

2 SHQ1 && 6' .

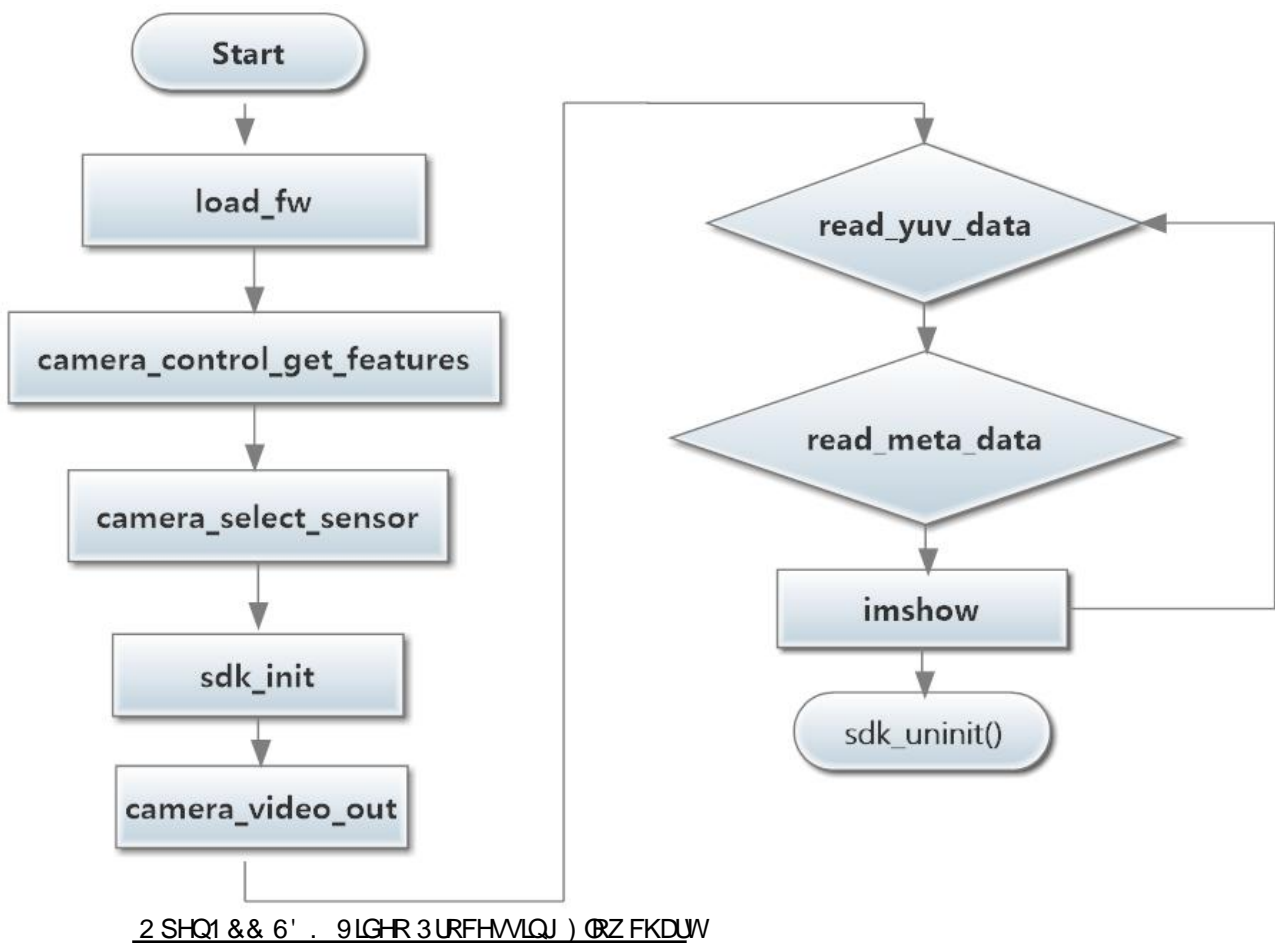
\$ 3, [, QMUDFH' RFXP HQMDMRQ

Version History

版本	日期	修改	变更摘要	sdk 版本
1.0.0	2020/1/10	王新华	Initial version	1.0.0
1.0.1	2020/3/16	王洋	Optimized version	1.0.1
2.0.0	2020/4/7	左文平	修订接口, 添加 python 接口	2.0.0
3.0.0	2020/10/14	左文平	1:添加支持 2 个模型接口 sdk_net2_init () 以及对应的结构体. 2: meta 数据格式增加了 64 字节。 3:移除了读取红外数据和深度数据接口。	3.0.1

一： & & 6' . ,QMUDFH' HMFUSWRQ

The interface files are primarily contained in 3 files: VGNK FDP HUD&WCK and) S &RQYHUW



' HYLFIH ,QWDOQ DMRQ 5 HODMG ,QMUDFHV

/ RDG' HYLFIH) IWP Z DUH

Interface Name	Interface Parameters	Description
ØRDBIZ	const char* bootExe	Path to USB boot program
	FRQWFKDU IWP Z DUH	Path to firmware file

([DP SØI
load_fw("./moviUsbBoot", "./fw/flicRefApp.mvcmnd");

Returns:0 if successful, -1 otherwise

' HMFUSWRQ
Automatically loads device firmware, device boots, host (PC) opens USB device.

* HM&RQYHUW 8 6%,QIRUP DMRQ

Interface Name	Interface Parameters	Description
get_usb_version()	void	N/A

([DP SØI
version = get_usb_version();

Returns:30 if USB 3.0, 20 if USB 2.0

' HMFUSMRQ

Returns USB Version Information (Port and USB Cable) connected to the device.

, QWDDQ IQJ \$, &DP HUD 3DUDP HMLU

Interface Name	Interface Parameters	Description
sdk_init()	vscRecvCb cb	Callback
	void* param	Callback function parameters
	const char *blob_path	Path to AI Model (.blob)
	CameraInfo *cam	Camera Configuration Parameters (See below)
	int cam_Len	Camera Configuration Structure Length

There are two ways to get media and metadata. 1: Passively obtained through callback function, 2: Actively obtained through read_XXX_data() without setting the callback function and callback parameters.

([DP SQ

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("13l4f"))

print("xlink_init ret=%d " % ret)

if (ret<0):

return

' HMFUSMRQ

Specifies the AI Vision model file and calculation parameters, initializes the device algorithm model, camera function switch selection, sets the video encoding parameters (if the function switch is turned on). Video output is controlled by camera_video_out().

1.4 , QWDDQ IQJ \$, &DP HUD P RGHV 3DUDP HMLU

Interface Name	Interface Parameters	Description
sdk_net2_init()	vscRecvCb cb	Callback

	void* param	Callback function parameters
	const char *blob1_path	AI Mode 1 file(blob)
	Network1Par* par1	Model param
	int par1_Len	Param1 length
	const char *blob2_path	AI Mode 2 file(blob)
	Network2Par* par2	Model param
	int par2_Len	Param2 length

There are two ways to get media and metadata. 1: Passively obtained through callback function, 2: Actively obtained through read_XXX_data() without setting the callback function and callback parameters.

' HMFUSMRQ

Specifies the 2 AI Vision model files and calculation parameters, initializes the device algorithm model, camera function switch selection, sets the video encoding parameters (if the function switch is turned on). Video output is controlled by camera_video_out().

* HMD6UH

Interface Name	Interface Parameters	Description
get_meta_size()	void	N/A

([DP S

Omitted

Returns:Size of the CNN calculation result's metadata.

' HMFUSMRQ

Turn off the camera, reload the model, and call before changing the model.it only support one AI mode net now.

5 HP RYH6' .

Interface Name	Interface Parameters	Description
sdk_uninit()	void	N/A

([DP S

sdk_uninit();

Returns: N/A

' HMFUSMRQ

Turn off the camera, reload the model, and call before changing the model.

* H6' . 9HMRQ,QIRUP DMRQ

Interface Name	Interface Parameters	Description
get_sdk_version()	char* version	Version Information

([DP S

char version[100];
get_sdk_version(version);

Returns:void

' HMFUSMRQ

Gets SDK version information.

9 LG-R 6 WHP LQJ 5 HDWVG , QMUDFHV

* HWK89 ' DMD

Interface Name	Parameters	Description
read_yuv_data()	char* pbuf	Receive Buffer
	int * size	Input and output parameters. Input is the size of the input buffer , output is the size of the returned video data.
	int blocked	0: If there is no data, return immediately. 1: Wait until data is read to return.

([DP S0H
read_yuv_data(data_yuv,&size,1)

Returns:0 if successful, -1 otherwise

' HMFUSWRQ

Gets a YUV data stream from the device. Content: struct frameSpecOut+YUV (NV12).

* HWK RU+ ' DMD

Interface Name	Parameters	Description
read_26x_data()	char* pbuf	Receive Buffer
	int * size	Input and output parameters. Input is the size of the input buffer , output is the size of the returned video data.
	int blocked	0: If there is no data, return immediately. 1: Wait until data is read to return.

([DP S0H
read_26x_data(data_26x,&size,1)

Returns:0 if successful, -1 otherwise

' HMFUSWRQ

Gets a H.264 or H.265 data stream from the device.Content: struct frameSpecOut+H26X data.

* HW 3(* GDM

Interface Name	Parameters	Description
read_jpg_data()	char* pbuf	Receive Buffer
	int * size	Input and output parameters. Input is the size of the input buffer area, output is the size of the returned video data.
	int blocked	0: If there is no data, return immediately. 1: Wait until data is read to return.

([DP S0H
read_jpg_data(yuv420p,&size,1)

Returns:0 if successful, -1 otherwise

' HMFUSWRQ

Gets a JPEG data stream from the device.Content: struct frameSpecOut+MJPEG data.

* HWKH RXV SXVRI WKH \$, 1 HZ RUN DO RUMK P

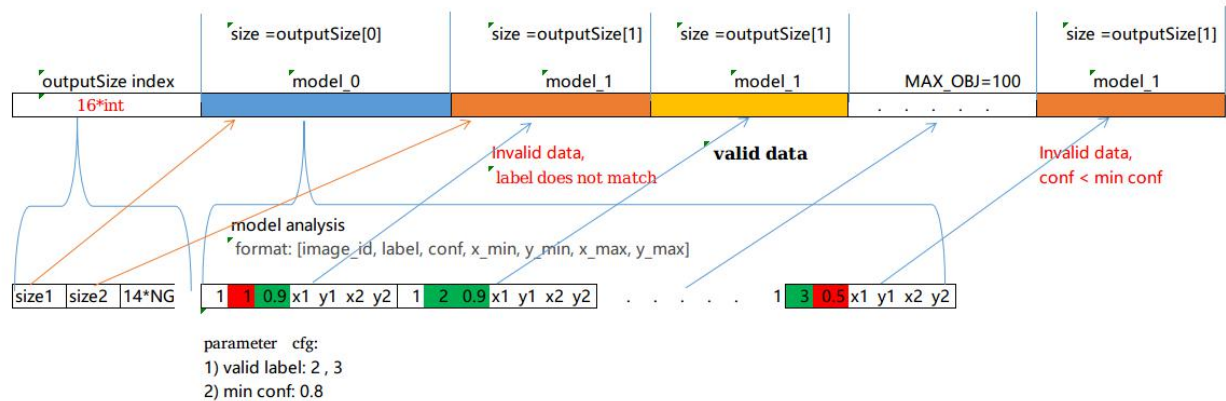
Interface Name	Parameters	Description
read_meta_data()	char* pbuf	Receive Buffer
	int * size	Input and output parameters. Input is the size of the input buffer area, output is the size of the returned video data.

	int blocked	0: If there is no data, return immediately. 1: Wait until data is read to return.
--	-------------	--

```
( [ DP S0H
read_meta_data(data_meta,&size,1)

Returns:0 if successful, -1 otherwise
```

' HMFUSWRQ
Get the number of operations from the device's AI Network.Content: struct frameSpecOut+AI data.and AI data format is below:



& DP HUD & RQMR05 H0DMG , QMU DFHV

2 EVMLQ & DP HUD 0 RGX0H , QIRUP DMRQ

Interface Name	Parameters	Description
camera_control_get_features()	SensorModesConfig *	Device information structure pointer

```
( [ DP S0H
SensorModesConfig cameraCfg;
camera_control_get_features(&cameraCfg);

Returns:0 if successful, -1 otherwise
```

cameraCfg.moduleName: Camera Module Name
cameraCfg.camWidth: Image Width
cameraCfg.camHeight: Image Height
cameraCfg.camFps: Camera Frame Rate
cameraCfg.AFmode: Autofocus (1 if supported, 0 otherwise)
cameraCfg.maxEXP: Maximum exposure time in microsecond (μs)
cameraCfg.minGain: Minimum Gain Multiplier
cameraCfg.maxGain: Maximum Gain Multiplier

Description:
Obtain information about the mode of the camera (获取相机可见光模组模式信息). Some cameras will support multiple video modes, which can be selected through camera_select_sensor().

6 H0FW0 RGX0H : RUNQ 0 RGH

Interface Name	Parameters	Description
camera_select_sensor()	int sensorid	camera_control_get_features() obtains the array of information of supported camera modes. sensorid is the serial number of the array.

```
( [ DP S0H
camera_select_sensor(0);
```

Returns: 0 if successful, -1 otherwise

```
' HMFUSMRQ
    Sets the working mode of the camera's visible light module.
```

&RQMRO&DP HUDIV 9 LGHR 2 XVSXW

Interface Name	Parameters	Description
camera_video_out()	int video_type	YUV Data output mode
	camera_ctrl_VIDEO_out mode	Disabled, Single (For photos), Continuous

```
typedef enum
{
    VIDEO_OUT_DISABLE,    /* Output Disabled */
    VIDEO_OUT_SINGLE,     /* Single Output */
    VIDEO_OUT_CONTINUOUS, /* Continuous Output */
}camera_ctrl_video_out;
```

```
( [ DP S0H
camera_video_out(YUV420p,VIDEO_OUT_CONTINUOUS);
```

Returns:0 if successful, -1 otherwise

```
' HMFUSMRQ
    Sets the device to output video data. This works for YUV420p, H26X, JPEG. H26X does not support single output.
```

6 HM&DP HUDIV) RFXV 0 RGH

Interface Name	Parameters	Description
camera_control_af_mode()	camera_ctrl_af_mode af_mode	CAMERA_CONTROL__AF_MODE E_OFF :Manual Focus CAMERA_CONTROL__AF_MODE E_AUTO:Automatic Focus

```
( [ DP S0H
camera_control_af_mode(CAMERA_CONTROL__AF_MODE_OFF);
Returns:0 if successful, -1 otherwise
```

```
' HMFUSMRQ
    Sets the camera to manual focus. Using camera_control_get_features() one can check if the camera supports manual focusing (cameraCfg.AFmode). If not supported, the setting is invalid and the camera defaults to automatic.
```

6 HM&DP HUDIV / HQV' LMDQFH

Interface Name	Parameters	Description
camera_control_lens_move()	uint32_t lens_position	Range of distances (1-100)

```
( [ DP S0H
camera_control_lens_move(10);
```

Returns:0 if successful, -1 otherwise

```
' HMFUSMRQ
    Used when focusing manually, greater value is a greater distance.
```

7UJJHU6LQJ0H) RFXV

Interface Name	Parameters	Description
----------------	------------	-------------

camera_control_focus_trigger()	void	N/A
--------------------------------	------	-----

([DP S0H

camera_control_focus_trigger();

Returns:0 if successful, -1 otherwise

' HMFUSWRQ

Focuses the camera once

6 HM&DP HUDV ([SRVXUH 0 RGH

Interface Name	Parameters	Description
camera_control_ae_mode()	camera_ctrl_ae_mode flash_mode	Manual or Automatic

([DP S0H

camera_control_ae_mode(CAMERA_CONTROL__AE_AUTO__FLASH_MODE__AUTO);

Returns: 0 if successful, -1 otherwise

' HMFUSWRQ

Sets exposure mode.

6 HM ([SRVXUH 7 LP H

Interface Name	Parameters	Description
camera_control_ae_set_exp()	uint32_t exp_compensation	Exposure duration in microsecond (µs) range (1-1 / fps)

([DP S0H

camera_control_ae_set_exp(20000);

Returns: 0 if successful, -1 otherwise

' HMFUSWRQ

Sets the exposure time for the manual exposure mode.

6 HM&DP HUD * DLQ

Interface Name	Parameters	Description
camera_control_ae_set_gain ()	uint32_t iso_val	Gain value

([DP S0H

camera_control_ae_set_gain(100);

Returns: 0 if successful, -1 otherwise

' HMFUSWRQ

Sets the gain in manual exposure mode. Min/max gain values can be retrieved through *camera_control_get_features()* and set manually.

6 HM&DP HUD : KLM %D0QFH 0 RGH

Interface Name	Parameters	Description
camera_control_awb_mode()	camera_ctrl_awb_mode awb_mode	Manual or Auto

([DP S0H

camera_control_awb_mode(CAMERA_CONTROL__AWB_MODE__AUTO);

Returns: 0 if successful, -1 otherwise

' HMFUSWRQ

Sets camera to automatic white balance mode.

) 0x00000000

Interface Name	Parameters	Description
f16Tof32()	unsigned int x	16-bit Data

```
( [ DP S0H
Float f=f16Tof32(100);
```

Returns: Float

```
' HMFUSMRQ
    Converts 16-bit short data to a floating point number. Used for metadata calculations and analysis.
```

二 3\VRQ6' . ,QMUDFH' RFXP HQMDMRQ

Starting from version 2.0.X onwards, the API will support Python. The SDK Interface can be found in the openncc.py file. To use the module, import it using: import openncc as ncc

```
' HYL FH , QMDQ DMRQ 5 HQMDMG , QMUDFHV
```

```
* HMF' . 9HJMRQ
```

Interface Name	Parameters	Description
get_sdk_version()	void	N/A

```
( [ DP S0H
print("get usb %d sdk versin %s" % (ncc.get_usb_version(), ncc.get_sdk_version()))
Returns: SDK Version
```

```
' HMFUSMRQ
    Gets the SDK version.
```

```
* HMFHQRI &RQHFMDMG 86%' HYL FH
```

Interface Name	Parameters	Description
get_usb_version()	void	N/A

```
( [ DP S0H
print("get usb %d sdk versin %s" % (ncc.get_usb_version(), ncc.get_sdk_version()))
Returns: 30 if USB 3.0, 20 if USB 2.0
```

```
' HMFUSMRQ
    Returns USB Version Information (Port and USB Cable) connected to the device
```

```
/ RDG' HYL FH) ILP Z DUH
```

Interface Name	Parameters	Description
0x00000000	bootExe	Path to USB boot program
	ILP Z DUH	Path to firmware file

```
( [ DP S0H
res = ncc.load_fw("./moviUsbBoot", "fw/flicRefApp.mvcmnd")
if res<0:
    printf('load firmware error!')
    sys.exit(1)
Returns: 0 if successful, -1 otherwise
```

' HMFUSMRQ

Automatically loads device firmware, device boots, host (PC) opens USB device.

, QWDDQ LQJ & DP HUD 3 DUDP HMLUJ

Interface Name	Parameters	Description
sdk_init()	vscRecvCb cb	Callback
	param	Callback function parameters
	blob_path	Path to AI Model (.blob)
	cam	Camera Configuration Parameters (See below)
	Cam_len	Camera configuration structure length

There are two ways to get Media and Metadata. 1: Passively obtained through callback function, 2: Actively obtained through read_XXX_data() without setting the callback function and callback parameters.

([DP SDH

```
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("13l4f"))
```

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

' HMFUSMRQ

Specifies the AI Vision model file and calculation parameters, initializes the device algorithm model, camera function switch selection, sets the video encoding parameters (if the function switch is turned on). Video output is controlled by camera_video_out().

1.4 , QWDDQ LQJ \$, & DP HUD P RGHV 3 DUDP HMLUJ

Interface Name	Interface Parameters	Description
sdk_net2_init()	vscRecvCb cb	Callback
	void* param	Callback function parameters
	const char *blob_path	AI Mode 1 file(blob)
	Network1Par* par1	Model param
	int par1_len	Param1 length
	const char *blob2_path	AI Mode 2 file(blob)
	Network2Par* par2	Mode2 param
	int par2_len	Param2 length

There are two ways to get media and metadata. 1: Passively obtained through callback function, 2: Actively

obtained through read_XXX_data() without setting the callback function and callback parameters.

```
( [ DP S0H
char *blob =  "./blob/vehicle-license-plate-detection-barrier-0106/vehicle-license-plate-
detection-barrier-0106.blob";
char *blob2 =  "./blob/license-plate-recognition-barrier-0001/license-plate-recognition-barrier-
0001.blob";
```

```
ret = sdk_net2_init(0,0,\
blob, &cnn1PrmSet, sizeof(cnn1PrmSet), \
blob2, &cnn2PrmSet, sizeof(cnn2PrmSet));
```

Returns:0 if successful. -1 otherwise

' HVFUSWRQ
Specifies the 2 AI Vision model files and calculation parameters, initializes the device algorithm model, camera function switch selection, sets the video encoding parameters (if the function switch is turned on). Video output is controlled by camera_video_out().

, QWDDQ LQJ & DP HUD Z LMK \$, 3 DUDP HMMU/

Interface Name	Parameters	Description
sdk_init()	vscRecvCb cb	Callback
	param	Callback function parameters
	Blob1_path	Path to AI Model (.blob)
	Par1	Model param
	Par1_len	Param1 length
	blob2_path	

There are two ways to get Media and Metadata. 1: Passively obtained through callback function, 2: Actively obtained through read_XXX_data() without setting the callback function and callback parameters.

```
( [ DP S0H
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
```

' HMFUSWRQ
Specifies the AI Vision model file and calculation parameters, initializes the device algorithm model, camera function switch selection, sets the video encoding parameters (if the function switch is turned on). Video output is controlled by camera_video_out().

JHMP HMD GMD VU H

Interface Name	Parameters	Description
get_meta_size()	void	N/A

```
( [ DP SØ
metasize=ncc.get_meta_size()
print("xlink_init ret= %d" % (metasize))
```

Returns:meta data size

' HMFUSWRQ
Turn off the camera, reload the model, and call before changing the model.only support one AI mode now.

8 QQMDQ H6' .

Interface Name	Parameters	Description
sdk_uninit()	void	N/A

```
( [ DP SØ
sdk_uninit();
```

Returns:N/A

' HMFUSWRQ
Turn off the camera, reload the model, and call before changing the model.

9 LGHR 6 WHDP LQ 5 HDMG ,QMUDFHV

* HW89 ' DM

Interface Name	Parameters	Description
GetYuvData()	yuvbuf	Bytearray receive buffer

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

Returns:Size of the YUV data.

' HMFUSWRQ
Gets a YUV data stream from the device.Content: struct frameSpecOut+YUV(NV12) data.

* HW+ RU+ ' DMD

Interface Name	Parameters	Description
GetH26xData()	databuf	Bytearray receive buffer

([DP S0H
Same as 2.1.

' HMFUSMRQ

Gets a H.264 or H.265 data stream from the device.Content: struct frameSpecOut+H26X data.

* HW 3(* GDM

Interface Name	Parameters	Description
GetJpegData()	databuf	Bytearray receive buffer

([DP S0H
Same as 2.1

' HMFUSMRQ

Gets a JPEG data stream from the device.Content: struct frameSpecOut+MJPEG data.

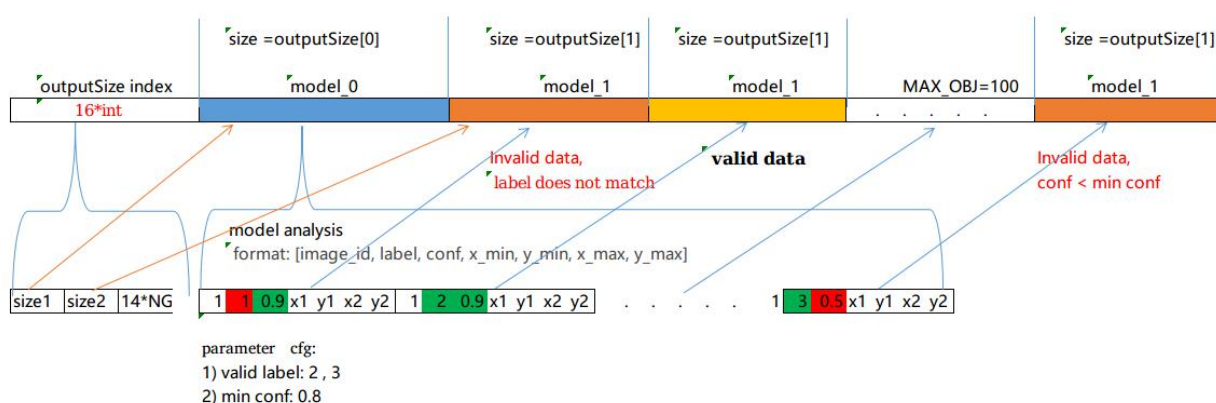
2.4 Get the AI Network inference (获取设备 AI 网络数据运算结果?)

Interface Name	Parameters	Description
GetMetaData()	databuf	Bytearray receive buffer

([DP S0H
Same as 2.1

' HMFUSMRQ

Get the number of operations from the device's AI Network.Content: struct frameSpecOut+AI data..and AI data format is below:



& DP HUD & RQMRC5 HODMG, QMU DFHV

2 EVMLQ & DP HUD 0 RGX0H, QIRUP DMRQ

Interface Name	Parameters	Description
CameraSensor Class	GetFirstSensor () , GetNextSensor ()	

([DP S0H
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))

```

' HMFUSMRQ

Obtains information about the mode of the camera (获取相机可见光模组模式信息). Some cameras will support multiple video modes, which can be selected through camera_select_sensor().

6H0FW0 RGX0M: RUNQ 0 RGH

Interface Name	Parameters	Description
camera_select_sensor()	sensorid	camera_control_get_features() obtains the array of information of supported camera modes. sensorid is the serial number of the array.

([DP S0H

ncc.camera_select_sensor(0)

Returns:0 if successful, -1 otherwise

' HMFUSMRQ

Sets the working mode of the camera's visible light module.

&RQMROMH&DP HUDV 9LGR 2 XISXW

Interface Name	Parameters	Description
camera_video_out()	video_type	Video data type
	out mode	Disabled, Single (For photos), or Continuous

([DP S0H

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

Returns:0 if successfuk, -1 otherwise

' HMFUSMRQ

Sets the device to output video data. This works for YUV420p, H26X, JPEG. H26X does not support single output.

6HM&DP HUDV) RFXV 0 RGH

Interface Name	Parameters	Description
camera_control_af_mode()	camera_ctrl_af_mode af_mode	CAMERA_CONTROL__AF_MODE_E_OFF : Manual Focus CAMERA_CONTROL__AF_MODE_E_AUTO:Automatic Focus

([DP S0H

ncc.camera_control_af_mode(ncc.CAMERA_CONTROL__AF_MODE_AUTO);

Returns:0 if successful, -1 otherwise

' HMFUSWRQ

Sets the camera to manual focus. Using camera_control_get_features() one can check if the camera supports manual focusing (cameraCfg.AFmode). If not supported, the setting is invalid and the camera defaults to automatic focusing.

6 HWMH&DP HUDV / HQV ' LMDQFH

Interface Name	Parameters	Description
camera_control_lens_move()	lens_position	Range of distances (1-100)

([DP SØ

ncc.camera_control_lens_move(10);

Returns:0 if successful, -1 otherwise

' HMFUSWRQ

Used when focusing manually, greater value is a greater distance.

7 UJ JHU6 IQJ Ø) RFXV

Interface Name	Parameters	Description
camera_control_focus_trigger()	void	N/A

([DP SØ

camera_control_focus_trigger();

Returns:0 if successful, -1 otherwise

' HMFUSWRQ

Focuses the camera once.

6 HM&DP HUDV ([SRVXUH 0 RGH

Interface Name	Parameters	Description
camera_control_ae_mode()	camera_ctrl_ae_mode flash_mode	Manual or Automatic

([DP SØ

ncc.camera_control_ae_mode(ncc.CAMERA_CONTROL__AE_AUTO__FLASH_MODE__AUTO);

Returns:0 if successful, -1 otherwise

' HMFUSWRQ

Sets exposure mode.

6 HM [SRVXUH 7 LP H

Interface Name	Parameters	Description
camera_control_ae_set_exp()	exp_compensation	Exposure duration in microsecond (µs) range (1-1 / fps)

([DP SØ

ncc.camera_control_ae_set_exp(20000);

Returns:0 if successful, -1 otherwise

' HMFUSWRQ

Sets the exposure time for the manual exposure mode.

6 HM&DP HUD * DQ

Interface Name	Parameters	Description
----------------	------------	-------------

camera_control_ae_set_gain ()	iso_val	Gain value
--------------------------------	---------	------------

```
( [ DP S0H
ncc.camera_control_lens_move(100);
```

Returns:0 if successful, -1 otherwise

```
' HVFUSMRQ
    Sets the gain in manual exposure mode. Min/max gain values can be retrieved through
camera_control_get_features() and set manually.
```

6 HW&DP HUD: KLM %D0QFH 0 RGH

Interface Name	Parameters	Description
camera_control_awb_mode()	camera_ctrl_awb_mode awb_mode	Manual or Automatic

```
( [ DP S0H
ncc.camera_control_awb_mode(ncc.CAMERA_CONTROL__AWB_MODE__AUTO);
Returns:0 if successful, -1 otherwise
```

```
' HVFUSMRQ
    Sets camera to automatic white balance mode.
) 0RDV&RQYHU/LRQ
```

Interface Name	Parameters	Description
f16Tof32()	x	16-bit Data

```
( [ DP S0H
f=f16Tof32(100);
Returns:Float
```

```
' HVFUSMRQ
Converts 16-bit short data to a floating point number. Used for metadata calculations and analysis.
```