

OpenNCC SDK

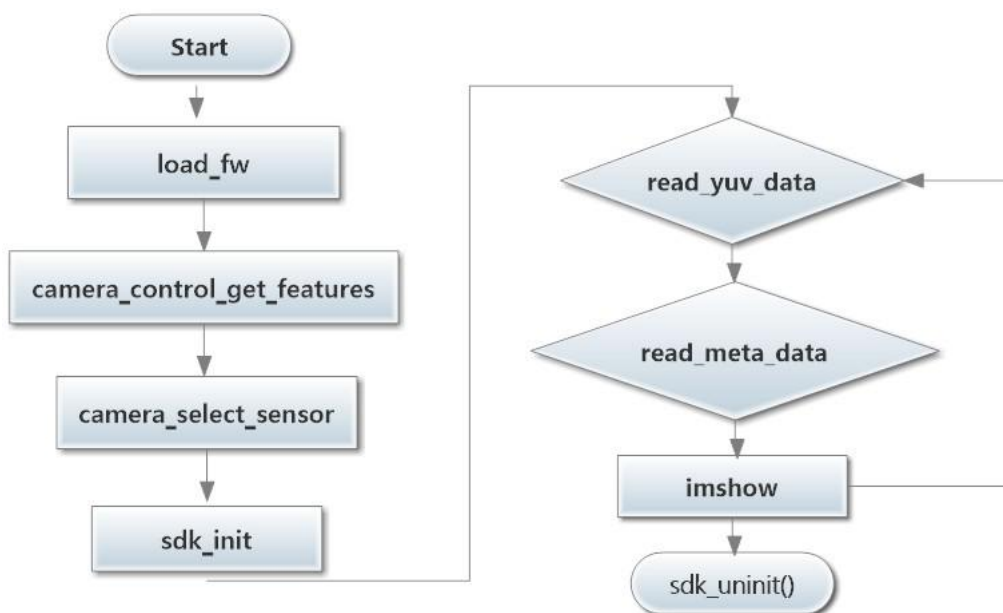
API2.0.x 接口定义

历史版本

版本	日期	修改	变更摘要
1.0.0	2020/1/10	王新华	Initial version
1.0.1	2020/3/16	王洋	Optimized version
2.0.0	2020/4/7	左文平	修订接口, 添加 python 接口

一： SDK C/C++接口说明

接口包含文件主要在 **sdk.h**、**cameraCtrl.h** 、**Fp16Convert 3** 个文件。



OpenNCC sdk 视频处理流程图

1.设备初始化相关接口

1.1 加载设备固件

接口名称	接口参数	参数说明
	const char* bootExe	Usb boot 程序路径
load_fw()	const char* firmware	固件文件放置路径

接口调用示例:

```
load_fw("./moviUsbBoot", "./fw/flicRefApp.mvcmd");
```

返回值: 0: 成功; -1 失败

接口功能说明:

自动加载设备固件, 设备 boot 运行, host (PC) 打开 usb 设备。

1.2 获取连接设备 usb 版本信息

接口名称	接口参数	参数说明
get_usb_version()	void	无

接口调用示例:

```
get_usb_version();
```

返回值：30：usb3.0、20：usb2.0

接口功能说明：

获取设备连接的 usb 版本信息（端口和 usb 线）

1.3 初始化相机参数

接口名称	接口参数	参数说明
sdk_init()	vscRecvCb cb	回调函数
	void* param	回调函数参数
	char *blob_path	AI 模型文件(blob 格式)路径
	CameraInfo* cam	相机配置参数，具体内容见下方
	int cam_Len	相机配置结构体长度

媒体数据和 meta 数据有 2 种方式获取，一：通过回调函数被动获取，二：通过 read_XXX_data() 主动获取，使用第二种方法不用设置回调函数以及回调参数。

```
typedef struct{
    int imageWidth;           //图像宽度
    int imageHeight;         //图像高度
    int startX;               //Ai 运算起点 x 坐标
    int startY;               //Ai 运算起点 y 坐标
    int endX;                 //Ai 运算终点 x 坐标
    int endY;                 //Ai 运算终点 y 坐标
    int inputDimWidth;        /* 缩放后模型输入宽，如果<=0,自动从模型的 xml 获取*/
    int inputDimHeight;       /* 缩放后模型输入高，如果<=0,自动从模型的 xml 获取 */
    IMAGE_FORMAT inputFormat; /* 模型输入格式，只支持
    RGB/RGB_PLANAR/BGR/BGR_PLANAR */
    float meanValue[3];       /* 缩放后的数据二次预处理如果 inputFormat 为 RGB:
                                R = (R-meanValue[0])/stdValue
                                G = (G-meanValue[0])/stdValue
                                B = (B-meanValue[0])/stdValue */
    float stdValue;
    int isOutputYUV;          //使能开关 1 : open 0 :close
    int isOutputH26X;
    int isOutputJPEG;
    encodeMode mode;          /* H264/H265 */
} CameraInfo;
```

接口调用示例：

```
sdk_init(NULL, NULL, (char*) "./blob/face-detection.blob", &cam_info, sizeof(cam_info));
```

返回值：0：成功；-1 失败

接口功能说明：

指定相机 AI 模型文件，AI 计算参数，初始化设备算法模型，相机功能开关选择，通过 mode 参数设置视频

压缩编码参数 (**ENCODE_H264_MODE**, **ENCODE_H265_MODE**) , 注意, 此处仅仅是功能开关是否开启, 视频输出还要通过 **camera_video_out** () 控制是否输出。

1.4 获取 meta data 大小

接口名称	接口参数	参数说明
get_meta_size()	void	无

接口调用示例: 略。

返回值: cnn 计算结果 meta data 数据大小。

接口功能说明:

相机关闭, 重新加载算法模型, 更换模型前调用

1.5 移除 sdk

接口名称	接口参数	参数说明
sdk_uninit()	void	无

接口调用示例:

```
sdk_uninit();
```

返回值: 无

接口功能说明:

相机关闭, 重新加载算法模型, 更换模型前调用

1.6 获取 sdk 版本信息

接口名称	接口参数	参数说明
get_sdk_version()	char* version	版本信息

接口调用示例:

```
char version[100];
```

```
get_sdk_version(version);
```

返回值: void

接口功能说明:

获取 sdk 版本信息。

2.视频流相关接口

2.1 获取设备 yuv 数据

接口名称	接口参数	参数说明
read_yuv_data()	char* pbuf	接收缓存区
	int * size	输入输出参数，输入时表示输入缓存区大小，输出时表示返回视频数据大小
	int blocked	数据返回 0：若无数据立即返回；1：阻塞直到读取到数据才返回

接口调用示例：

```
read_yuv_data(data_yuv,&size,1)
```

返回值：0：成功；-1 失败

接口功能说明：

获取设备 yuv 数据流

2.2 获取设备 H.264 或 H.265 数据流

接口名称	接口参数	参数说明
read_26x_data()	char* pbuf	接收缓存区
	int * size	输入输出参数，输入时表示输入缓存区大小，输出时表示返回视频数据大小
	int blocked	数据返回 0：若无数据立即返回；1：阻塞直到读取到数据才返回

接口调用示例：

```
read_26x_data(data_26x,&size,1)
```

返回值：0：成功；-1 失败

接口功能说明：

获取设备 H.264 或 H.265 数据流

2.3 获取设备 jpeg 数据

接口名称	接口参数	参数说明
read_jpg_data()	char* pbuf	接收缓存区
	int * size	输入输出参数，输入时表示输入缓存区大小，输出时表示返回视频数据大小
	int blocked	数据返回 0：若无数据立即返回；1：阻塞直到读取到数据才返回

接口调用示例：

read_jpg_data(yuv420p,&size,1)

返回值：0：成功；-1 失败

接口功能说明：

获取设备 jpeg 数据流

2.4 获取设备 AI 算法运算结果

接口名称	接口参数	参数说明
read_meta_data()	char* pbuf	接收缓存区
	int * size	输入输出参数，输入时表示输入缓存区大小，输出时表示返回视频数据大小
	int blocked	数据返回 0：若无数据立即返回；1：阻塞直到读取到数据才返回

接口调用示例：

read_meta_data(data_mate,&size,1)

返回值：0：成功；-1 失败

接口功能说明：

获取设备 AI 网络运算结果数据

2.5 获取设备 ir 数据

接口名称	接口参数	参数说明
read_ir_data()	char* pbuf	接收缓存区
	int * size	输入输出参数，输入时表示输入缓存区大小，输出时表示返回视频数据大小
	int blocked	数据返回 0：若无数据立即返回；1：阻塞直到读取到数据才返回

接口调用示例：

read_ir_data(yuv420p,&size,1)

返回值：0：成功；-1 失败

接口功能说明：

获取设备红外图像数据流，只有 3D 相机才支持该接口。

2.6 获取设备 depth 数据

接口名称	接口参数	参数说明
read_depth_data()	char* pbuf	接收缓存区
	int * size	输入输出参数，输入时表示输入缓存区大小，输出时表示返回视频数据大小

	int blocked	数据返回 0：若无数据立即返回；1：阻塞直到读取到数据才返回
--	-------------	--------------------------------

接口调用示例：

```
read_depth_data(data_depth,&size,1)
```

返回值：0：成功；-1 失败

接口功能说明：

获取设备深度图像数据流，只有 3D 相机才支持该接口。

3. 相机控制相关接口

3.1 获取相机模组信息

接口名称	接口参数	参数说明
camera_control_get_features()	SensorModesConfig *	设备信息的结构体指针

接口调用示例：

```
SensorModesConfig cameraCfg;
```

```
camera_control_get_features(&cameraCfg);
```

返回值：0：成功；-1 失败

cameraCfg.moduleName 相机模组名称

cameraCfg.camWidth 图像宽

cameraCfg.camHeight 图像高

cameraCfg.camFps 相机帧率

cameraCfg.AFmode 是否支持自动聚焦 1 支持，0 不支持

cameraCfg.maxEXP 最大曝光时间，单位微秒 us

cameraCfg.minGain 最小增益倍数

cameraCfg.maxGain 最大增益倍数

接口功能说明：

获取相机可见光模组模式信息，有的相机支持多种视频模式，可以通过 camera_select_sensor () 选择使用。

3.2 选择模组工作模式

接口名称	接口参数	参数说明
camera_select_sensor()	int sensorid	camera_control_get_features 获取到相机支持的模组信息数组，sensorid 为数组的序号。

接口调用示例：

```
camera_select_sensor(0);
```

返回值：0：成功；-1 失败

接口功能说明：

设置相机可见光模组的工作模式。

3.3 控制相机视频输出方式

接口名称	接口参数	参数说明
camera_video_out()	int video_type	Yuv 数据输出模式
	camera_ctrl_VIDEO_out mode	禁止, 单次 (拍照用), 连续

```
typedef enum
{
    VIDEO_OUT_DISABLE,      /* 禁止输出 */
    VIDEO_OUT_SINGLE,      /* 输出一次 */
    VIDEO_OUT_CONTINUOUS,   /* 连续输出 */
}camera_ctrl_video_out;
```

接口调用示例：

```
camera_video_out(YUV420p,VIDEO_OUT_CONTINUOUS);
```

返回值：0：成功；-1 失败

接口功能说明：

控制设备输出视频数据的模式，该设置当前对 YUV420p, H26X, JPEG 有效，其中好 H26X 不支持单次输出。

3.4 选择相机聚焦模式

接口名称	接口参数	参数说明
camera_control_af_mode()	camera_ctrl_af_mode af_mode	CAMERA_CONTROL_AF_MODE_OFF : 手动 CAMERA_CONTROL_AF_MODE_AUTO: 自动

接口调用示例：

```
camera_control_af_mode(CAMERA_CONTROL_AF_MODE_OFF);
```

返回值：0：成功；-1 失败

接口功能说明：

设置相机聚焦模式，通过 camera_control_get_features () 获取到相机是否支持手动模式

(cameraCfg.AFmode) ，只有支持手动才可以设置，否则设置无效，相机不执行该命令，默认自动。

3.5 设置相机镜头距离

接口名称	接口参数	参数说明
camera_control_lens_move()	uint32_t lens_position	距离大小, 范围 (1-100)

接口调用示例:

```
camera_control_lens_move(10);
```

返回值: 0: 成功; -1 失败

接口功能说明:

手动聚焦时候用, 距离越大, 值越大。

3.6 触发单次聚焦

接口名称	接口参数	参数说明
camera_control_focus_trigger()	无	

接口调用示例:

```
camera_control_focus_trigger();
```

返回值: 0: 成功; -1 失败

接口功能说明:

单次聚焦。

3.7 选择相机曝光模式

接口名称	接口参数	参数说明
camera_control_ae_mode()	camera_ctrl_ae_mode flash_mode	手动, 自动选择。

接口调用示例:

```
camera_control_ae_mode(CAMERA_CONTROL_AE_AUTO_FLASH_MODE_AUTO);
```

返回值: 0: 成功; -1 失败

接口功能说明:

曝光方法设置。

3.8 设置相机曝光时间

接口名称	接口参数	参数说明
camera_control_ae_set_exp()	uint32_t exp_compensation	曝光时间设置, 单位微秒(us) 范围 (1-1/fps)

接口调用示例:

```
camera_control_ae_set_exp(20000);
```

返回值: 0: 成功; -1 失败

接口功能说明：

手动曝光，设置曝光时间。

3.9 设置相机曝光增益

接口名称	接口参数	参数说明
camera_control_ae_set_gain ()	uint32_t iso_val	增益值

接口调用示例：

```
camera_control_lens_move(100);
```

返回值：0：成功；-1 失败

接口功能说明：

手动曝光时候，设置增益值，通过上面 3.1 API 接口 camera_control_get_features () 获取到 minGain, maxGain 值（见结构体 SensorModesConfig），手动设置。

3.10 选择相机白平衡模式

接口名称	接口参数	参数说明
camera_control_awb_mode()	camera_ctrl_awb_mode awb_mode	手动，自动

接口调用示例：

```
camera_control_awb_mode(CAMERA_CONTROL_AWB_MODE_AUTO);
```

返回值：0：成功；-1 失败

接口功能说明：

相机白平衡设置，手动，自动选择。

3.11 浮点数转化

接口名称	接口参数	参数说明
f16Tof32()	unsigned int x	16 位数据

接口调用示例：

```
Float f=f16Tof32(100);
```

返回值：浮点数

接口功能说明：

16 位 short 数据转浮点数，用于 meta data 计算分析。

二： SDK Python 接口说明

从 **API2.0.x** 开始支持 **python API**, **sdk** 接口见 **openncc.py** 文件, 使用时候导入该模块即可, 如:
import openncc as ncc.

1.设备初始化相关接口

1.1 获取 sdk 版本信息

接口名称	接口参数	参数说明
get_sdk_version()	无	

接口调用示例:

```
print("get usb %d sdk versin %s" % (ncc.get_usb_version() ,ncc.get_sdk_version()))
```

返回值: 版本信息

接口功能说明:

获取 sdk 版本信息。

1.2 获取设备连接 usb 信息

接口名称	接口参数	参数说明
get_usb_version()	无	无

返回值: 30: usb3.0、20: usb2.0

接口调用示例:

```
print("get usb %d sdk versin %s" % (ncc.get_usb_version() ,ncc.get_sdk_version()))
```

接口功能说明:

获取设备连接的 usb 版本信息 (端口和 usb 线)

1.3 加载设备固件

接口名称	接口参数	参数说明
	bootExe	Usb boot 程序路径
load_fw()	firmware	固件文件放置路径

返回值: 0: 成功; -1 失败

接口调用示例:

```
res = ncc.load_fw("./moviUsbBoot","fw/flicRefApp.mvcmd")
if res<0:
    printf('load firmware error!')
    sys.exit(1)
```

接口功能说明：

自动加载设备固件，设备 boot 运行，host (PC) 打开 usb 设备。

1.4 初始化相机参数

接口名称	接口参数	参数说明
sdk_init()	vscRecvCb cb	回调函数
	param	回调函数参数
	blob_path	模型文件路径
	cam	相机配置参数具体内容见下方
	Cam_len	相机配置结构体长度

媒体数据以及 **meta** 数据有 **2** 种获取方法，具体见 **c/c++** 对应的该接口描述。

接口调用示例：

```
cam_info=ncc.CameraInfo()
cam_info.inputFormat=ncc.IMG_FORMAT_BGR_PLANAR
cam_info.stdValue=1
```

```
cam_info.isOutputYUV=1
cam_info.isOutputH26X=1
cam_info.isOutputJPEG=1
```

```
cam_info.imageWidth = cameraCfg.camWidth
cam_info.imageHeight = cameraCfg.camHeight
cam_info.startX      = 0
cam_info.startY      = 0
cam_info.endX        = cameraCfg.camWidth
cam_info.endY        = cameraCfg.camHeight
cam_info.inputDimWidth =0
cam_info.inputDimHeight =0
ncc.SetMeanValue(cam_info,0.0,0.0,0.0)
```

```
ret = ncc.sdk_init(None, None, "./blob/face-detection-retail-0004-fp16.blob",cam_info,
struct.calcsize("1314f"))
```

```
print("xlink_init ret=%d " % ret)
if (ret<0):
    return
```

接口功能说明：

指定相机 **AI** 模型文件，**AI** 计算参数，初始化设备算法模型，相机功能开关选择，通过 **mode** 参数设置视频

压缩编码参数 (**ENCODE_H264_MODE**, **ENCODE_H265_MODE**) , 注意, 此处仅仅是功能开关是否开启, 视频输出还要通过 **camera_video_out ()** 控制是否输出。

1.5 移除 sdk

接口名称	接口参数	参数说明
sdk_uninit()	无	无

接口调用示例:

```
sdk_uninit();
```

返回值: 无

接口功能说明:

相机关闭, 重新加载算法模型, 更换模型前调用

2. 视频流相关接口

2.1 获取设备 yuv 数据

接口名称	接口参数	参数说明
GetYuvData()	yuvbuf	接收缓存区, bytearray 类型

接口调用示例:

```
metasize=ncc.get_meta_size()
offset=struct.calcsize(media_head)
yuvsz=cameraCfg.camWidth*cameraCfg.camHeight*2
yuvbuf = bytearray(yuvsz+offset)
metabuf = bytearray(metasize+offset)
size = ncc.GetYuvData(yuvbuf)
```

返回值: yuv 实际数据大小。

接口功能说明:

获取设备 yuv 数据流

2.2 获取设备 H.264 或 H.265 数据流

接口名称	接口参数	参数说明
GetH26xData()	databuf	接收缓存区, bytearray 类型

接口调用示例:

用法同获 2.1 获取 yuv 数据

接口功能说明:

获取设备 H.264 或 H.265 数据流

2.3 获取设备 jpeg 数据

接口名称	接口参数	参数说明
GetJpegData()	databuf	接收缓存区,bytearray 类型

接口调用示例:

用法同获 2.1 获取 yuv 数据

接口功能说明:

获取设备 jpeg 数据流

2.4 获取设备 AI 网络数据运算结果

接口名称	接口参数	参数说明
GetMetaData()	databuf	接收缓存区,bytearray 类型

接口调用示例:

用法同获 2.1 获取 yuv 数据

接口功能说明:

获取设备 AI 网络运算结果数据

3. 相机控制相关接口

3.1 获取相机模组信息

接口名称	接口参数	参数说明
CameraSensor 类	GetFirstSensor () , GetNextSensor ()	

接口调用示例:

```
sensors=ncc.CameraSensor()
sensor1 = ncc.SensorModesConfig()
if sensors.GetFirstSensor(sensor1)==0:
    print("camera: %s, %dX%d@%dfps, AFmode:%d,
maxEXP:%dus,gain[%d, %d]\n" % (
    sensor1.moduleName, sensor1.camWidth, sensor1.camHeight, sensor1.camFps,
    sensor1.AFmode, sensor1.maxEXP, sensor1.minGain, sensor1.maxGain))

sensor2 = ncc.SensorModesConfig()
while sensors.GetNextSensor(sensor2)==0:
    print("camera: %s, %dX%d@%dfps, AFmode:%d,
maxEXP:%dus,gain[%d, %d]\n" % (
    sensor2.moduleName, sensor2.camWidth, sensor2.camHeight, sensor2.camFps,
```

sensor2.AFmode, sensor2.maxEXP, sensor2.minGain, sensor2.maxGain))

接口功能说明:

获取相机可见光模组模式信息, 有的相机支持多种视频模式, 可以通过 camera_select_sensor () 选择使用。

3.2 选择模组工作模式

接口名称	接口参数	参数说明
camera_select_sensor()	sensorid	camera_control_get_features 获取到相机支持的模组信息数组, sensorid 为数组的序号。

接口调用示例:

```
ncc.camera_select_sensor(0)
```

返回值: 0: 成功; -1 失败

接口功能说明:

设置相机可见光模组的工作模式。

3.3 控制相机视频输出方式

接口名称	接口参数	参数说明
camera_video_out()	video_type	数据类型
	out mode	禁止, 单次 (拍照用), 连续

接口调用示例:

```
ncc.camera_video_out(ncc.YUV420p,ncc.VIDEO_OUT_CONTINUOUS)
```

返回值: 0: 成功; -1 失败

接口功能说明:

控制设备输出视频数据的模式, 该设置当前对 YUV420p, H26X, JPEG 有效, 其中好 H26X 不支持单次输出。

3.4 设置相机聚焦模式

接口名称	接口参数	参数说明
camera_control_af_mode()	camera_ctrl_af_mode af_mode	CAMERA_CONTROL_AF_MODE_OFF : 手动 CAMERA_CONTROL_AF_MODE_AUTO: 自动

接口调用示例:

```
ncc.camera_control_af_mode(ncc.CAMERA_CONTROL_AF_MODE_AUTO);
```

返回值：0：成功；-1 失败

接口功能说明：

设置相机聚焦模式，通过 camera_control_get_features () 获取到相机是否支持手动模式 (cameraCfg.AFmode) ，只有支持手动才可以设置，默认自动。

3.5 选择相机镜头距离

接口名称	接口参数	参数说明
camera_control_lens_move()	lens_position	距离大小，范围 (1-100)

接口调用示例：

```
ncc.camera_control_lens_move(10);
```

返回值：0：成功；-1 失败

接口功能说明：

手动聚焦时候用，距离越大，值越大。

3.6 触发单次聚焦

接口名称	接口参数	参数说明
camera_control_focus_trigger()	无	

接口调用示例：

```
camera_control_focus_trigger();
```

返回值：0：成功；-1 失败

接口功能说明：

单次聚焦。

3.7 选择相机曝光模式

接口名称	接口参数	参数说明
camera_control_ae_mode()	camera_ctrl_ae_mode flash_mode	手动，自动选择。

接口调用示例：

```
ncc.camera_control_ae_mode( ncc.CAMERA_CONTROL_AE_AUTO_FLASH_MODE_AUTO);
```

返回值：0：成功；-1 失败

接口功能说明：

曝光方法设置。

3.8 选择相机曝光时间

接口名称	接口参数	参数说明
camera_control_ae_set_exp()	exp_compensation	曝光时间设置, 单位微秒(us) 范围 (1-1/fps)

接口调用示例:

```
ncc.camera_control_ae_set_exp(20000);
```

返回值: 0: 成功; -1 失败

接口功能说明:

手动曝光, 设置曝光时间。

3.9 选择相机曝光增益大小

接口名称	接口参数	参数说明
camera_control_ae_set_gain ()	iso_val	增益值

接口调用示例:

```
ncc.camera_control_lens_move(100);
```

返回值: 0: 成功; -1 失败

接口功能说明:

手动曝光时候, 设置增益值, 通过上面 3.1 API 接口 camera_control_get_features () 获取到 minGain, maxGain 值 (见对象 SensorModesConfig), 手动设置。

3.10 选择相机白平衡模式

接口名称	接口参数	参数说明
camera_control_awb_mode()	camera_ctrl_awb_mode awb_mode	手动, 自动

接口调用示例:

```
ncc.camera_control_awb_mode(ncc.CAMERA_CONTROL_AWB_MODE_AUTO);
```

返回值: 0: 成功; -1 失败

接口功能说明:

相机白平衡设置, 手动, 自动选择。

3.11 浮点数转化

接口名称	接口参数	参数说明
f16Tof32()	x	16 位数据

接口调用示例:

```
f=f16Tof32(100);
```

返回值: 浮点数

接口功能说明:

16 位 short 数据转浮点数, 用于 meta data 计算分析。