int CartoonInit(Handle* handle, const char* ModelPath, int ForwardType=0);
// 人脸漫画风
int ANFaceInit(Handle* handle, const char* ModelPath, int ForwardType=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	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

2.4.3 CartoonInfer、ANFaceInfer

- 函数声明

```
// 整张图片漫画风
int CartoonInfer(Handle* handle, const unsigned char* src, int size, tiorb_ImageCartoon_info* cartoon_info
// 人脸漫画风
int ANFaceInfer(Handle* handle, const unsigned char* src, int size, tiorb_ImageCartoon_info* aface_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 unsigned char*	无	二进制图片
入参3	size	是	int	无	图片大小
入参4	cartoon_info	是	结构体指针	无	返回结果

	aface_info				
出参	*返回值*		*int*		*状态码*

2.4.4 CartoonDestroyStruct、ANFaceDestroyStruct

- 函数声明

```
// 整张图片漫画风
int CartoonDestroyStruct(tiorb_ImageCartoon_info* cartoon_info);
// 人脸漫画风
int ANFaceDestroyStruct(tiorb_ImageCartoon_info* aface_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	cartoon_info aface_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
  - 释放结构体所占的内存
  - 每次推理后都需要调用

2.4.5 CartoonDestroyModel、ANFaceDestroyModel

- 函数声明

```
// 整张图片漫画风
int CartoonDestroyModel(Handle* handle);
// 人脸漫画风
int ANFaceDestroyModel(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*	*Handle **	*无*	*释放Handle 指针*
出参	*返回值*		*int*		*状态码*

- 函数功能
  - 释放模型所占的内存
  - 释放后无法继续推理

2.5 目标检测与主体检测

2.5.1 ObjectInit

- 函数声明

```
struct tiorb_ObjectDetect_info {
    int num_objects    = 0;
    float* score  = nullptr;
    int* label    = nullptr;
    int* location = nullptr;
    int* MajorObject=nullptr;
};
struct Handle* OBDetectHandle();
```

- 函数功能

  获取句柄，用于多线程场景。

2.5.2 ObjectInit

- 函数声明

```
int ObjectInit(Handle* handle, const char* ModelPath, int ForwardType);
```

- 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	*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 ObjectDetect

- 函数声明

```
int ObjectDetect(Handle* handle, const unsigned char* src, int size, tiorb_ObjectDetect_info* object_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 unsigned char*	*无*	*二进制图片*
入参3	*size*	*是*	*int*	*无*	*图片大小*
入参4	object_info	*是*	*结构体指针*	*无*	*返回结果*
出参	*返回值*		*int*		*状态码*

2.5.4 ObjectDestroyStruct

- 函数声明

```
int ObjectDestroyStruct(tiorb_ObjectDetect_info* object_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	object_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
  - 释放结构体所占的内存
  - 每次推理后都需要调用

2.5.5 ObjectDestroyModel

- 函数声明

```
int ObjectDestroyModel(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*	*Handle **	*无*	*释放Handle 指针*
出参	*返回值*		*int*		*状态码*

- 函数功能
  - 释放模型所占的内存
  - 释放后无法继续推理

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