



# ACCELERATE DEEP LEARNING INFERENCE USING INTEL TECHNOLOGIES

## OPTIMIZATION: TOOLS AND TECHNIQUES

February 2020

# OPTIMIZATION NOTICE

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness or any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice Revision #20110804.



# LEGAL NOTICES AND DISCLAIMERS (1 OF 2)

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software, or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at [www.intel.com](http://www.intel.com).

Performance estimates were obtained prior to implementation of recent software patches and firmware updates intended to address exploits referred to as "Spectre" and "Meltdown." Implementation of these updates may make these results inapplicable to your device or system.

Cost reduction scenarios described are intended as examples of how a given Intel-based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

This document contains information on products, services, and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications, and roadmaps.

Any forecasts of goods and services needed for Intel's operations are provided for discussion purposes only. Intel will have no liability to make any purchase in connection with forecasts published in this document.

Arduino\*101 and the Arduino\* infinity logo are trademarks or registered trademarks of Arduino, LLC.

Altera, Arria, the Arria logo, Intel, the Intel logo, Intel Atom, Intel Core, Intel Nervana, Intel Xeon Phi, Movidius, Saffron, and Xeon are trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

Copyright © 2018, Intel Corporation. All rights reserved.



# LEGAL NOTICES AND DISCLAIMERS (2 OF 2)

This document contains information on products, services, and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications, and roadmaps. Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software, or service activation. Learn more at [intel.com](https://www.intel.com) or from the OEM or retailer. No computer system can be absolutely secure.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit [www.intel.com/performance](https://www.intel.com/performance).

Cost reduction scenarios described are intended as examples of how a given Intel-based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

Statements in this document that refer to Intel's plans and expectations for the quarter, the year, and the future, are forward-looking statements that involve a number of risks and uncertainties. A detailed discussion of the factors that could affect Intel's results and plans is included in Intel's SEC filings, including the annual report on Form 10-K.

The products described may contain design defects or errors, known as errata, which may cause the product to deviate from published specifications. Current characterized errata are available on request. Performance estimates were obtained prior to implementation of recent software patches and firmware updates intended to address exploits referred to as "Spectre" and "Meltdown." Implementation of these updates may make these results inapplicable to your device or system.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document. Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

Intel, the Intel logo, Pentium, Celeron, Atom, Core, Xeon, Movidius, Saffron, and others are trademarks of Intel Corporation in the United States and other countries.

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

Copyright © 2018, Intel Corporation. All rights reserved.



# SMART VIDEO WORKSHOP OVERVIEW

## INTRODUCTION

1. Introduction to Intel technologies for deep learning inference
2. Hardware acceleration techniques

Each module contains a hands-on lab exercise that introduces various Intel technologies to accelerate computer vision application with hardware heterogeneity.

## INTEL® DISTRIBUTION OF OPENVINO™ 101

## HARDWARE ACCELERATION

2. Basic End-to-End Object Detection Example

- 3./4./5. Hardware Acceleration with CPU, Integrated GPU, Intel® Movidius™ NCS, FPGA

## OPTIMIZATION

6. Optimization Tools and Techniques

## APPLICATION

7. Advanced Video Analytics



# AGENDA

- Pick the Right Model
- Use DL Workbench
- Don't Infer If Not Needed
- Use Command Line Deployment Manager

PICK THE RIGHT MODEL

# USE/TRAIN A MODEL WITH THE RIGHT PERFORMANCE PLUS ACCURACY TRADEOFFS.

Performance is based on many factors:

- Topography complexity/layer implementation plus scheduling
- Number of color channels (that is, BGR vs. grayscale)
- Model resolution



# EXERCISE: RANGE OF MODEL PERFORMANCE

FOCUS ON THE INFERENCE TIMING

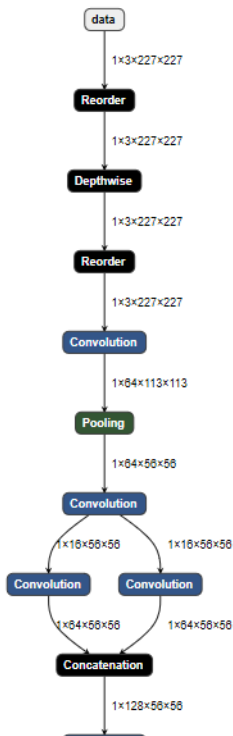
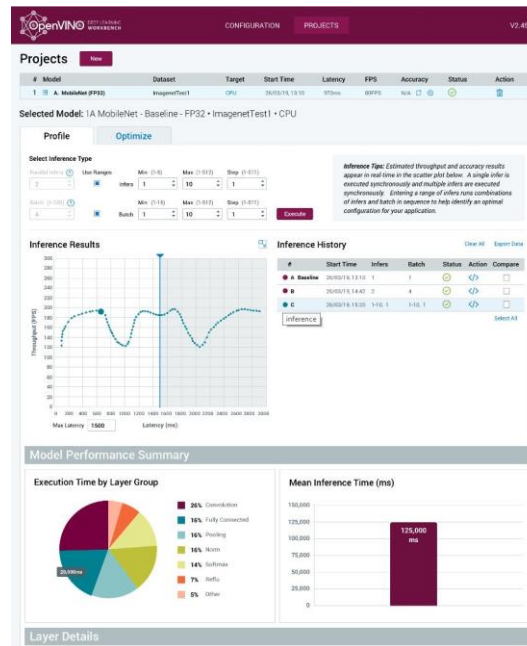
	CPU ms/frame	GPU ms/frame	Intel® Movidius™ Myriad™ X ms/frame
ssd512			
ssd300			
Mobilnet-ssd*			

USE DL WORKBENCH

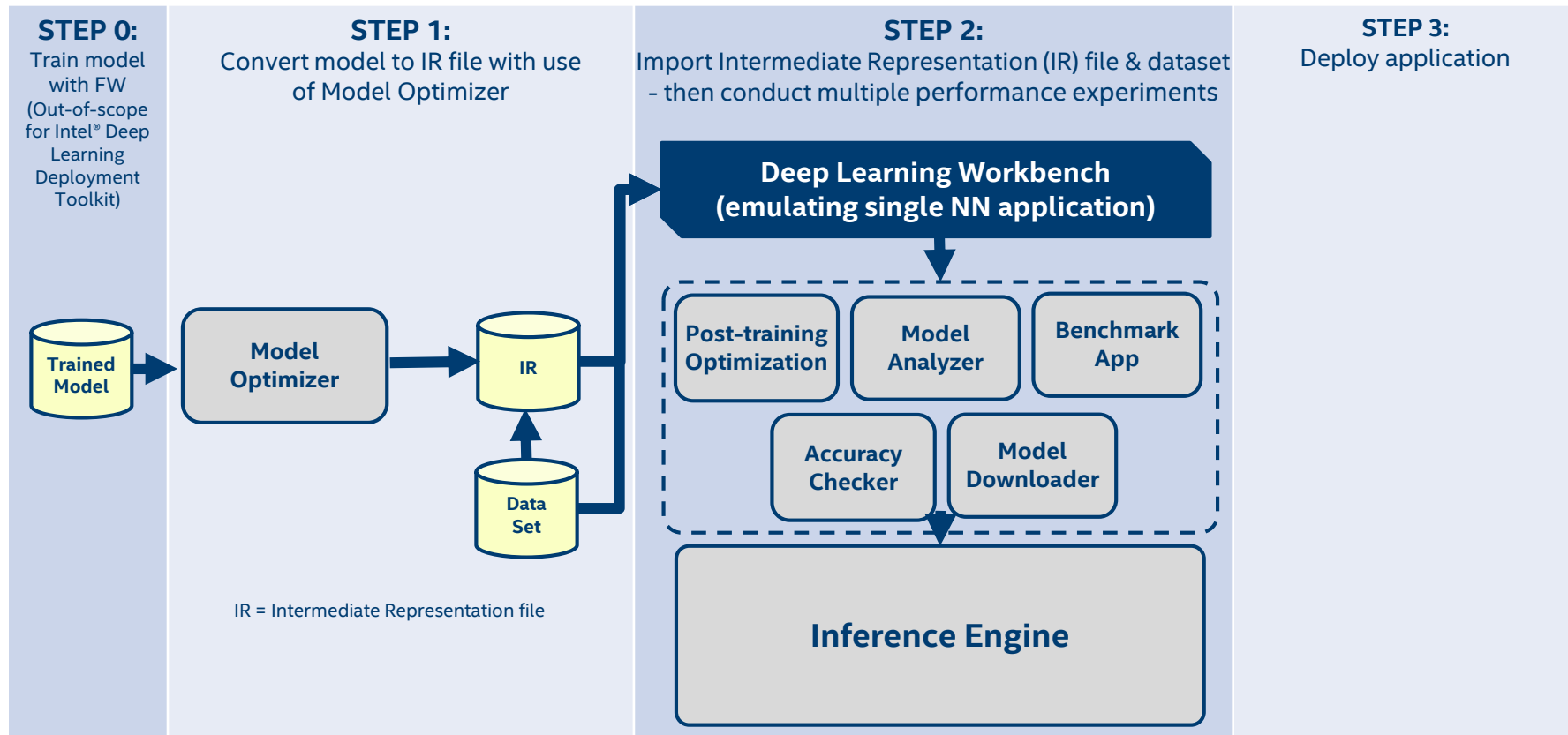
# DEEP LEARNING WORKBENCH

## Deep Learning Workbench capabilities

- Web-based tool - UI extension of Intel® Distribution of OpenVINO™ toolkit functionality
- Visualizes performance data for topologies/ layers to aid in model analysis
- Automate analysis for optimal performance configuration (streams, batches, latency)
- Experiment with int8 calibration for optimal tuning
- Provide accuracy info through accuracy checker
- Direct access to Models from public set of Open Model Zoo



# DEEP LEARNING WORKBENCH DATA FLOW



# WORKBENCH INTERFACES WITH KEY COMPONENTS

- **Post-Training Optimization Toolkit** – Convert a model into a more hardware-friendly representation by applying specific methods that do not require re-training, for example, post-training quantization.
- **Model Analyzer** – Provides theoretical data on models: computational complexity (flops), number of neurons, memory consumption.
- **Benchmark App** – Helps measure performance (throughput, latency) of a model, get performance metrics per layer and overall basis.
- **Accuracy Checker Tool** – Check for accuracy of the model (original and after conversion) to IR file using a known data set.
- **Model Downloader** – Provides an easy way of accessing a number of public neural network models as well as a set of pre-trained Intel models

## Installation & Distribution

Intel® Distribution of OpenVINO™ toolkit

- Build your local docker image from package (build scripts)
- Build your local docker image by copying and running dockerfile from documentation

Download docker image from DockerHub

- Work in progress – will depend on decision of OpenSource PDT
- Can be used for internal experiments (at least)

IR = Intermediate Representation file




# CONVERT MODEL TO INT8 USING 2 NEW CALIBRATION ALGORITHMS

**Audience:** New and experienced users of OpenVINO and DL Workbench

**Problem:** Provide an easier to use (non-command line) interface to new Post training optimization (calibration) tool and calibration algorithms

**UseCase:** OpenVINO user can convert her model to Int8 using 2 new algorithms achieving this goal using purely UI.

 **OpenVINO** DEEP LEARNING WORKBENCH

Version: 1.0.2493.f92ad399 ...

### Calibration options

squeezenet1.1 • Imagenet\_200\_224x224 • CPU

Subset of images: ⓘ

100 %

### Select optimization method:

☐ Optimization method: Default  
Uncontrollable minor drop of model accuracy  
Significant increase of model speed

☒ Optimization method: AccuracyAware  
Optimization method: AccuracyAware  
Controllable drop of model accuracy  
Increase of model speed

Max Accuracy Drop: ⓘ

1%

**Optimize Tips:**  
Maximum performance calibration (optimization method: Default) optimizes your model to achieve best performance. The algorithm usually produces the fastest model and usually but not always results in accuracy drop within 1%. Also, this algorithm takes less time than the AccuracyAware optimization method.

Maximum accuracy calibration (optimization method: AccuracyAware) optimizes your model to achieve best performance possible with the specified maximum acceptable accuracy drop. Maximum accuracy calibration might result in lower performance compared to the Maximum performance calibration, while the accuracy drop is predictable. Accuracy drop is the difference between the original model accuracy and the optimized model accuracy. Accuracy of the optimized model is guaranteed to be not smaller than the difference between the original model accuracy and the accuracy drop.

As a rule, the smaller the calibration subset, the less time the algorithms take. It is recommended to use at least a 3-5% subset of the validation dataset (300-1000 images).

A model optimized by the Default method translates all layers that support INT8 execution into the INT8 precision, while the AccuracyAware method translates only those layers that both can be executed in the INT8 precision and almost do not increase accuracy drop.

Cancel

Optimize

# IMPORT DATASET IN COCO FORMAT TO USE WITH MODEL

**Audience:** New and experienced users of OpenVINO and DL Workbench

**Problem:** Currently DL Workbench supports only ImageNet format dataset(s) or Pascal VOC format dataset(s) in flow. COCO dataset extends the list of supported formats giving additional freedom in selecting dataset for experiments.

**UseCase:** *Applicable for all use cases provided by DL Workbench.*

The screenshot shows the OpenVINO Deep Learning Workbench interface. At the top, the logo and version number 'Version: 1.0.2493.f92ad399' are visible. The main window displays the 'Import Local Dataset' dialog box. The dialog box has a title bar and a content area. The content area contains the text: 'Import a Dataset formatted in the ImageNet, VOC or COCO formats (tar.gz or .zip file)'. Below this text are two input fields: 'Dataset File:' with a 'Choose file' button, and 'Dataset Name:' with a text input field. A red error message 'This field is required' is displayed below the 'Dataset Name' field. At the bottom of the dialog box are two buttons: 'Cancel' and 'Import Dataset'. To the right of the dialog box, there is a section titled 'Import Dataset Tips:'. This section contains a list of dataset types: 'ImageNet', 'Pascal VOC', and 'COCO'. The 'COCO' item is expanded, showing a description: 'Common Objects in Context (COCO) dataset is used for object detection, segmentation, person keypoints detection, stuff segmentation, and caption generation. Currently, the DL Workbench supports only object-detection COCO datasets.' Below the description is a link to 'Dataset Types' for details. Further down, it states: 'A dataset is downloaded as two separate archives, but you have to create one archive with a folder for images and a folder for annotations, where images go to the val folder.' Below this is a code block showing the directory structure for a COCO dataset: 

```
|-- coco.zip
|   |-- val
|       |-- 0001.jpg
|       |-- 0002.jpg
|       ...
|       |-- n.jpg
|   |-- annotations
|       |-- instances_val.json
```

 At the bottom of the tips section, it states: 'For Pascal VOC and COCO datasets, only the Object-Detection task is supported.'

# IMPROVED PER-LAYER DATA VISUALIZATION AND COMPARISON MODE. MULTIPLE UX IMPROVEMENTS.

**Audience:** New and existing users of DL Workbench

**Problem:** Address the number of UX items.  
Improve visualization of table with per-layer information

**UseCase:** User have more intuitive visualization of the per-layer information including fusing of layers on target HW and comparison mode for the models.

The screenshot displays the OpenVINO Deep Learning Workbench interface. At the top, the header shows the OpenVINO logo and version: 1.0.2493.f92ad399. Below the header, there are filters for 'Select Column' and 'Select Filter', along with 'Add new filter', 'Clear Filter', and 'Apply Filter' buttons.

Layer	Execution Time, ms
Depthwise	
<a href="#">relu3_11/x1</a> Activation	0.008
<a href="#">Add1_22808/Fused_Add_</a> Convolution	0.136
<a href="#">conv3_11/x2</a> Convolution	0.091
<a href="#">Add1_22820/Fused_Add_</a> Depthwise	0.019
<a href="#">relu3_12/x1</a> Activation	0.009
<a href="#">Add1_22832/Fused_Add_</a> Convolution	0.147
<a href="#">conv3_12/x2</a> Convolution	0.091
<a href="#">Add1_22856/Fused_Add_</a> Depthwise	0.004
<a href="#">relu4_1/x1</a> Activation	0.002

The right panel shows a detailed view of the selected layer: [Add1\\_22832/Fused\\_Add\\_](#). It includes the following information:

- Execution parameters:**
  - Execution Order: 96
  - Execution Time, ms: 0.147
  - Output Layouts: nChw8c
  - Output Precisions: FP32
  - Primitive Type: jlt\_avx2\_1x1\_FP32
- Fusing information:**

IR Layers [Add1\\_22832/Fused\\_Add\\_](#), [relu3\\_12/x2](#) were transformed on device to single layer [Add1\\_22832/Fused\\_Add\\_](#). This is called layer fusion and the diagram below demonstrates the fusion scheme and information on each layer from original IR.

```
graph LR; A[Add1_22832/Fused_Add_] --> B[relu3_12/x2]; B --> C[Add1_22832/Fused_Add_];
```
- Layer Parameters:**
  - Spatial Parameters: No data
  - Specific Parameters: No data
  - Positional Data:
    - Input 0: 1, 128, 28, 28
    - Input 1: 1, 128, 28, 28



DON'T INFER IF NOT NEEDED

# DETERMINE IF THERE IS NOTHING TO SEE

Inference is expensive to run each frame. It can save time to not run when there is nothing to identify.

- Check motion vectors
- Frame sizes
- bgsubmog
- SAD

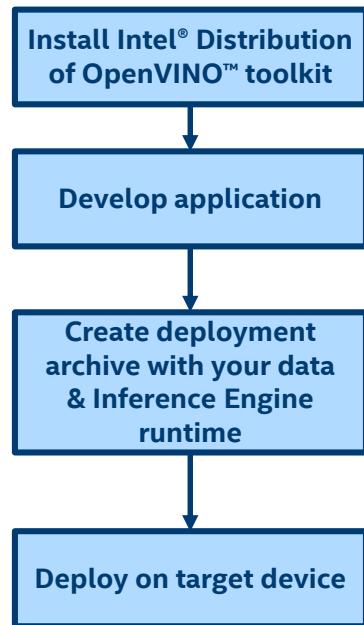
These methods can be several orders of magnitude less expensive than inference. Use techniques to increase the total # of streams a system can watch.

# USE COMMAND LINE DEPLOYMENT MANAGER

# COMMAND LINE DEPLOYMENT MANAGER

- Generate an optimal, minimized runtime package for selected target device.
- Deploy Inference Engine with pre-compiled application-specific data such as models, config, and a subset of required hardware plugins.
- Achieve deployment footprint to be several times smaller than the development footprint.

For more details, see [Introduction to CLI Deployment Manager](#)



Target	Size, MB
CPU only	65
GPU only	26
Myriad only	22
HDDL only	27
GNA only	15

Measurements for deployment archives based on 2019 R3

# LAB5 - OPTIMIZING COMPUTER VISION APPLICATIONS

URL: <https://github.com/intel-iot-devkit/smart-video-workshop/blob/master/optimization-tools-and-techniques/README.md>

**Objective:** This tutorial shows some techniques to get better performance for computer vision applications with the Intel® Distribution of OpenVINO™ toolkit.

**Estimated Complete Time:** 40min





# ADVANCED VIDEO ANALYTICS

## SECURITY BARRIER DEMO

February 2020

## VIDEO ANALYTICS IN INTEL® DISTRIBUTION OF OPENVINO™ TOOLKIT

Topology	Type	Description
<a href="#"><u>license-plate-recognition-barrier-0001</u></a>	ocr	Chinese license plate recognition.
<a href="#"><u>vehicle-attributes-recognition-barrier-0010</u></a>	object_attributes	Vehicle attributes recognition with modified RESNET10* backbone.
<a href="#"><u>vehicle-license-plate-detection-barrier-0007</u></a>	detection	Multiclass (vehicle, license plates) detector based on RESNET10 plus SSD.



# VEHICLE-ATTRIBUTES-RECOGNITION-BARRIER-0010

## USE CASE/HIGH-LEVEL DESCRIPTION

Vehicle attributes classification algorithm for a traffic analysis scenario.



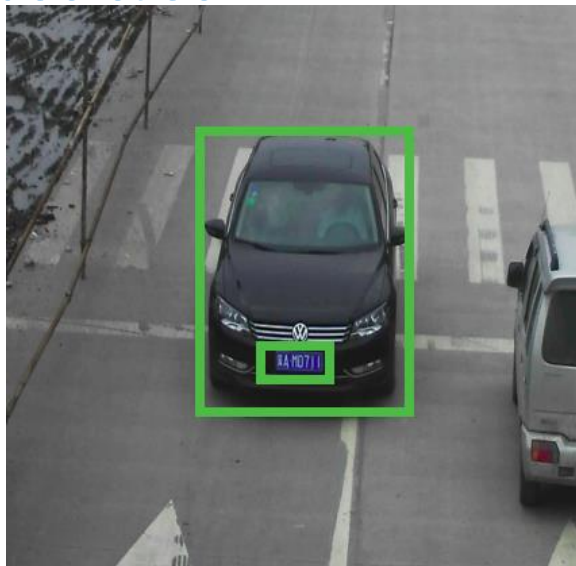
Type: regular  
Color: black



# VEHICLE-LICENSE-PLATE-DETECTION-BARRIER-007

## USE CASE/HIGH-LEVEL DESCRIPTION

RESNET\* 10 plus SSD-based vehicle and (Chinese) license plate detector for "Barrier" use case.



# LICENSE-PLATE-RECOGNITION-BARRIER-0001

## USE CASE/HIGH-LEVEL DESCRIPTION

Small-footprint network trained E2E to recognize Chinese license plates in traffic scenarios.

Note: The license plates in the image are modified from the originals.





## LAB7 - ADVANCED VIDEO ANALYTICS

URL: [https://github.com/intel-iot-devkit/smart-video-workshop/blob/master/advanced-video-analytics/multiple\\_models.md](https://github.com/intel-iot-devkit/smart-video-workshop/blob/master/advanced-video-analytics/multiple_models.md)

**Objective:** The tutorial shows some techniques for developing advanced video analytics applications.

**Estimated Complete Time:** 20min



