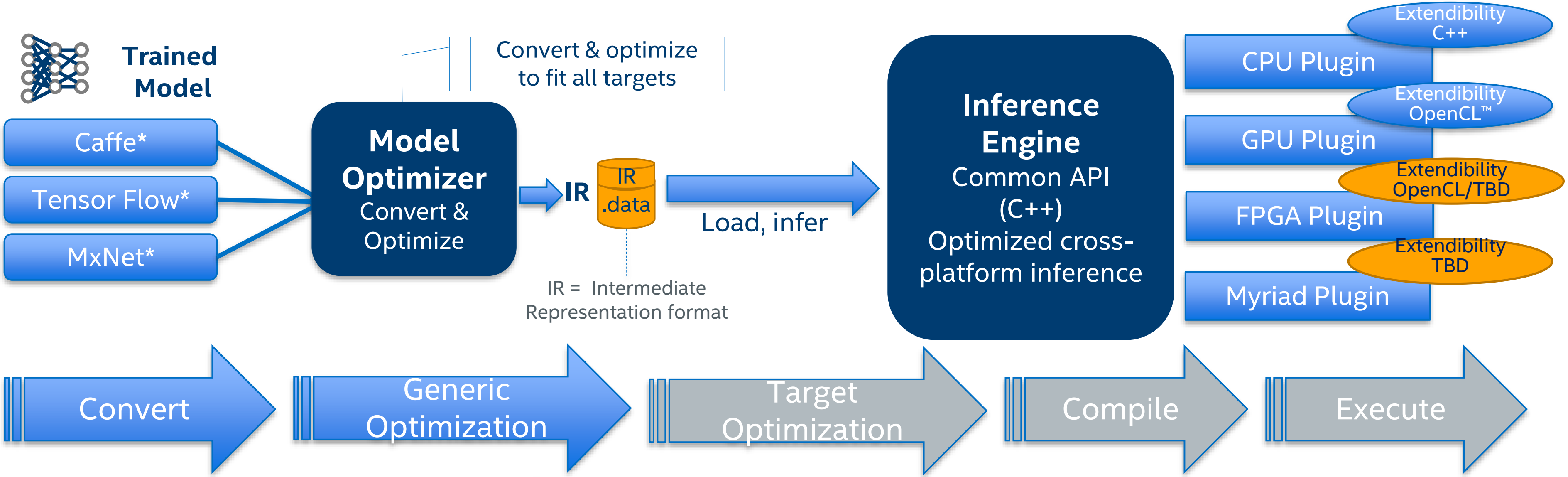
OpenVINO: ModelOptimizer

Maple Chou / 2019.01

DLDK

Model Optimizer

- **What it is**: Preparation step -> imports trained models
- **Why important**: Optimizes for performance/space with conservative topology transformations; biggest boost is from conversion to data types matching hardware.



Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

The things you must to know

- Model Optimizer is offline tool
  - You can execute mo in your laptop/ server or anywhere
- Model Optimizer generate .xml and .bin
  - .xml = topology (IR)
  - .bin = weight (matrix/filter)
- After MO
  - You can use .xml/.bin to inference (Inference Engine)

Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

Model Optimizer

Model Optimizer is a cross platform tool written in Python

- Python 3.5 or higher
- Pip packages:
  - Mxnet
  - Tensorflow
  - Protobuff
  - Etc. Check list of required packages in requirements.txt

Example usage

- python mo.py --framework tf --input_model /models/optimized.pb

Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

Model Optimizer Purposes

Convert

- Map Framework (FW) specific model format to unified IR format
- IR format is DLDT serialization format that consist of two files:
  - XML file for topology description (human-readable)
  - BIN file for weights
- There is *NO* one-to-one correspondence between every framework layer and some IR layer
- Need for framework-specific translation techniques (easy for Caffe, hard for TensorFlow)

Optimize

- Hardware independent optimization
- No need to implement similar optimization techniques in each HW plug-in inside IE
- Frequently model conversion means optimization: TensorFlow patterns

Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

Model Optimizer stages

Load

- Parse a FW specific model. Original FW may or may not be used.
- Build NetworkX graph for further transformations

Front

- Decode FW specific attributes to represent them in a unified way
- Replace FW specific patterns to represent them with unified set of operations

Middle

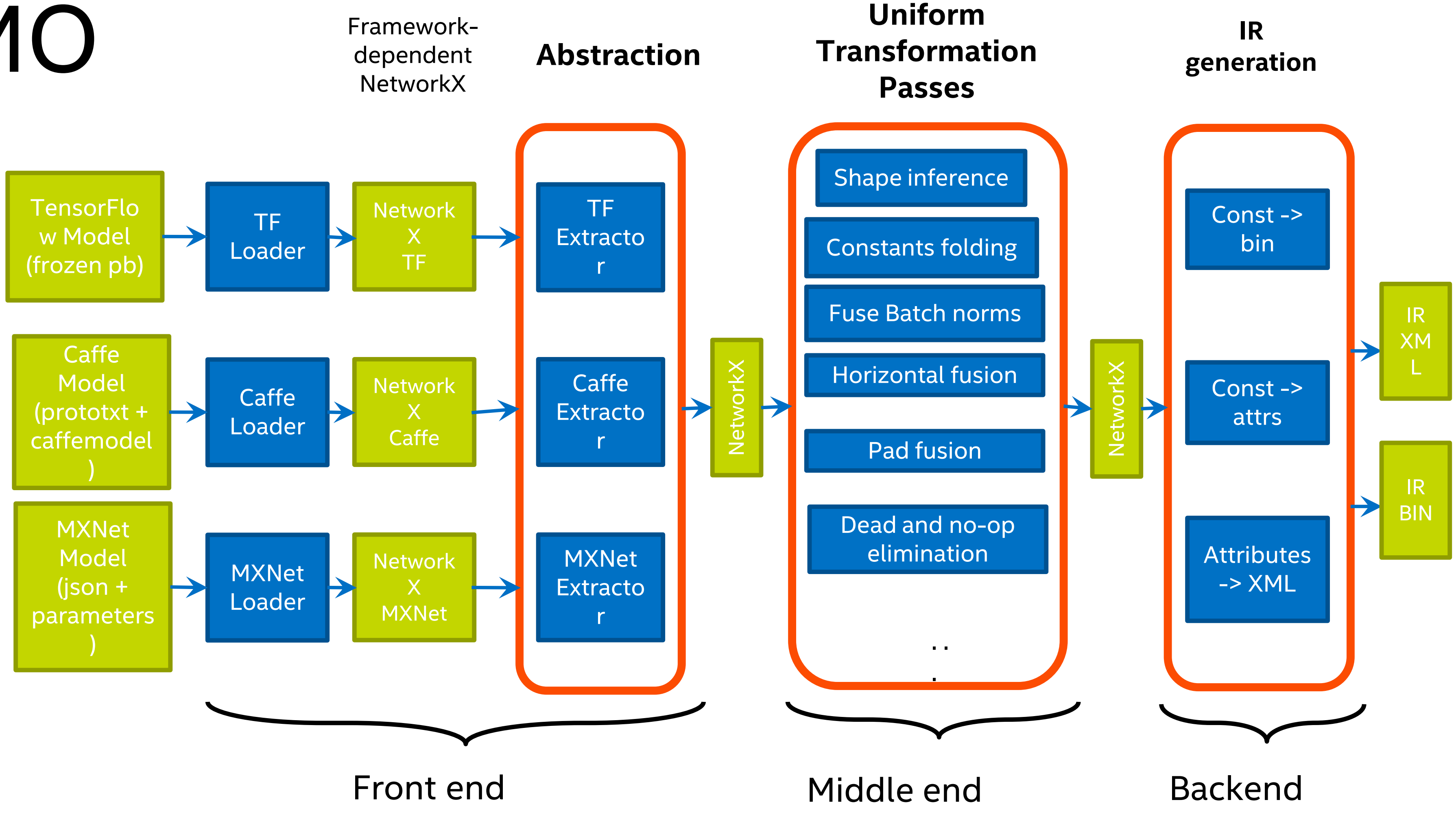
- Calculate and propagate shapes
- Transform graph to leave only ops that are supported by the target IR format
- **Optimize**: propagate constants, fuse operations, eliminate dead parts and ops that don’t have effect

Back

- Finalize graph transformation to completely fit to IR requirements
- Emit final XML and BIN files

Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

MO



Framework-dependent NetworkX

Abstraction

Uniform Transformation Passes

IR generation

TensorFlow Model (frozen pb)

TF Loader

Network X TF

TF Extractor

Caffe Model (prototxt + caffemodel)

Caffe Loader

Network X Caffe

Caffe Extractor

MXNet Model (json + parameters)

MXNet Loader

Network X MXNet

MXNet Extractor

NetworkX

Shape inference

Constants folding

Fuse Batch norms

Horizontal fusion

Pad fusion

Dead and no-op elimination

. . .

NetworkX

Const -> bin

Const -> attrs

Attributes -> XML

IR XML

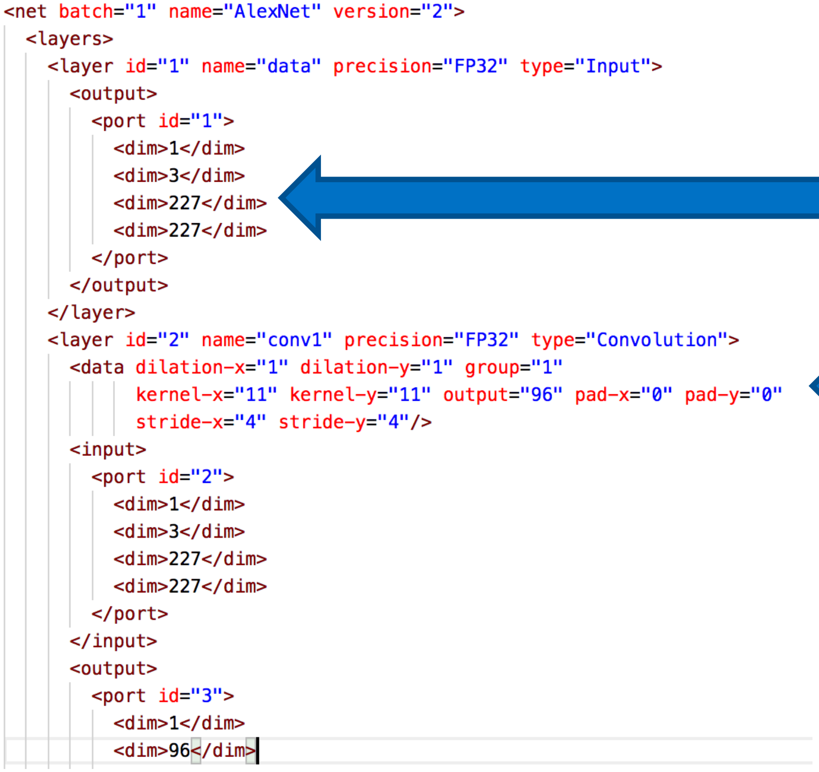
IR BIN

Front end

Middle end

Backend

XML



```xml
<net batch="1" name="AlexNet" version="2">
  <layers>
    <layer id="1" name="data" precision="FP32" type="Input">
      <output>
        <port id="1">
          <dim>1</dim>
          <dim>3</dim>
          <dim>227</dim>
          <dim>227</dim>
        </port>
      </output>
    </layer>
    <layer id="2" name="conv1" precision="FP32" type="Convolution">
      <data dilation-x="1" dilation-y="1" group="1"
            kernel-x="11" kernel-y="11" output="96" pad-x="0" pad-y="0"
            stride-x="4" stride-y="4"/>
      <input>
        <port id="2">
          <dim>1</dim>
          <dim>3</dim>
          <dim>227</dim>
          <dim>227</dim>
        </port>
      </input>
      <output>
        <port id="3">
          <dim>1</dim>
          <dim>96</dim>
```

Input Dimensions

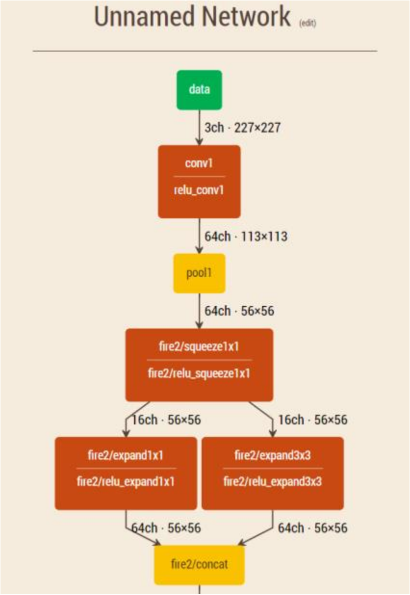
Convolution Parameters

Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

IR Visualizer

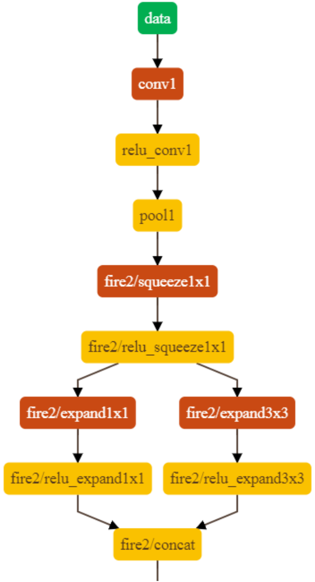
NetScope – CNN Visualizer for Caffe

https://dgschwend.github.io/netscope/quickstart.html



IR Visualizer – still **INTERNAL PREVIEW ONLY, not in public product**

http://goto/cvsdk-ir

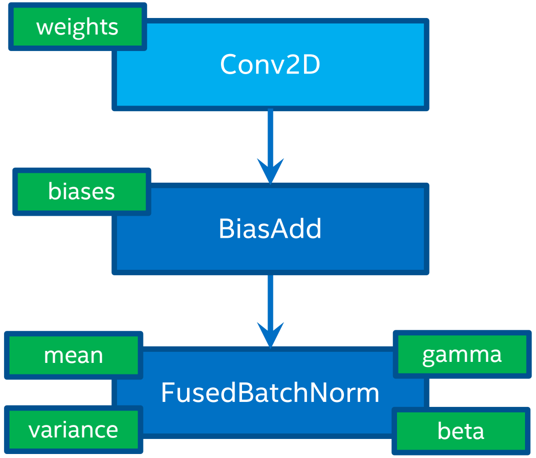


Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

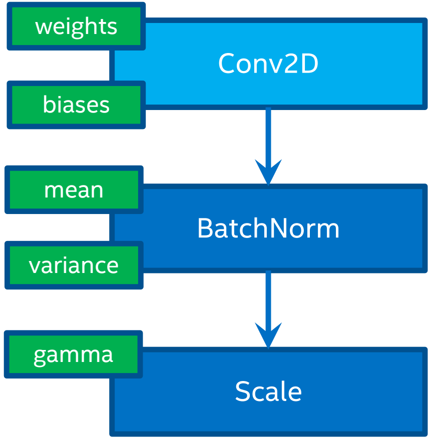
Transformation Examples

Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

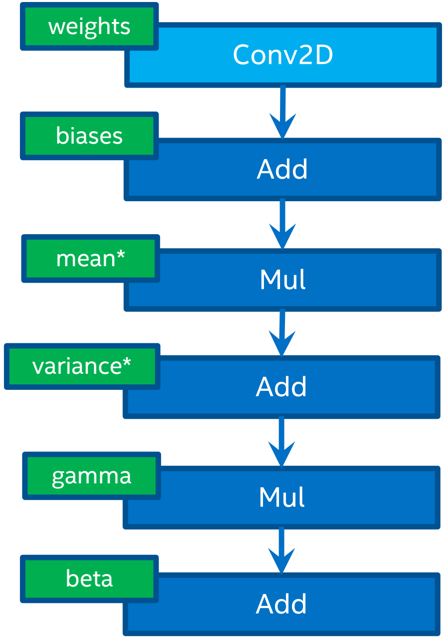
Example: Batch normalization fusion (TF)



TensorFlow



Caffe



TensorFlow
(before 1.5)

Optimization Notice

Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

Intel® DL Deployment Toolkit Functionality

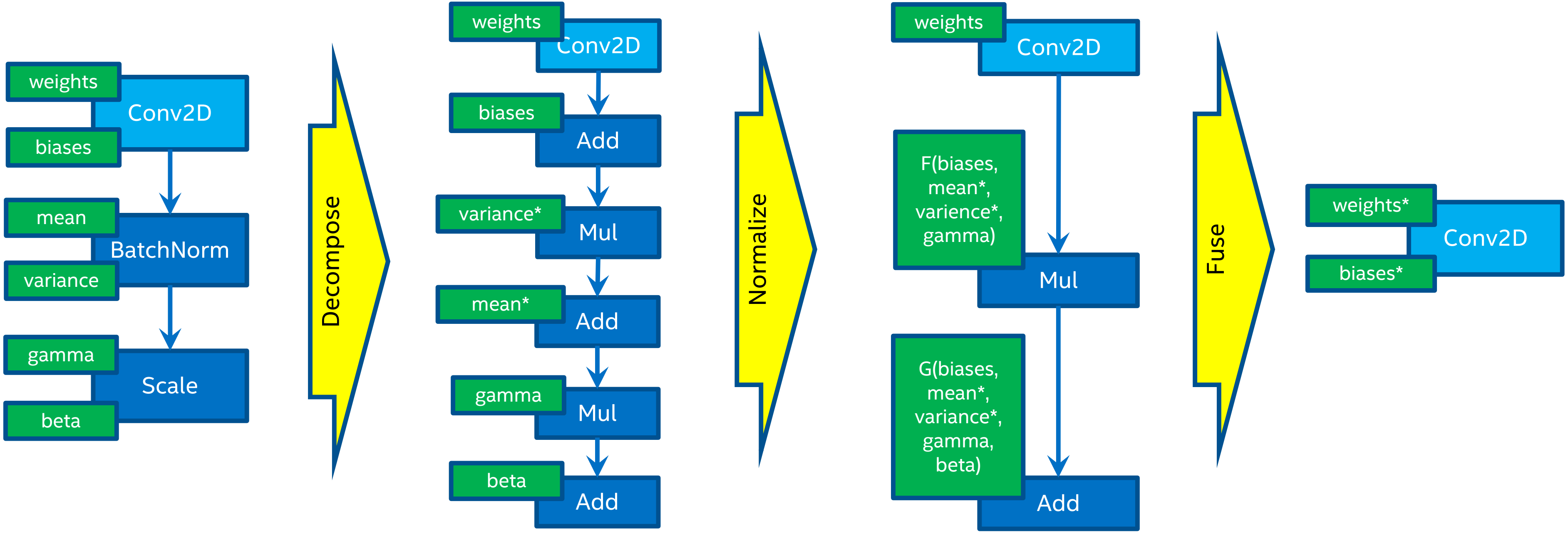
Model Optimizer/Optimization Tech

- Linear Operation Fusing
  - **BatchNorm and ScaleShift decomposition:** *BN* layers decomposes to *Mul->Add->Mul->Add* sequence; ScaleShift layers decomposes to *Mul->Add* sequence.
  - **Linear operations merge:** Merges sequences of Mul and Add operations to the **single** Mul->Add instance.
  - **Linear operations fusion:** Fuses Mul and Add operations to Convolution or FullybConnected layers.
- Grouped Convolutions Fusing
  - Specific optimization that applies for TensorFlow* topologies. (Xception*)

Optimization Notice

Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

Decompose, normalize and fuse



Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

Batch normalization fusion: Results

Model	CPU FP32, FPS		GPU FP16, FPS		CPU Speedup %	GPU Speedup %
	No fusion	Fusion	No fusion	Fusion		
Alexnet	76	76	137	137	0.0	0.0
TF Inception v1	86	99	59	105	15.4	77.0
TF Inception v2	66	76	50	78	13.9	57.3
TF InceptionResnet v2	9	12	7	9	30.0	23.8
TF Resnet v2 50	36	43	28	41	20.3	48.4
TF MobileNet v1 1_1_0_224	190	231	67	137	21.9	102.8
TF MobileNet v2 1_1_0_224	213	338	52	109	58.6	111.4
TF SSD MobileNet v1	94	97	47	59	3.6	24.8
TF SSD Inception V2	30	30	33	33	0.0	0.0
Caffe SqueezeNet v1.1	352	352	214	239	0.0	11.7

Discuss horiz fusion,

- example Inception V2
- “Split” is not good FPGA

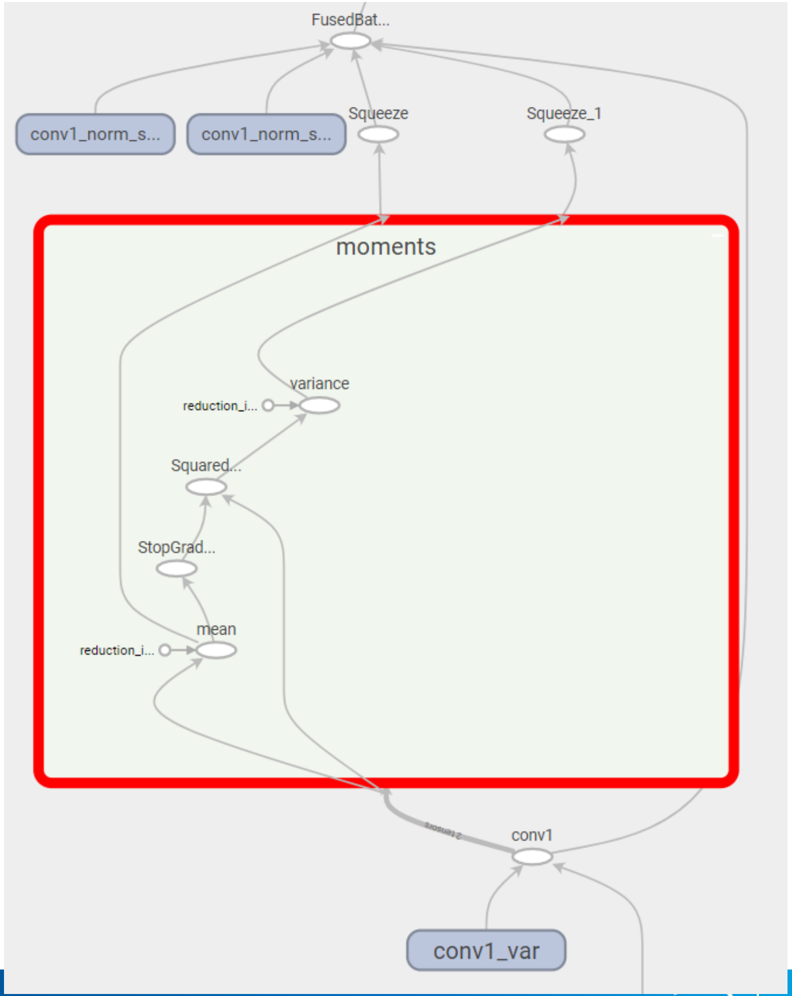
Optimization Notice

Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

MVN translation alternatives

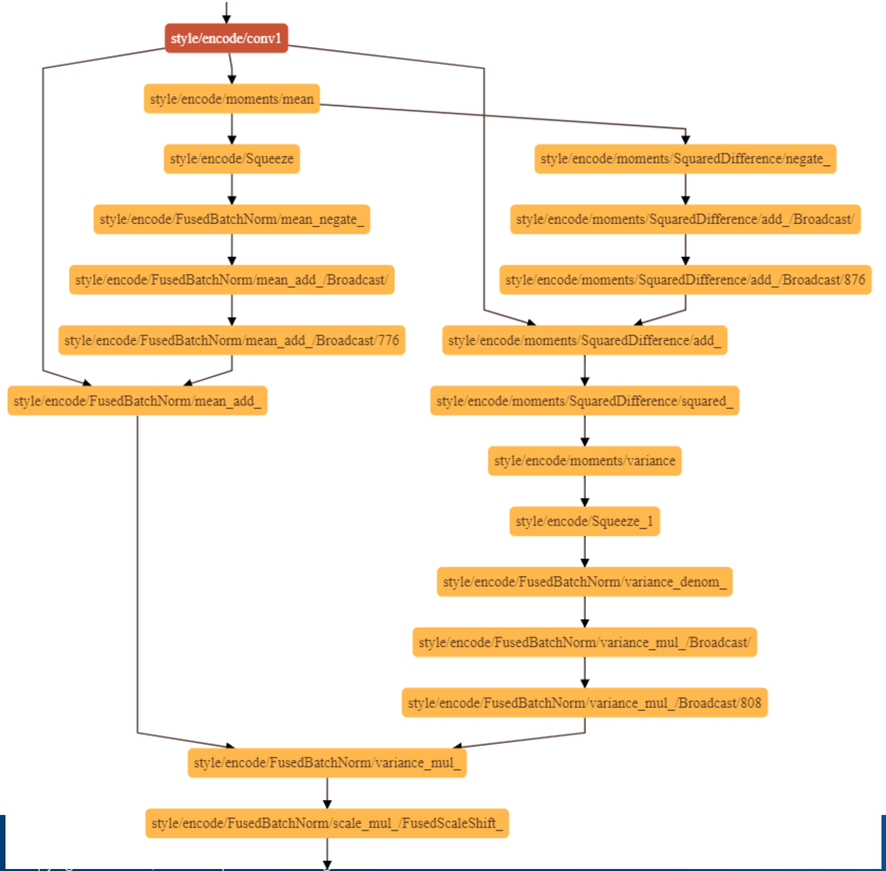
Two options:

- Translate op-by-op literally expressing a sub-graph in IR
  - Pros: simple, always work
  - Cons: Mean/SquaredDifference have no direct mapping to a single IE layer; many layers – poor performance
- Recognize MVN pattern and replace by a single IR layer
  - Pros: performance!
  - Cons: fragile; works only when exactly matches



Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

Not optimized



style/encode/conv1

style/encode/moments/mean

style/encode/Squeeze

style/encode/FusedBatchNorm/mean_negate_

style/encode/FusedBatchNorm/mean_add_/Broadcast/

style/encode/FusedBatchNorm/mean_add_/Broadcast/776

style/encode/FusedBatchNorm/mean_add_

style/encode/moments/SquaredDifference/negate_

style/encode/moments/SquaredDifference/add_/Broadcast/

style/encode/moments/SquaredDifference/add_/Broadcast/876

style/encode/moments/SquaredDifference/add_

style/encode/moments/SquaredDifference/squared_

style/encode/moments/variance

style/encode/Squeeze_1

style/encode/FusedBatchNorm/variance_denom_

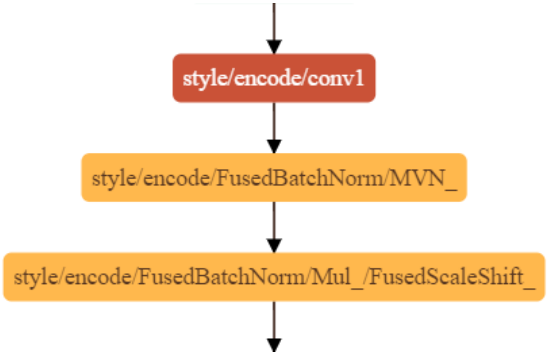
style/encode/FusedBatchNorm/variance_mul_/Broadcast/

style/encode/FusedBatchNorm/variance_mul_/Broadcast/808

style/encode/FusedBatchNorm/variance_mul_

style/encode/FusedBatchNorm/scale_mul_/FusedScaleShift_

Optimized



style/encode/conv1

style/encode/FusedBatchNorm/MVN_

style/encode/FusedBatchNorm/Mul_/FusedScaleShift_

Model	CPU FP32, ms		CPU Speedup %
	Base	Optimized	
Cyberlink Style Transfer	1252	504	148.4

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

MO EXAMPLES

Use MO help and learn about first MO mistake: using python 2.7

```
$ python mo.py --help

Traceback (most recent call last):

  File "mo.py", line 21, in <module>

    from mo.utils.versions_checker import check_python_version

  File "/localdisk/myshevts/computer_vision_sdk_fpga_2018.3.325/deployment_tools/model_optimizer/mo/utils/versions_checker.py", li
44

    def get_module_version_list_from_file(file_name: str):

                                         ^

SyntaxError: invalid syntax
```

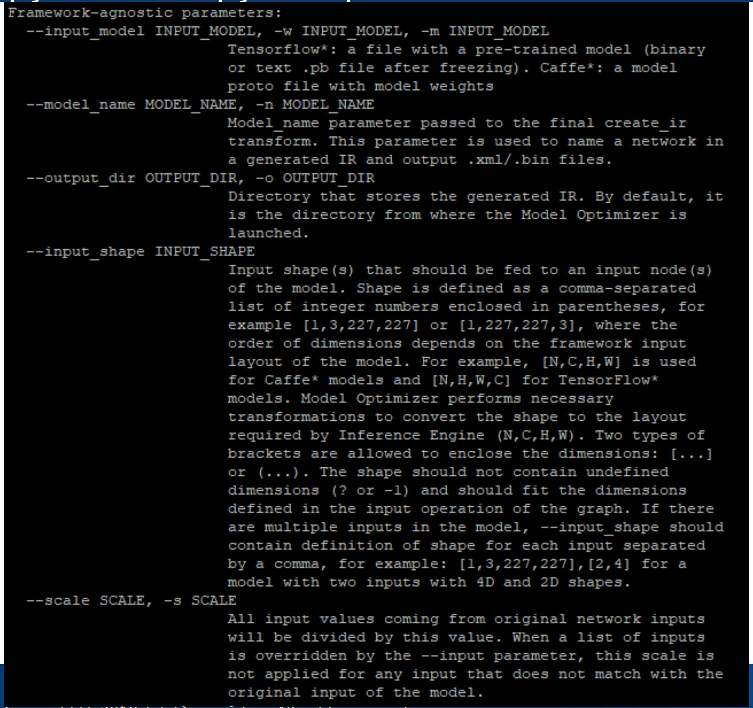
Fix: use python3

```
$ python --version
Python 2.7.5
$ python3 --version
Python 3.5.5
```

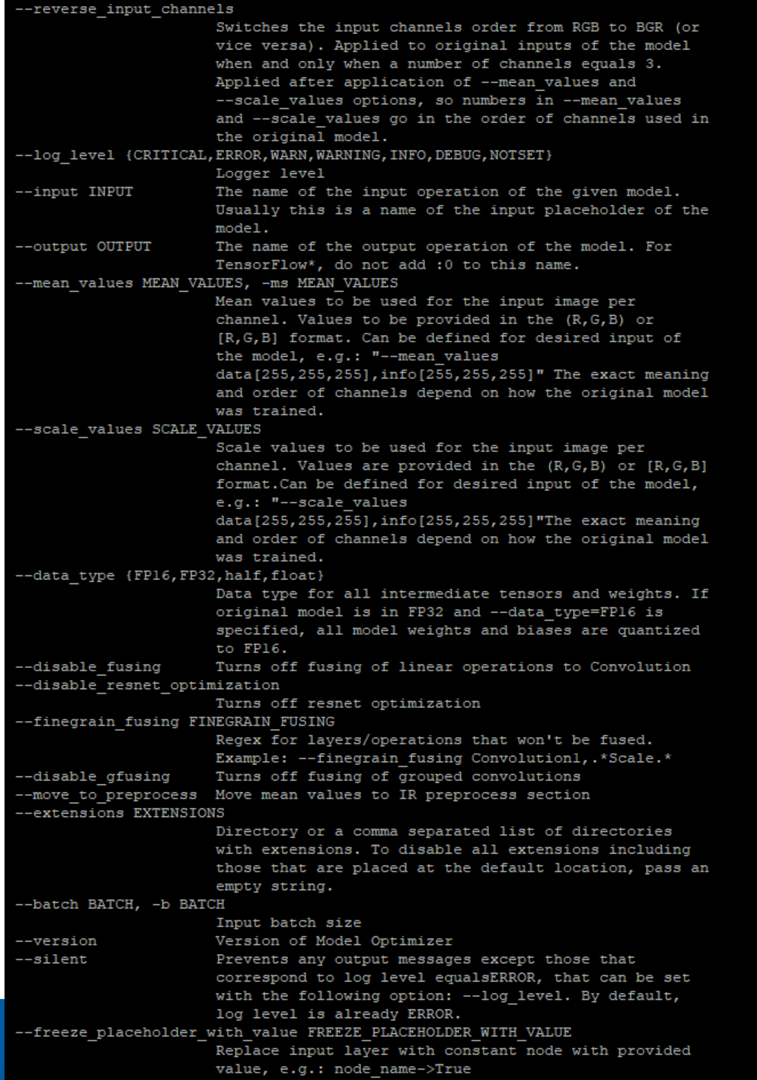
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

Model Optimizer: general command-line options

```
$ cd /opt/intel/computer_vision_sdk/deployment_tools/model_optimizer
$ python3 mo.py --help
```



```
Framework-agnostic parameters:
  --input_model INPUT_MODEL, -w INPUT_MODEL, -m INPUT_MODEL
                        Tensorflow*: a file with a pre-trained model (binary
                        or text .pb file after freezing). Caffe*: a model
                        proto file with model weights
  --model_name MODEL_NAME, -n MODEL_NAME
                        Model_name parameter passed to the final create_ir
                        transform. This parameter is used to name a network in
                        a generated IR and output .xml/.bin files.
  --output_dir OUTPUT_DIR, -o OUTPUT_DIR
                        Directory that stores the generated IR. By default, it
                        is the directory from where the Model Optimizer is
                        launched.
  --input_shape INPUT_SHAPE
                        Input shape(s) that should be fed to an input node(s)
                        of the model. Shape is defined as a comma-separated
                        list of integer numbers enclosed in parentheses, for
                        example [1,3,227,227] or [1,227,227,3], where the
                        order of dimensions depends on the framework input
                        layout of the model. For example, [N,C,H,W] is used
                        for Caffe* models and [N,H,W,C] for TensorFlow*
                        models. Model Optimizer performs necessary
                        transformations to convert the shape to the layout
                        required by Inference Engine (N,C,H,W). Two types of
                        brackets are allowed to enclose the dimensions: [...]
                        or (...). The shape should not contain undefined
                        dimensions (? or -1) and should fit the dimensions
                        defined in the input operation of the graph. If there
                        are multiple inputs in the model, --input_shape should
                        contain definition of shape for each input separated
                        by a comma, for example: [1,3,227,227],[2,4] for a
                        model with two inputs with 4D and 2D shapes.
  --scale SCALE, -s SCALE
                        All input values coming from original network inputs
                        will be divided by this value. When a list of inputs
                        is overridden by the --input parameter, this scale is
                        not applied for any input that does not match with the
                        original input of the model.
```



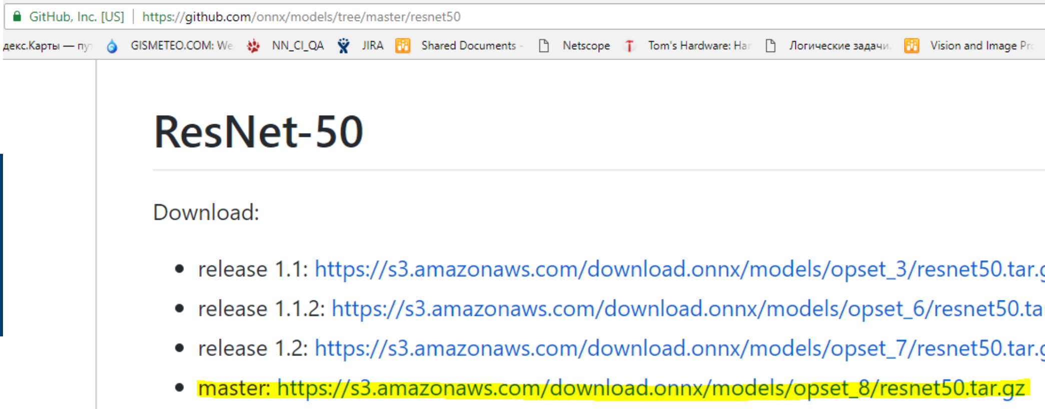
```
--reverse_input_channels
                      Switches the input channels order from RGB to BGR (or
                      vice versa). Applied to original inputs of the model
                      when and only when a number of channels equals 3.
                      Applied after application of --mean_values and
                      --scale_values options, so numbers in --mean_values
                      and --scale_values go in the order of channels used in
                      the original model.
--log_level {CRITICAL,ERROR,WARN,WARNING,INFO,DEBUG,NOTSET}
                      Logger level
--input INPUT         The name of the input operation of the given model.
                      Usually this is a name of the input placeholder of the
                      model.
--output OUTPUT       The name of the output operation of the model. For
                      TensorFlow*, do not add :0 to this name.
--mean_values MEAN_VALUES, -ms MEAN_VALUES
                      Mean values to be used for the input image per
                      channel. Values to be provided in the (R,G,B) or
                      [R,G,B] format. Can be defined for desired input of
                      the model, e.g.: "--mean_values
                      data[255,255,255],info[255,255,255]" The exact meaning
                      and order of channels depend on how the original model
                      was trained.
--scale_values SCALE_VALUES
                      Scale values to be used for the input image per
                      channel. Values are provided in the (R,G,B) or [R,G,B]
                      format.Can be defined for desired input of the model,
                      e.g.: "--scale_values
                      data[255,255,255],info[255,255,255]"The exact meaning
                      and order of channels depend on how the original model
                      was trained.
--data_type {FP16,FP32,half,float}
                      Data type for all intermediate tensors and weights. If
                      original model is in FP32 and --data_type=FP16 is
                      specified, all model weights and biases are quantized
                      to FP16.
--disable_fusing      Turns off fusing of linear operations to Convolution
--disable_resnet_optimization
                      Turns off resnet optimization
--finegrain_fusing FINEGRAIN_FUSING
                      Regex for layers/operations that won't be fused.
                      Example: --finegrain_fusing Convolution1,.*Scale.*
--disable_gfusing     Turns off fusing of grouped convolutions
--move_to_preprocess  Move mean values to IR preprocess section
--extensions EXTENSIONS
                      Directory or a comma separated list of directories
                      with extensions. To disable all extensions including
                      those that are placed at the default location, pass an
                      empty string.
--batch BATCH, -b BATCH
                      Input batch size
--version             Version of Model Optimizer
--silent              Prevents any output messages except those that
                      correspond to log level equalsERROR, that can be set
                      with the following option: --log_level. By default,
                      log level is already ERROR.
--freeze_placeholder_with_value FREEZE_PLACEHOLDER_WITH_VALUE
                      Replace input layer with constant node with provided
                      value, e.g.: node_name->True
```

Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

ONNX EXAMPLE

ResNet

```
$ python3 mo.py --input_model <path>/resnet50.onnx
```

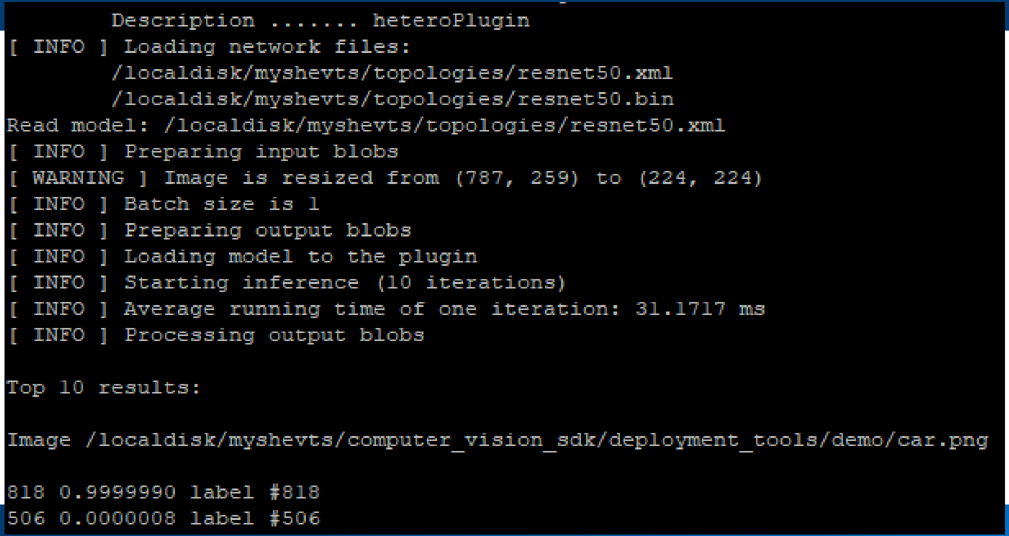


ResNet-50

Download:

- release 1.1: https://s3.amazonaws.com/download.onnx/models/opset_3/resnet50.tar.g
- release 1.1.2: https://s3.amazonaws.com/download.onnx/models/opset_6/resnet50.ta
- release 1.2: https://s3.amazonaws.com/download.onnx/models/opset_7/resnet50.tar.g
- master: https://s3.amazonaws.com/download.onnx/models/opset_8/resnet50.tar.gz

```
$ classification_sample -i ./car.png -m resnet50.xml –d HETERO:FPGA,CPU -ni 10
```



```
        Description ....... heteroPlugin
[ INFO ] Loading network files:
        /localdisk/myshevts/topologies/resnet50.xml
        /localdisk/myshevts/topologies/resnet50.bin
Read model: /localdisk/myshevts/topologies/resnet50.xml
[ INFO ] Preparing input blobs
[ WARNING ] Image is resized from (787, 259) to (224, 224)
[ INFO ] Batch size is 1
[ INFO ] Preparing output blobs
[ INFO ] Loading model to the plugin
[ INFO ] Starting inference (10 iterations)
[ INFO ] Average running time of one iteration: 31.1717 ms
[ INFO ] Processing output blobs

Top 10 results:

Image /localdisk/myshevts/computer_vision_sdk/deployment_tools/demo/car.png

818 0.9999990 label #818
506 0.0000008 label #506
```

Optimization Notice

Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

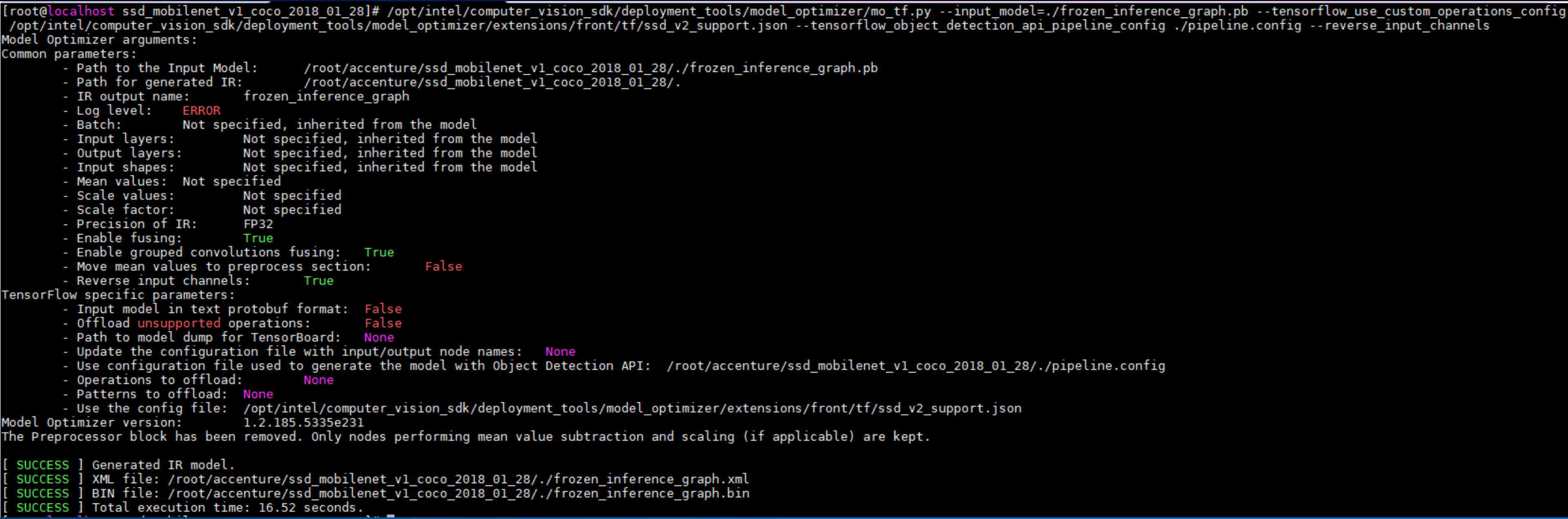
TENSORFLOW EXAMPLE

SSD_MOBILENET_v1_COCO

```
#> 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
```

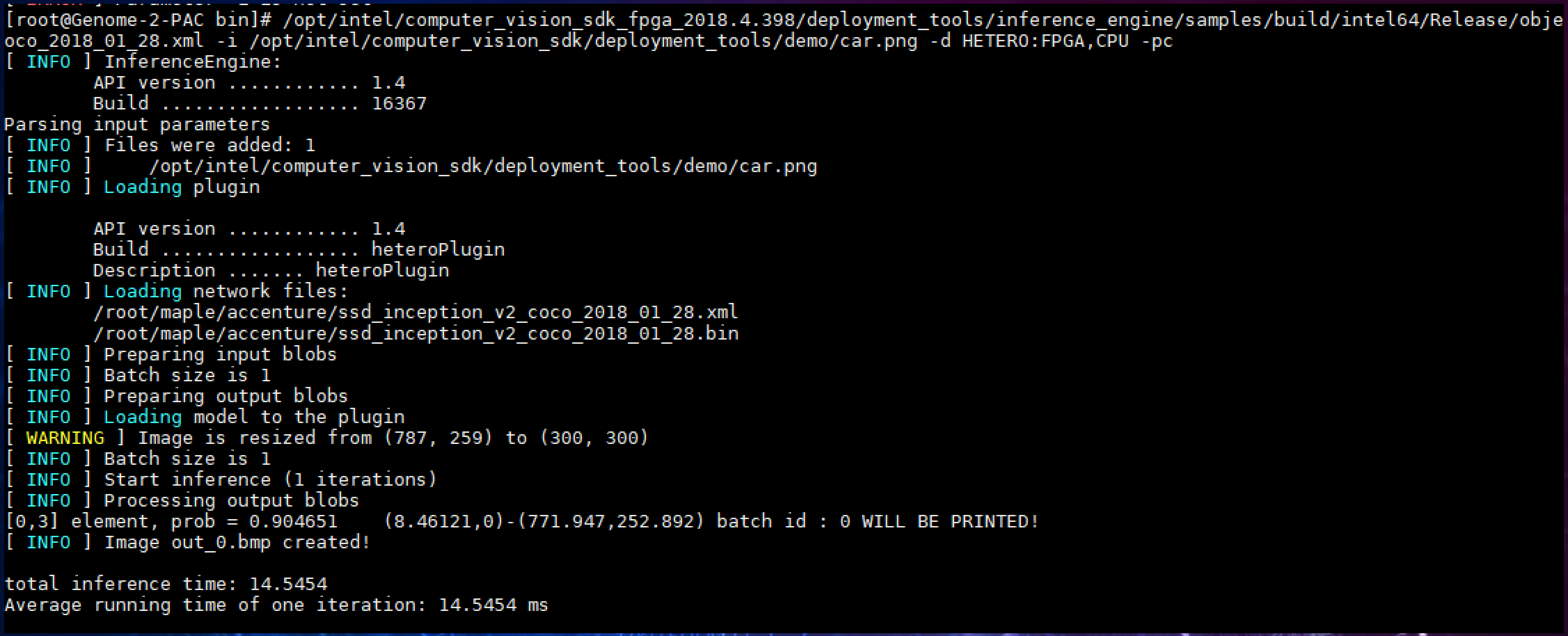
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

MO Result – SSD_Mobilenet @ R3



```
[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.
```

IE Result – SSD_Mobilenet @ R4

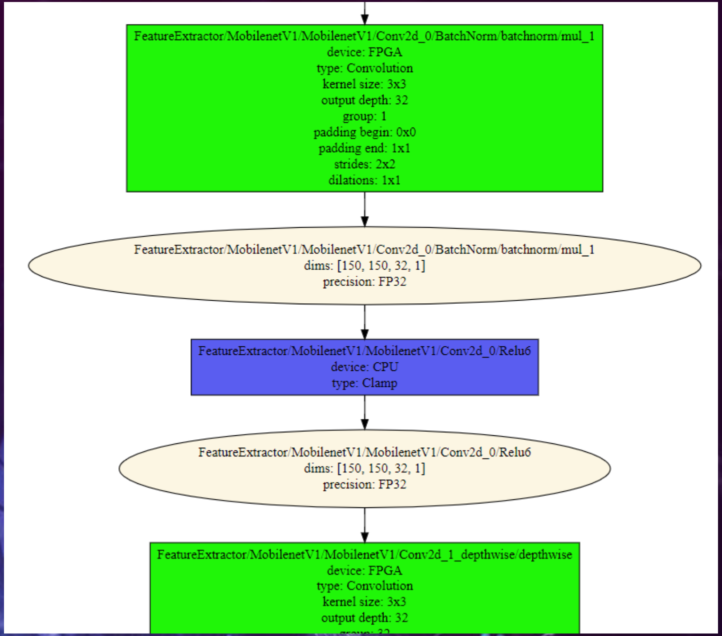


```
[root@Genome-2-PAC bin]# /opt/intel/computer_vision_sdk_fpga_2018.4.398/deployment_tools/inference_engine/samples/build/intel64/Release/obje
oco_2018_01_28.xml -i /opt/intel/computer_vision_sdk/deployment_tools/demo/car.png -d HETERO:FPGA,CPU -pc
[ INFO ] InferenceEngine:
        API version ............ 1.4
        Build .................. 16367
Parsing input parameters
[ INFO ] Files were added: 1
[ INFO ]     /opt/intel/computer_vision_sdk/deployment_tools/demo/car.png
[ INFO ] Loading plugin

        API version ............ 1.4
        Build .................. heteroPlugin
        Description ....... heteroPlugin
[ INFO ] Loading network files:
        /root/maple/accenture/ssd_inception_v2_coco_2018_01_28.xml
        /root/maple/accenture/ssd_inception_v2_coco_2018_01_28.bin
[ INFO ] Preparing input blobs
[ INFO ] Batch size is 1
[ INFO ] Preparing output blobs
[ INFO ] Loading model to the plugin
[ WARNING ] Image is resized from (787, 259) to (300, 300)
[ INFO ] Batch size is 1
[ INFO ] Start inference (1 iterations)
[ INFO ] Processing output blobs
[0,3] element, prob = 0.904651    (8.46121,0)-(771.947,252.892) batch id : 0 WILL BE PRINTED!
[ INFO ] Image out_0.bmp created!

total inference time: 14.5454
Average running time of one iteration: 14.5454 ms
```

HETERO Affinity @ R3

Green FPGA

Blue CPU



FeatureExtractor/MobilenetV1/MobilenetV1/Conv2d_0/BatchNorm/batchnorm/mul_1
device: FPGA
type: Convolution
kernel size: 3x3
output depth: 32
group: 1
padding begin: 0x0
padding end: 1x1
strides: 2x2
dilations: 1x1

FeatureExtractor/MobilenetV1/MobilenetV1/Conv2d_0/BatchNorm/batchnorm/mul_1
dims: [150, 150, 32, 1]
precision: FP32

FeatureExtractor/MobilenetV1/MobilenetV1/Conv2d_0/Relu6
device: CPU
type: Clamp

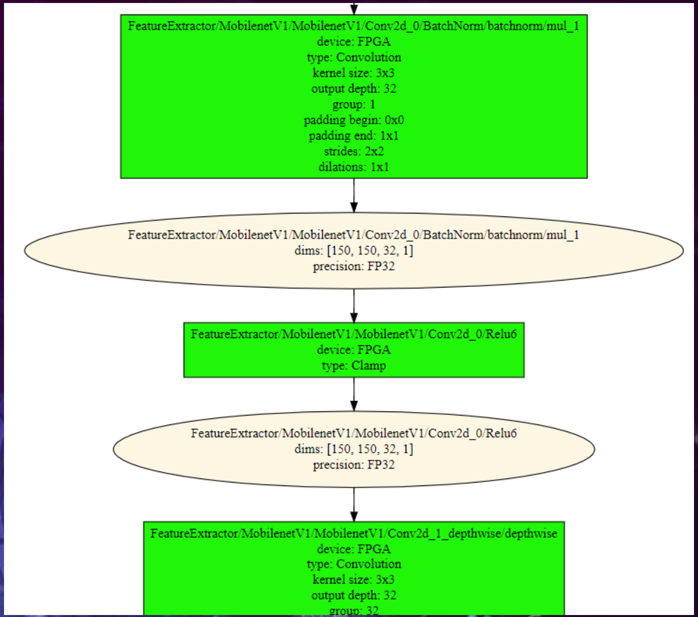
FeatureExtractor/MobilenetV1/MobilenetV1/Conv2d_0/Relu6
dims: [150, 150, 32, 1]
precision: FP32

FeatureExtractor/MobilenetV1/MobilenetV1/Conv2d_1_depthwise/depthwise
device: FPGA
type: Convolution
kernel size: 3x3
output depth: 32

HETERO Affinity @ R4

Green FPGA

Blue CPU



FeatureExtractor/MobilenetV1/MobilenetV1/Conv2d_0/BatchNorm/batchnorm/mul_1
device: FPGA
type: Convolution
kernel size: 3x3
output depth: 32
group: 1
padding begin: 0x0
padding end: 1x1
strides: 2x2
dilations: 1x1

FeatureExtractor/MobilenetV1/MobilenetV1/Conv2d_0/BatchNorm/batchnorm/mul_1
dims: [150, 150, 32, 1]
precision: FP32

FeatureExtractor/MobilenetV1/MobilenetV1/Conv2d_0/Relu6
device: FPGA
type: Clamp

FeatureExtractor/MobilenetV1/MobilenetV1/Conv2d_0/Relu6
dims: [150, 150, 32, 1]
precision: FP32

FeatureExtractor/MobilenetV1/MobilenetV1/Conv2d_1_depthwise/depthwise
device: FPGA
type: Convolution
kernel size: 3x3
output depth: 32
group: 32