ACCELERATE DEEP LEARNING INFERENCE USING INTEL® TECHNOLOGIES

INTRODUCTION: SMART VIDEO

INTEL® DISTRIBUTION OF OPENVINO™ TOOLKIT 2020.R4 VERSION

July 2020

AGENDA

- Intel® Smart Video/Computer vision Tools Overview
- Model Optimizer
- Inference Engine
- 15 Minute Break
- Accelerators based on Intel® Movidius™ Vision Processing Unit
- Accelerators based on Intel® Arria® FPGA
- Multiple Models in One Application
- DL Workbench + Demo
- DL Streamer
- Register for access to Intel® DevCloud for the Edge
- **Lab1 - DevCloud Sample Application: Accelerated Object Detection**
- **Lab2 – DevCloud Advanced Tutorials: DL Streamer Benchmark**

NOTICES AND DISCLAIMER

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

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.

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.

OPTIMIZATION NOTICE

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 to learn more.

The benchmark results reported in this deck may need to be revised as additional testing is conducted. The results depend on the specific platform configurations and workloads utilized in the testing, and may not be applicable to any particular user's components, computer system or workloads. The results are not necessarily representative of other benchmarks and other benchmark results may show greater or lesser impact from mitigations.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit www.intel.com/benchmarks.

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.

Other names and brands may be claimed as the property of others. Any third-party information referenced on this document is provided for information only. Intel does not endorse any specific third-party product or entity mentioned on this document. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Copyright Intel Corporation.

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 of 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. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804

OpenVX and the OpenVX logo are trademarks of the Khronos Group Inc.

OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission by Khronos

AI CHANGING AND ENABLING EVERY INDUSTRY

AI software market is projected to reach USD 126.0 billion in annual worldwide revenue by 2025[3]

Deep learning software revenue is estimated to grow to USD 67.2 billion by 2025[4]

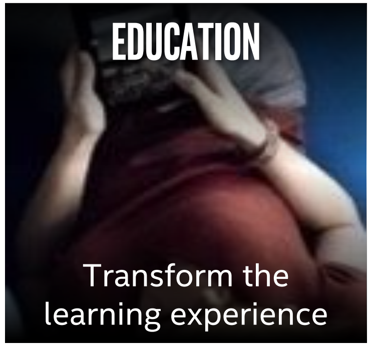
Global deep learning chip market is expected to reach USD 29.4 billion by 2025[5]

AGRICULTURE

Achieve higher yields and increase efficiency

ENERGY

Maximize production and uptime



EDUCATION

Transform the learning experience

GOVERNMENT

Enhance safety, research, and more

FINANCE

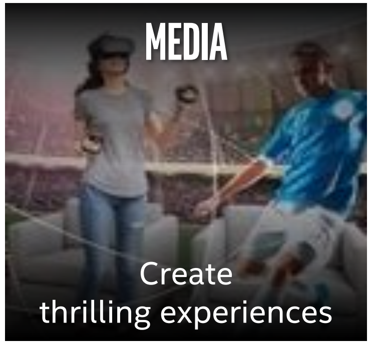
Turn data into valuable intelligence

HEALTH

Revolutionize patient outcomes

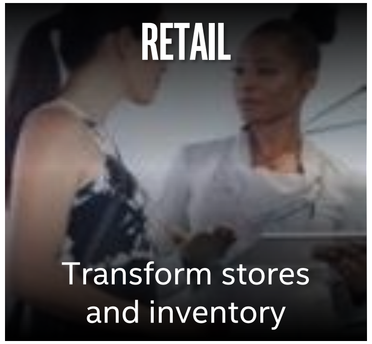
INDUSTRIAL

Empower truly intelligent Industry 4.0



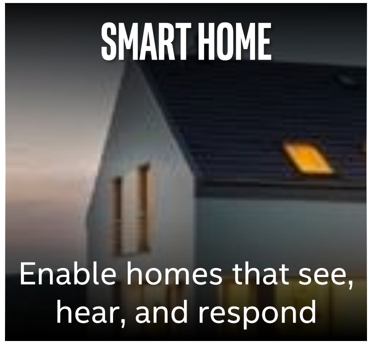
MEDIA

Create thrilling experiences



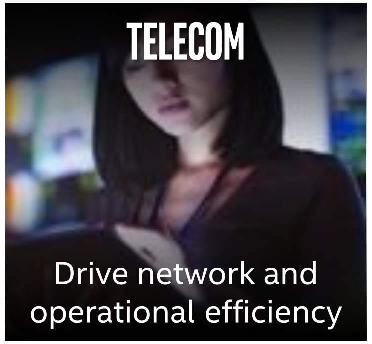
RETAIL

Transform stores and inventory



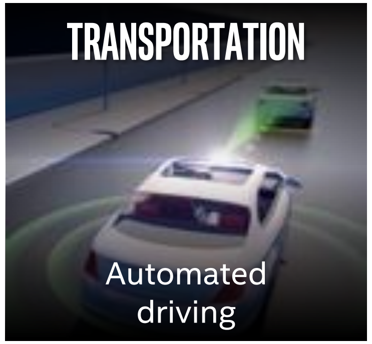
SMART HOME

Enable homes that see, hear, and respond



TELECOM

Drive network and operational efficiency



TRANSPORTATION

Automated driving

[3] Tractica, Artificial Intelligence Software Market, 2020

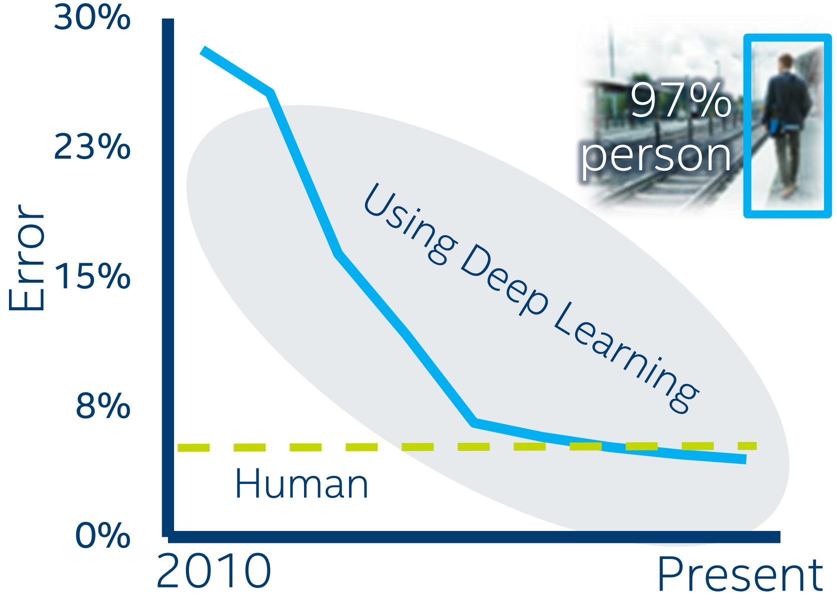
[4] Tractica, deep learning research, 2018

[5] AlliedMarketResearch, Deep Learning Chip Market, 2018

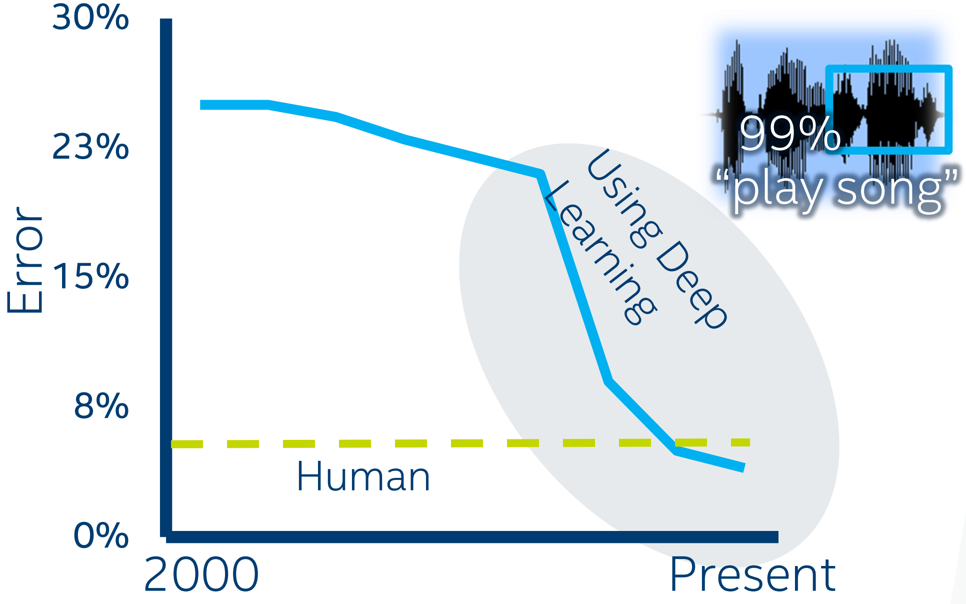
DEEP LEARNING BREAKTHROUGHS AND OPPORTUNITIES

Machines able to meet or exceed human image and speech recognition

Image Recognition



Speech Recognition

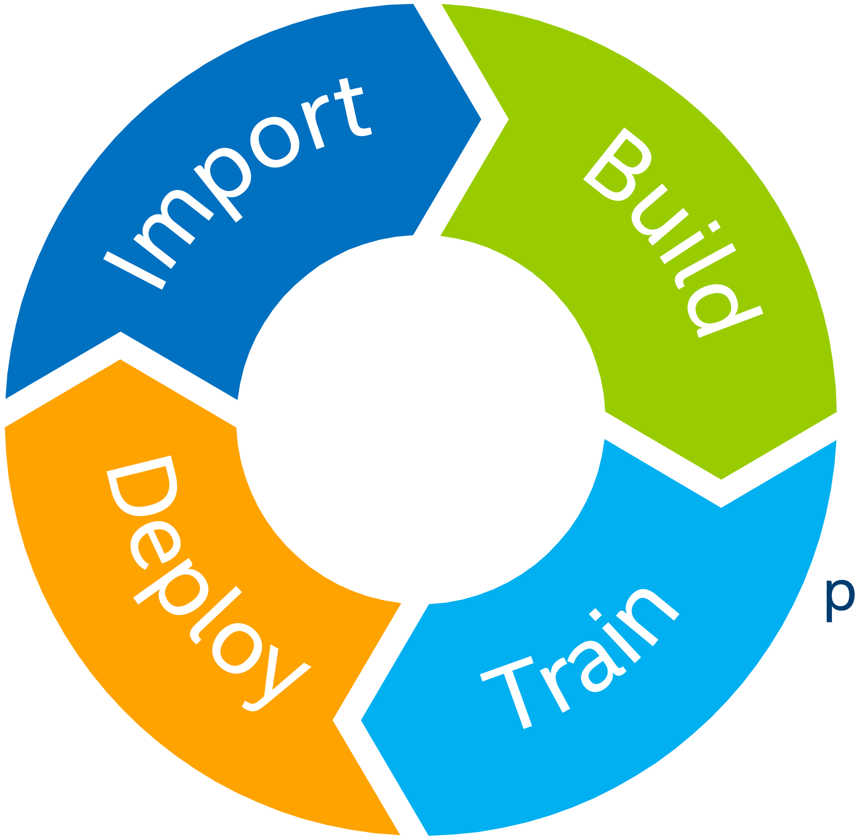


ADDITIONAL ECONOMIC IMPACT DRIVEN BY AI

$13 TRILLION IN 2030

Source: ILSVRC ImageNet winning entry classification error rate each year 2010-2016 (Left), https://www.microsoft.com/en-us/research/blog/microsoft-researchers-achieve-new-conversational-speech-recognition-milestone/ (Right)
Source: https://www.mckinsey.com/featured-insights/artificial-intelligence/notes-from-the-ai-frontier-applications-and-value-of-deep-learning

DEEP LEARNING DEVELOPMENT CYCLE



Data acquisition and organization

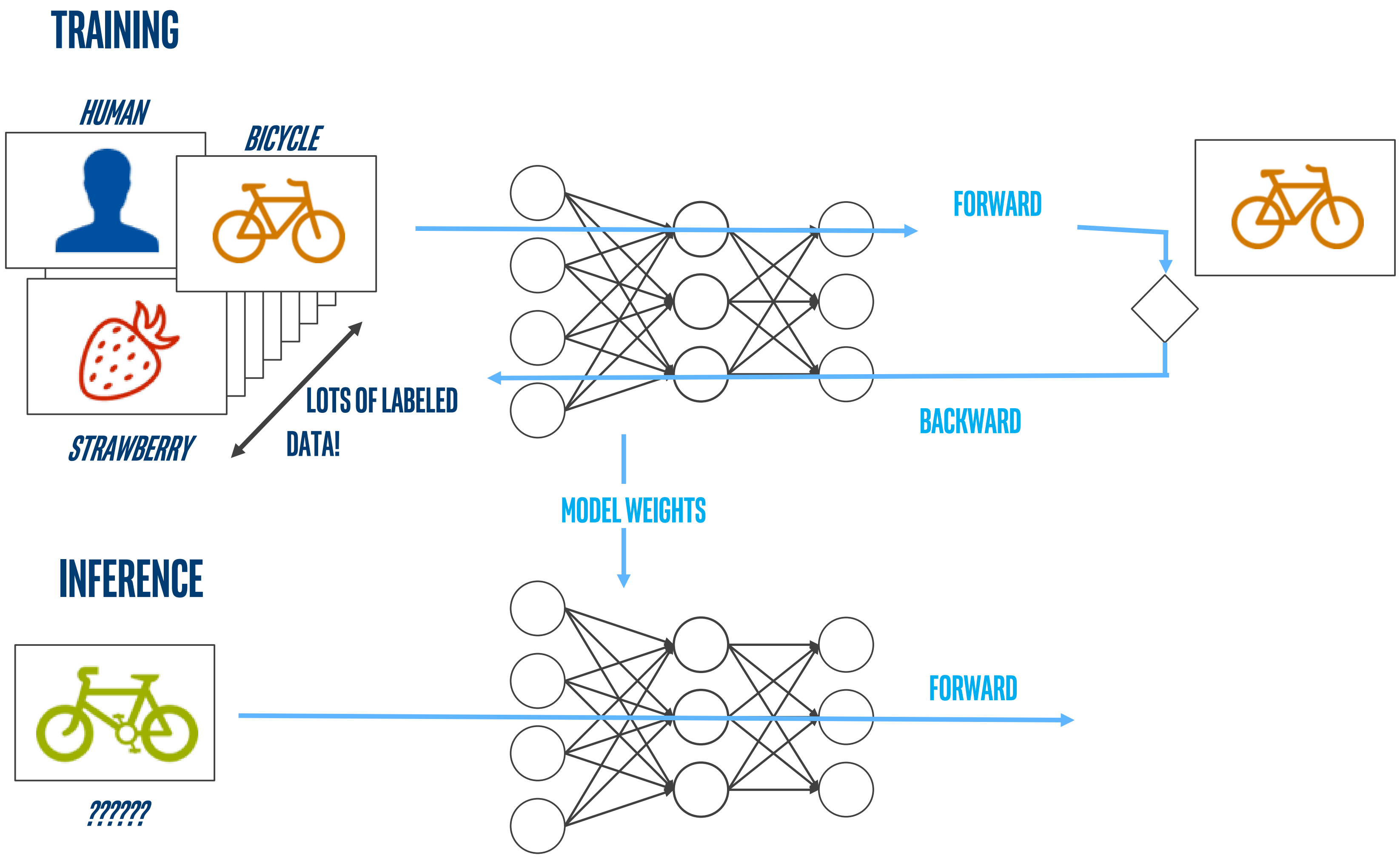
Create models

Adjust models to meet performance and accuracy objectives

Integrate trained models with application code

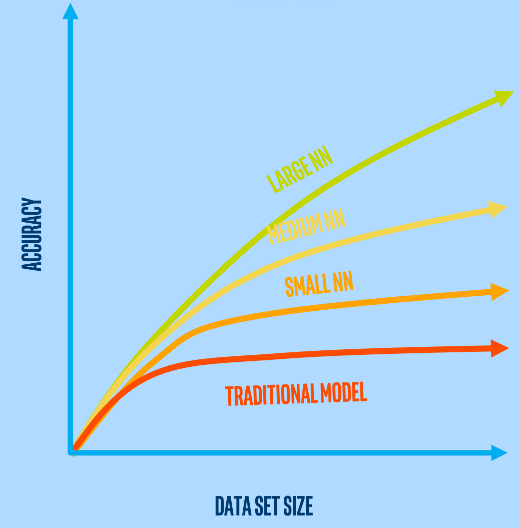
Intel® Distribution OpenVINO™ Toolkit Provides Deployment from Intel® Edge to Cloud

DEEP LEARNING: TRAINING VS. INFERENCE



DID YOU KNOW?

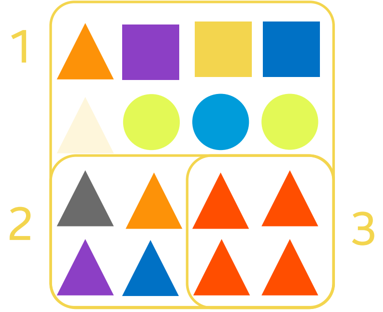
Training requires a very large data set and deep neural network (many layers) to achieve the highest accuracy in most cases



AI COMPUTE CONSIDERATIONS

How do you determine the right computing for your AI needs?

WORKLOADS



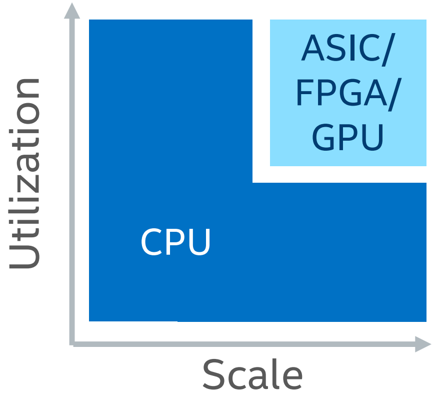
What is my workload profile?

REQUIREMENTS

RESPONSE TIME
SECURITY
DATA
VERSATILITY
SCALABILITY
COST
POWER
SIZE
RELIABILITY
THROUGHPUT
EFFICIENCY
MANAGEABILITY
FLEXIBILITY
ACCURACY

What are my use case requirements?

DEMAND

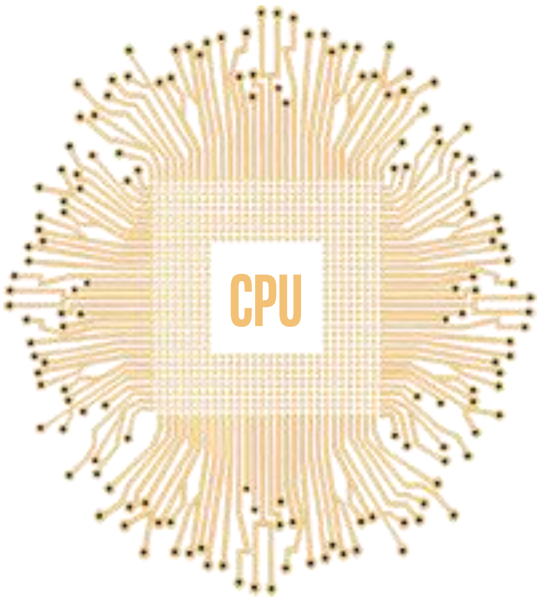


How prevalent is AI in my environment?

Optimization Notice

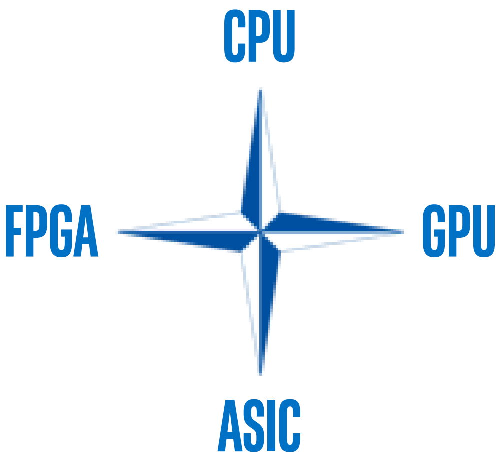
WHY INTEL AI COMPUTE?

MAXIMIZE



Get the most out of the foundation for AI from the CPU leader

OPTIMIZE



Choose the right compute for you from the one with all the options

SIMPLIFY

OPTIMIZED SW

DATA PIPELINE

ANALYTICS & AI

SUPPORT

MOVE/STORE

Reduce “moving parts” by building on an optimized AI platform

LEAD

Lead your industry by aligning with the builder of next-gen AI solutions

INTEL® DISTRIBUTION OF OPENVINO™ TOOLKIT

Tool Suite for High-Performance, Deep Learning Inference

Fast, accurate real-world results using high-performance, AI and computer vision inference deployed into production across Intel® architecture from edge to cloud

High-Performance,
Deep Learning Inference

Streamlined Development,
Ease of Use

Write Once,
Deploy Anywhere

USING THE INTEL® DISTRIBUTION OF OPENVINO™ TOOLKIT

Advanced capabilities to streamline deep learning deployments

1. BUILD

Trained Model

TensorFlow Caffe

KALDI mxnet

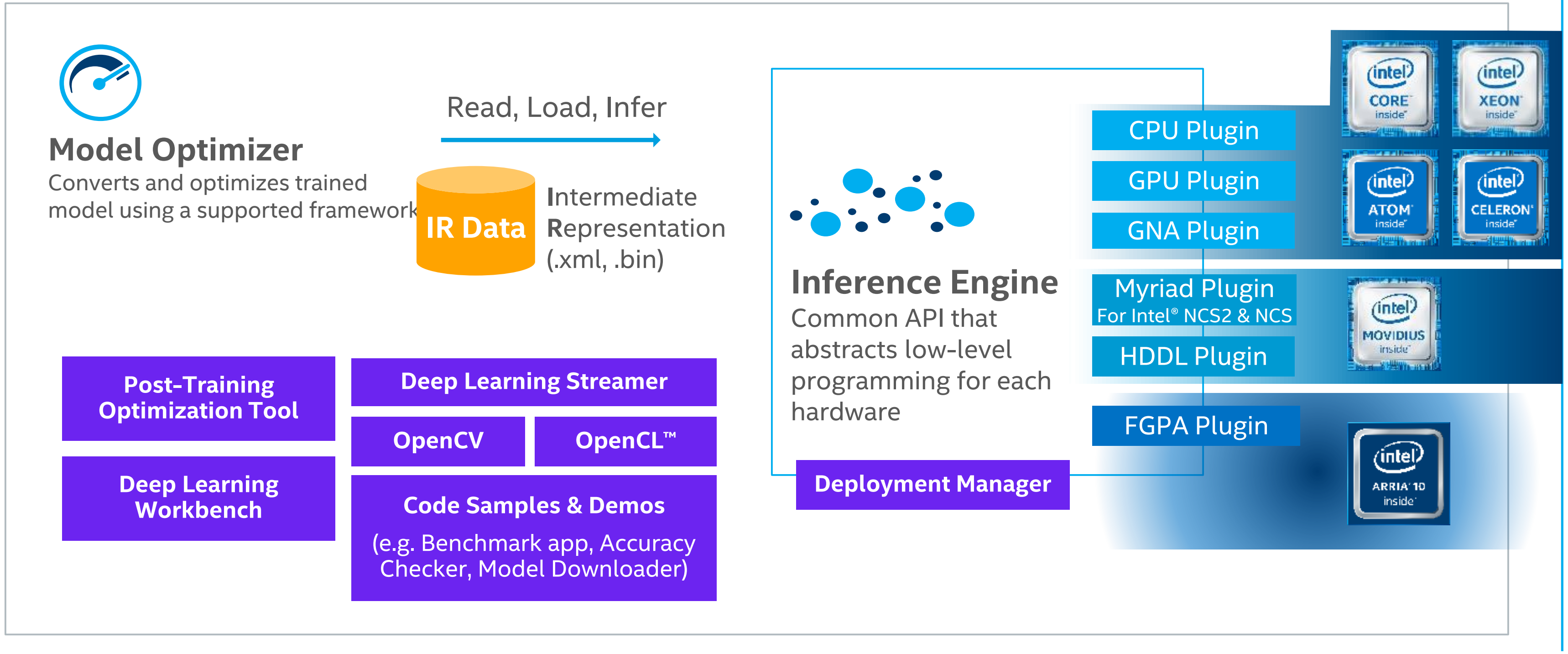
ONNX

Open Model Zoo

100+ open sourced and optimized pre-trained models; 80+ supported public models

2. OPTIMIZE

3. DEPLOY



Model Optimizer

Converts and optimizes trained model using a supported framework

Read, Load, Infer

IR Data

Intermediate Representation (.xml, .bin)

Inference Engine

Common API that abstracts low-level programming for each hardware

- Post-Training Optimization Tool
- Deep Learning Workbench
- Deep Learning Streamer
- OpenCV
- OpenCL™
- Code Samples & Demos (e.g. Benchmark app, Accuracy Checker, Model Downloader)
- Deployment Manager

- CPU Plugin
- GPU Plugin
- GNA Plugin
- Myriad Plugin For Intel® NCS2 & NCS
- HDDL Plugin
- FGPA Plugin

AGENDA

- Intel® Smart Video/Computer vision Tools Overview
- Model Optimizer
- Inference Engine
- 15 Minute Break
- Accelerators based on Intel® Movidius™ Vision Processing Unit
- Accelerators based on Intel® Arria® FPGA
- Multiple Models in One Application
- DL Workbench + Demo
- DL Streamer
- Register for access to Intel® DevCloud for the Edge
- **Lab1 - DevCloud Sample Application: Accelerated Object Detection**
- **Lab2 – DevCloud Advanced Tutorials: DL Streamer Benchmark**

MODEL OPTIMIZER

INTEL® DEEP LEARNING DEPLOYMENT TOOLKIT

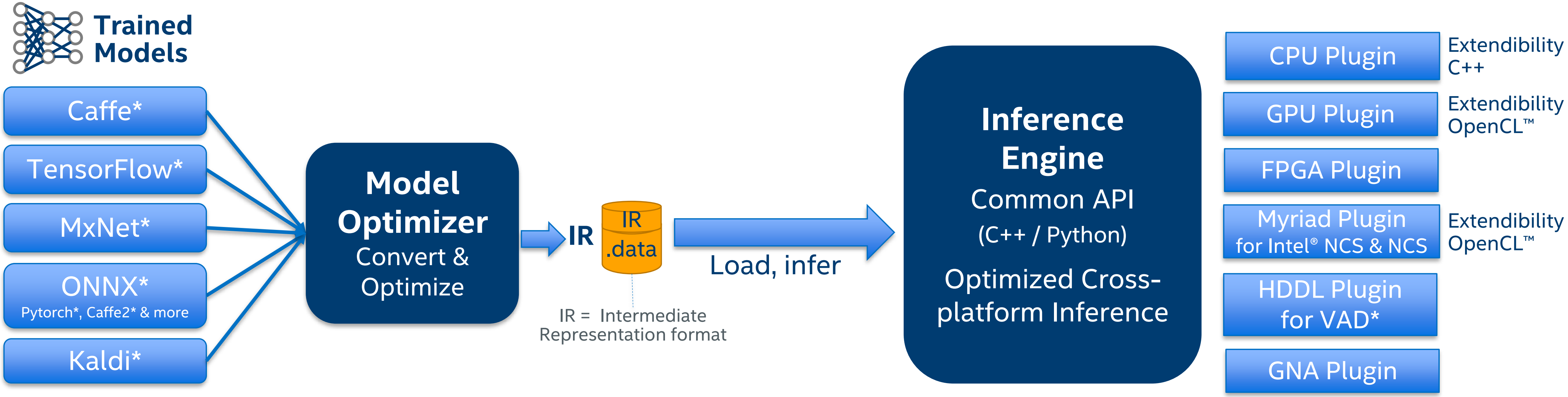
FOR DEEP LEARNING INFERENCE

Model Optimizer

- A Python* based tool to **import** trained models and **convert** them to Intermediate Representation
- **Optimizes for performance** or space with conservative topology transformations
- **Hardware-agnostic** optimizations

Inference Engine

- High-level, C/C++ and Python, inference **runtime API**
- Interface is implemented as **dynamically loaded plugins** for each hardware type
- Delivers advanced performance for each type **without requiring users to implement and maintain multiple code pathways**



GPU = Intel® CPU with integrated GPU/Intel® Processor Graphics, Intel® NCS = Intel® Neural Compute Stick (VPU)
*VAD = Intel® Vision Accelerator Design Products (HDDL-R)

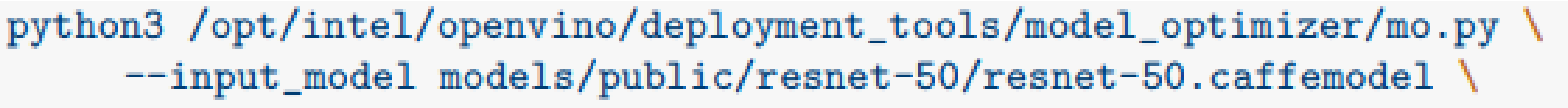
MODEL OPTIMIZER: GENERIC OPTIMIZATION

Model optimizer performs generic optimization

- Node merging
- Horizontal fusion
- Batch normalization to scale shift
- Fold scale shift with convolution
- Drop unused layers (dropout)

The simplest way to convert a model is to run mo.py with a path to the input model file

- By default, generic optimization will be automatically applied, unless manually set disable



```
python3 /opt/intel/openvino/deployment_tools/model_optimizer/mo.py \
     --input_model models/public/resnet-50/resnet-50.caffemodel \
```

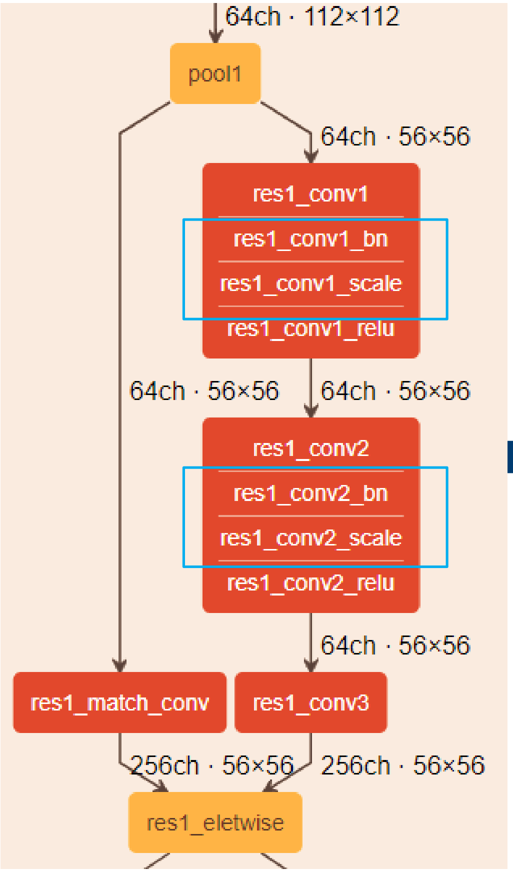
MODEL OPTIMIZATION TECHNIQUES

Linear Operation Fusing: 3 stages

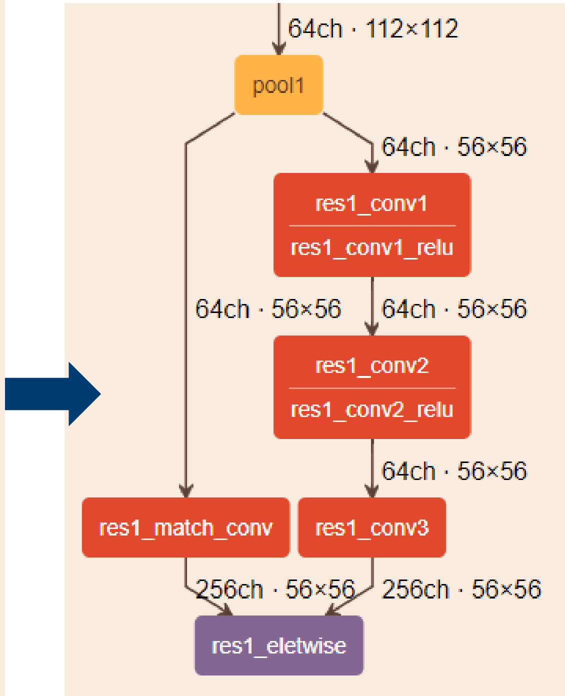
1. **BatchNorm and ScaleShift decomposition:** *BN* layers decomposes to *Mul->Add->Mul->Add* sequence; ScaleShift layers decomposes to *Mul->Add* sequence.

2. **Linear operations merge:** Merges sequences of Mul and Add operations to the **single** Mul->Add instance.

3. **Linear operations fusion:** Fuses Mul and Add operations to Convolution or FullybConnected layers.



Caffe* Resnet269 block (from Netscope)

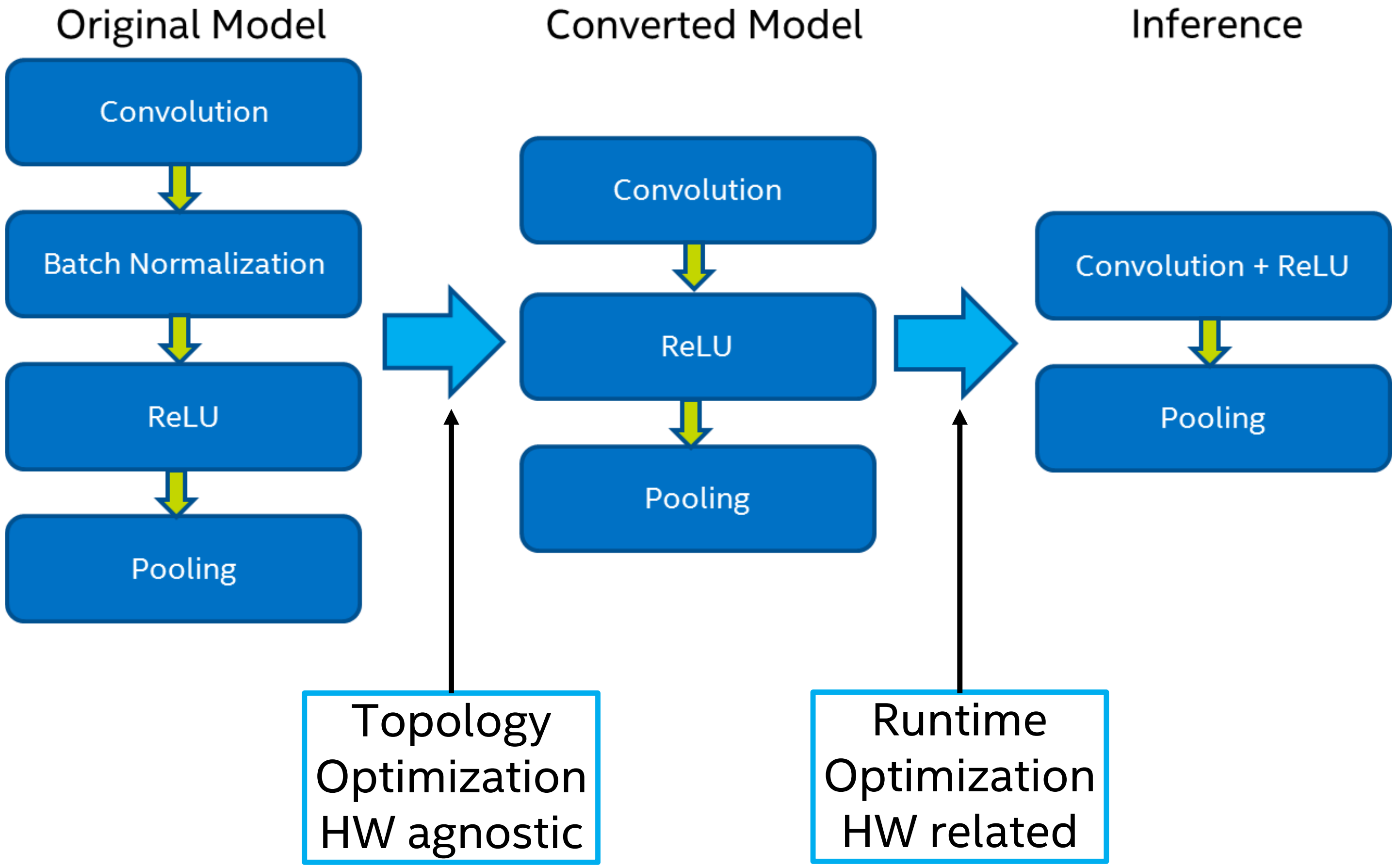


Merged Caffe* Resnet269 block
(from Netscope*)

MODEL OPTIMIZER: LINEAR OPERATION FUSING

Example

1. Remove Batch normalization stage.
2. Recalculate the weights to ‘include’ the operation.
3. Merge Convolution and ReLU into one optimized kernel.



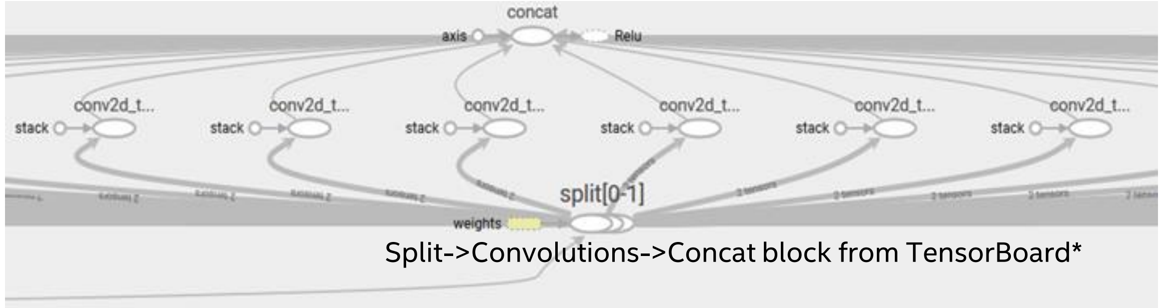
MODEL OPTIMIZER: FRAMEWORK OR TOPOLOGY SPECIFIC OPTIMIZATION

Grouped Convolutions Fusing

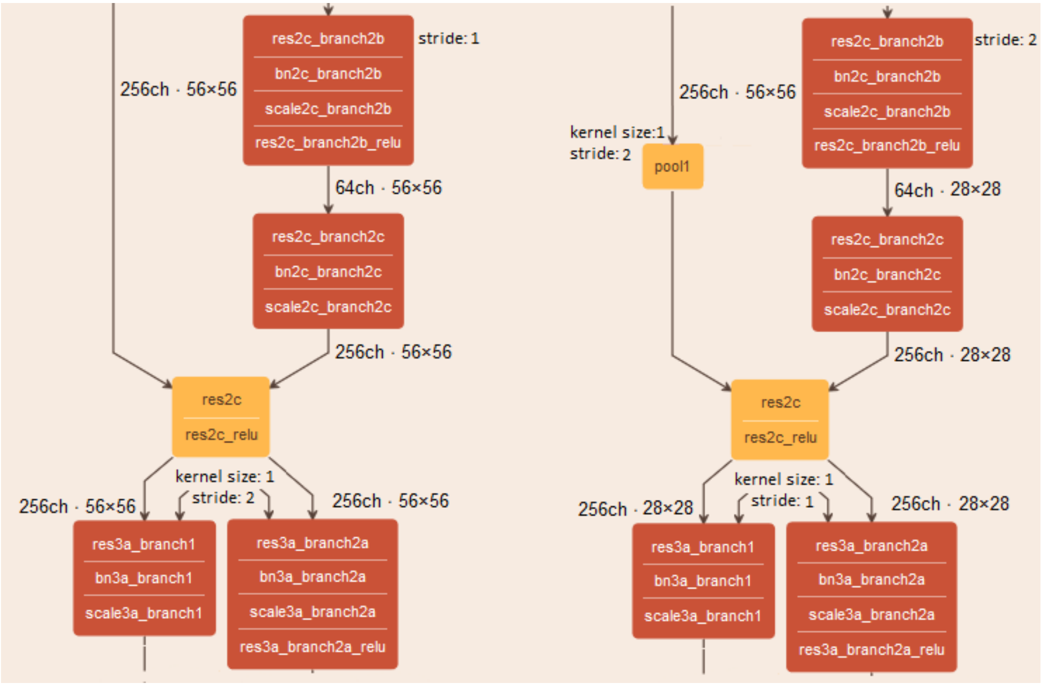
- Grouped convolution fusing is a specific optimization that applies for TensorFlow* topologies. The main idea of this optimization is to combine convolutions results for the Split outputs and then recombine them using **Concat** operation in the same order as they were out from **Split**.

ResNet* optimization (stride optimization)

- This optimization is to move the stride that is greater than 1 from Convolution layers with the kernel size = 1 to upper Convolution layers. In addition, the Model Optimizer adds a Pooling layer to align the input shape for a Eltwise layer, if it was changed during the optimization.



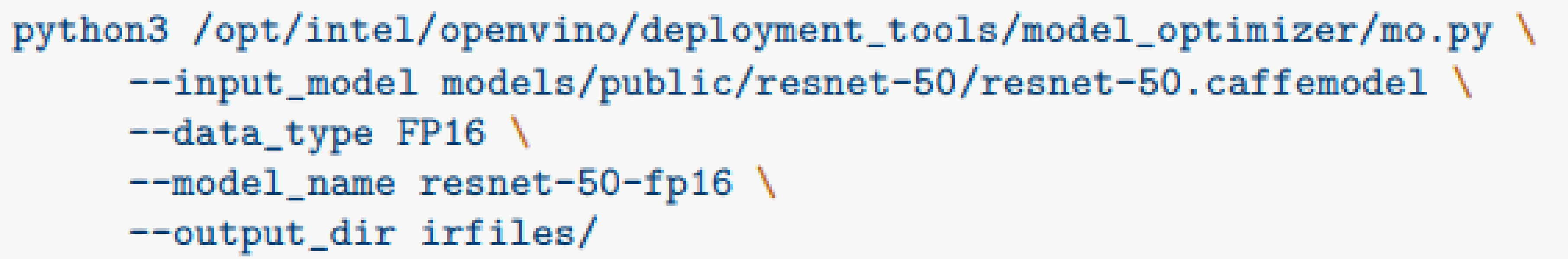
Split->Convolutions->Concat block from TensorBoard*



MODEL OPTIMIZER: QUANTIZATION

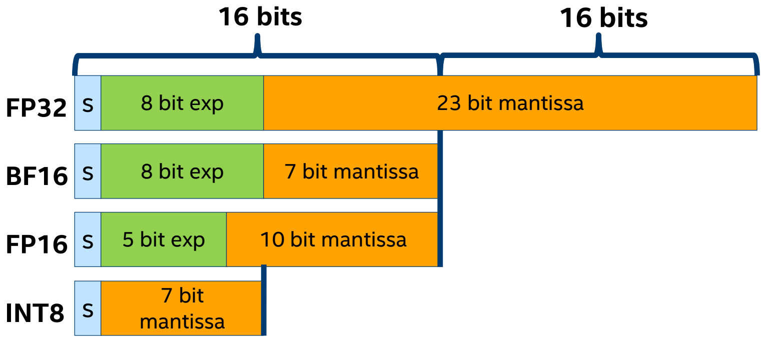
--data_type {FP16,FP32,half,float}

- 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.



```
python3 /opt/intel/openvino/deployment_tools/model_optimizer/mo.py \
     --input_model models/public/resnet-50/resnet-50.caffemodel \
     --data_type FP16 \
     --model_name resnet-50-fp16 \
     --output_dir irfiles/
```

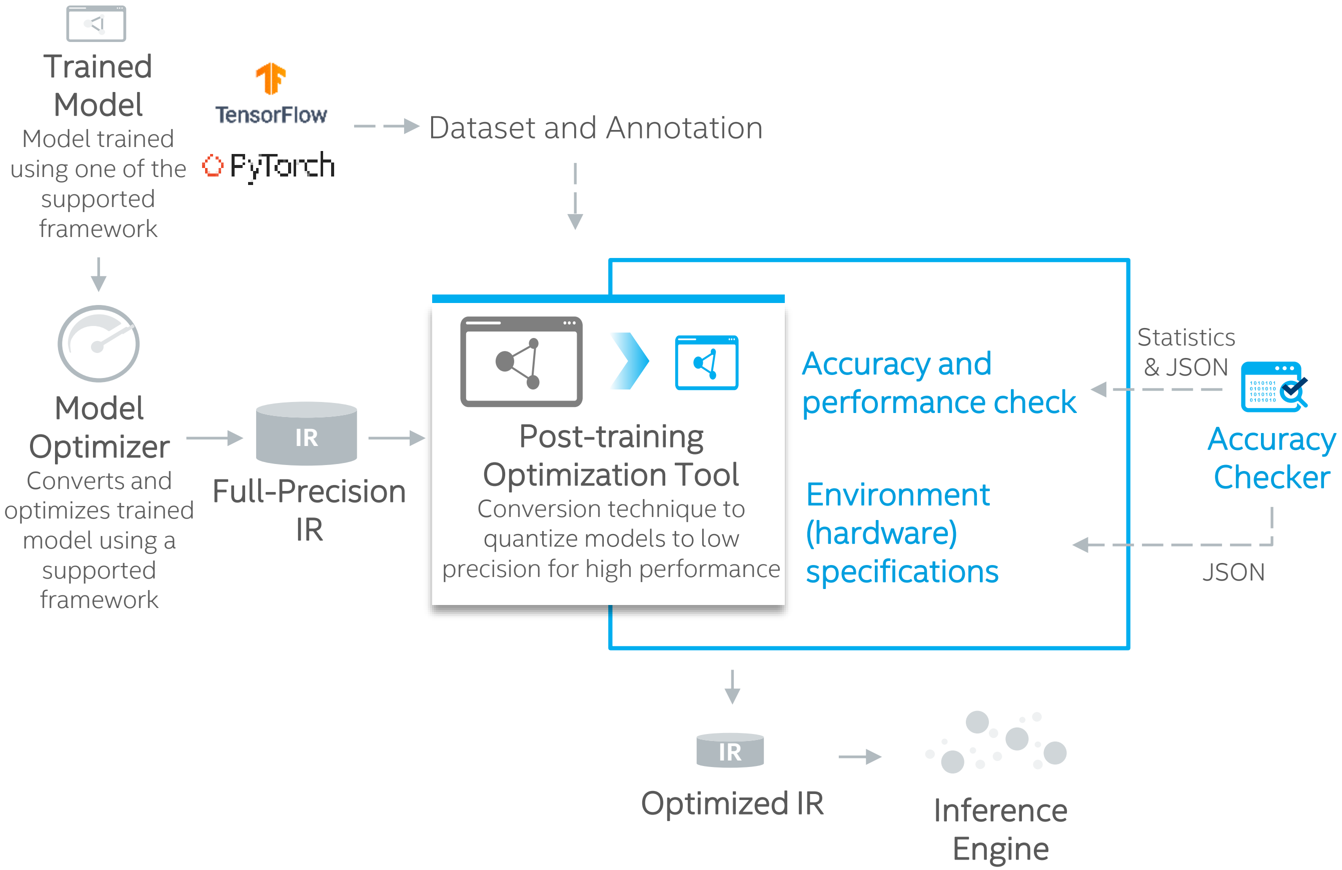
PLUGIN	FP32	FP16	INT8
CPU plugin	Supported and preferred	Supported	Supported
GPU plugin	Supported	Supported and preferred	Supported*
VPU plugins	Not supported	Supported	Not supported
GNA plugin	Supported	Supported	Not supported
FPGA plugin	Supported	Supported	Not supported



Note:
1. To create INT8 models, you will need DL Workbench or Post Training Optimization Tool
2. FPGA also support FP11, convert happens on FPGA

Post-Training Optimization Tool

- Using the Python API, the Post-training Optimization Tool integrates with the Model Optimizer, DL Workbench and accuracy checker tools to streamline the development process
- Enables a conversion technique of deep learning model that **reduces model size into low precision data types**, such as INT8, without re-training
- Reduces model size **while also improving latency, with little degradation** in model accuracy and without model re-training.
- Different optimization approaches are supported: quantization algorithms, sparsity, etc.


Trained Model
Model trained using one of the supported framework
TensorFlow
PyTorch
Dataset and Annotation
Model Optimizer
Converts and optimizes trained model using a supported framework
IR
Full-Precision IR
Post-training Optimization Tool
Conversion technique to quantize models to low precision for high performance
Accuracy and performance check
Environment (hardware) specifications
Statistics & JSON
Accuracy Checker
JSON
IR
Optimized IR
Inference Engine

Performance Benchmarks ▸
https://docs.openvinotoolkit.org/latest/_docs_performance_int8_vs_fp32.html

Optimization Notice

SPEED UP DEVELOPMENT WITH OPEN SOURCE RESOURCES

Open source resources with pre-trained models, samples and demos

Computer Vision

[Object detection](https://github.com/opencv/open_model_zoo) [Human pose estimation]

[Object recognition] [Image processing]

[Reidentification] [Action recognition]

Volumetric segmentation

Image super resolution

[Semantic segmentation]

[Instance segmentation]

3D reconstruction

Audio, Speech, Language

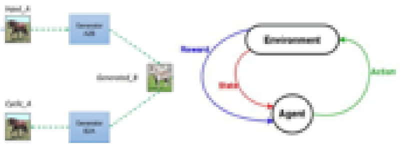
Language processing

Speech to text

[Text detection]

[Text recognition]

Natural Language Processing



Other

(Data Generation, Reinforcement Learning)

[Compressed models]

[Image retrieval]

And more..

Model Downloader

- Provides an easy way of accessing a number of public models as well as a set of pre-trained Intel models

Accuracy Checker

- Check for accuracy of the model (original and after conversion) to IR file using a known data set

PRE-TRAINED MODELS

https://github.com/opencv/open_model_zoo

AGENDA

- Intel® Smart Video/Computer vision Tools Overview
- Model Optimizer
- Inference Engine
- 15 Minute Break
- Accelerators based on Intel® Movidius™ Vision Processing Unit
- Accelerators based on Intel® Arria® FPGA
- Multiple Models in One Application
- DL Workbench + Demo
- DL Streamer
- Register for access to Intel® DevCloud for the Edge
- **Lab1 - DevCloud Sample Application: Accelerated Object Detection**
- **Lab2 – DevCloud Advanced Tutorials: DL Streamer Benchmark**

INFERENCE ENGINE

INTEL® DEEP LEARNING DEPLOYMENT TOOLKIT

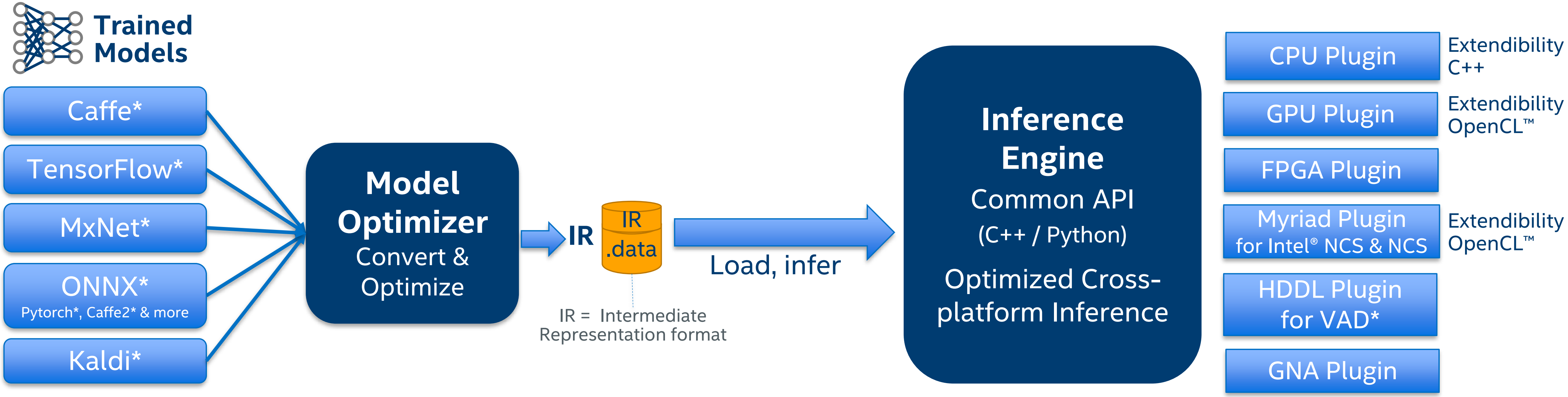
FOR DEEP LEARNING INFERENCE

Model Optimizer

- A Python* based tool to **import** trained models and **convert** them to Intermediate Representation
- **Optimizes for performance** or space with conservative topology transformations
- **Hardware-agnostic** optimizations

Inference Engine

- High-level, C/C++ and Python, inference **runtime API**
- Interface is implemented as **dynamically loaded plugins** for each hardware type
- Delivers advanced performance for each type **without requiring users to implement and maintain multiple code pathways**



GPU = Intel® CPU with integrated GPU/Intel® Processor Graphics, Intel® NCS = Intel® Neural Compute Stick (VPU)
*VAD = Intel® Vision Accelerator Design Products (HDDL-R)

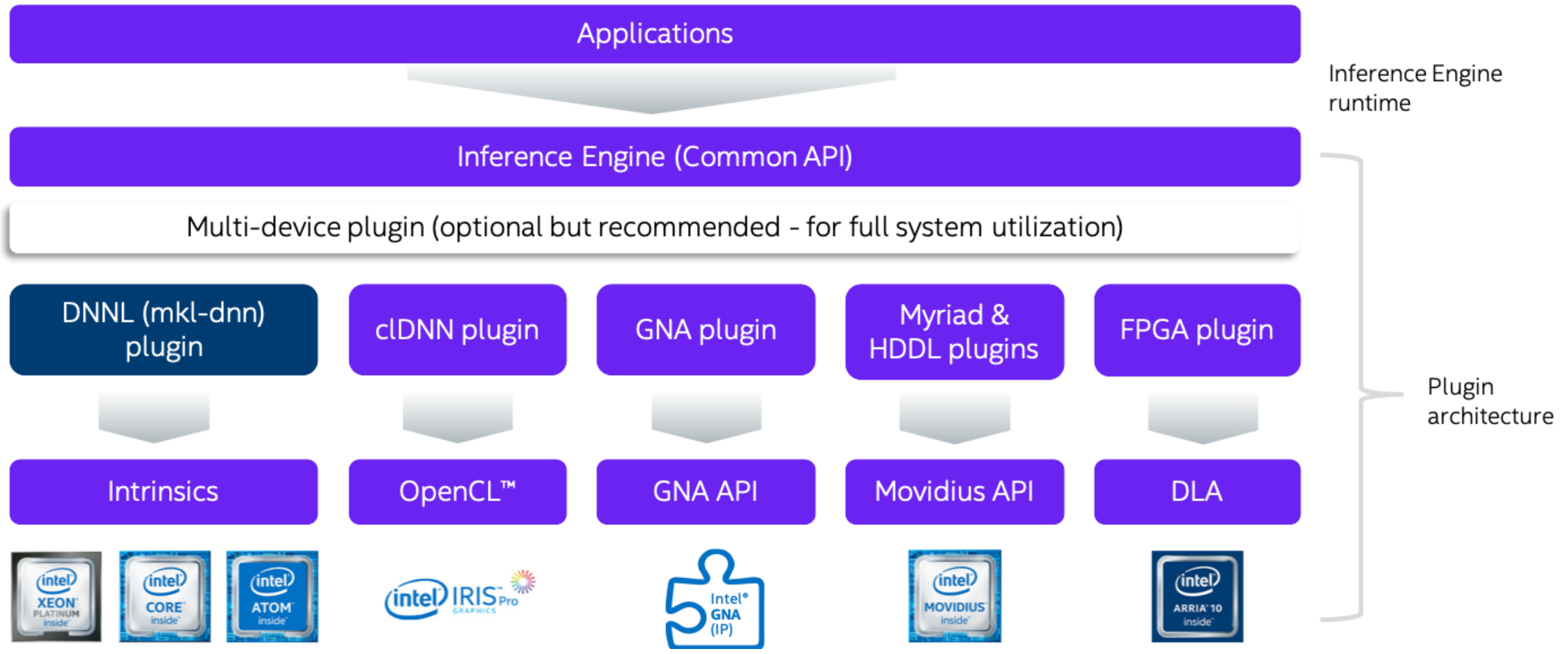
OPTIMAL MODEL PERFORMANCE USING THE INFERENCE ENGINE

Core Inference Engine Libraries

- Create Inference Engine Core object to work with devices
- Read the network
- Manipulate network information
- Execute and pass inputs and outputs

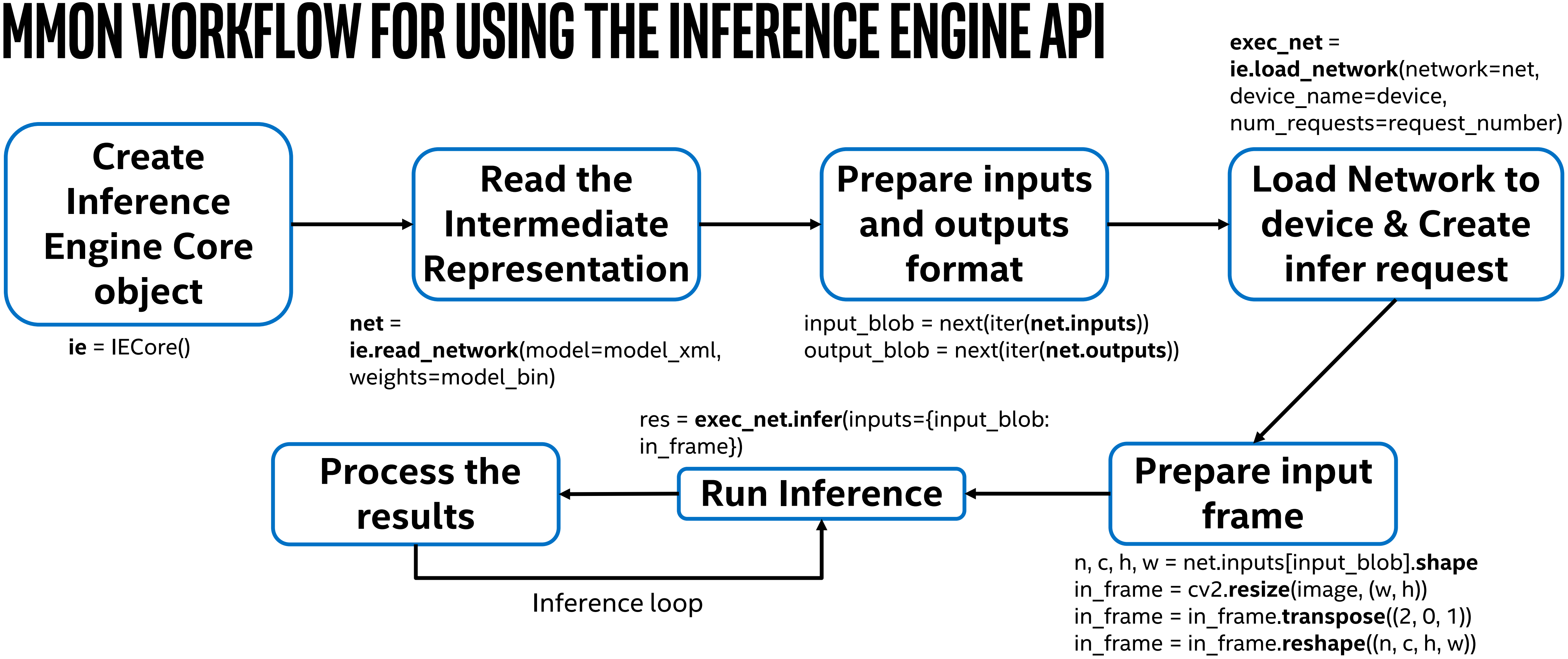
Device-specific Plugin Libraries

- For each supported target device, Inference Engine provides a plugin — a DLL/shared library that contains complete implementation for inference on this device.



GPU = Intel CPU with integrated graphics/Intel® Processor Graphics/GEN
GNA = Gaussian mixture model and Neural Network Accelerator

COMMON WORKFLOW FOR USING THE INFERENCE ENGINE API



Create Inference Engine Core object

ie = IECore()

Read the Intermediate Representation

net =
ie.read_network(model=model_xml,
weights=model_bin)

Prepare inputs and outputs format

input_blob = next(iter(**net.inputs**))
output_blob = next(iter(**net.outputs**))

Load Network to device & Create infer request

exec_net =
ie.load_network(network=net,
device_name=device,
num_requests=request_number)

Prepare input frame

n, c, h, w = net.inputs[input_blob]**.shape**
in_frame = cv2.**resize**(image, (w, h))
in_frame = in_frame.**transpose**((2, 0, 1))
in_frame = in_frame.**reshape**((n, c, h, w))

Run Inference

res = **exec_net.infer**(inputs={input_blob:
in_frame})

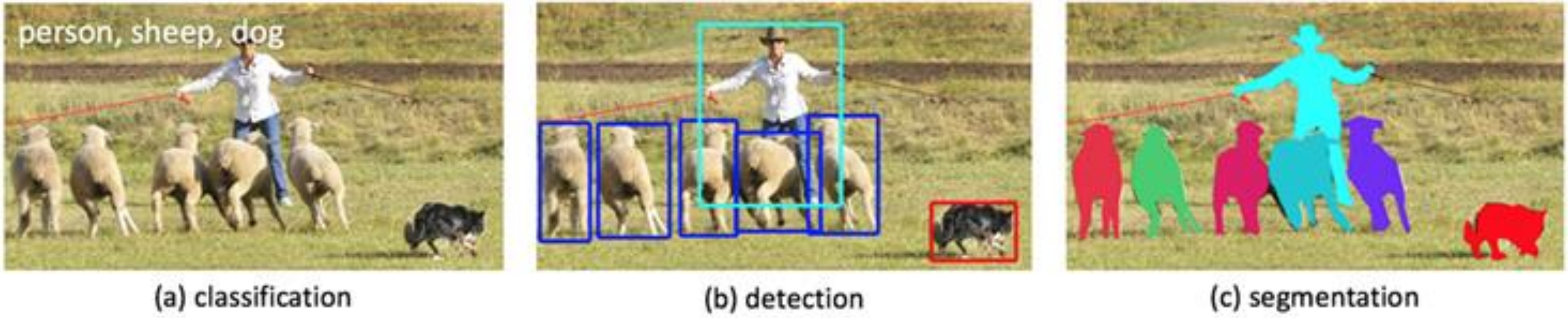
Process the results

Inference loop

http://docs.openvinotoolkit.org/latest/_docs_IE_DG_Integrate_with_customer_application_new_API.html

INFERENCE ON AN INTEL® EDGE SYSTEM

Many deep learning networks are available—choose the one you need.

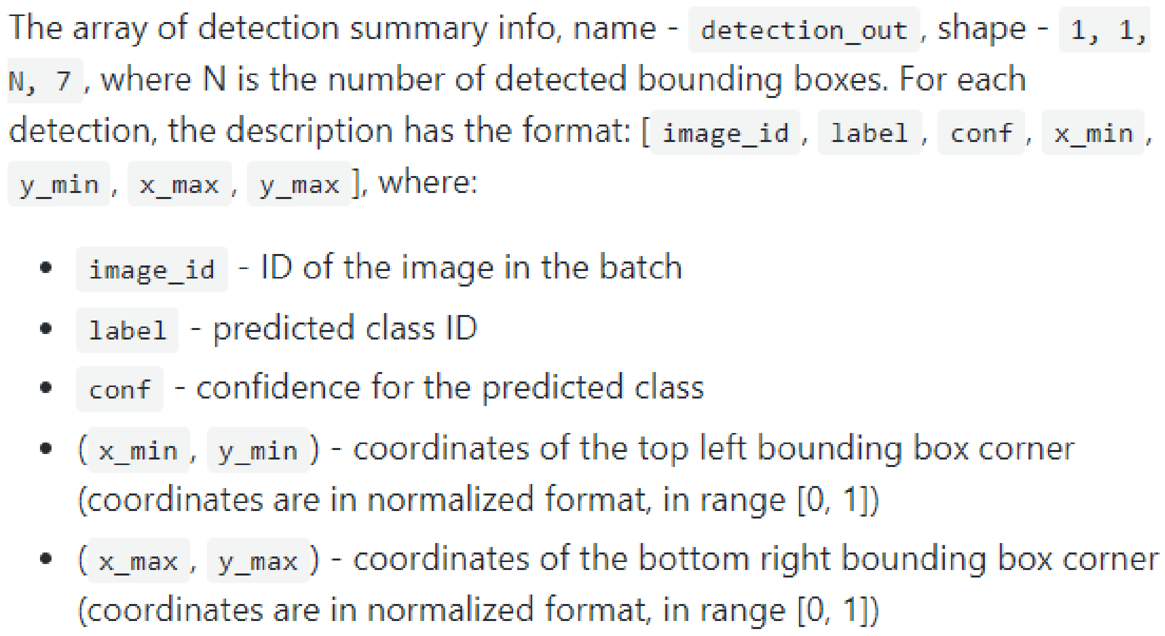


The complexity of the problem (data set) dictates the network structure. The more complex the problem, the more 'features' required, the deeper the network.

PROCESS THE RESULTS

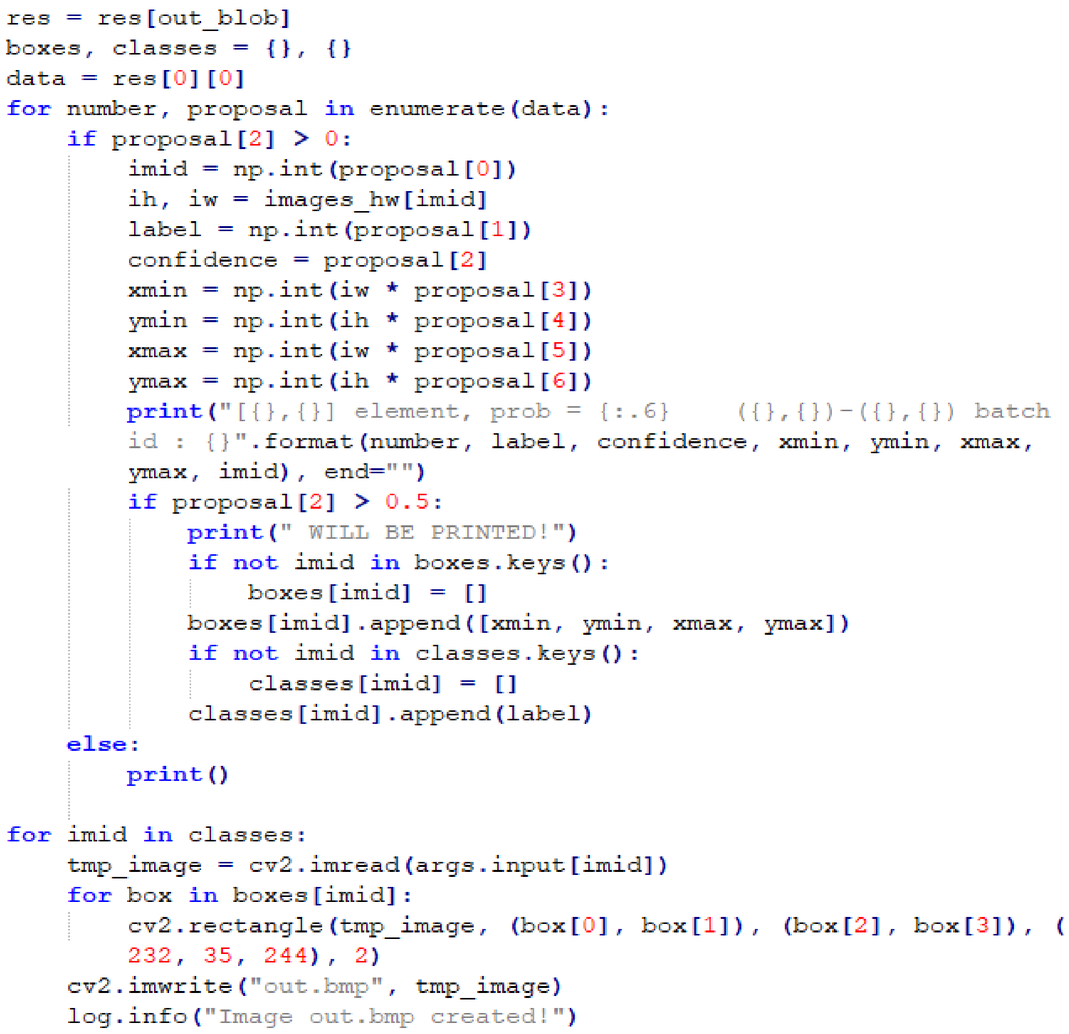
OBJECT DETECTION SSD EXAMPLE

Process the results (Post-processing)



The array of detection summary info, name - `detection_out` , shape - `1, 1, N, 7` , where N is the number of detected bounding boxes. For each detection, the description has the format: [`image_id` , `label` , `conf` , `x_min` , `y_min` , `x_max` , `y_max`], where:

- `image_id` - ID of the image in the batch
- `label` - predicted class ID
- `conf` - confidence for the predicted class
- (`x_min` , `y_min`) - coordinates of the top left bounding box corner (coordinates are in normalized format, in range [0, 1])
- (`x_max` , `y_max`) - coordinates of the bottom right bounding box corner (coordinates are in normalized format, in range [0, 1])



```python
res = res[out_blob]
boxes, classes = {}, {}
data = res[0][0]
for number, proposal in enumerate(data):
    if proposal[2] > 0:
        imid = np.int(proposal[0])
        ih, iw = images_hw[imid]
        label = np.int(proposal[1])
        confidence = proposal[2]
        xmin = np.int(iw * proposal[3])
        ymin = np.int(ih * proposal[4])
        xmax = np.int(iw * proposal[5])
        ymax = np.int(ih * proposal[6])
        print("[{},{}] element, prob = {:.6}    ({},{})-({},{}) batch
        id : {}".format(number, label, confidence, xmin, ymin, xmax,
        ymax, imid), end="")
        if proposal[2] > 0.5:
            print(" WILL BE PRINTED!")
            if not imid in boxes.keys():
                boxes[imid] = []
            boxes[imid].append([xmin, ymin, xmax, ymax])
            if not imid in classes.keys():
                classes[imid] = []
            classes[imid].append(label)
    else:
        print()

for imid in classes:
    tmp_image = cv2.imread(args.input[imid])
    for box in boxes[imid]:
        cv2.rectangle(tmp_image, (box[0], box[1]), (box[2], box[3]), (
        232, 35, 244), 2)
    cv2.imwrite("out.bmp", tmp_image)
    log.info("Image out.bmp created!")
```

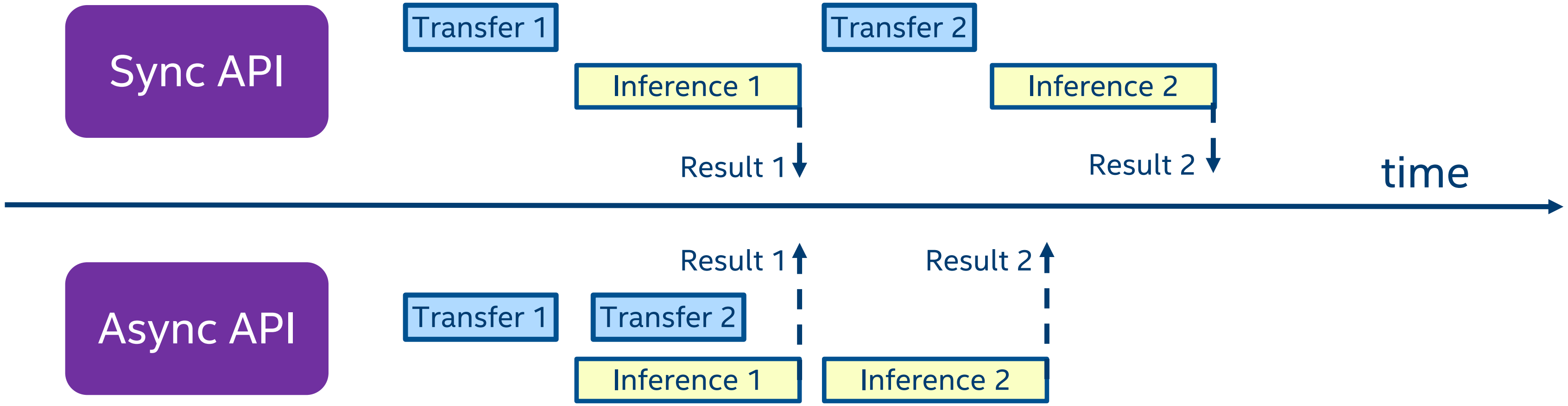
INFERENCE ENGINE

Synchronous vs Asynchronous Execution

In IE API model can be executed by **Infer Request** which can be:

- **Synchronous -** blocks until inference is completed.
  - exec_net.infer(inputs = {input_blob: in_frame})
- **Asynchronous** – checks the execution status with the wait, or specify a completion callback *(recommended way)*.
  - exec_net.start_async(request_id = id, inputs={input_blob: in_frame})
  - If exec_net.requests[id].wait() != 0

    do something



[Optimization Notice](#)

INFERENCE ENGINE

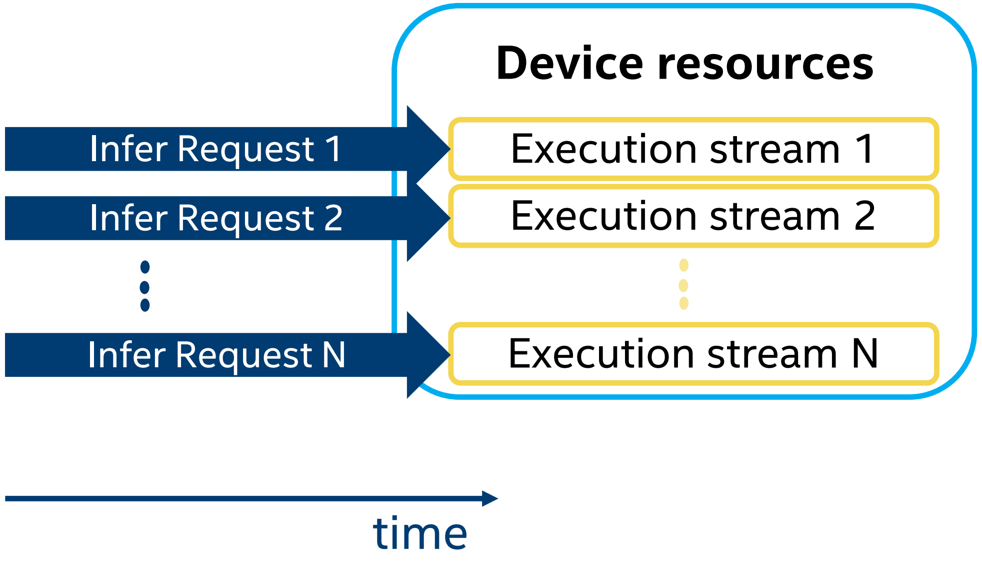
Throughput Mode for CPU, iGPU and VPU

Latency – inference time of 1 frame (ms).

Throughput – overall amount of frames inferred per 1 second (FPS)

“Throughput" mode allows the Inference Engine to efficiently run multiple infer requests simultaneously, greatly improving the overall throughput.

Device resources are divided into execution “**streams**” – parts which runs infer requests in parallel



CPU Example:
ie = IECore()
ie.GetConfig(CPU, KEY_CPU_THROUGHPUT_STREAMS)

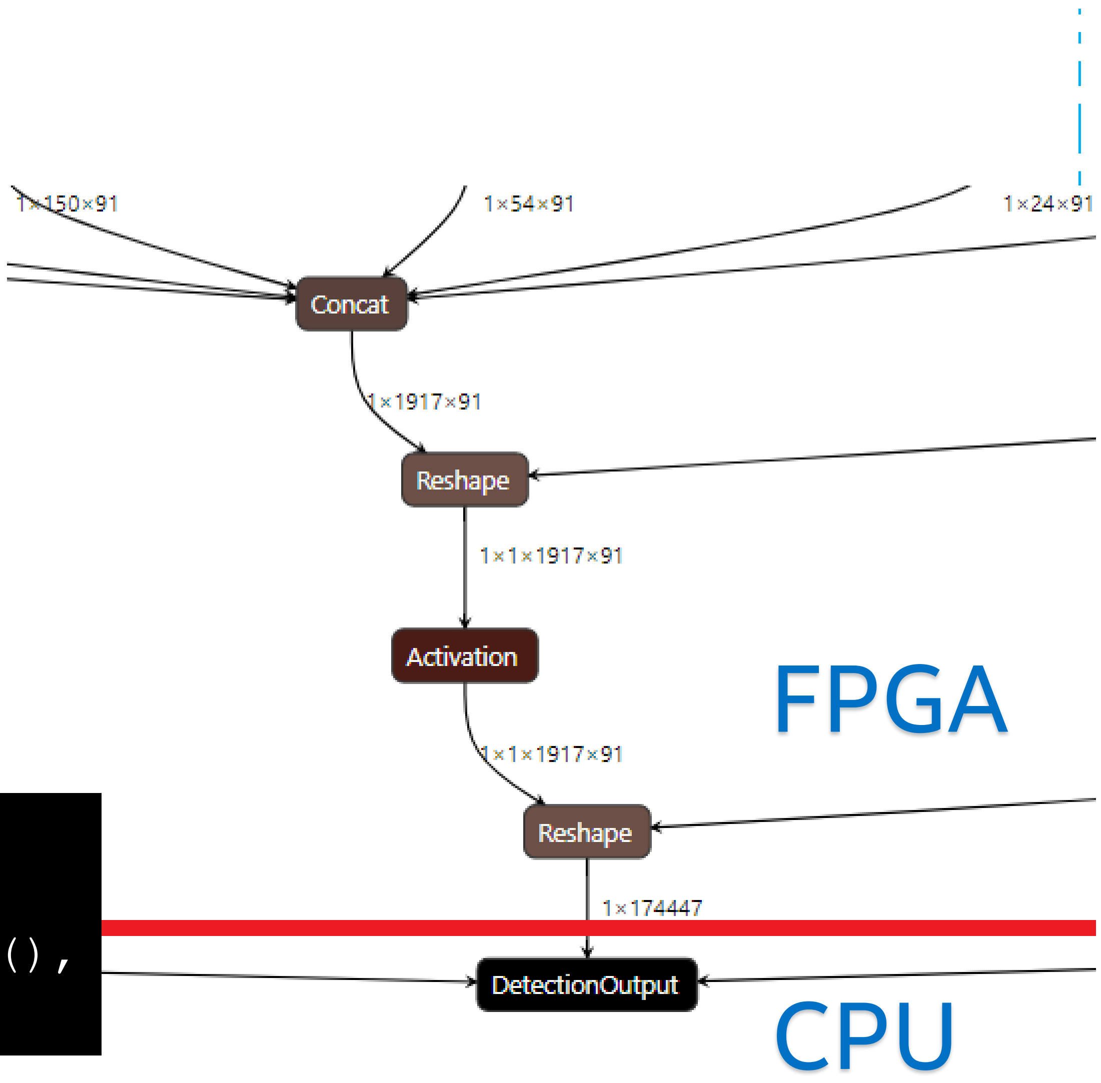
Optimization Notice

INFERENCE ENGINE

Heterogeneous Support

- You can execute different layers on different HW units
- Offload unsupported layers on fallback devices:
  - Default affinity policy
  - Setting affinity manually (`CNNLayer::affinity`)
- All device combinations are supported (CPU, GPU, FPGA, MYRIAD, HDDL)
- Samples/demos usage “`-d HETERO:FPGA,CPU`”

```
InferenceEngine::Core core;
      auto executable_network =
      core.LoadNetwork(reader.getNetwork(),
      "HETERO:FPGA,CPU");
```

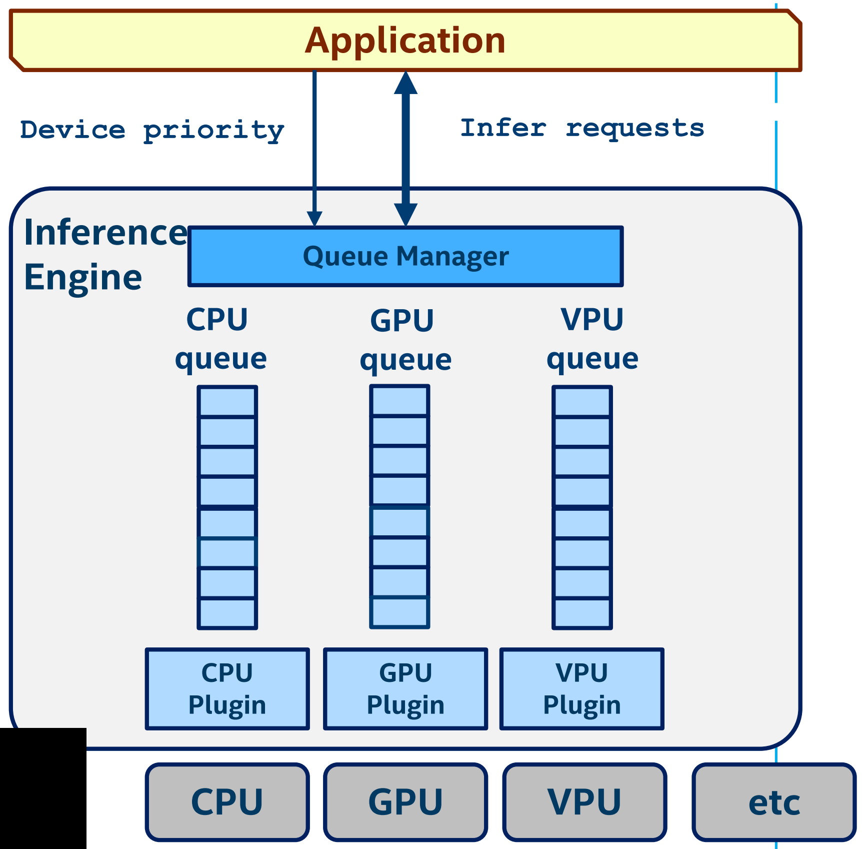


INFERENCE ENGINE

Multi-device Support

Automatic load-balancing between devices (inference requests level) for full system utilization

- Any combinations of the following devices are supported (CPU, iGPU, VPU, HDDL)
- As easy as “-d MULTI:CPU,GPU” for cmd-line option of your favorite sample/demo
- C++ example (Python is similar)



```
Core ie;
ExecutableNetwork exec =
ie.LoadNetwork(network,{{“DEVICE_PRIORITIES”, “CPU,GPU”}}, “MULTI”)
```

SPEED UP DEVELOPMENT WITH OPEN SOURCE RESOURCES

Open source resources with pre-trained models, demos, and tools

The Open Model Zoo demo applications are console applications that demonstrate how you can use your applications to solve specific use-cases.

Smart Classroom
Recognition and action detection demo for classroom settings

Multi-Camera, Multi-Person
Tracking multiple people on multiple cameras for public safety use cases

Gaze Estimation
Face detection followed by gaze estimation, head pose estimation and facial landmarks regression.

Weld Porosity Detection
Demonstrates how to find defects in welding

Person Inpainting
Removes unwanted people in images or videos

And more..

DEMO APPLICATIONS

https://github.com/opencv/open_model_zoo

15 MINS BREAK

Survey: https://bit.ly/VINOsurvey

Download the Intel® Distribution of OpenVINO(TM) toolkit
https://software.intel.com/content/www/us/en/develop/tools/openvino-toolkit/choose-download.html

Intel® Edge Software Hub – Edge Computing Software and Packages
https://www.intel.com/content/www/us/en/edge-computing/edge-software-hub.html

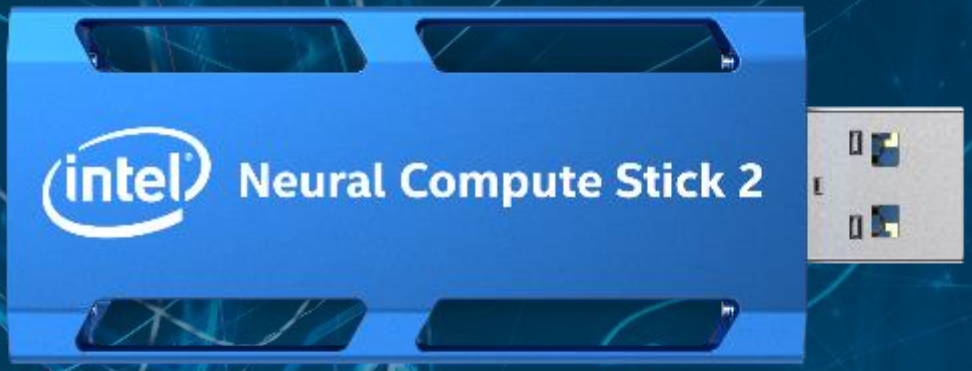
Schedule for the Intel® Distribution of OpenVINO™ Toolkit Virtual Workshops
https://software.seek.intel.com/OpenVINOworkshops

Go to Market with the Intel® Distribution of OpenVINO™ Toolkit
https://software.intel.com/content/www/us/en/develop/topics/iot/training/go-to-market-with-openvino.html

AGENDA

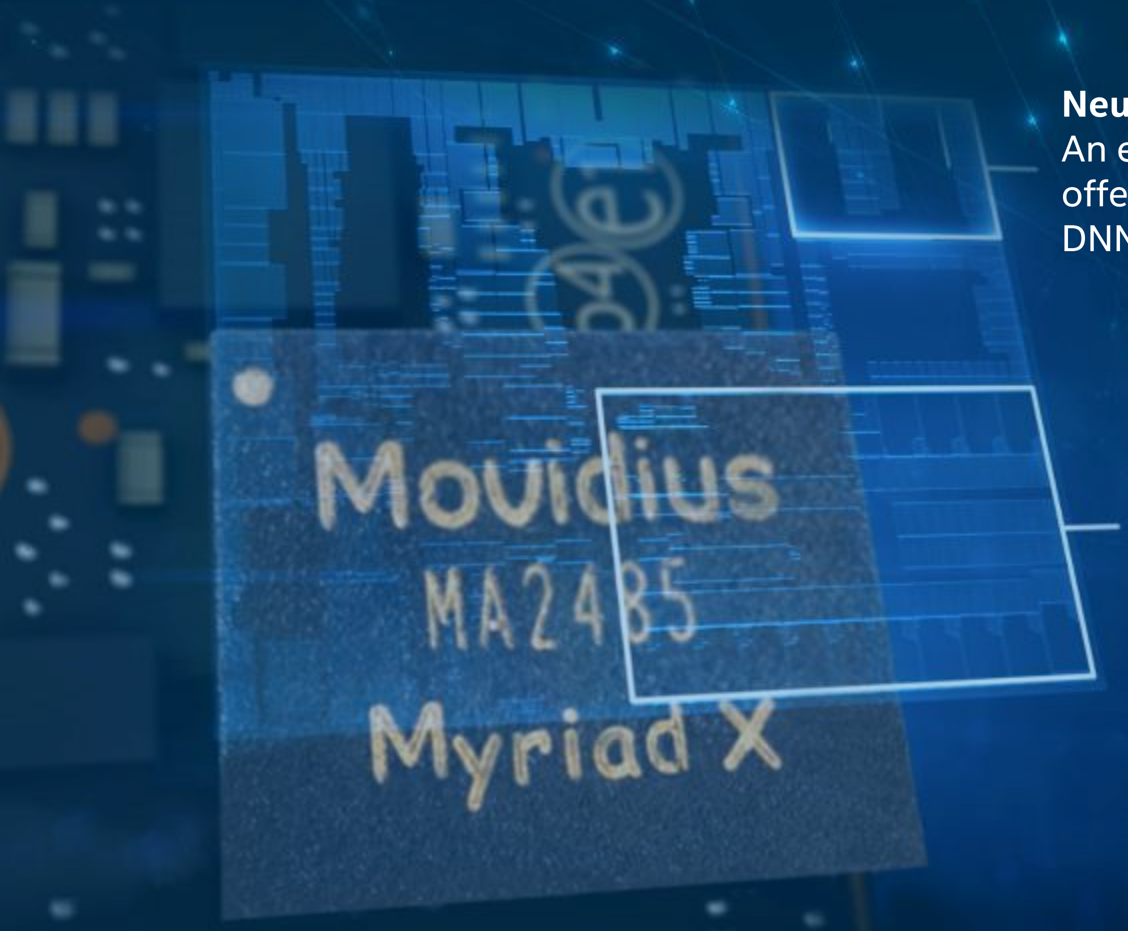
- Intel® Smart Video/Computer vision Tools Overview
- Model Optimizer
- Inference Engine
- 15 Minute Break
- Accelerators based on Intel® Movidius™ Vision Processing Unit
- Accelerators based on Intel® Arria® FPGA
- Multiple Models in One Application
- DL Workbench + Demo
- DL Streamer
- Register for access to Intel® DevCloud for the Edge
- **Lab1 - DevCloud Sample Application: Accelerated Object Detection**
- **Lab2 – DevCloud Advanced Tutorials: DL Streamer Benchmark**

REDEFINING THE AI DEVELOPMENT KIT
INTEL® NEURAL COMPUTE STICK 2



Vision Processing Unit (VPU)	Intel® Movidius™ Myriad™ X VPU
Software Development Kit	Intel® Distribution of OpenVINO™ toolkit
Operating Software Support	Ubuntu* 16.04 or 18.04 LTS (64 bit), Windows® 10 (64 bit), CentOS* 7.4 (64 bit), macOS* 10.4.4, Raspbian*, and other via the open-source distribution of OpenVINO™ toolkit
Supported Framework	TensorFlow*, Caffe*, MXNet*, ONNX*, and PyTorch* / PaddlePaddle* via ONNX* conversion
Connectivity	USB 3.1 Type-A
Dimensions	72.5mm X 27mm X 14mm
Operating Temperature	0° - 40° C
Material Master Number	964486
MSRP	$69 as of July 14th 2019

NEXT GENERATION AI INFERENCE
INTEL® MOVIDIUS™ MYRIAD™ X VPU

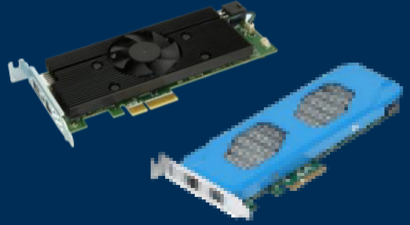


Neural Compute Engine
An entirely new deep neural network (DNN) inferencing engine that offers flexible interconnect and ease of configuration for on-device DNNs and computer vision applications

16 SHAVE Cores
VLIW (DSP) programmable processors are optimized for complex vision & imaging workloads

EXAMPLES OF INTEL® VISION ACCELERATOR DESIGN PRODUCTS

Accelerators based on Intel® Movidius™ VPU

EXAMPLE CARD BASED ON VISION ACCELERATOR DESIGNS	 1 Intel® Movidius™ VPU	 2 Intel® Movidius™ VPUs	 8 Intel® Movidius™ VPUs
INTERFACE	M.2, Key E	miniPCIe**	PCIe x4
CURRENTLY MANUFACTURED BY*	AAEON an ASUS assoc. co. ADLINK Leading EDGE COMPUTING ADVANTECH Enabling an Intelligent Planet iEi JWIPC NEXCOM tinyGO uzel		
SOFTWARE TOOLS	**INTEL® DISTRIBUTION OF OPENVINO™ TOOLKIT** Develop NN Model; Deploy across Intel® CPU, GPU, VPU, FPGA; Leverage common algorithms		

*Please contact Intel representative for complete list of ODM manufacturers. Other names and brands may be claimed as the property of others.

Optimization Notice

Click here for Latest Publicly Posted Benchmarks

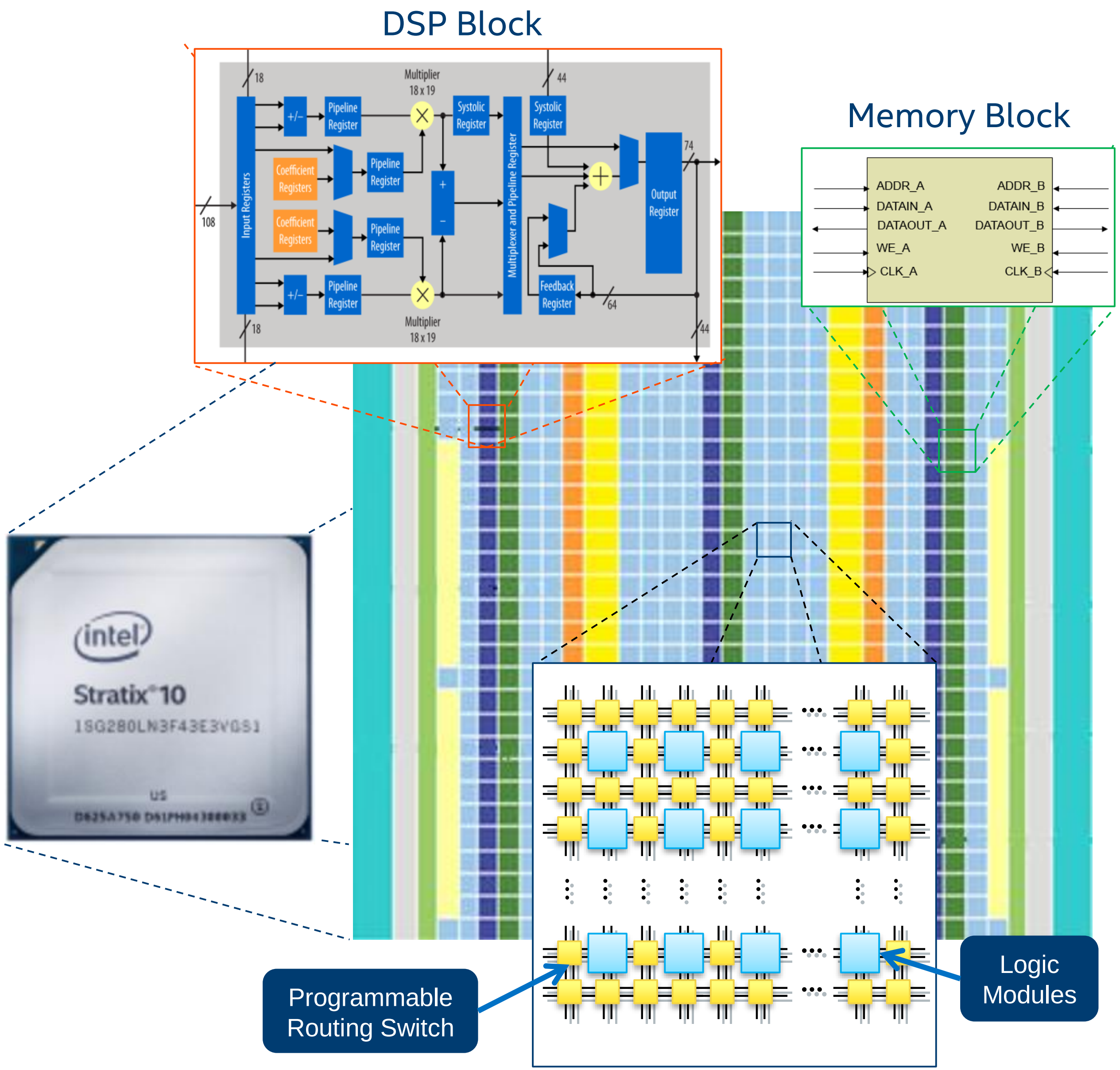
Click here for Programing Guide for Use with Intel® Distribution of OpenVINO toolkit

AGENDA

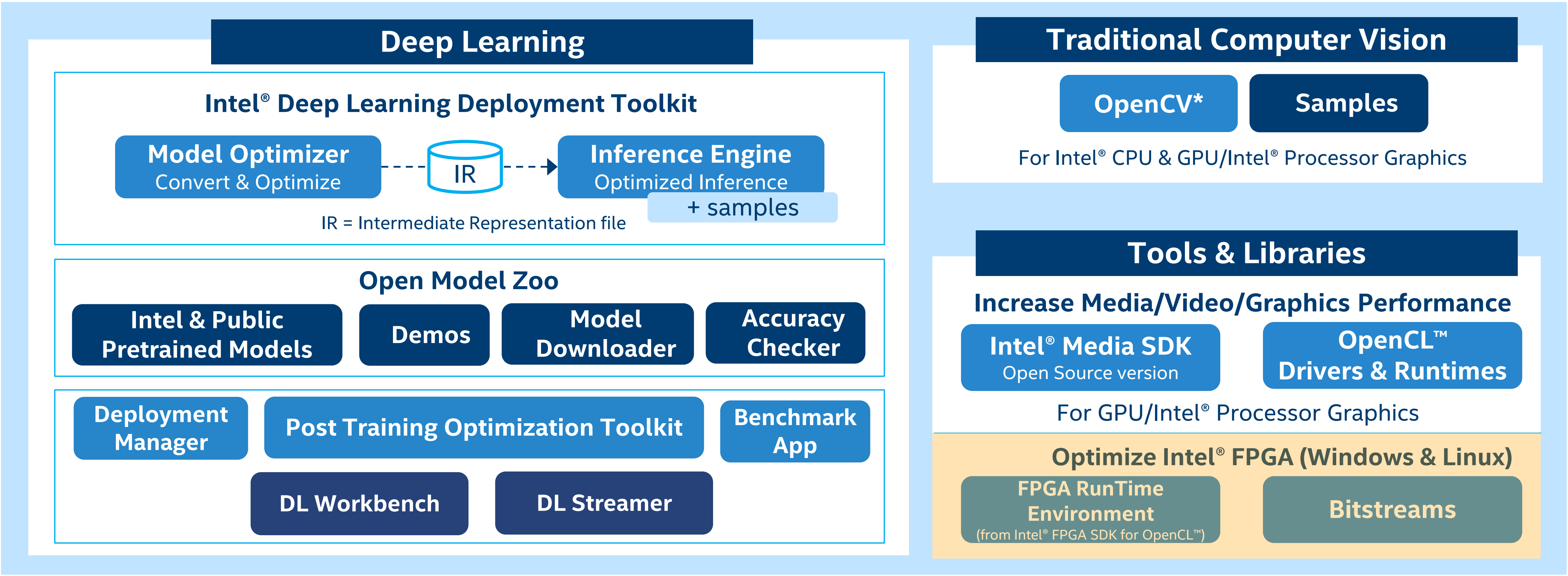
- Intel® Smart Video/Computer vision Tools Overview
- Model Optimizer
- Inference Engine
- 15 Minute Break
- Accelerators based on Intel® Movidius™ Vision Processing Unit
- Accelerators based on Intel® Arria® FPGA
- Multiple Models in One Application
- DL Workbench + Demo
- DL Streamer
- Register for access to Intel® DevCloud for the Edge
- **Lab1 - DevCloud Sample Application: Accelerated Object Detection**
- **Lab2 – DevCloud Advanced Tutorials: DL Streamer Benchmark**

FPGA OVERVIEW

- Field Programmable Gate Array (FPGA)
  - Millions of logic elements
  - Thousands of embedded memory blocks
  - Thousands of DSP blocks
  - Programmable routing
  - High speed transceivers
  - Various built-in hardened IP
- Used to create **Custom Hardware!**



INSIDE INTEL® DISTRIBUTION OF OPENVINO™ TOOLKIT



Deep Learning

Intel® Deep Learning Deployment Toolkit

Model Optimizer
Convert & Optimize

IR

Inference Engine
Optimized Inference

+ samples

IR = Intermediate Representation file

Open Model Zoo

Intel & Public Pretrained Models

Demos

Model Downloader

Accuracy Checker

Deployment Manager

Post Training Optimization Toolkit

Benchmark App

DL Workbench

DL Streamer

Traditional Computer Vision

OpenCV*

Samples

For Intel® CPU & GPU/Intel® Processor Graphics

Tools & Libraries

Increase Media/Video/Graphics Performance

Intel® Media SDK
Open Source version

OpenCL™ Drivers & Runtimes

For GPU/Intel® Processor Graphics

Optimize Intel® FPGA (Windows & Linux)

FPGA RunTime Environment
(from Intel® FPGA SDK for OpenCL™)

Bitstreams

OS Support: CentOS* 7.4 (64 bit), Ubuntu* 16.04.3 LTS (64 bit), Microsoft Windows® 10 (64 bit), Yocto Project* version Poky Jethro v2.0.3 (64 bit), macOS* 10.13 & 10.14 (64 bit)

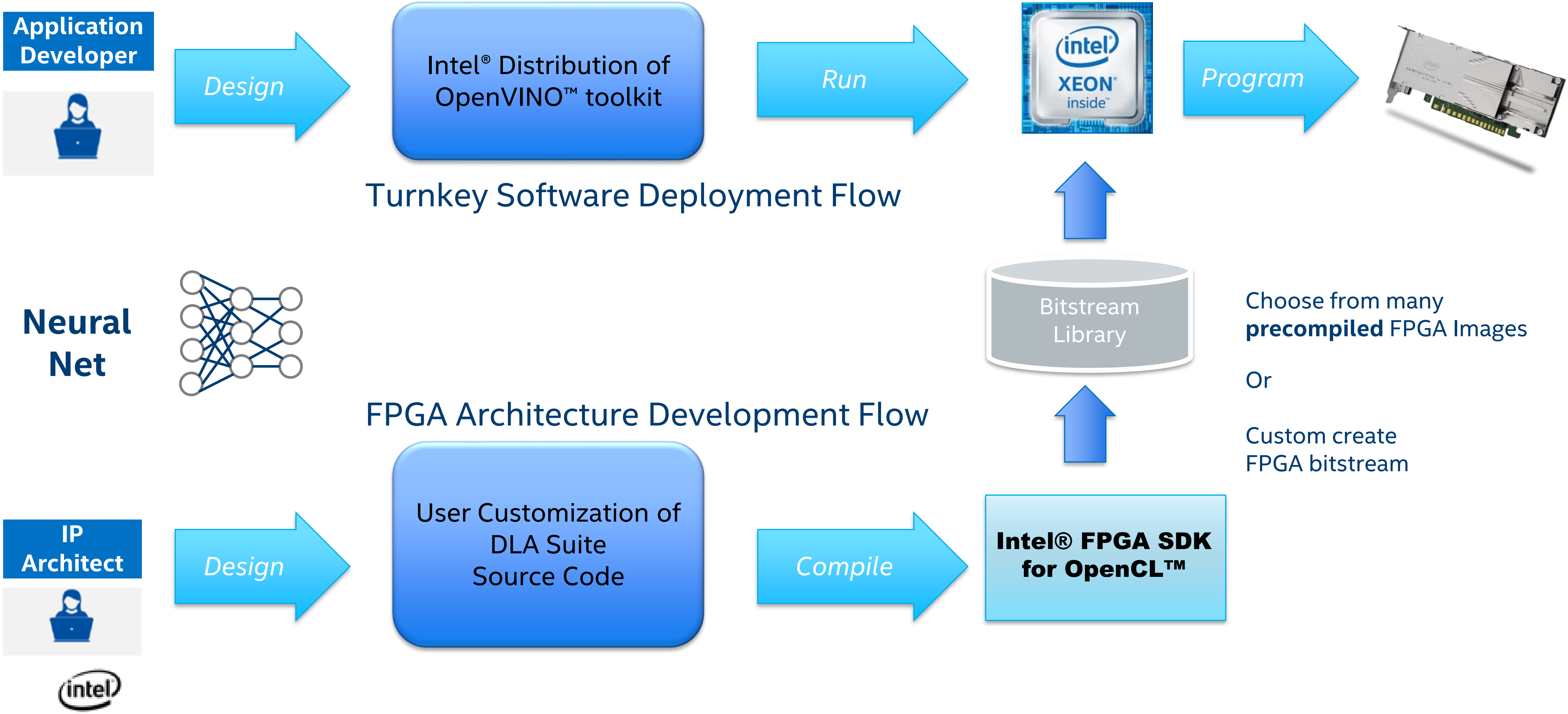
Intel® Architecture-Based
Platforms Support

Intel® Vision Accelerator Design Products & AI in Production/ Developer Kits

An open source version is available at 01.org/openvinotoolkit (some deep learning functions support Intel CPU/GPU only).

OpenVX and the OpenVX logo are trademarks of the Khronos Group Inc.
OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission by Khronos

INTEL® DISTRIBUTION OF OPENVINO™ TOOLKIT WITH DLA USER FLOWS



Application Developer

Design

Intel® Distribution of OpenVINO™ toolkit

Run

Program

Turnkey Software Deployment Flow

Neural Net

Bitstream Library

Choose from many **precompiled** FPGA Images

Or

Custom create FPGA bitstream

FPGA Architecture Development Flow

IP Architect

Design

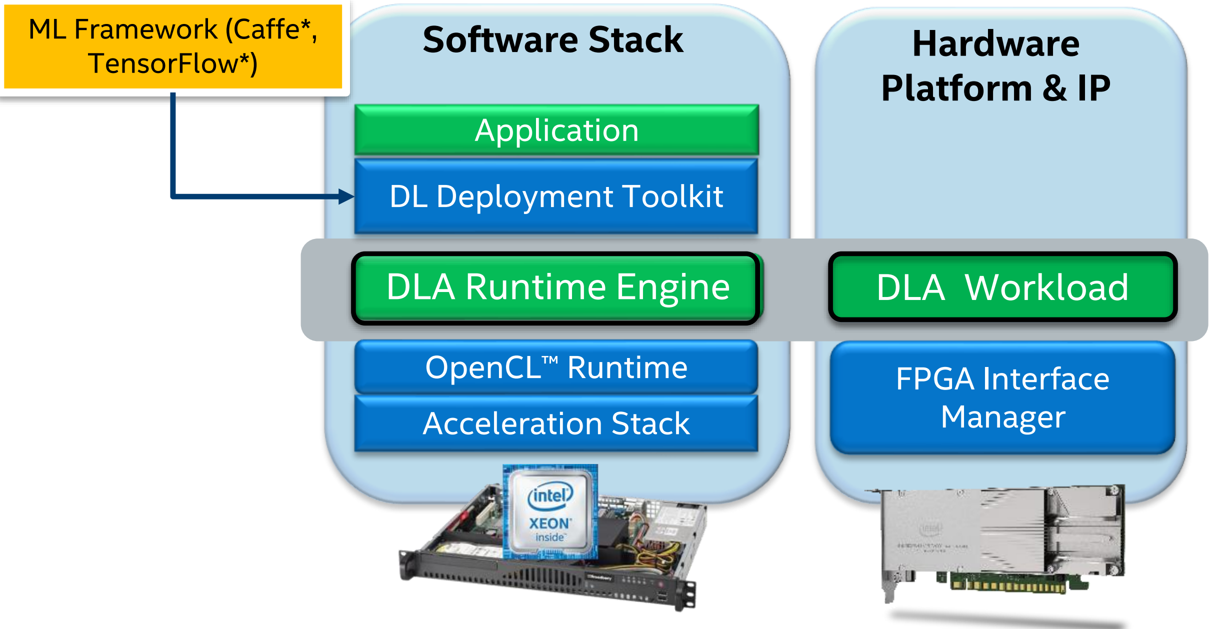
User Customization of DLA Suite Source Code

Compile

Intel® FPGA SDK for OpenCL™

MACHINE LEARNING ON INTEL® FPGA PLATFORM

Acceleration Stack Platform Solution



ML Framework (Caffe*, TensorFlow*)

Software Stack

Application

DL Deployment Toolkit

DLA Runtime Engine

OpenCL™ Runtime

Acceleration Stack

Hardware Platform & IP

DLA Workload

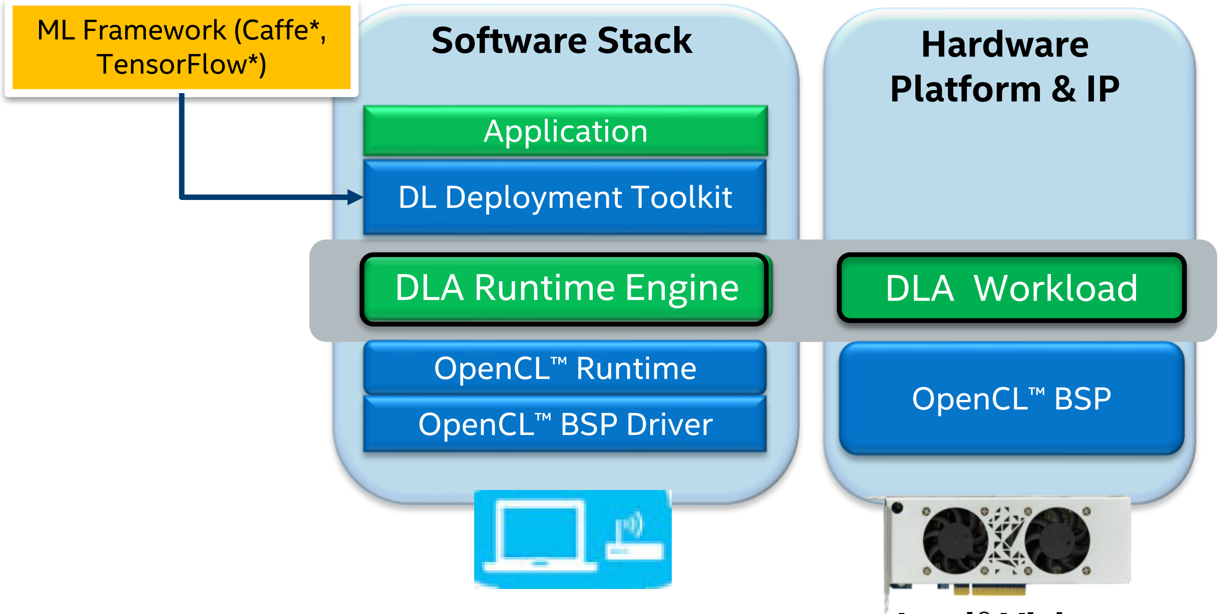
FPGA Interface Manager

Intel® Xeon® CPU

Intel® Programmable Acceleration Card

Intel® FPGA Acceleration Hub

Edge Computing Solution



ML Framework (Caffe*, TensorFlow*)

Software Stack

Application

DL Deployment Toolkit

DLA Runtime Engine

OpenCL™ Runtime

OpenCL™ BSP Driver

Hardware Platform & IP

DLA Workload

OpenCL™ BSP

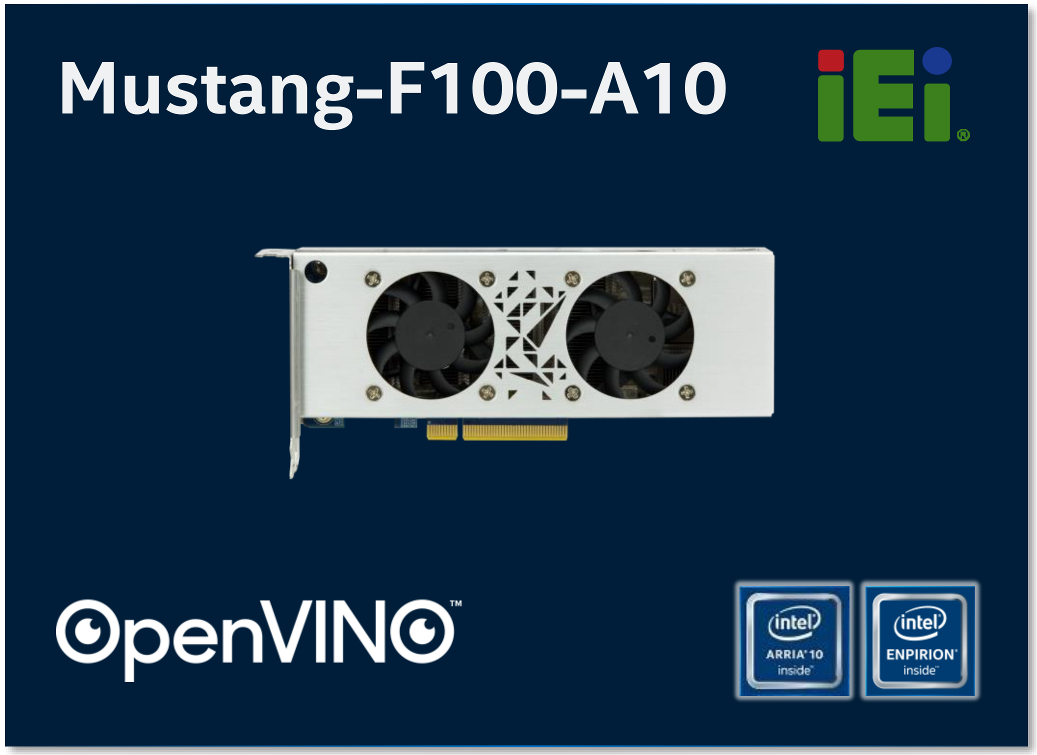
Intel® CPU

Intel® Vision Accelerator Design with FPGA

Intel® Vision Accelerator Design Products

INTEL® VISION ACCELERATION DESIGN WITH INTEL® ARRIA® 10 FPGA

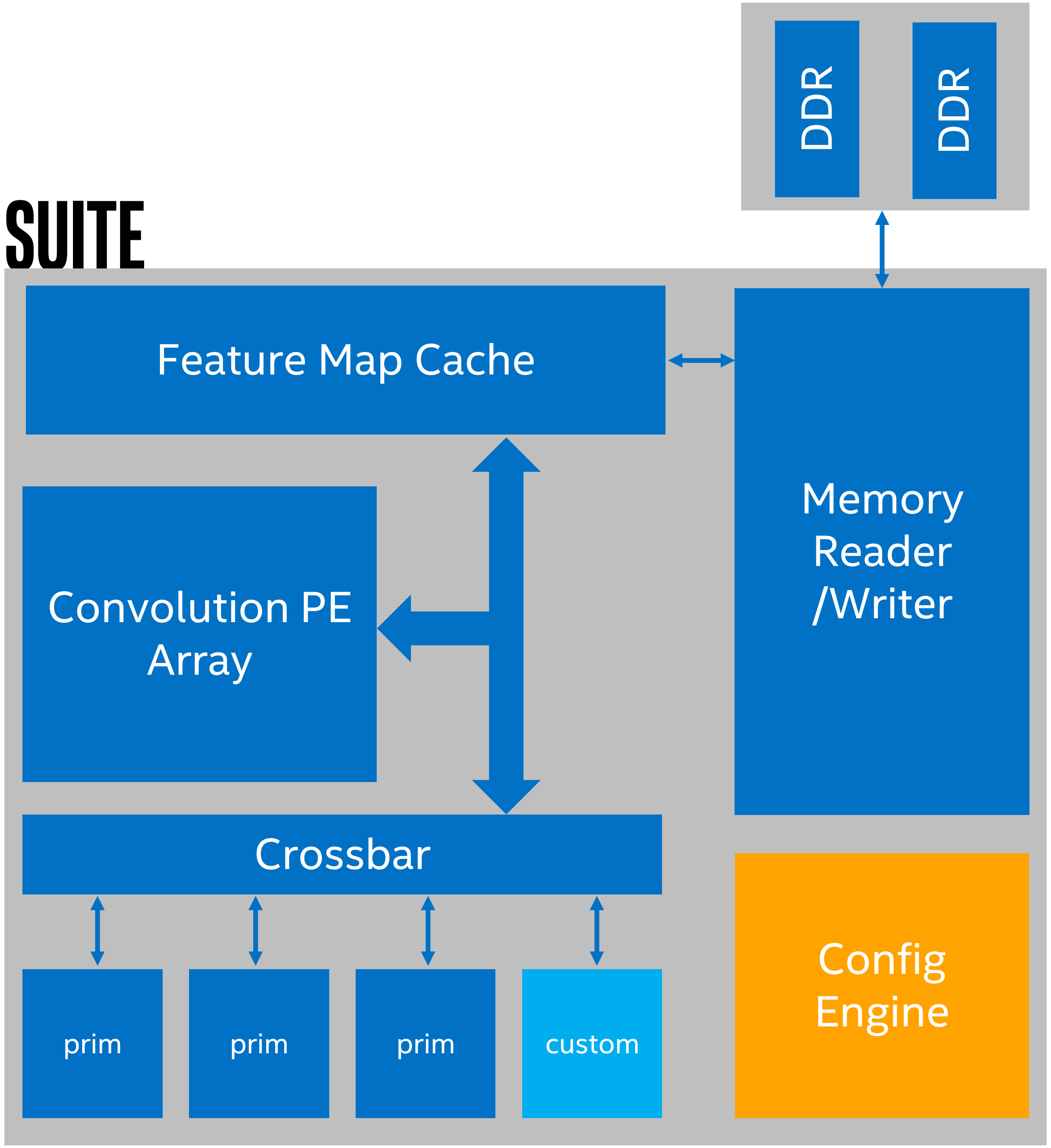
KEY DIFFERENTIATORS



- High performance, low latency
- Flexibility to adapt to new, evolving, and custom networks
- Supports large image sizes (e.g., 4K)
- Large networks (up to 4 billion parameters)
- Wide ambient temperature range (0° C to 65° C)
- 24/7/365 operation
- Long lifespan (8–10 years)

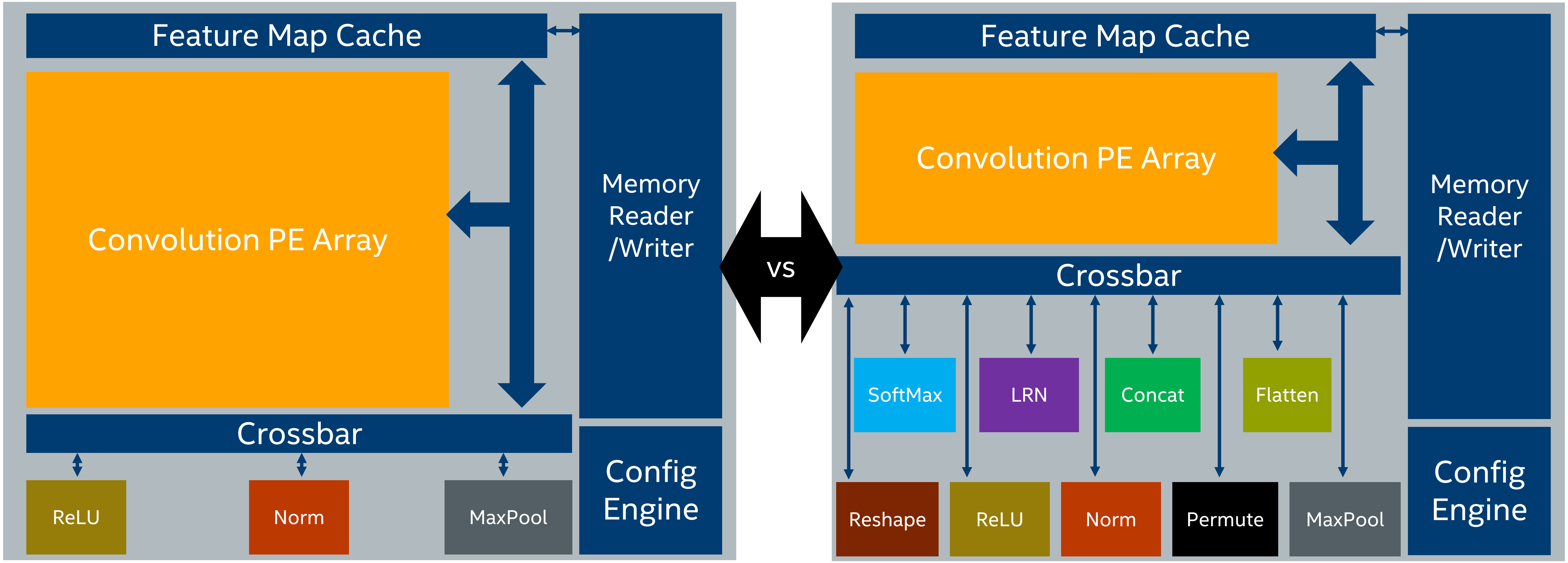
INTEL® FPGA DEEP LEARNING ACCELERATION SUITE

- CNN inference acceleration engine for topologies executed in a graph loop architecture
  - AlexNet, GoogleNet, SqueezeNet, VGG, ResNet*, MobileNet*, Yolo, SSD, …
- Software Deployment
  - No FPGA compile required
  - Run-time reconfigurable
- Customized Hardware Development
  - **Custom architecture creation w/ parameters**
  - Custom primitives using OpenCL™ flow



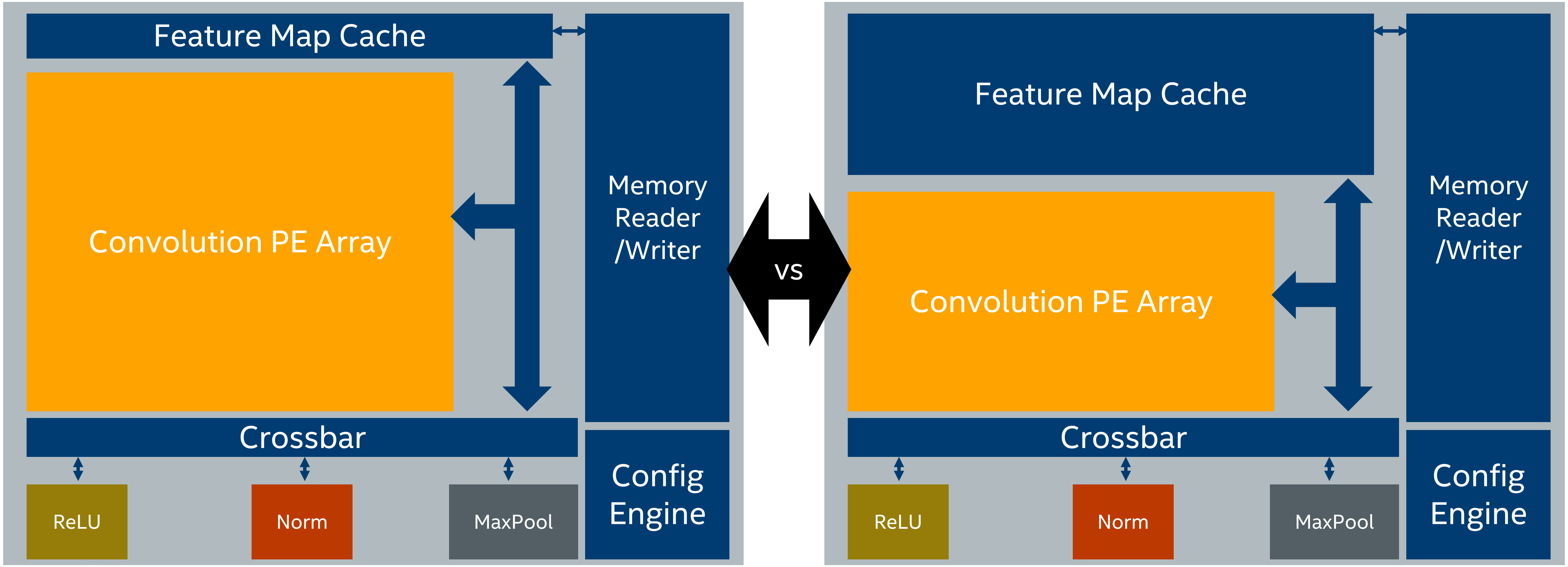
SUPPORT FOR DIFFERENT TOPOLOGIES

Adapts to support new or evolving networks



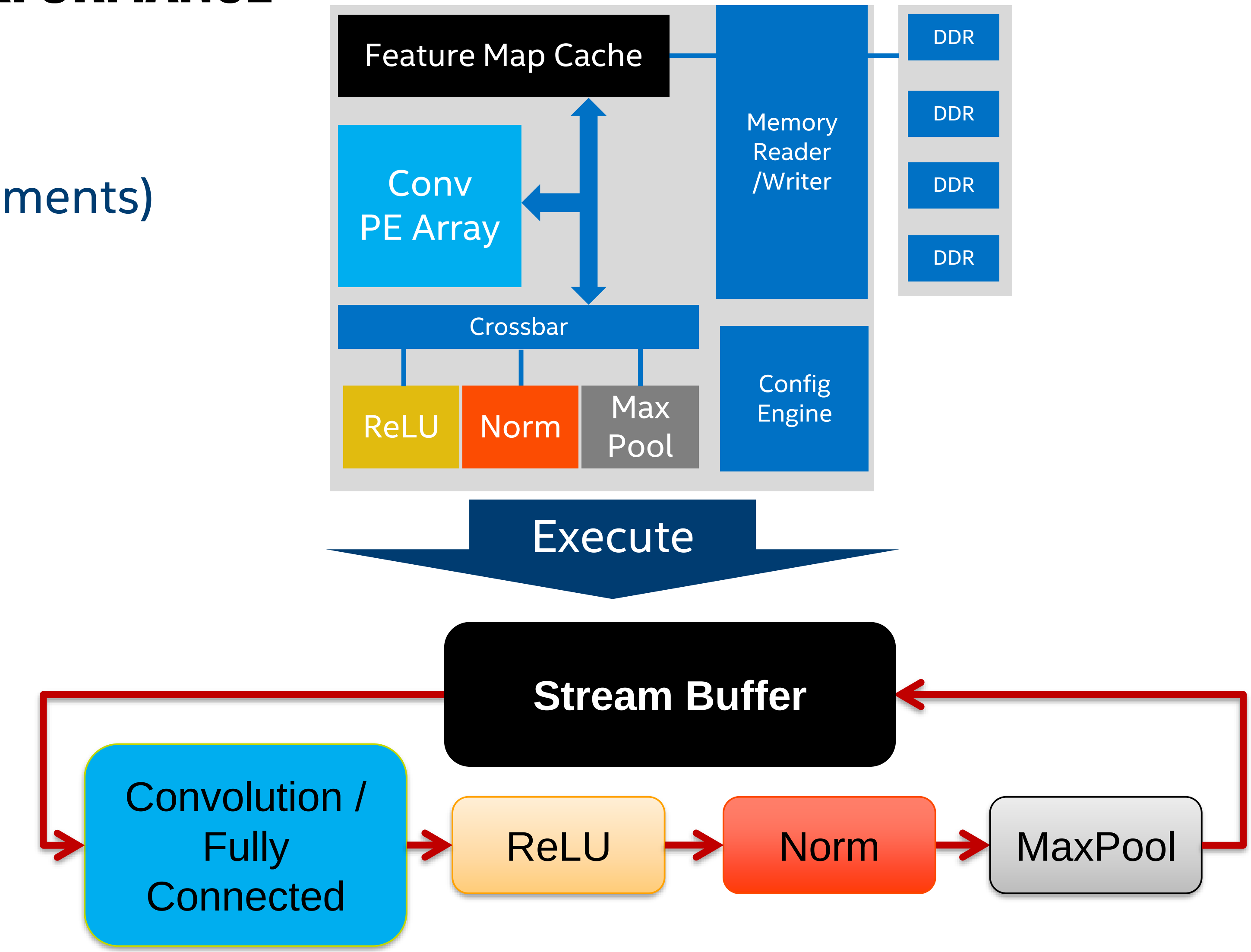
OPTIMIZE FOR BEST PERFORMANCE

Tradeoff between size of Feature Map cache and convolutional PE array



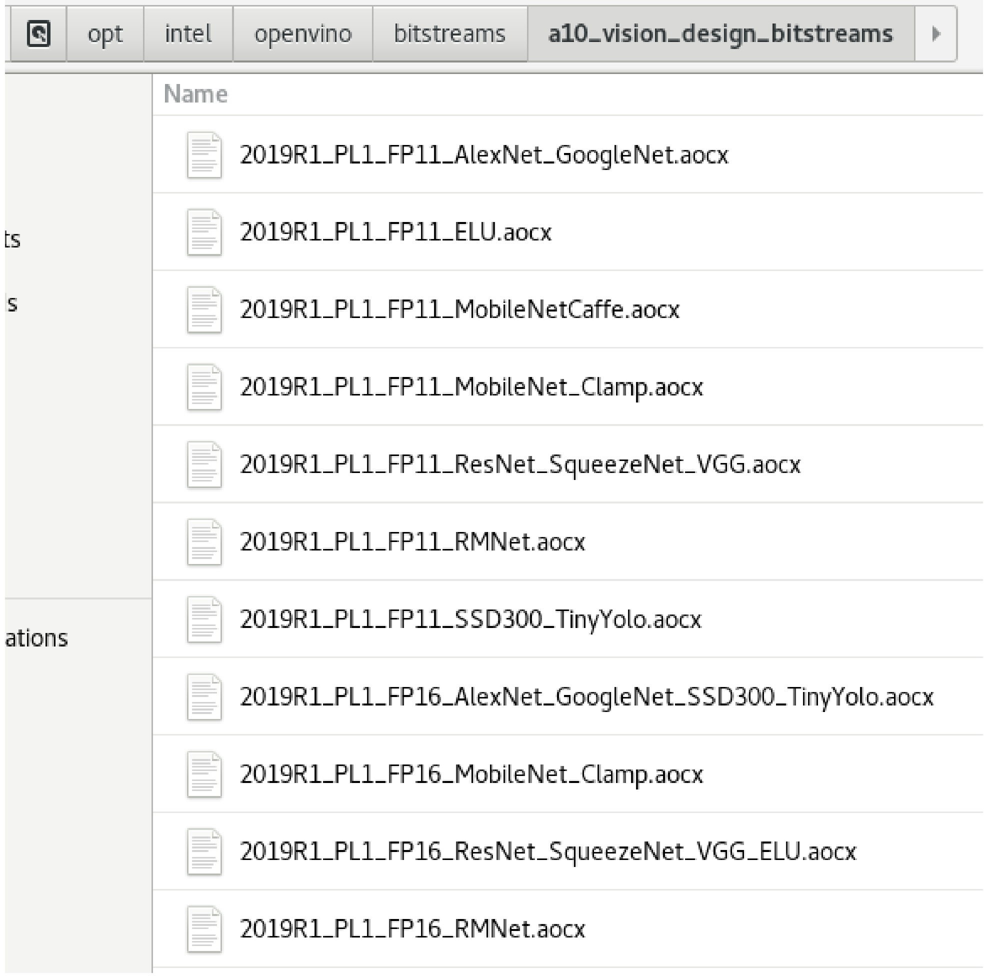
DLA ARCHITECTURE: BUILT FOR PERFORMANCE

- Maximize Parallelism on the FPGA
  - Filter Parallelism (Processing Elements)
  - Input-Depth Parallelism
  - Winograd Transformation
  - Batching
  - Feature Stream Buffer
  - Filter Cache
- Choosing FPGA Bitstream
  - Data Type / Design Exploration
  - Primitive Support



DLA ARCHITECTURE SELECTION

- Intel® Distribution of OpenVINO™ toolkit ships with many FPGA images for various boards/data types/topologies
  - <version>_<board>_<data type>_<Topologies/Feature>.aocx
- Find ideal FPGA image that meets your needs
- Check documentation for list of FPGA images and supported topologies
  - https://docs.openvinotoolkit.org/latest/_docs_IE_DG_supported_plugins_FPGA.html
- Example: ResNet* focused image does not have Norm (better performance)



Name

2019R1_PL1_FP11_AlexNet_GoogleNet.aocx

2019R1_PL1_FP11_ELU.aocx

2019R1_PL1_FP11_MobileNetCaffe.aocx

2019R1_PL1_FP11_MobileNet_Clamp.aocx

2019R1_PL1_FP11_ResNet_SqueezeNet_VGG.aocx

2019R1_PL1_FP11_RMNet.aocx

2019R1_PL1_FP11_SSD300_TinyYolo.aocx

2019R1_PL1_FP16_AlexNet_GoogleNet_SSD300_TinyYolo.aocx

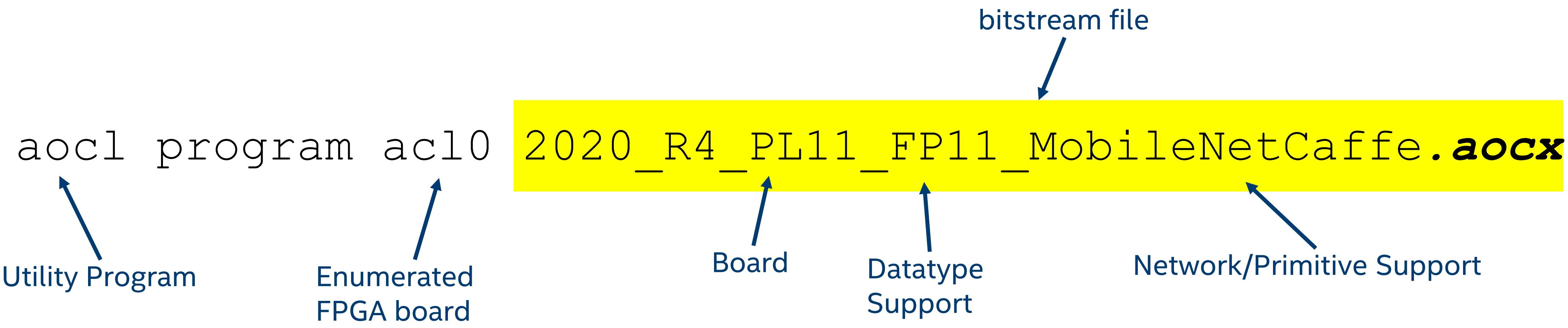
2019R1_PL1_FP16_MobileNet_Clamp.aocx

2019R1_PL1_FP16_ResNet_SqueezeNet_VGG_ELU.aocx

2019R1_PL1_FP16_RMNet.aocx

LOAD SELECTED BITSTREAM PRIOR TO EXECUTION

- Program the FPGA with the selected FPGA bitstream



bitstream file

```
aocl program acl0 2020_R4_PL11_FP11_MobileNetCaffe.aocx
```

Utility Program

Enumerated FPGA board

Board

Datatype Support

Network/Primitive Support

INTEL® DISTRIBUTION OF OPENVINO™ TOOLKIT FOR INTEL® VISION ACCELERATOR DESIGN WITH AN INTEL® ARRIA® 10 FPGA AND THE INTEL® PROGRAMMABLE ACCELERATION CARD WITH INTEL® ARRIA® 10 GX FPGA SUPPORT CHANGE

Intel will be transitioning to the next-generation programmable deep-learning solution based on FPGAs in order to increase the level of customization possible in FPGA deep-learning.

As part of this transition, future standard releases (i.e., non-LTS releases) of Intel® Distribution of OpenVINO™ toolkit will no longer include the Intel® Vision Accelerator Design with an Intel® Arria® 10 FPGA and the Intel® Programmable Acceleration Card with Intel® Arria® 10 GX FPGA.

Intel® Distribution of OpenVINO™ toolkit 2020.3.X LTS release will continue to support Intel® Vision Accelerator Design with an Intel® Arria® 10 FPGA and the Intel® Programmable Acceleration Card with Intel® Arria® 10 GX FPGA.

AGENDA

- Intel® Smart Video/Computer vision Tools Overview
- Model Optimizer
- Inference Engine
- 15 Minute Break
- Accelerators based on Intel® Movidius™ Vision Processing Unit
- Accelerators based on Intel® Arria® FPGA
- Multiple Models in One Application
- DL Workbench + Demo
- DL Streamer
- Register for access to Intel® DevCloud for the Edge
- **Lab1 - DevCloud Sample Application: Accelerated Object Detection**
- **Lab2 – DevCloud Advanced Tutorials: DL Streamer Benchmark**

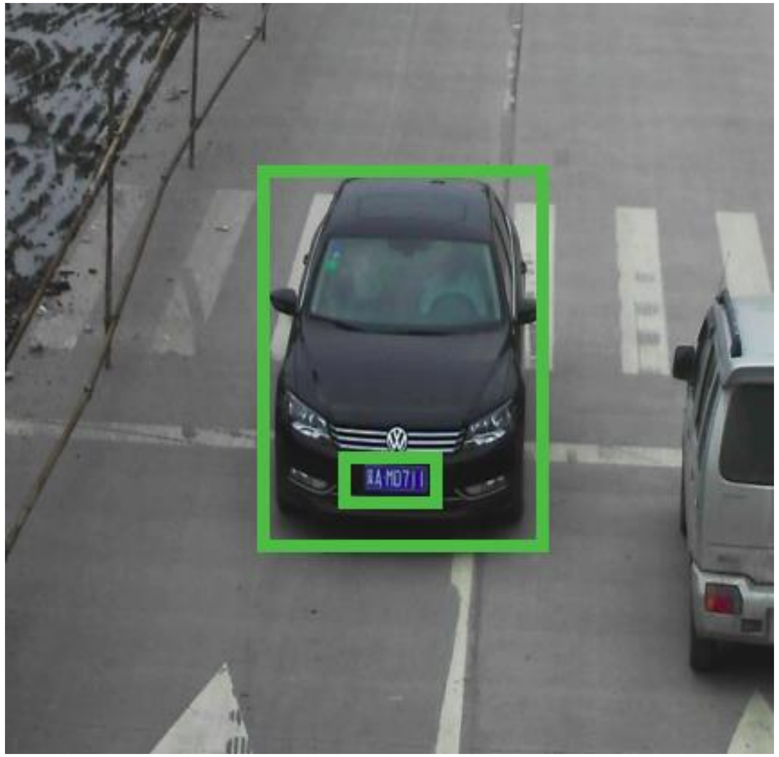
MULTIPLE MODELS IN ONE APPLICATION
SECURITY BARRIER DEMO

VIDEO ANALYTICS IN INTEL® DISTRIBUTION OF OPENVINO™ TOOLKIT

Topology	Type	Description
vehicle-license-plate-detection-barrier-0007	detection	Multiclass (vehicle, license plates) detector based on RESNET* 10 plus SSD.
vehicle-attributes-recognition-barrier-0010	object_attributes	Vehicle attributes recognition with modified RESNET 10 backbone.
license-plate-recognition-barrier-0001	ocr	Chinese license plate recognition.

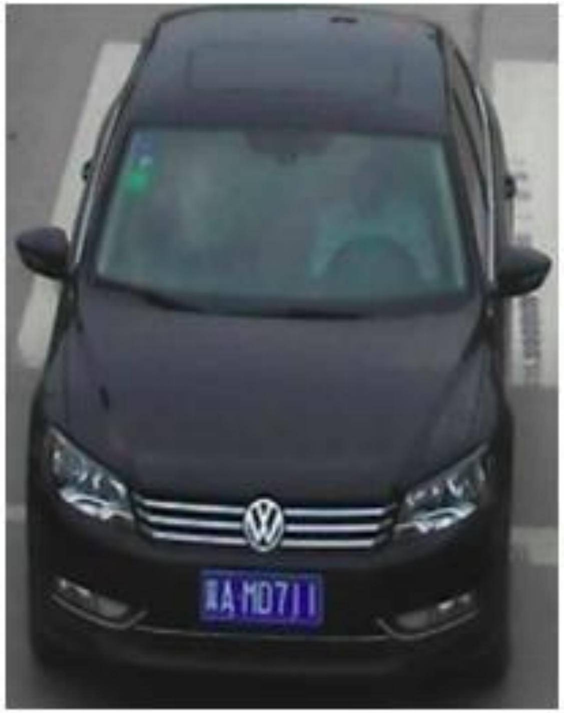
VEHICLE-LICENSE-PLATE-DETECTION-BARRIER-007
USE CASE/HIGH-LEVEL DESCRIPTION

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



VEHICLE-ATTRIBUTES-RECOGNITION-BARRIER-0010
USE CASE/HIGH-LEVEL DESCRIPTION

Vehicle attributes classification algorithm for a traffic analysis scenario.

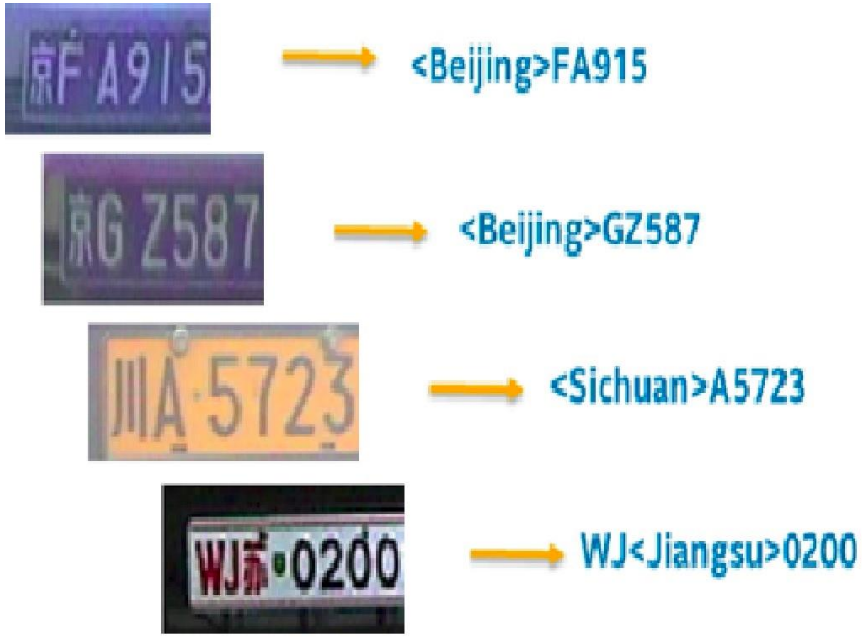


Type: regular
Color: black

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.



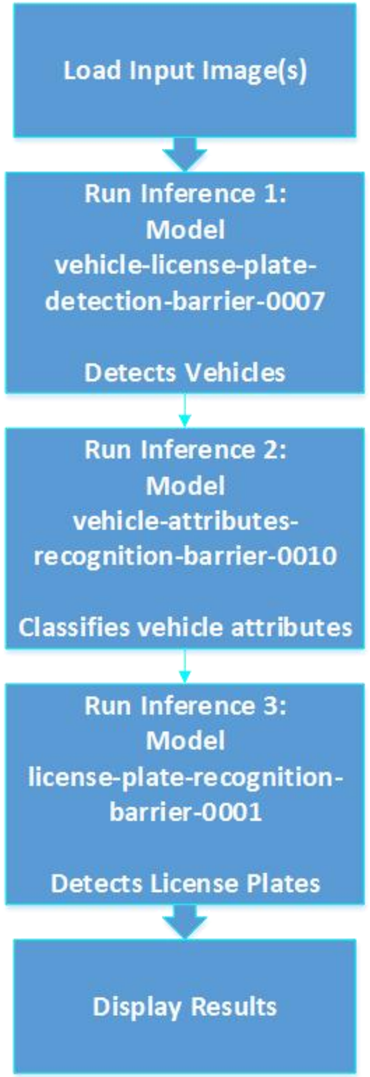
<Beijing>FA915

<Beijing>GZ587

<Sichuan>A5723

WJ<Jiangsu>0200

SECURITY BARRIER DEMO


Load Input Image(s)

Run Inference 1:
Model
vehicle-license-plate-detection-barrier-0007

Detects Vehicles

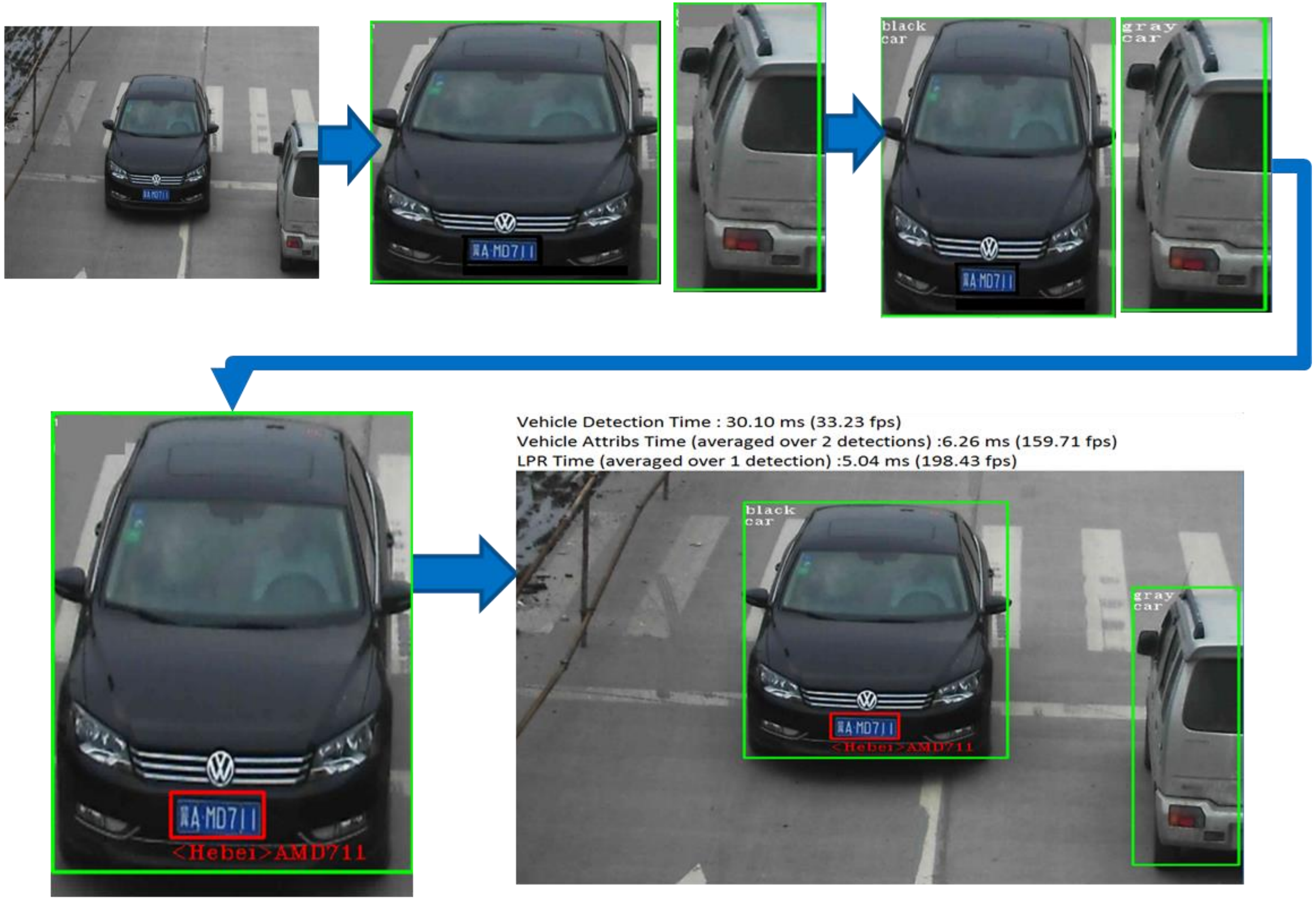
Run Inference 2:
Model
vehicle-attributes-recognition-barrier-0010

Classifies vehicle attributes

Run Inference 3:
Model
license-plate-recognition-barrier-0001

Detects License Plates

Display Results



```
Vehicle Detection Time : 30.10 ms (33.23 fps)
Vehicle Attribs Time (averaged over 2 detections) :6.26 ms (159.71 fps)
LPR Time (averaged over 1 detection) :5.04 ms (198.43 fps)
```

AGENDA

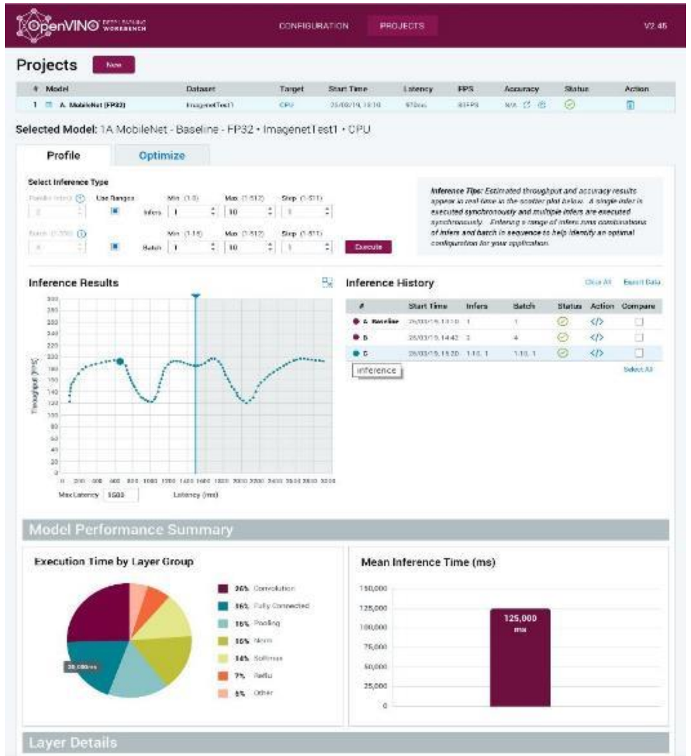
- Intel® Smart Video/Computer vision Tools Overview
- Model Optimizer
- Inference Engine
- 15 Minute Break
- Accelerators based on Intel® Movidius™ Vision Processing Unit
- Accelerators based on Intel® Arria® FPGA
- Multiple Models in One Application
- DL Workbench + Demo
- DL Streamer
- Register for access to Intel® DevCloud for the Edge
- **Lab1 - DevCloud Sample Application: Accelerated Object Detection**
- **Lab2 – DevCloud Advanced Tutorials: DL Streamer Benchmark**

DEEP LEARNING WORKBENCH

Deep Learning Workbench

- Web-based, **UI extension tool** of the Intel® Distribution of OpenVINO™ toolkit
- **Visualizes performance data for** topologies and layers to aid in model analysis
- **Automates analysis** for optimal performance configuration (streams, batches, latency)
- **Experiment with INT8 or Winograd calibration** for optimal tuning using the Post Training Optimization Tool
- Provide **accuracy informatio**n through accuracy checker
- **Direct access to models** from public set of Open Model Zoo
- Enables **remote profiling**, allowing the collection of performance data from multiple different machines without any additional set-up.

Development Guide ▸
https://docs.openvinotoolkit.org/latest/_docs_Workbench_DG_Introduction.html



Projects

#	Model	Dataset	Target	Start Time	Latency	FPS	Accuracy	Status	Action
1	A. MobileNet (FP32)	ImagenetTest1	CPU	25/03/19, 13:10	970ms	83FPS	N/A		

Selected Model: 1A MobileNet - Baseline - FP32 • ImagenetTest1 • CPU

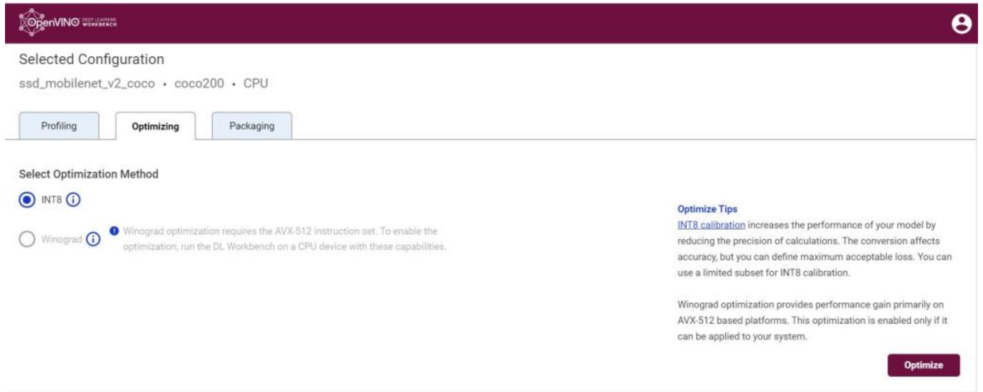
Profile | Optimize

Select Inference Type

Use Ranges ☑ Infers: Min (1-9) 1, Max (1-512) 10, Step (1-511) 1

☑ Batch: Min (1-15) 1, Max (1-512) 10, Step (1-511) 1

Inference Tips: Estimated throughput and accuracy results appear in real time in the scatter plot below. A single infer is executed synchronously and multiple infers are executed asynchronously. Entering a range of infers runs combinations of infers and batch in sequence to help identify an optimal configuration for your application.



Selected Configuration

ssd_mobilenet_v2_coco • coco200 • CPU

Select Optimization Method

● INT8

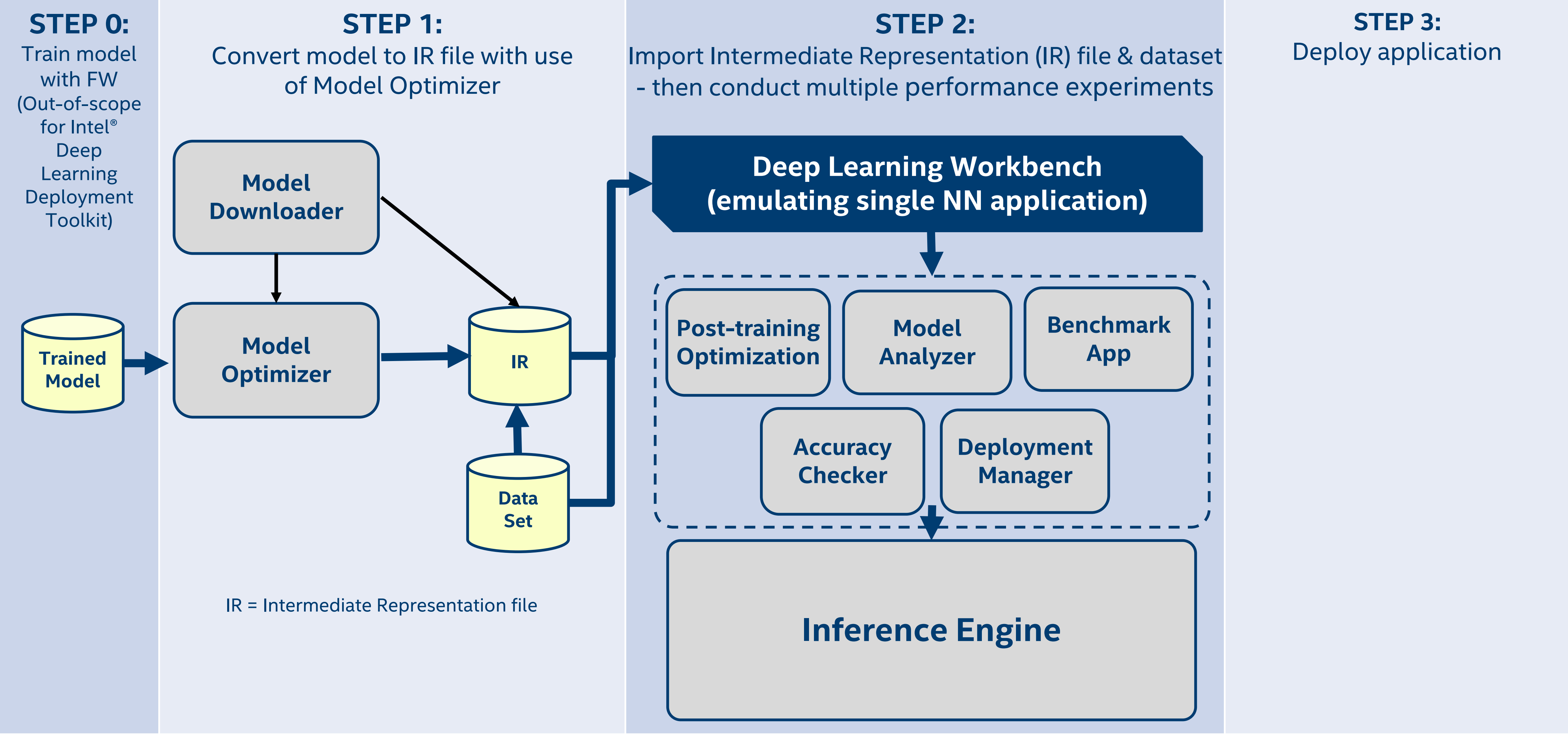
○ Winograd — Winograd optimization requires the AVX-512 instruction set. To enable the optimization, run the DL Workbench on a CPU device with these capabilities.

Optimize Tips

INT8 calibration increases the performance of your model by reducing the precision of calculations. The conversion affects accuracy, but you can define maximum acceptable loss. You can use a limited subset for INT8 calibration.

Winograd optimization provides performance gain primarily on AVX-512 based platforms. This optimization is enabled only if it can be applied to your system.

DEEP LEARNING WORKBENCH DATA FLOW



STEP 0:
Train model with FW (Out-of-scope for Intel® Deep Learning Deployment Toolkit)

STEP 1:
Convert model to IR file with use of Model Optimizer

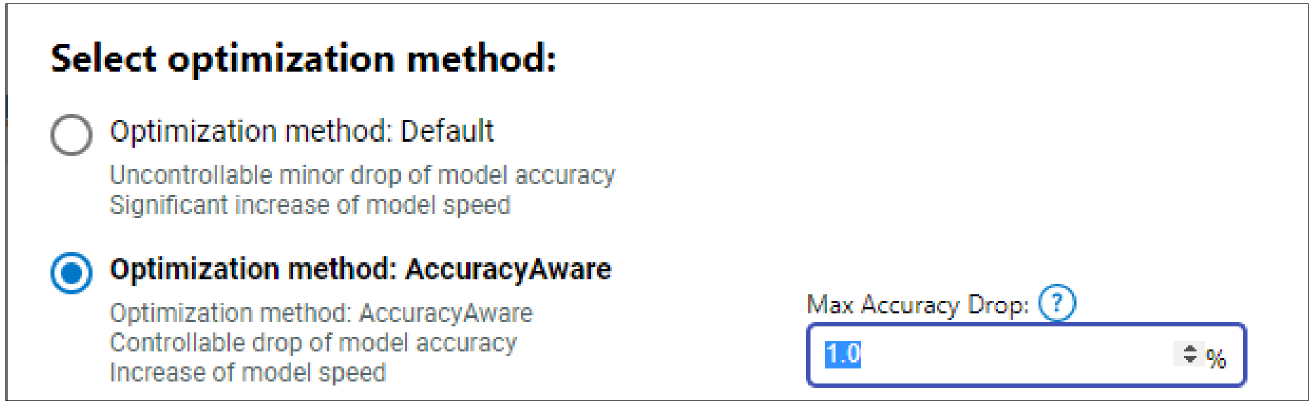
STEP 2:
Import Intermediate Representation (IR) file & dataset - then conduct multiple performance experiments

STEP 3:
Deploy application

IR = Intermediate Representation file

DEEP LEARNING WORKBENCH : FEATURES

CONVERT MODEL TO INT8 USING 2 NEW CALIBRATION ALGORITHMS



Select optimization method:

○ Optimization method: Default

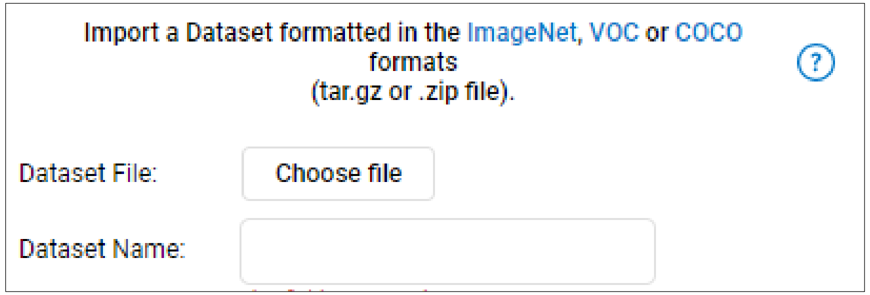
Uncontrollable minor drop of model accuracy
Significant increase of model speed

● **Optimization method: AccuracyAware**

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

Max Accuracy Drop: 1.0 %

IMPORT DATASET IN COCO FORMAT TO USE WITH MODEL

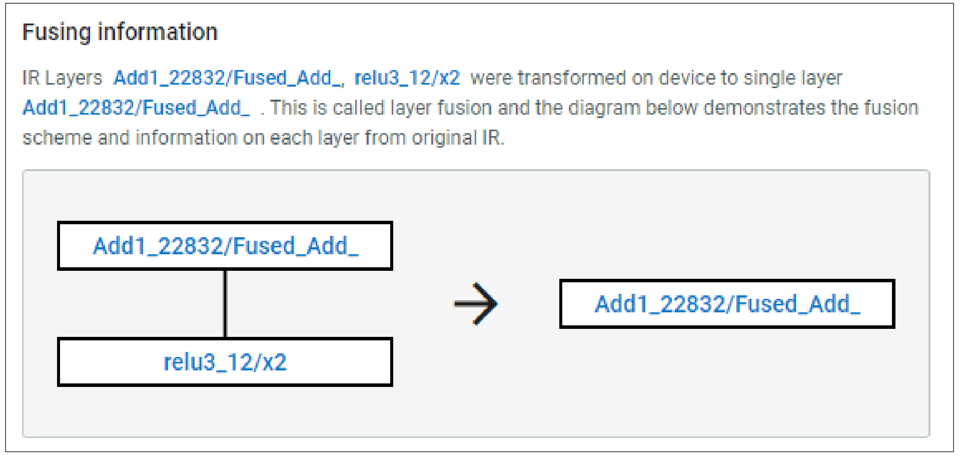


Import a Dataset formatted in the ImageNet, VOC or COCO formats
(tar.gz or .zip file).

Dataset File: Choose file

Dataset Name:

IMPROVED PER-LAYER DATA VISUALIZATION AND COMPARISON MODE.

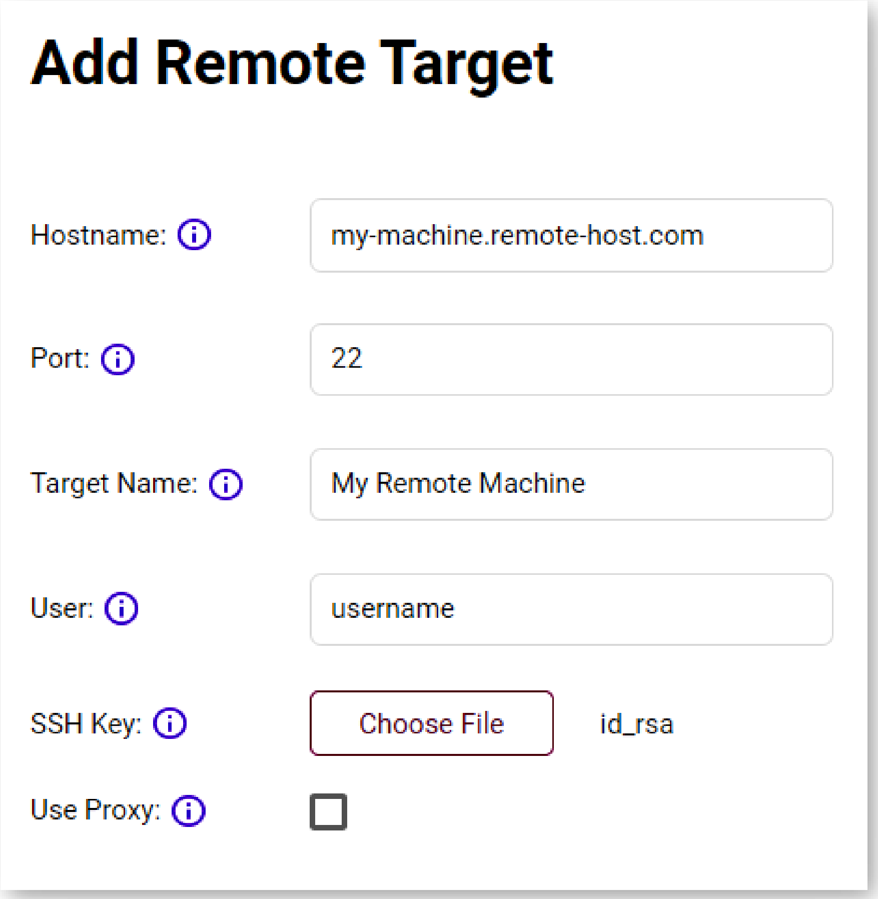


Fusing information

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.

DEEP LEARNING WORKBENCH : NEW FEATURES

REMOTE PROFILING SUPPORT



Add Remote Target

Hostname: my-machine.remote-host.com

Port: 22

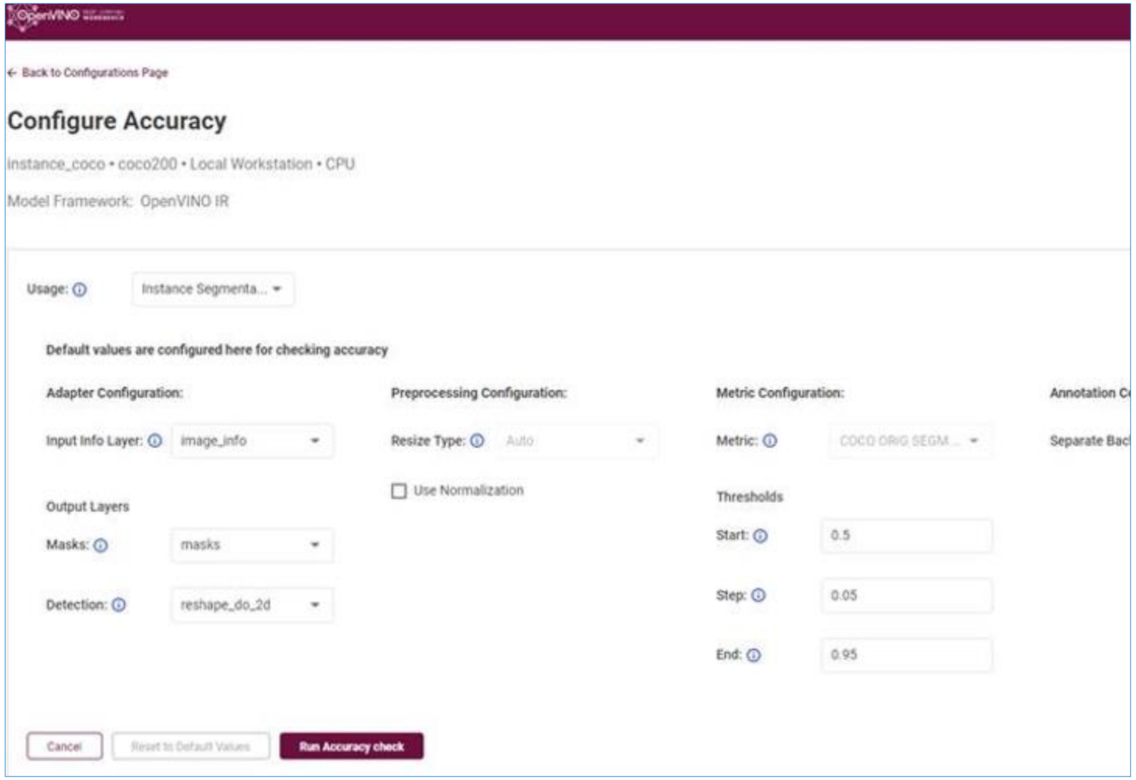
Target Name: My Remote Machine

User: username

SSH Key: id_rsa

Use Proxy: □

SUPPORT FOR SEGMENTATION USE CASES



Configure Accuracy

instance_coco • coco200 • Local Workstation • CPU

Model Framework: OpenVINO IR

Usage: Instance Segmenta...

Default values are configured here for checking accuracy

Adapter Configuration:

Input Info Layer: image_info

Output Layers

Masks: masks

Detection: reshape_do_2d

Preprocessing Configuration:

Resize Type: Auto

□ Use Normalization

Metric Configuration:

Metric: COCO ORIG SEGM ...

Thresholds

Start: 0.5

Step: 0.05

End: 0.95

Annotation C

Separate Bac

DEMO - DL WORKBENCH WALKTHROUGH

AGENDA

- Intel® Smart Video/Computer vision Tools Overview
- Model Optimizer
- Inference Engine
- 15 Minute Break
- Accelerators based on Intel® Movidius™ Vision Processing Unit
- Accelerators based on Intel® Arria® FPGA
- Multiple Models in One Application
- DL Workbench + Demo
- DL Streamer
- Register for access to Intel® DevCloud for the Edge
- **Lab1 - DevCloud Sample Application: Accelerated Object Detection**
- **Lab2 – DevCloud Advanced Tutorials: DL Streamer Benchmark**

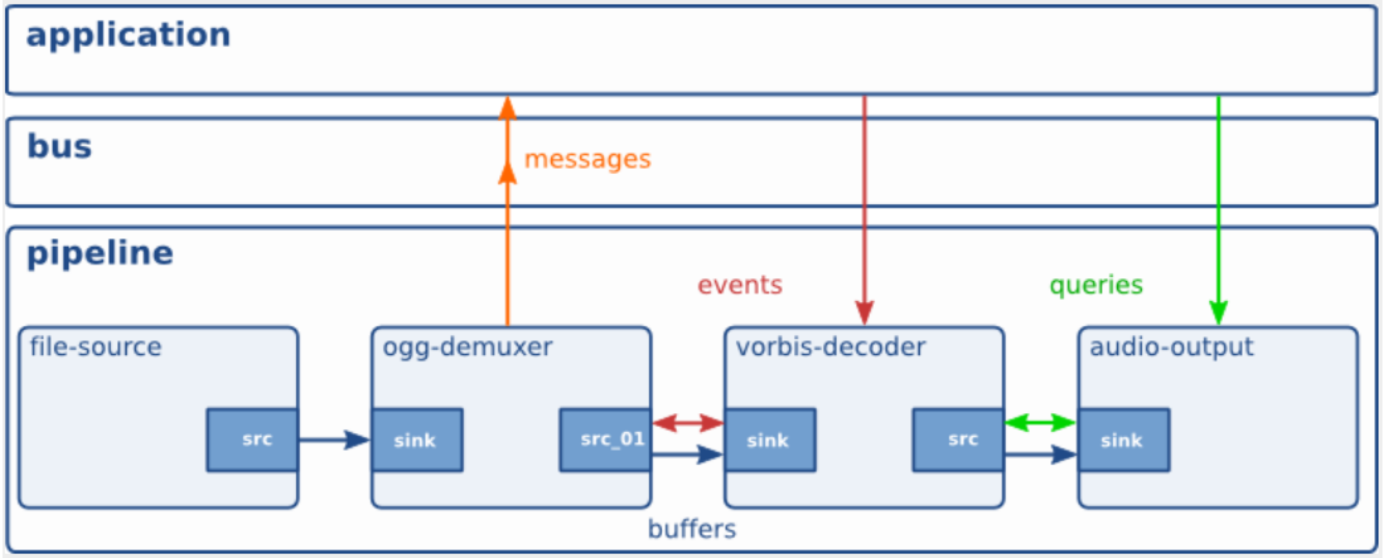
DEEP LEARNING STREAMER

INTRODUCING.. DL STREAMER

- Intel® Distribution of OpenVINO™ toolkit Deep Learning (DL) Streamer, now part of the **default installation** package
- Enables developers to **create and deploy optimized streaming media analytics pipelines** across Intel® architecture from edge to cloud
- Optimal pipeline interoperability with a **familiar developer experience** built using the GStreamer multimedia framework

WHAT IS GSTREAMER?

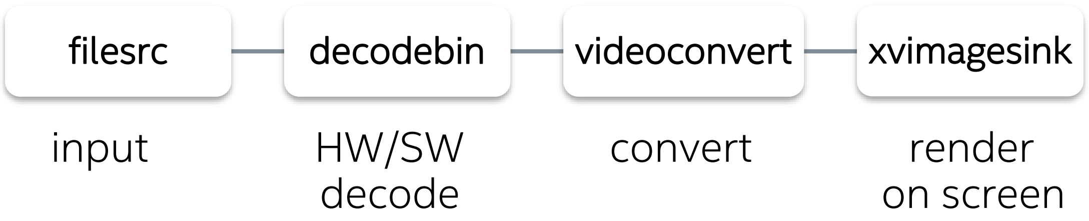
- A pipeline consists of **connected processing elements**
- Each element is provided by **a plug-in** and can be **grouped into bins**
- Elements communicate by means of **pads** – source pad and sink pad
- Data buffers flow **from Source** element **to Sink** element & from source pad to sink pad



Ref: https://gstreamer.freedesktop.org/data/doc/gstreamer/head/manual/manual.pdf

MEDIA PROCESSING PIPELINE

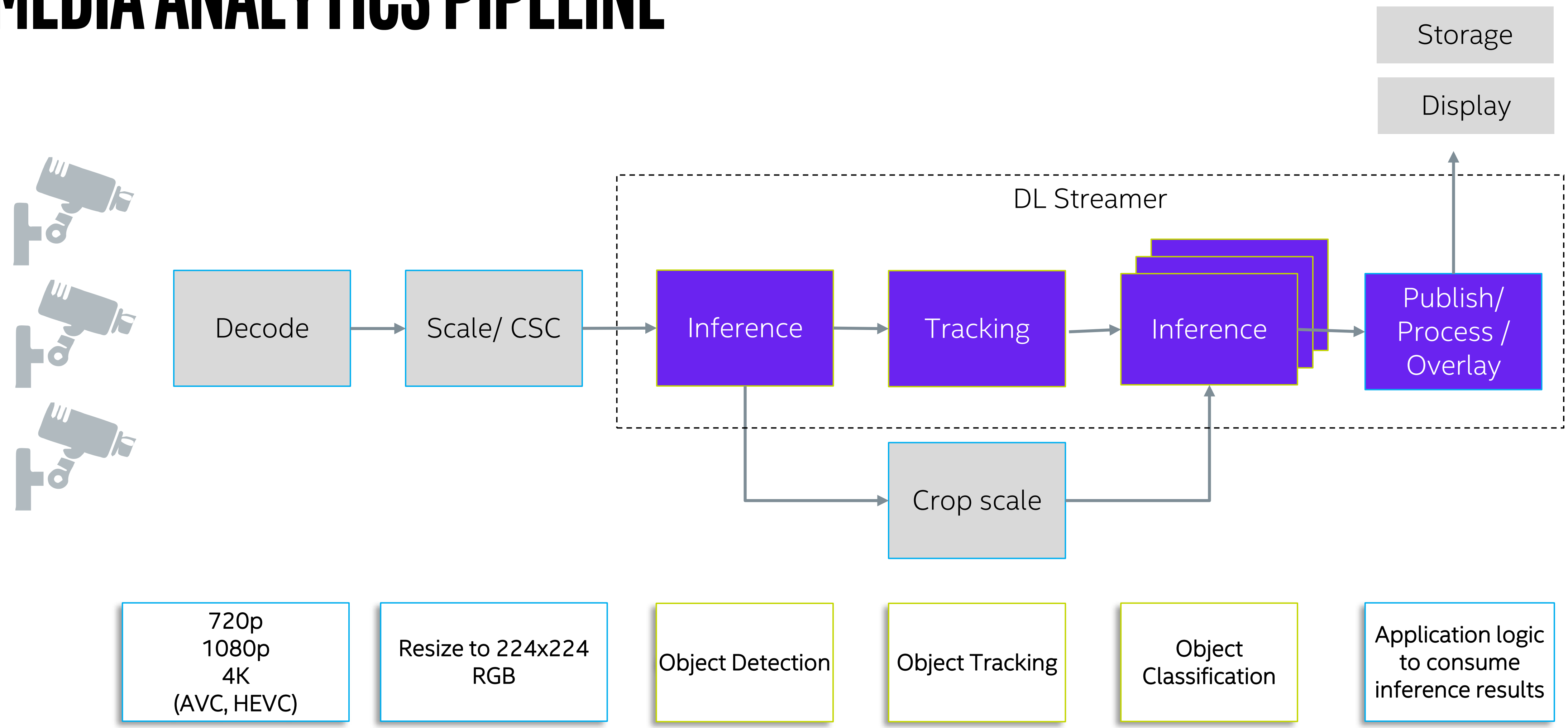
Video Pipeline – decode, convert, render



```
gst-launch-1.0 filesrc location=/path/to/video.mp4 ! decodebin ! videoconvert ! xvimagesink
```

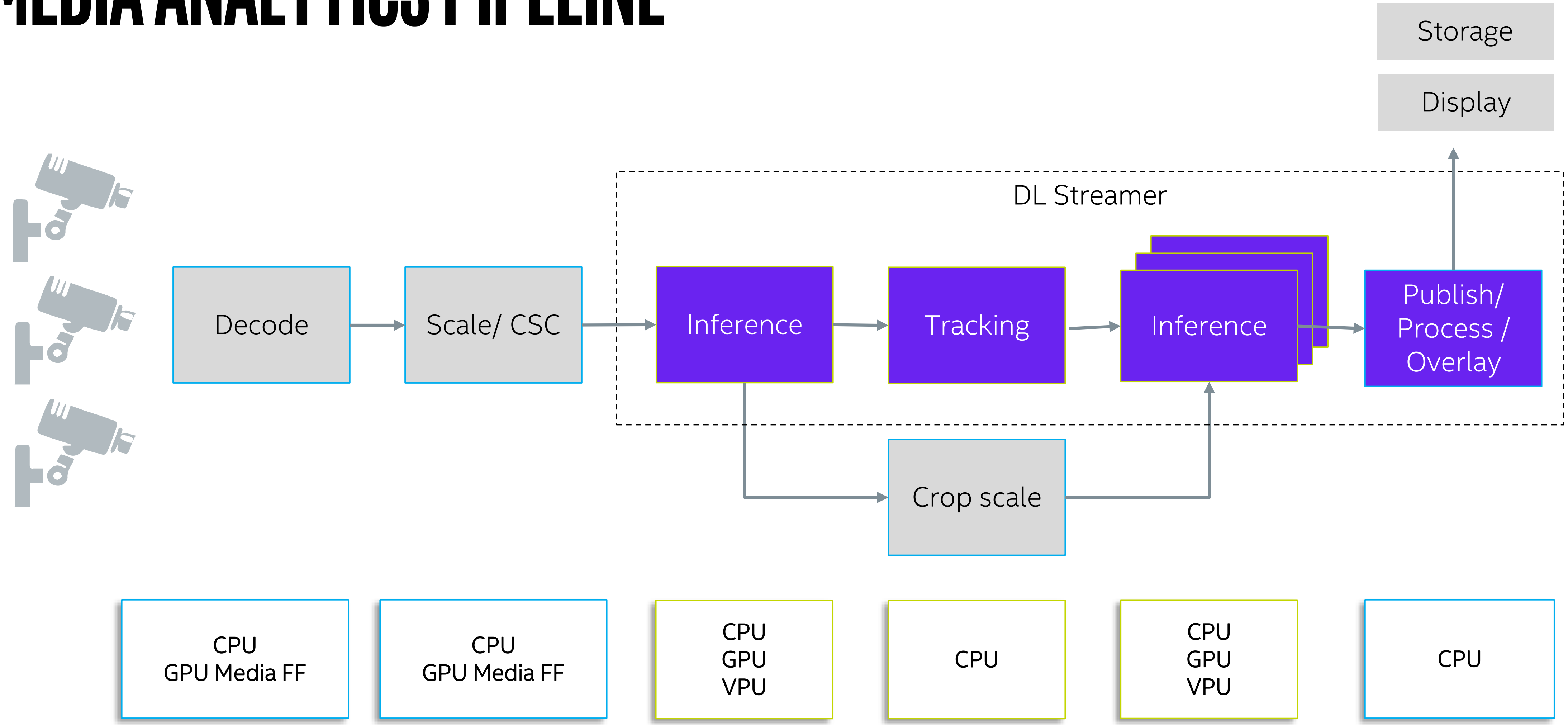
Optimization Notice

MEDIA ANALYTICS PIPELINE



Optimization Notice

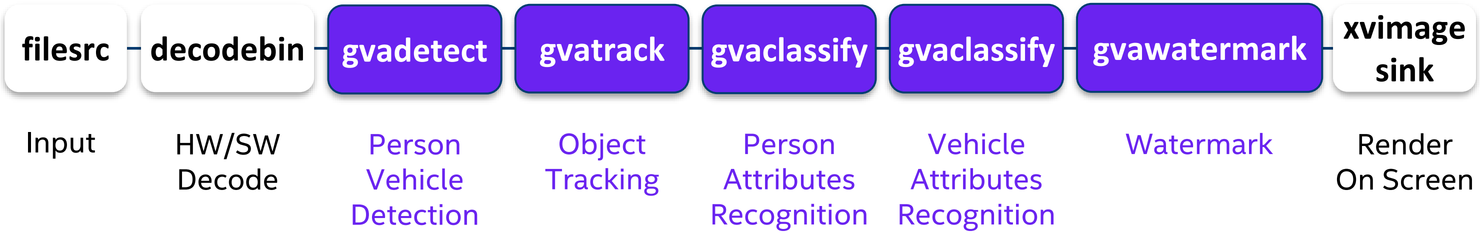
MEDIA ANALYTICS PIPELINE

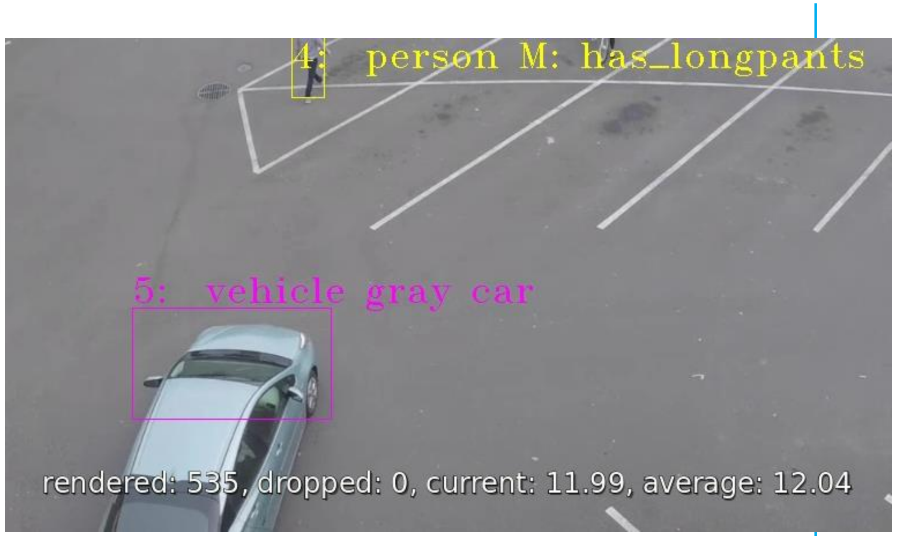


Optimization Notice

USING THE DL STREAMER

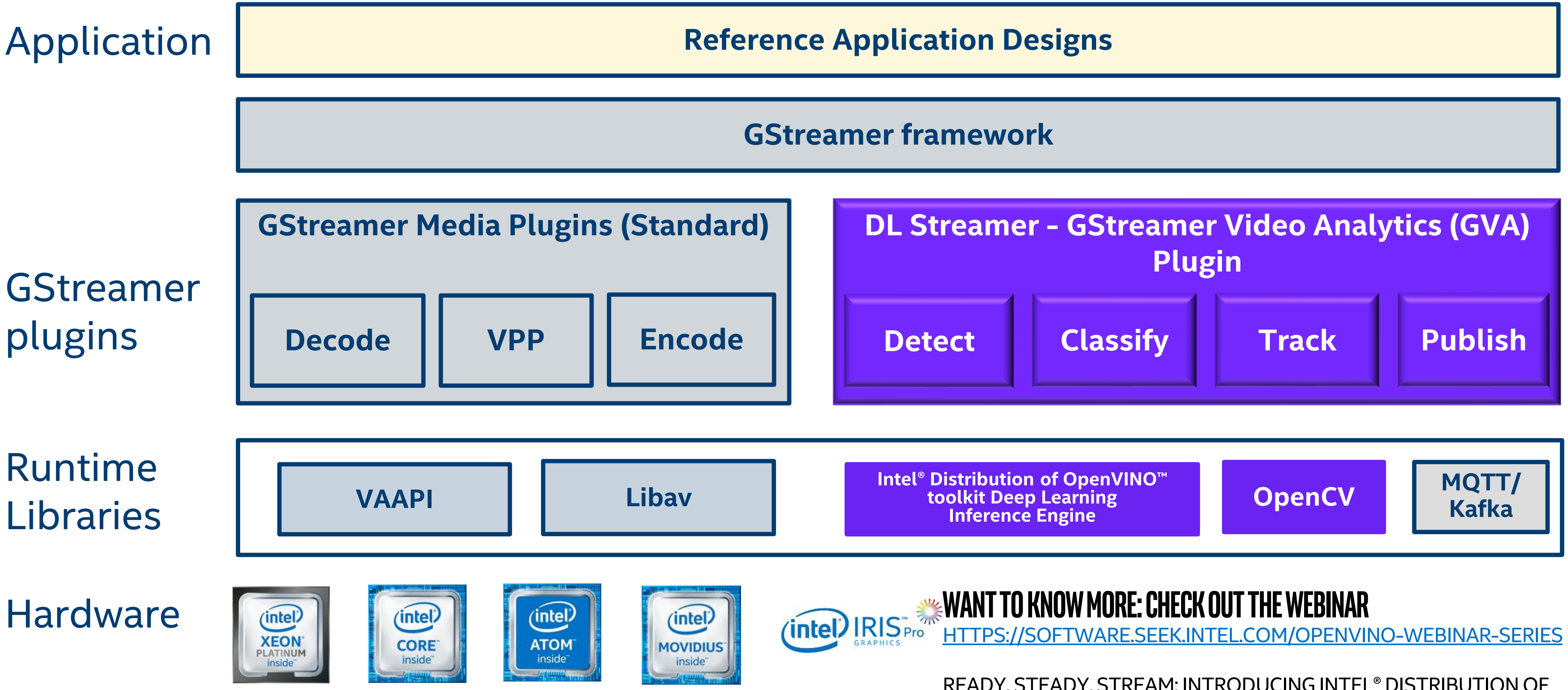
Video Analytics pipeline – person and vehicle detection, person, vehicle attributes classification




4: person M: has_longpants
5: vehicle gray car
rendered: 535, dropped: 0, current: 11.99, average: 12.04

```
 gst-launch-1.0 filesrc location=/path/to/video.mp4 !
 decodebin ! videoconvert ! video/x-raw,format=BGRx ! \
 gvadetect model=person-vehicle-bike-detection-crossroad-0078.xml model-proc=person-vehicle-bike-detection-
crossroad-0078.json inference-interval=10 threshold=0.6 device=CPU ! queue ! \
 gvatrack tracking-type="short-term" ! queue ! \
 gvaclassify model= person-attributes-recognition-crossroad-0230.xml model-proc= person-attributes-recognition-
crossroad-0230.json reclassify-interval=10 device=CPU object-class=person ! queue ! \
 gvaclassify model= vehicle-attributes-recognition-barrier-0039.xml model-proc= vehicle-attributes-recognition-
barrier-0039.json reclassify-interval=10 device=CPU object-class=vehicle ! queue ! \
 gvawatermark ! videoconvert ! fpsdisplaysink video-sink=xvimagesink sync=true
```

UNDER THE HOOD: DL STREAMER



Application

Reference Application Designs

GStreamer framework

GStreamer plugins

GStreamer Media Plugins (Standard)

Decode

VPP

Encode

DL Streamer - GStreamer Video Analytics (GVA) Plugin

Detect

Classify

Track

Publish

Runtime Libraries

VAAPI

Libav

Intel® Distribution of OpenVINO™ toolkit Deep Learning Inference Engine

OpenCV

MQTT/ Kafka

Hardware

WANT TO KNOW MORE: CHECK OUT THE WEBINAR

HTTPS://SOFTWARE.SEEK.INTEL.COM/OPENVINO-WEBINAR-SERIES

READY, STEADY, STREAM: INTRODUCING INTEL® DISTRIBUTION OF OPENVINO™ TOOLKIT DEEP LEARNING STREAMER

AGENDA

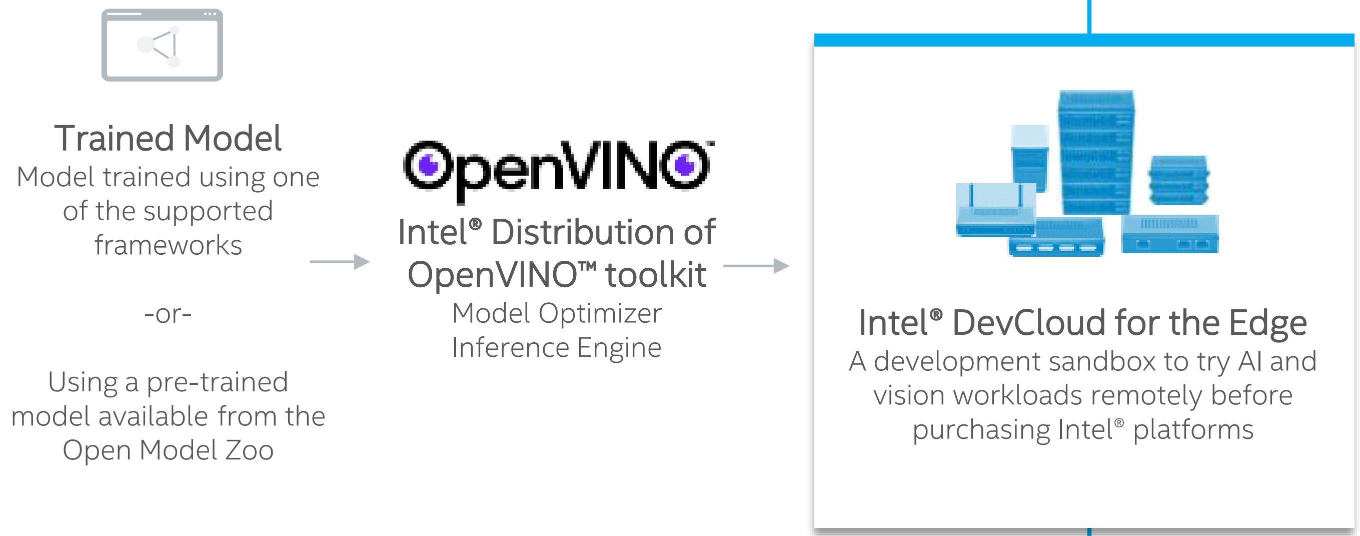
- Intel® Smart Video/Computer vision Tools Overview
- Model Optimizer
- Inference Engine
- 15 Minute Break
- Accelerators based on Intel® Movidius™ Vision Processing Unit
- Accelerators based on Intel® Arria® FPGA
- Multiple Models in One Application
- DL Workbench + Demo
- DL Streamer
- Register for access to Intel® DevCloud for the Edge
- **Lab1 - DevCloud Sample Application: Accelerated Object Detection**
- **Lab2 – DevCloud Advanced Tutorials: DL Streamer Benchmark**

INTEL® DEVCLOUD FOR THE EDGE

Sign Up Here: https://devcloud.intel.com/edge

TEST HARDWARE WITH THE INTEL® DEVCLOUD FOR THE EDGE

Powered by Intel® Distribution of OpenVINO™ toolkit



Trained Model
Model trained using one of the supported frameworks

-or-

Using a pre-trained model available from the Open Model Zoo

→

OpenVINO
Intel® Distribution of OpenVINO™ toolkit
Model Optimizer
Inference Engine

→

Intel® DevCloud for the Edge
A development sandbox to try AI and vision workloads remotely before purchasing Intel® platforms

- Prototype on the latest hardware and software to future proof your solution
- Benchmark your customized AI application
- Run AI applications from anywhere in the world
- Help to reduce development time and cost

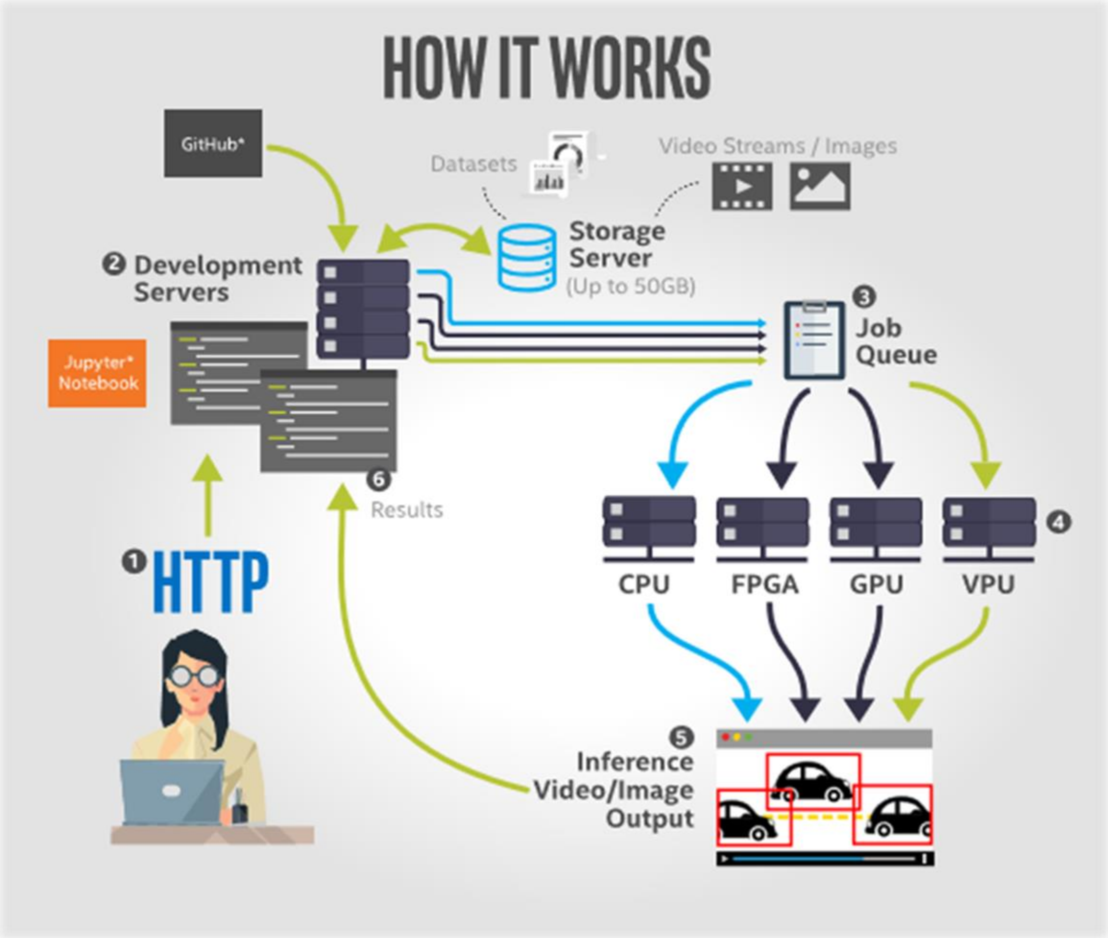
https://devcloud.intel.com/edge/

Deploy and Scale

OpenVINO

ACCELERATE TIME TO PRODUCTION WITH INTEL® DEVCLOUD FOR THE EDGE

SEE IMMEDIATE AI APPLICATION PERFORMANCE ACROSS INTEL'S VAST ARRAY OF EDGE SOLUTIONS



- **Instant, Global Access**
  Run AI applications from anywhere in the world
- **Prototype on the Latest Hardware and Software**
  Develop knowing you're using the latest Intel technology
- **Benchmark your Customized AI Application**
  Immediate feedback – frames per second, performance
- **Reduce Development Time and Cost**
  Quickly find the right compute for your edge solution

Sign up now for access

Optimization Notice

Signup for Access to the Intel® DevCloud for Edge

Sign Up Here: https://devcloud.intel.com/edge/

Intel's Registration Passcode:

Code Valid From:

Code Valid To:

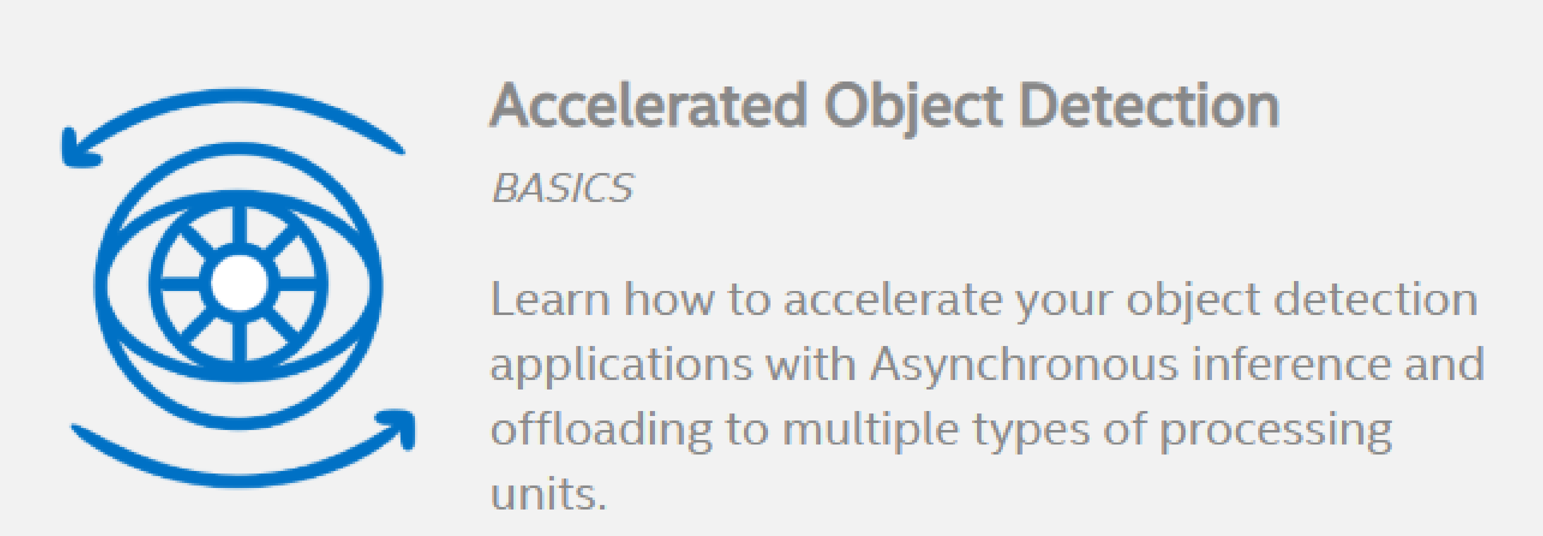
Account Activation:

Account Deactivation: Valid for 30 days

AGENDA

- Intel® Smart Video/Computer vision Tools Overview
- Model Optimizer
- Inference Engine
- 15 Minute Break
- Accelerators based on Intel® Movidius™ Vision Processing Unit
- Accelerators based on Intel® Arria® FPGA
- Multiple Models in One Application
- DL Workbench + Demo
- DL Streamer
- Register for access to Intel® DevCloud for the Edge
- **Lab1 - DevCloud Sample Application: Accelerated Object Detection**
- **Lab2 – DevCloud Advanced Tutorials: DL Streamer Benchmark**

LAB1 – DEVCLOUD SAMPLE APPLICATIONS

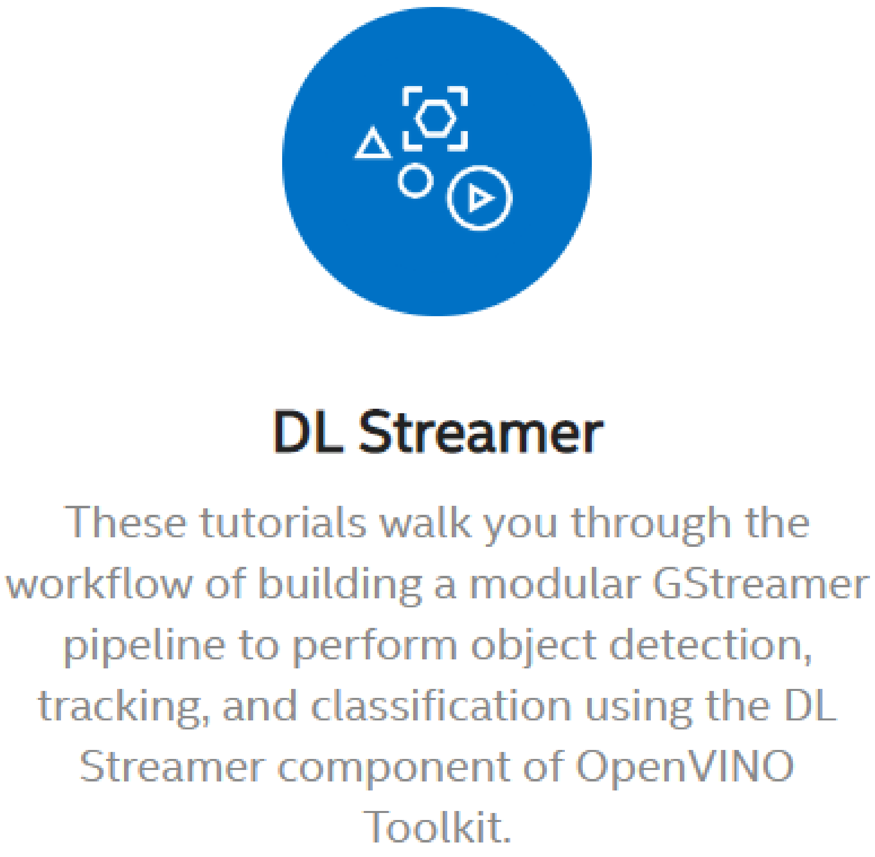


Accelerated Object Detection

BASICS

Learn how to accelerate your object detection applications with Asynchronous inference and offloading to multiple types of processing units.

LAB2 – DEVCLOUD ADVANCED TUTORIALS



DL Streamer

These tutorials walk you through the workflow of building a modular GStreamer pipeline to perform object detection, tracking, and classification using the DL Streamer component of OpenVINO Toolkit.

AGENDA

- Register for access to Intel® DevCloud for the Edge
- Intel® Smart Video/Computer vision Tools Overview
- Model Optimizer + Demo
- Inference Engine
- **Lab1 - DevCloud Tutorial: Classification**
- 15 Minute Break
- Accelerators based on Intel® Movidius™ Vision Processing Unit
- Accelerators based on Intel® Arria® FPGA
- **Lab2 - DevCloud Sample Application: Accelerated Object Detection**
- Multiple Models in One Application
- DL Workbench + Demo
- DL Streamer
- **Lab3 – DevCloud Advanced Tutorials: DL Streamer Benchmark**

GETTING STARTED WITH INTEL® DISTRIBUTION OF OPENVINO™ TOOLKIT

Recommendations to the customer or developer

QUALIFY

- Use a trained model and check if framework is supported

  – *or* –

- Take advantage of a pre-trained model from the Open Model Zoo

INSTALLATION

- Download the Intel® OpenVINO™ toolkit package from Intel® Developer Zone, or by YUM or APT repositories
- Utilize the Getting Started Guide

PREPARE

- Understand sample demos and tools included
- Understand performance
- Choose hardware option with Performance Benchmarks
- Build, test and remotely run workloads on the Intel® DevCloud for the Edge before buying hardware

HANDS ON

- Visualize metrics with the Deep Learning Workbench
- Utilize prebuilt, Reference Implementations to become familiar with capabilities
- Optimize workloads with these performance best practices
- Use the Deployment Manager to minimize deployment package

SUPPORT

- Ask questions and share information with others through the Community Forum
- Engage using #OpenVINO on Stack Overflow
- Visit documentation site for guides, how to's, and resources
- Attend training and get certified
- Ready to go to market? Tell us how we can help

https://bit.ly/VINOsurvey