LAB1- Prepare Environment

<Step 0 - Install>

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

<Step 1 - Run demo script>

```
#> source /opt/intel/computer_vision_sdk/bin/setupvars.sh
#> cd /opt/intel/computer_vision_sdk/deployment_tools/demo/
#> ./demo_squeezenet_download_convert_run.sh
```

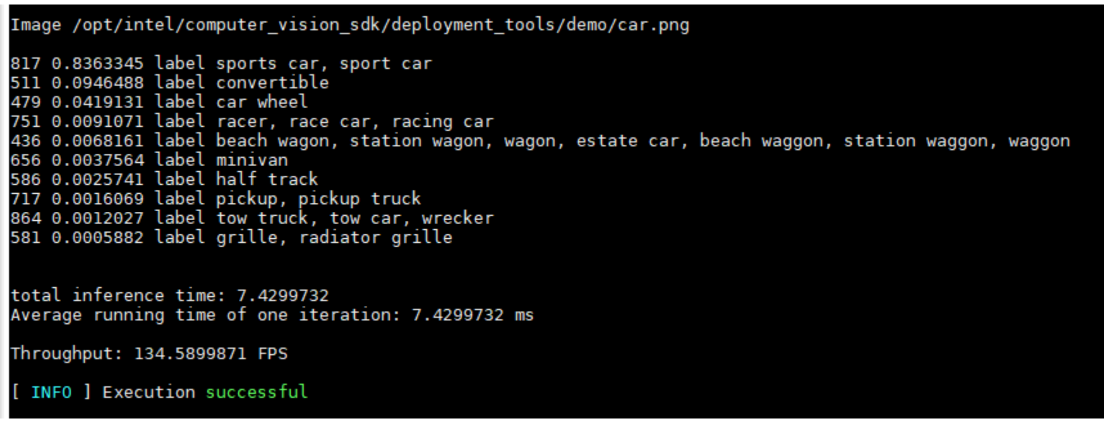
This script will

1. Download/ install needed packages
2. Download squeeznet into $home/openvino_models folder
3. run model optimizer to compile caffe model into intel IR format

```
/opt/intel/computer_vision_sdk_2018.5.445/deployment_tools/model_optimizer/mo.py --input_model
/root/openvino_models/models/FP32/classification/squeezenet/1.1/caffe/squeezenet1.1.caffemodel
/root/openvino_models/ir/FP32//classification/squeezenet/1.1/caffe --data_type FP3
```

4. Compile and Run and classification sample for card

Result -



```
Image /opt/intel/computer_vision_sdk/deployment_tools/demo/car.png

817 0.8363345 label sports car, sport car
511 0.0946488 label convertible
479 0.0419131 label car wheel
751 0.0091071 label racer, race car, racing car
436 0.0068161 label beach wagon, station wagon, wagon, estate car, beach waggon, station waggon, waggon
656 0.0037564 label minivan
586 0.0025741 label half track
717 0.0016069 label pickup, pickup truck
864 0.0012027 label tow truck, tow car, wrecker
581 0.0005882 label grille, radiator grille


total inference time: 7.4299732
Average running time of one iteration: 7.4299732 ms

Throughput: 134.5899871 FPS

[ INFO ] Execution successful
```

<Step 2 - Following Steps>

After exection, there's two folder appear in $HOME. (1) Inference_engine_simpile (2) openvino_models

Please use other images to run 'classification_simple'

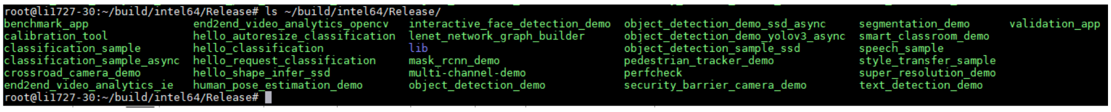
LAB2 – Run Intel Pretrained models

<Step1 build Sample Code>

```
#> mkdir ~/build/
#> cd ~/build/
#> cmake /opt/intel/computer_vision_sdk/deployment_tools/inference_engine/samples/
#> make
```

<Step2 – check the result>

```
#> ls ~/build/intel64/build/
```



```
root@li1727-30:~/build/intel64/Release# ls ~/build/intel64/Release/
benchmark_app                  end2end_video_analytics_opencv   interactive_face_detection_demo  object_detection_demo_ssd_async     segmentation_demo        validation_app
calibration_tool               hello_autoresize_classification  lenet_network_graph_builder      object_detection_demo_yolov3_async  smart_classroom_demo
classification_sample          hello_classification             lib                              object_detection_sample_ssd         speech_sample
classification_sample_async    hello_request_classification     mask_rcnn_demo                   pedestrian_tracker_demo             style_transfer_sample
crossroad_camera_demo          hello_shape_infer_ssd            multi-channel-demo               perfcheck                           super_resolution_demo
end2end_video_analytics_ie     human_pose_estimation_demo       object_detection_demo            security_barrier_camera_demo        text_detection_demo
root@li1727-30:~/build/intel64/Release# 
```

<Step3 execute a command>

```
#> cd ~/build/intel64/Release/
#> ./object_detection_demo_ssd_async -d CPU -m /opt/intel/computer_vision_sdk/deployment_tools/intel_models/face-detection-retail-0004/FP32/face-detection-retail-0004.xml -i ~/Movie/face.mp4
#> ./interactive_face_detection_demo -d CPU -m /opt/intel/computer_vision_sdk/deployment_tools/intel_models/face-detection-retail-0004/FP32/face-detection-retail-0004.xml -i ~/Movie/face.mp4
#> ./interactive_face_detection_demo -d CPU -m /opt/intel/computer_vision_sdk/deployment_tools/intel_models/face-detection-retail-0004/FP32/face-detection-retail-0004.xml -m_ag /opt/intel/computer_vision_sdk/deployment_tools/intel_models/age-gender-recognition-retail-0013/FP32/age-gender-recognition-retail-0013.xml -m_hp /opt/intel/computer_vision_sdk/deployment_tools//intel_models/head-pose-estimation-adas-0001/FP32/head-pose-estimation-adas-0001.xml -m_em /opt/intel/computer_vision_sdk/deployment_tools/intel_models/emotions-recognition-retail-0003/FP32/emotions-recognition-retail-0003.xml -i ~/Movie/face.mp4
```

<step4 Following Study>

Goto the online doc - https://software.intel.com/en-us/articles/OpenVINO-IE-Samples and check how to use IE sample code

LAB3 – Run a real Example

<step 1 choose a model >

Please check OpenVINO online documents https://software.intel.com/en-us/articles/OpenVINO-ModelOptimizer or use the model from tensorflow/ caffe or the model which you're using now.

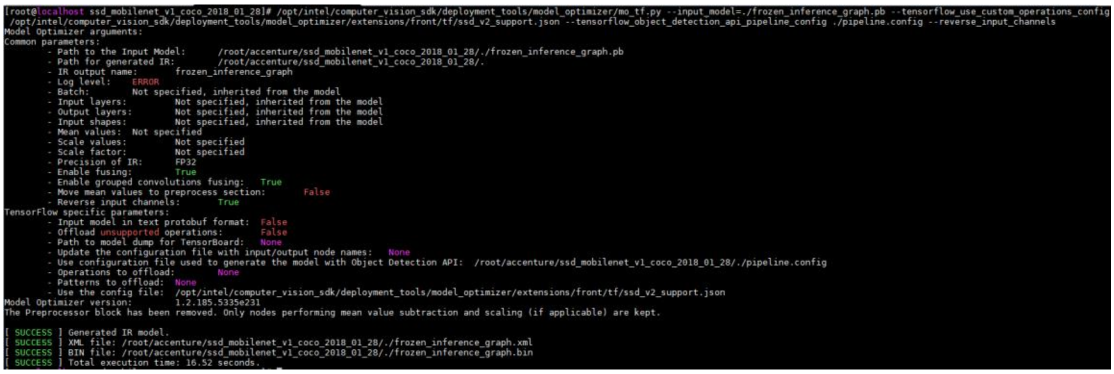
<step 2 Convert by MO / example commands for MO>

```
#> cd /opt/intel/computer_vision_sdk/deployment_tools/model_optimizer/install_prerequisites/ && ./install_prerequisites_tf.sh && cd ~
```

```
#> wget http://download.tensorflow.org/models/object_detection/ssd_mobilenet_v1_coco_2018_01_28.tar.gz
```

```
#> tar zxvf ssd_mobilenet_v1_coco_2018_01_28.tar.gz && cd ssd_mobilenet_v1_coco_2018_01_28
```

```
#> /opt/intel/computer_vision_sdk/deployment_tools/model_optimizer/mo_tf.py --input_model=./frozen_inference_graph.pb --tensorflow_use_custom_operations_config /opt/intel/computer_vision_sdk/deployment_tools/model_optimizer/extensions/front/tf/ssd_v2_support.json --tensorflow_object_detection_api_pipeline_config ./pipeline.config --reverse_input_channels
```



```
[root@localhost ssd_mobilenet_v1_coco_2018_01_28]# /opt/intel/computer_vision_sdk/deployment_tools/model_optimizer/mo_tf.py --input_model=./frozen_inference_graph.pb --tensorflow_use_custom_operations_config
 /opt/intel/computer_vision_sdk/deployment_tools/model_optimizer/extensions/front/tf/ssd_v2_support.json --tensorflow_object_detection_api_pipeline_config ./pipeline.config --reverse_input_channels
Model Optimizer arguments:
Common parameters:
        - Path to the Input Model:      /root/accenture/ssd_mobilenet_v1_coco_2018_01_28/./frozen_inference_graph.pb
        - Path for generated IR:        /root/accenture/ssd_mobilenet_v1_coco_2018_01_28/.
        - IR output name:       frozen_inference_graph
        - Log level:    ERROR
        - Batch:        Not specified, inherited from the model
        - Input layers:         Not specified, inherited from the model
        - Output layers:        Not specified, inherited from the model
        - Input shapes:         Not specified, inherited from the model
        - Mean values:  Not specified
        - Scale values:         Not specified
        - Scale factor:         Not specified
        - Precision of IR:      FP32
        - Enable fusing:        True
        - Enable grouped convolutions fusing:   True
        - Move mean values to preprocess section:       False
        - Reverse input channels:       True
TensorFlow specific parameters:
        - Input model in text protobuf format:  False
        - Offload unsupported operations:       False
        - Path to model dump for TensorBoard:   None
        - Update the configuration file with input/output node names:   None
        - Use configuration file used to generate the model with Object Detection API:  /root/accenture/ssd_mobilenet_v1_coco_2018_01_28/./pipeline.config
        - Operations to offload:        None
        - Patterns to offload:  None
        - Use the config file:  /opt/intel/computer_vision_sdk/deployment_tools/model_optimizer/extensions/front/tf/ssd_v2_support.json
Model Optimizer version:        1.2.185.5335e231
The Preprocessor block has been removed. Only nodes performing mean value subtraction and scaling (if applicable) are kept.

[ SUCCESS ] Generated IR model.
[ SUCCESS ] XML file: /root/accenture/ssd_mobilenet_v1_coco_2018_01_28/./frozen_inference_graph.xml
[ SUCCESS ] BIN file: /root/accenture/ssd_mobilenet_v1_coco_2018_01_28/./frozen_inference_graph.bin
[ SUCCESS ] Total execution time: 16.52 seconds.
```

<step 2 Run IE / example commands for IE>

```
#> ~/build/intel64/Release/object_detection_demo_ssd_async -d CPU -m ./frozen_inference_graph.xml -i ~/Movie/face.mp4
```

LAB4 – Run it on FPGA

Use the server in the classroom the test the model and commands in Lab1/2/3 with hetero plugin