**Hands-on Lab1-1：**

**Inference Engine Classification Example**

**Step 1: Log into your laptops**

Username: **intel**

Password: **intel123**

**Step 2: Run Model Optimizer against DL model**

1. **Set the environmental variables**

$ export SV=/opt/intel/workshop/smart-video-workshop/

$ source /opt/intel/computer\_vision\_sdk/bin/setupvars.sh

[setupvars.sh] OpenVINO environment initialized

1. **Convert classification model to IR using Model Optimizer**

$ cd $SV/object-detection/

$ mkdir -p squeezenet1.1/FP32

$ cd /opt/intel/computer\_vision\_sdk/deployment\_tools/model\_optimizer

$ python3 mo\_caffe.py --input\_model /opt/intel/computer\_vision\_sdk/deployment\_tools/model\_downloader/classification/squeezenet/1.1/caffe/squeezenet1.1.caffemodel -o $SV/object-detection/squeezenet1.1/FP32

Model Optimizer arguments:

Common parameters:

- Path to the Input Model: /opt/intel/computer\_vision\_sdk/deployment\_tools/model\_downloader/classification/squeezenet/1.1/caffe/squeezenet1.1.caffemodel

- Path for generated IR: /opt/intel/workshop/smart-video-workshop//object-detection/squeezenet1.1/FP32

- IR output name: squeezenet1.1

- 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: False

Caffe specific parameters:

- Enable resnet optimization: True

- Path to the Input prototxt: /opt/intel/computer\_vision\_sdk/deployment\_tools/model\_downloader/classification/squeezenet/1.1/caffe/squeezenet1.1.prototxt

- Path to CustomLayersMapping.xml: Default

- Path to a mean file: Not specified

- Offsets for a mean file: Not specified

Model Optimizer version: 1.2.110.59f62983

[ SUCCESS ] Generated IR model.

[ SUCCESS ] XML file: /opt/intel/workshop/smart-video-workshop//object-detection/squeezenet1.1/FP32/squeezenet1.1.xml

[ SUCCESS ] BIN file: /opt/intel/workshop/smart-video-workshop//object-detection/squeezenet1.1/FP32/squeezenet1.1.bin

[ SUCCESS ] Total execution time: 0.55 seconds.

1. **Navigate to the sample model directory and check outputs**

$ cd $SV/object-detection/squeezenet1.1/FP32

$ ll

ttotal 4884

drwxrwxr-x 2 filly filly 4096 Sep 3 13:26 ./

drwxrwxr-x 3 filly filly 4096 Sep 3 13:26 ../

-rw-rw-r-- 1 filly filly 4941984 Sep 3 13:26 **squeezenet1.1.bin**

-rw-rw-r-- 1 filly filly 9081 Sep 3 13:26 squeezenet1.1.mapping

-rw-rw-r-- 1 filly filly 36745 Sep 3 13:26 **squeezenet1.1.xml**

1. **Copy label file to working directory**

$ cp /opt/intel/computer\_vision\_sdk/deployment\_tools/demo/squeezenet1.1.labels $SV/object-detection/squeezenet1.1/FP32

**Step 3: Use Inference Engine in the application**

1. **Open the sample app (main.cpp) in the editor of your choice to view the lines that call the Inference Engine.**

$ gedit /opt/intel/computer\_vision\_sdk/inference\_engine/samples/classification\_sample/main.cpp

* Line 79 — loads the Inference Engine plugin for use within the application
* Line 114 — initializes the network object
* Line 203 — loads model to the plugin
* Line 216 — Prepare input
* Line 243 — runs inference using the optimized model
* Line 269 — allocate output blobs

1. **Run the classification sample application to use the Inference Engine on the test pic**

$ cd $SV/object-detection

$ ./lab.py classification\_sample -i car.png -m squeezenet1.1

#########

/opt/intel/computer\_vision\_sdk/deployment\_tools/inference\_engine/samples/build/intel64/Release/classification\_sample -i /opt/intel/computer\_vision\_sdk/deployment\_tools/demo/car.png -m /opt/intel/workshop/smart-video-workshop/object-detection/squeezenet1.1/FP32/squeezenet1.1.xml -d CPU

#########

hit enter to run the above command...

[ INFO ] InferenceEngine:

API version ............ 1.1

Build .................. 12419

[ INFO ] Parsing input parameters

[ INFO ] Loading plugin

API version ............ 1.1

Build .................. lnx\_20180510

Description ....... MKLDNNPlugin

[ INFO ] Loading network files:

/opt/intel/workshop/smart-video-workshop/object-detection/squeezenet1.1/FP32/squeezenet1.1.xml

/opt/intel/workshop/smart-video-workshop/object-detection/squeezenet1.1/FP32/squeezenet1.1.bin

[ INFO ] Preparing input blobs

[ WARNING ] Image is resized from (787, 259) to (227, 227)

[ INFO ] Batch size is 1

[ INFO ] Preparing output blobs

[ INFO ] Loading model to the plugin

[ INFO ] Starting inference (1 iterations)

[ INFO ] Average running time of one iteration: 5.11079 ms

[ INFO ] Processing output blobs

Top 10 results:

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

[ INFO ] Execution successful



1. **Run the example on different hardware ( i.e. CPU and GPU )**

$ ./lab.py classification\_sample -i car.png -m squeezenet1.1 -d GPU

#########

/opt/intel/computer\_vision\_sdk/deployment\_tools/inference\_engine/samples/build/intel64/Release/classification\_sample -i /opt/intel/computer\_vision\_sdk/deployment\_tools/demo/car.png -m /opt/intel/workshop/smart-video-workshop/object-detection/squeezenet1.1/FP32/squeezenet1.1.xml -d GPU

#########

hit enter to run the above command...

[ INFO ] InferenceEngine:

API version ............ 1.1

Build .................. 12419

[ INFO ] Parsing input parameters

[ INFO ] Loading plugin

API version ............ 1.1

Build .................. ci-main-03703

Description ....... clDNNPlugin

[ INFO ] Loading network files:

/opt/intel/computer\_vision\_sdk/deployment\_tools/demo/ir/squeezenet1.1/squeezenet1.1.xml

/opt/intel/computer\_vision\_sdk/deployment\_tools/demo/ir/squeezenet1.1/squeezenet1.1.bin

[ INFO ] Preparing input blobs

[ WARNING ] Image is resized from (787, 259) to (227, 227)

[ INFO ] Batch size is 1

[ INFO ] Preparing output blobs

[ INFO ] Loading model to the plugin

[ INFO ] Starting inference (1 iterations)

[ INFO ] Average running time of one iteration: 5.75039 ms

[ INFO ] Processing output blobs

Top 10 results:

Image /opt/intel/computer\_vision\_sdk/deployment\_tools/demo/car.png

817 0.8363329 label sports car, sport car

511 0.0946493 label convertible

479 0.0419136 label car wheel

751 0.0091072 label racer, race car, racing car

436 0.0068162 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

[ INFO ] Execution successful

**Hands-on Lab1-2：**

**Interactive Face Detection Sample with Intel Models**

**Step 1: Verify the IR files in model directory**

$ cd /opt/intel/computer\_vision\_sdk/deployment\_tools/intel\_models/face-detection-retail-0004/FP32

$ ll

drwxr-xr-x 2 root root 4096 Aug 2 12:23 ./

drwxr-xr-x 5 root root 4096 Aug 2 12:23 ../

-rw-rw-r-- 1 root root 2352880 Aug 2 12:23 **face-detection-retail-0004.bin**

-rw-rw-r-- 1 root root 48040 Aug 2 12:23 **face-detection-retail-0004.xml**

**Step 2: Check the camera available**

ls /dev/video\*

/dev/video0

**Step 2: Run the application with this model**

1. **Go to lab runner directory**

$ cd $SV/object-detection

1. **Run interactive\_face\_detection\_sample**

$ ./lab.py interactive\_face\_detection\_sample -i /dev/video0 -m face-detection-retail-0004

using camera stream for infer: /dev/video0

#########

/opt/intel/computer\_vision\_sdk/deployment\_tools/inference\_engine/samples/build/intel64/Release/interactive\_face\_detection\_sample -i /dev/video0 -m /opt/intel/computer\_vision\_sdk/deployment\_tools/intel\_models/face-detection-retail-0004/FP32/face-detection-retail-0004.xml -d CPU

#########

hit enter to run the above command...

InferenceEngine:

API version ............ 1.1

Build .................. 12419

[ INFO ] Parsing input parameters

[ INFO ] Reading input

[ INFO ] Loading plugin CPU

API version ............ 1.1

Build .................. lnx\_20180510

Description ....... MKLDNNPlugin

[ INFO ] Loading network files for Face Detection

[ INFO ] Batch size is set to 1

[ INFO ] Checking Face Detection inputs

[ INFO ] Checking Face Detection outputs

[ INFO ] Loading Face Detection model to the CPU plugin

[ INFO ] Age Gender DISABLED

[ INFO ] Head Pose DISABLED

[ INFO ] Emotions Recognition DISABLED

[ INFO ] Start inference

[ INFO ] Press any key to stop

