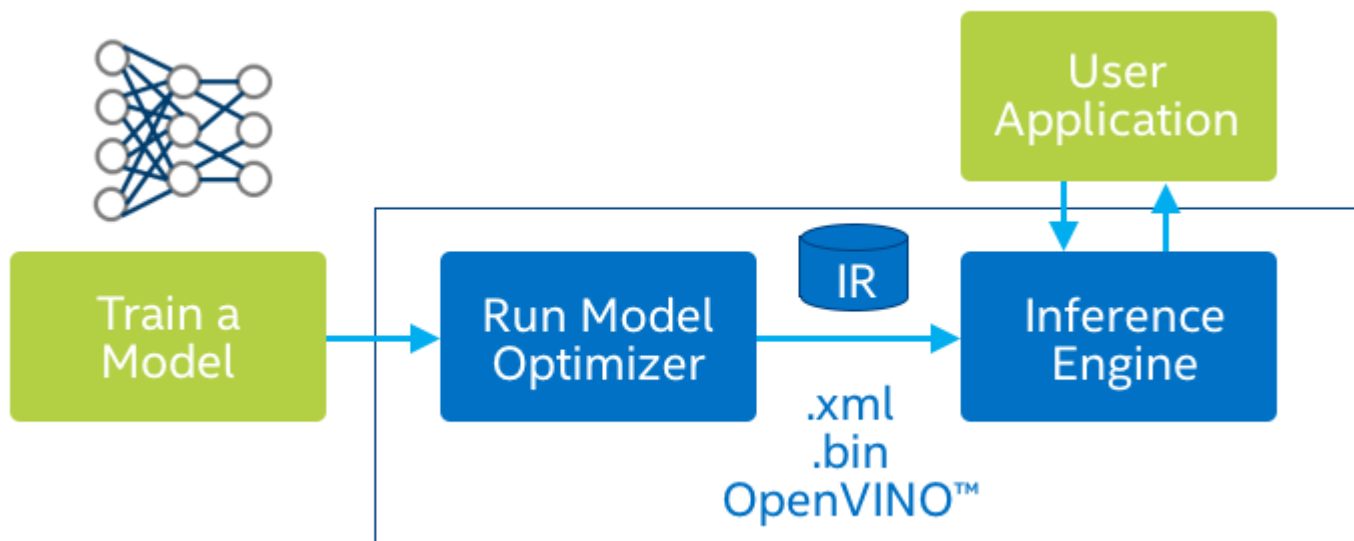


OpenVINO



Model floating-point format

	FP16	FP32
CPU		x
GPU (the one in the CPU)	x	x
NCS1 & NCS2	x	

USB rules

<https://software.intel.com/en-us/articles/OpenVINO-Install-Linux>

- Create vim 97-myrriad-usbboot.rules

```
cat <<EOF > 97-myrriad-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
```

- load rules

```
sudo udevadm control --reload-rules
```

```
sudo udevadm trigger
```

```
sudo ldconfig
```

Install drivers

<https://software.intel.com/en-us/articles/OpenVINO-Install-Linux#Install-Dependencies>

Optional:

<https://software.intel.com/en-us/articles/OpenVINO-Install-Linux#gpu-steps>

Pre-built face detection demo

- build samples

```
/opt/intel/computer_vision_sdk_2018.5.445/deployment_tools/inference_engine/samples/build_samples.sh
```

- face detection on CPU ~70FPS (i5-6300U)

```
~/inference_engine_samples/intel64/Release/interactive_face_detection_demo -m
```

```
/opt/intel/computer_vision_sdk_2018.5.445/deployment_tools/intel_models/face-detection-retail-0004/FP32/face-detection-retail-0004.xml
```

- face detection on NCS stick, ~14FPS on NCS1, ~44FPS on NCS2

```
~/inference_engine_samples/intel64/Release/interactive_face_detection_demo -m
```

```
/opt/intel/computer_vision_sdk_2018.5.445/deployment_tools/intel_models/face-detection-retail-0004/FP16/face-detection-retail-0004.xml -d MYRIAD
```

- face detection + head pose both on NCS

```
~/inference_engine_samples/intel64/Release/interactive_face_detection_demo -m
```

```
/opt/intel/computer_vision_sdk_2018.5.445/deployment_tools/intel_models/face-detection-retail-0004/FP16/face-detection-retail-0004.xml -d MYRIAD -m_hp /opt/intel/computer_vision_sdk_2018.5.445/deployment_tools/intel_models/head-pose-estimation-adas-0001/FP16/head-pose-estimation-adas-0001.xml -d_hp MYRIAD
```

Pre-built samples, face detection demo

- face detection + age, gender, both on NCS

```
~/inference_engine_samples/intel64/Release/interactive_face_detection_demo -m  
/opt/intel/computer_vision_sdk_2018.5.445/deployment_tools/intel_models/face-detection-retail-0004/FP16/face-detection-  
retail-0004.xml -d MYRIAD -m_ag /opt/intel/computer_vision_sdk_2018.5.445/deployment_tools/intel_models/age-gender-  
recognition-retail-0013/FP16/age-gender-recognition-retail-0013.xml -d_ag MYRIAD
```

- face detection + emotion, both on NCS

```
~/inference_engine_samples/intel64/Release/interactive_face_detection_demo -m  
/opt/intel/computer_vision_sdk_2018.5.445/deployment_tools/intel_models/face-detection-retail-0004/FP16/face-detection-  
retail-0004.xml -d MYRIAD -m_em /opt/intel/computer_vision_sdk_2018.5.445/deployment_tools/intel_models/emotions-  
recognition-retail-0003/FP16/emotions-recognition-retail-0003.xml -d_em MYRIAD
```

- face detection on NCS + landmark on CPU (on NCS2 there's a bug)

```
~/inference_engine_samples/intel64/Release/interactive_face_detection_demo -m  
/opt/intel/computer_vision_sdk_2018.5.445/deployment_tools/intel_models/face-detection-retail-0004/FP16/face-detection-  
retail-0004.xml -d MYRIAD -m_lm /opt/intel/computer_vision_sdk_2018.5.445/deployment_tools/intel_models/facial-  
landmarks-35-adas-0001/FP32/facial-landmarks-35-adas-0001.xml -d_lm CPU
```

Pre-built samples, Pose estimation

- pose estimation on NCS, ~3.6FPS on NCS2

```
~/inference_engine_samples/intel64/Release/human_pose_estimation_demo -m  
/opt/intel/computer_vision_sdk_2018.5.445/deployment_tools/intel_models/human-pose-estimation-0001/FP16/human-  
pose-estimation-0001.xml -d MYRIAD
```

- pose estimation on CPU, ~8.8FPS

```
~/inference_engine_samples/intel64/Release/human_pose_estimation_demo -m  
/opt/intel/computer_vision_sdk_2018.5.445/deployment_tools/intel_models/human-pose-estimation-0001/FP32/human-  
pose-estimation-0001.xml -d CPU
```

- pose estimation on GPU, FP32 ~10.3FPS

```
~/inference_engine_samples/intel64/Release/human_pose_estimation_demo -m  
/opt/intel/computer_vision_sdk_2018.5.445/deployment_tools/intel_models/human-pose-estimation-0001/FP32/human-  
pose-estimation-0001.xml -d GPU
```

- pose estimation on GPU, FP16, ~15.5FPS

```
~/inference_engine_samples/intel64/Release/human_pose_estimation_demo -m  
/opt/intel/computer_vision_sdk_2018.5.445/deployment_tools/intel_models/human-pose-estimation-0001/FP16/human-  
pose-estimation-0001.xml -d GPU
```


Visualize intermediate representation

repository <https://github.com/lutzroeder/netron>

online app <https://lutzroeder.github.io/netron/>

Compile your own network. GoogLeNet/InceptionV1

```
git clone https://github.com/tensorflow/models.git
```

```
wget http://download.tensorflow.org/models/inception\_v1\_2016\_08\_28.tar.gz
```

```
tar -xzf inception\_v1\_2016\_08\_28.tar.gz
```

- find mean values and scale from here <https://software.intel.com/en-us/articles/OpenVINO-Using-TensorFlow>

```
python3 /home/ntitus/github/models/research/slim/export_inference_graph.py --model_name=inception_v1 --  
output_file=inception_v1.pb
```

```
mo.py\
```

```
--input_model inception_v1.pb\
```

```
--input_checkpoint /home/ntitus/meetup/detection/output/inception_v1_2016_08_28/inception_v1.ckpt\
```

```
--input_shape [1,224,224,3]\
```

```
--data_type FP32\
```

```
--mean_values [127.5,127.5,127.5]\
```

```
--scale 127.5
```

```
~/inference_engine_samples/intel64/Release/classification_sample\
```

```
-m inception_v1.xml\
```

```
-d CPU\
```

```
-i cat.png
```

Compile your own network. Mobilenet v2 SSD

wget https://raw.githubusercontent.com/tensorflow/models/master/research/object_detection/samples/configs/ssd_mobilenet_v2_coco.config

wget http://download.tensorflow.org/models/object_detection/ssd_mobilenet_v2_coco_2018_03_29.tar.gz

tar -xvzf ssd_mobilenet_v2_coco_2018_03_29.tar.gz

<https://drive.google.com/open?id=1W81xyBmXQVFIEhFtIWteAclyzGy7BsJ6>

mo.py\

--input_model ssd_mobilenet_v2_coco_2018_03_29/frozen_inference_graph.pb\

--tensorflow_object_detection_api_pipeline_config ssd_mobilenet_v2_coco.config\

--tensorflow_use_custom_operations_config /opt/intel/computer_vision_sdk/deployment_tools/model_optimizer/extensions/front/tf/ssd_support.json\

--input_shape [1,360,640,3]\

--data_type FP16

(if you haven't already built the samples /opt/intel/computer_vision_sdk_2018.5.445/deployment_tools/inference_engine/samples/build_samples.sh
)

~/inference_engine_samples_build/intel64/Release/object_detection_demo_ssd_async\

-i Highway_Free_footage-cM1WpTC2Sp8.mp4\

-m frozen_inference_graph.xml\

-d MYRIAD -t 0.55

Useful links

FAQ <https://software.intel.com/en-us/articles/OpenVINO-ModelOptimizer#FAQ>

FORUM <https://software.intel.com/en-us/forums/computer-vision>

Tensorflow object detection zoo

https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md

where to get NCS2 <https://eu.mouser.com/new/Intel/intel-neural-compute-stick-2/>

Thanks DevPlant for hosting