Setting up of NCS2 on Ubuntu and Windows

Step 1. Gather Your Equipment

Before you start, make sure you have the following:

- An x86-64 host computer with Windows 10® or Ubuntu*16.04 for the Intel® Distribution of OpenVINO™ Toolkit.
- Intel® Neural Compute Stick 2 (Intel® NCS 2).
- An internet connection to download and install the Intel® Distribution of OpenVINO™ toolkit.

Step 2. Meet the minimum requirements

A Linux build environment needs these components:
   a. OpenCV 3.4 or higher
   b. Intel® HD Graphics Driver (Graphics are required only if you use a GPU)
   c. GNU Compiler Collection (GCC) 3.4 or higher
   d. CMake* 2.8 or higher
   e. Python* 3.5 or higher

A Windows build environment needs these components:

   a. Intel® HD Graphics Driver (Graphics are required only if you use a GPU)
   b. OpenCV 3.4 or higher
   c. Intel® C++ Compiler 2017 Update 4
   d. CMake* 2.8 or higher
   e. Python* 3.5 or higher
   f. Visual Studio* 2015 or 2017

Step 3. Install the OpenVINO™ Toolkit

   a. Download the Intel® Distribution of OpenVINO™ toolkit package file:

      1. Windows - download the latest version..

      2. Linux - download the latest version..

b. Make sure that the file is saved to the Downloads directory w_openvino_toolkit_p_<version>.exe. [either windows 10 or linux]

c. Run the following codes for linux

```
cd ~/Downloads
tar xvf l_openvino_toolkit_<VERSION>.tgz
cd l_openvino_toolkit_<VERSION>
./install_GUI.sh
```

d. For windows Double click the downloaded **w_openvino_toolkit_<VERSION>.exe** file.

e. To install external software dependencies run :

```
cd /opt/intel/openvino/install_dependencies
sudo -E ./install_openvino_dependencies.sh
```

Step 4. Configure Neural Compute Stick USB Driver (Windows users can skip this step)

- Inorder to configure the USB driver on Linux run the following commands:

```
source ~/intel/openvino/bin/setupvars.sh
cd ~/intel/openvino/install_dependencies ./install_NCS_udev_rules.sh
```

Step 5. Install External Software Dependencies (Windows users can skip this step)

Various dependencies are needed for building the applications using OpenCV library and for the Deep Learning Inference Engines and Model Optimizer tools

1. Move to the `install_dependencies` directory:

```
cd /opt/intel/openvino/install_dependencies
```

2. Run a script to download and install the external software dependencies:

```
sudo -E ./install_openvino_dependencies.sh
```

Step 6. Set the Environment Variables

LINUX - We should be able to see [setupvars.sh] OpenVINO environment initialized after running the following commands.

```
source /opt/intel/openvino/bin/setupvars.s
```

WINDOWS - Run the below commands :

```
cd C:\Program Files (x86)\IntelSWTools\openvino\bin\
setupvars.bat
```

Step 7. Configure the Model Optimizer

LINUX

1. Go to the Model Optimizer prerequisites directory:

```
cd/opt/intel/openvino/deployment_tools/model_optimizer/install_prerequisites
```

2. Run the script to configure the Model Optimizer for Caffe, TensorFlow, MXNet, Kaldi*, and ONNX:

```
sudo ./install_prerequisites.sh
```

1. Open a command prompt. To do so, type cmd in your **Search Windows** box and then press **Enter**. Type commands in the opened window:
2. Go to the Model Optimizer prerequisites directory.

```
cd C:\Program Files
(x86)\IntelSWTools\openvino\deployment_tools\model_optimizer\install_prerequisites
```

3. Run the following batch file to configure the Model Optimizer for Caffe*, TensorFlow*, MXNet*, Kaldi*, and ONNX*:

```
install_prerequisites.bat
```

Step 8. Verify Installation of OpenVINO Tool kit

● We will first verify the installation and compile samples, run the verification applications provided with the product on the CPU (This provides the validation for the installation of OpenVINO Toolkit)

LINUX

1. Go to the **Inference Engine demo** directory:
2. Run the **Image Classification verification script**:

```
cd /opt/intel/openvino/deployment_tools/demo
./demo_squeezenet_download_convert_run.sh
```

WINDOWS

1. Open a command prompt window.
2. Go to the Inference Engine demo directory:

```
cd C:\Program Files (x86)\IntelSWTools\openvino\deployment_tools\de
```

3. Run the verification scripts by following the instructions :To run the script, start the `demo_squeezenet_download_convert_run.bat` file:

```
demo_squeezenet_download_convert_run.bat
```

Step 9. System check for NCS2

Linux

First make sure the USB rules are set up.

```
cat <<EOF > 97-myriad-usbboot.rules
SUBSYSTEM=="usb", ATTRS{idProduct}=="2150", ATTRS{idVendor}=="03e7", GROUP="users",
MODE="0666", ENV{ID_MM_DEVICE_IGNORE}="1"
SUBSYSTEM=="usb", ATTRS{idProduct}=="2485", ATTRS{idVendor}=="03e7", GROUP="users",
MODE="0666", ENV{ID_MM_DEVICE_IGNORE}="1"
SUBSYSTEM=="usb", ATTRS{idProduct}=="f63b", ATTRS{idVendor}=="03e7", GROUP="users",
MODE="0666", ENV{ID_MM_DEVICE_IGNORE}="1"
EOF


sudo cp 97-myriad-usbboot.rules /etc/udev/rules.d/

sudo udevadm control --reload-rules

sudo udevadm trigger
```

Then check if the device is visible with lsusb.

```
lsusb
```

If using NCS2,

```
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 015: ID 03e7:2485
```

Here ID 03e7:2150 without a description string is the Movidius device.

Step 10. Verify of NCS2 with OpenVINO Tool kit

Linux

```
cd ~/intel/openvino/deployment_tools/demo
./demo_security_barrier_camera.sh -d MYRIAD
```

Windows

```
C:
cd C:\"Program Files (x86)"\IntelSWTools\openvino\deployment_tools\demo
.\demo_security_barrier_camera.bat -d MYRIAD
```

References

[1] https://blog.codecentric.de/en/2017/10/object-detection-neural-compute-stick/

[2] https://github.com/intel-iot-devkit/smart-video-workshop

[3] https://hypraptive.github.io/2018/08/13/optimized-inference-edge.html

[4] https://movidius.github.io/ncsdk/docker.html

[5] https://www.freecodecamp.org/news/how-to-set-up-the-intel-movidius-neural-compute-stick-b9db16d493a7/

[6] https://medium.com/@hamza.manai.mail/how-did-i-used-intels-neural-compute-stick-from-scratch-a9737c9da7b8

[7] https://github.com/MrEliptik/Keras_to_TF_NCS2

[8] https://blog.victormeunier.com/keras_to_tf_ncs2.html

[9] https://thenewstack.io/a-closer-look-at-intel-movidius-neural-compute-stick/

[10] https://software.intel.com/en-us/articles/get-started-with-neural-compute-stick

[11] https://docs.openvinotoolkit.org/latest/_docs_install_guides_installing_openvino_linux.html