Smart Video Evaluation Toolkit – Sample Application User Guide

Dec 2019

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting: http://www.intel.com/design/literature.htm

[Delete this paragraph if your document does not message any Intel® technologies that need anything else in order to work.] Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at http://www.intel.com/ or from the OEM or retailer.

Intel [include any Intel trademarks which are used in this document] and the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others.

Contents

Revision History

Date	Revision	Description
2019/12/26	1.0	Initial release

Note: Releases in the table are listed in reverse order so that the latest/newest is in the top row.

§

1.0 Installation Guide

1.1 System installation

Install Ubuntu 18.04 to a CFL device (e.g. NUC8i7BEH)

Set up the network correctly and run “sudo apt update”

1.2 Install Openvino R3

Download Openvino R3 package from https://software.intel.com/en-us/openvino-toolkit

Install Openvino R3 according to https://docs.openvinotoolkit.org/latest/_docs_install_guides_installing_openvino_linux.html#install-openvino

By default, Openvino R3 is installed to “/opt/intel/openvino”. It also can be installed to ~/intel/openvino. In this case, please replace “/opt/intel/openvino” with “~/intel/openvino” in following instructions.

Make sure the OpenCL driver is installed correctly by running “sudo /opt/intel/openvino/install_dependencies/install_NEO_OCL_driver.sh”. If you see below error message during install NEO OCL driver:

```
dpkg: dependency problems prevent removal of intel-igc-core:

 intel-igc-opencl depends on intel-igc-core (= 1.0.10-2407).

dpkg: error processing package intel-igc-core (--remove):

 dependency problems - not removing

Errors were encountered while processing:

 intel-igc-core
```

Please uninstall intel-igc-opencl and intel-igc-core manually by bellow command:

```
sudo dpkg -r intel-igc-opencl

sudo dpkg -r intel-igc-core
```

Then re-run command “sudo /opt/intel/openvino/install_dependencies/install_NEO_OCL_driver.sh”

Run “source /opt/intel/openvino/bin/setupvars.sh” and add “source /opt/intel/openvino/bin/setupvars.sh” to .bashrc under home directory.

1.3 Install SVET sample application and dependent libraries

Unzip svet_e2e_sample_l package and run bellow commands.

```
$cd  svet_e2e_sample_l
$sudo ./install_binary.sh
```

The installation script will run “apt install” to install dependent packages, copy the libva/media-driver binaries to /usr/lib/x86_64-linux-gnu/, MediaSDK libraires to /opt/intel/mediasdk. If you have installed libva/media-driver/MediaSDK libraries on the system, they will be overwritten. The installation script will also add several environment variable setting commands to ~/.bashrc

Run vainfo and you can see below output

```
$vainfo

error: can't connect to X server!

libva info: VA-API version 1.1.0

libva info: va_getDriverName() returns 0

libva info: User requested driver 'iHD'

libva info: Trying to open /usr/lib/x86_64-linux-gnu/dri/iHD_drv_video.so

libva info: Found init function __vaDriverInit_1_1

libva info: va_openDriver() returns 0

vainfo: VA-API version: 1.1 (libva 2.1.1.pre1)

vainfo: Driver version: Intel iHD driver - 2.0.0

vainfo: Supported profile and entrypoints

    VAProfileNone               : VAEntrypointVideoProc

    VAProfileNone               : VAEntrypointStats

    VAProfileMPEG2Simple        : VAEntrypointVLD
```

```
        VAProfileMPEG2Simple        : VAEntrypointEncSlice
        VAProfileMPEG2Main          : VAEntrypointVLD
        VAProfileMPEG2Main          : VAEntrypointEncSlice
        VAProfileH264Main           : VAEntrypointVLD
        VAProfileH264Main           : VAEntrypointEncSlice
        VAProfileH264Main           : VAEntrypointFEI
        VAProfileH264Main           : VAEntrypointEncSliceLP
        VAProfileH264High           : VAEntrypointVLD
        VAProfileH264High           : VAEntrypointEncSlice
        VAProfileH264High           : VAEntrypointFEI
        VAProfileH264High           : VAEntrypointEncSliceLP
....
```

Run “sudo ./download_models.sh” to download OpenVINO IR files for face detection, human pose estimation and vehicle detection.

If you don’t have video clip for testing, you can download sample video for face detection from https://raw.githubusercontent.com/intel-iot-devkit/sample-videos/master/head-pose-face-detection-male.mp4, human pose estimation from https://github.com/intel-iot-devkit/sample-videos/blob/master/classroom.mp4 and vehicle detection sample video from https://github.com/intel-iot-devkit/sample-videos/blob/master/car-detection.mp4. Since vaas sample application only support element stream, you can use bellow command to extract the element stream from MP4 file:

```
ffmpeg -i classroom.mp4 -vcodec copy -an -bsf:v h264_mp4toannexb
classroom.h264
```

After that, classroom.h264 can be used as input video stream.

2.0 Run SVET sample application

2.1 Check environment variables

Using below commands to check if environment variables LIBVA_DRIVERS_PATH and INTEL_OPENVINO_DIR set correctly.

```
$echo $LIBVA_DRIVERS_PATH

/usr/lib/x86_64-linux-gnu/dri

$echo $INTEL_OPENVINO_DIR

/home/dwang26/intel/openvino_2019.3.376
```

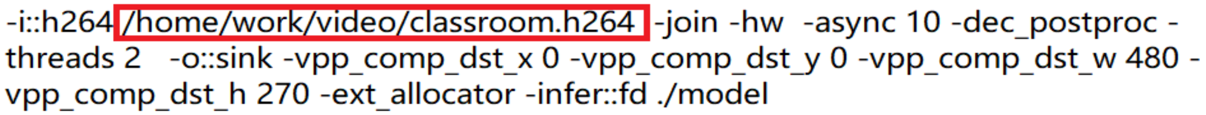
And use below command to check if there is any missing libraries:

```
ldd video_e2e_sample  | grep "not found"
```

If there is any library not found, it means the installation wasn't completed. Please contact your account manager from Intel and send the output of above command in email.

2.2 Modify parameter file

Modify the video path(following "-i::h264") of **every line** in par_file/face_detection_1080p_16_channel.par



```
-i::h264 /home/work/video/classroom.h264 -join -hw  -async 10 -dec_postproc -
threads 2   -o::sink -vpp_comp_dst_x 0 -vpp_comp_dst_y 0 -vpp_comp_dst_w 480 -
vpp_comp_dst_h 270 -ext_allocator -infer::fd ./model
```

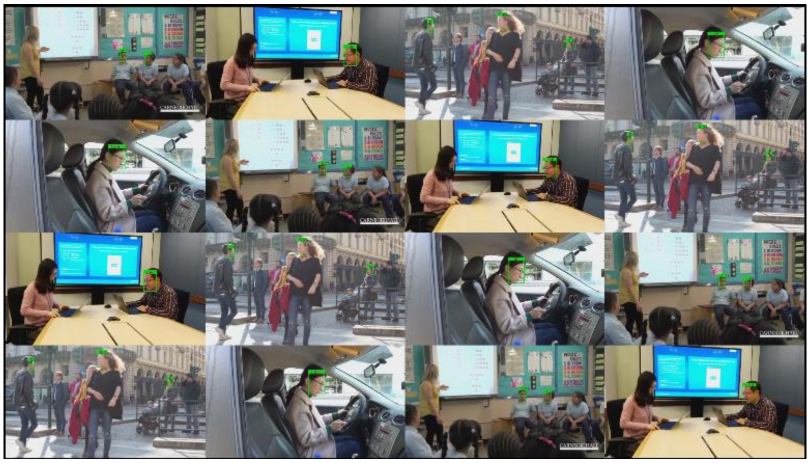
Otherwise you will see below error message when run the sample application

```
[ERROR], sts=MFX_ERR_NULL_PTR(-2), Init, m_fSource pointer is NULL at
/home/work/video_e2e_sample_l/MediaSDK/samples/video_e2e_sample/src/file_and_rtsp_bitstream_
```

2.3 Run video_e2e_sample application

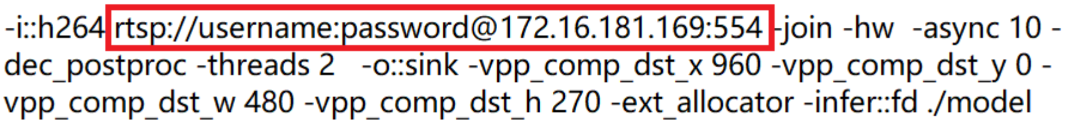
Before running the SVET sample application, you must switch to text mode by "Ctrl + Alt + F3". And then switch to root user by "su -p". The -p option is to keep the environment variables settings.

Then run “run_face_detection_test.sh” for testing 16 channel 1080p video decoding + face detection + one channel 1080p video encoding. The loading of face detection models is slow and you might need to wait for a minute until the video showing on display as below screenshot:



If you want to stop the application, press “Ctrl + c” in the bash shell.

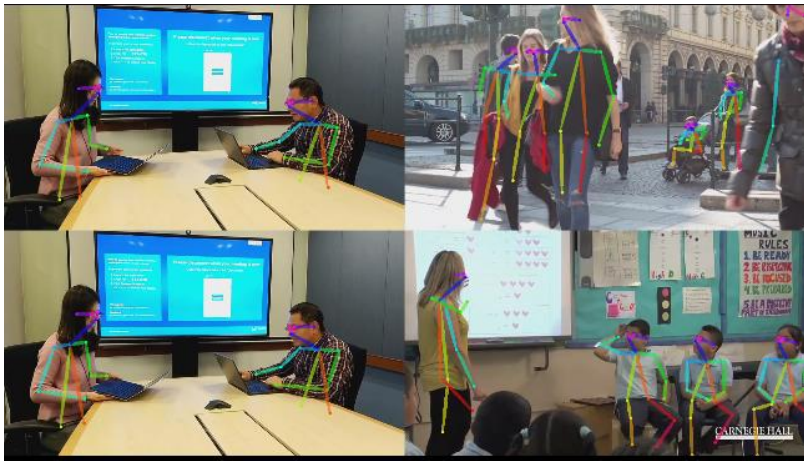
To use RTSP video stream instead of local video file, you can modify the par file and use RTSP url to replace local video file path.



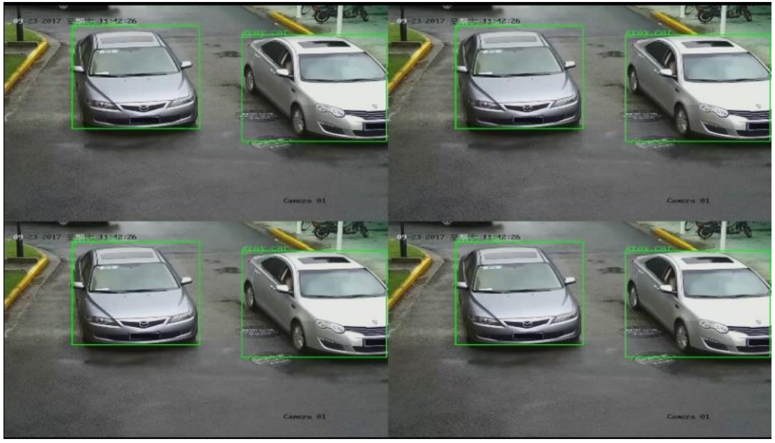
```
-i::h264 rtsp://username:password@172.16.181.169:554 -join -hw  -async 10 -
dec_postproc -threads 2   -o::sink -vpp_comp_dst_x 960 -vpp_comp_dst_y 0 -
vpp_comp_dst_w 480 -vpp_comp_dst_h 270 -ext_allocator -infer::fd ./model
```

You can modify the par file to change the position(-vpp_com_dst_x -vpp_comp_dst_y) and size(-vpp_comp_dst_w -vpp_comp_dst_h) of every video stream.

To run the 4 channel human pose estimation demo, please modify script run_face_detection_test.sh and use human_pose_estimation_1080p_4_channel.par as par file. Modify the video file paths in the par file firstly. Below picture is the screenshot of 4 channel human pose estimation demo.



To run the 4-channel vehicle demo, please modify script run_face_detection_test.sh and use vehical_detect_1080p_4_channel.par as par file. Modify the video file paths in the par file firstly. Below picture is the screenshot of 4 channel vehicle detection demo.


gray car
Camera 01