Using the Intel Distribution of the OpenVINO Toolkit for Deploying Accelerated Deep Learning Applications [2021.1]

Agenda

Part 1: Deploying Deep Learning-based Computer Vision Applications

- Intel® Smart Video/Computer vision Tools Overview
- Model Optimizer
- Post-Training Optimization Tool
- Inference Engine
- Accelerators based on Intel® Movidius™ Vision Processing Unit
- Multiple Models in One Application
- DL Workbench + Demo
- DL Streamer

15 Minutes Break

Part 2: DevCloud and Demos

- Intel® DevCloud for the Edge
- Demo - DevCloud Sample Application: Accelerated Object Detection

Part 3: Get a DevCloud Account

- Register for access to Intel® DevCloud for the Edge

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.

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.

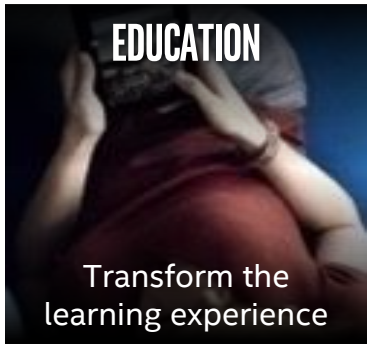
Notice revision #20110804

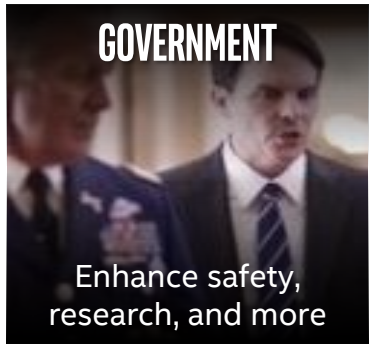
AI CHANGING AND ENABLING EVERY INDUSTRY

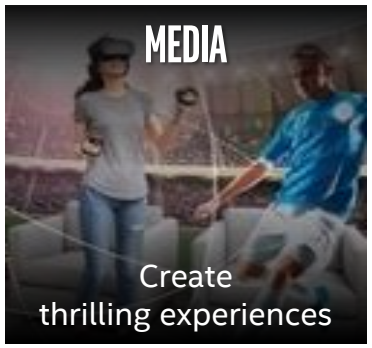
AI software market is projected to reach USD 126.0 billion in annual worldwide revenue by 2025[1]

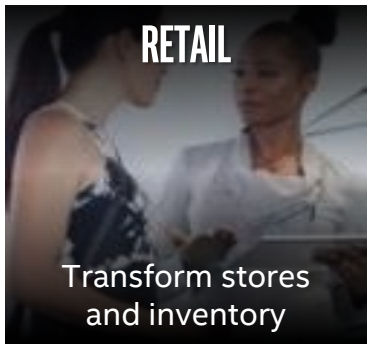
Deep learning software revenue is estimated to grow to USD 67.2 billion by 2025[2]

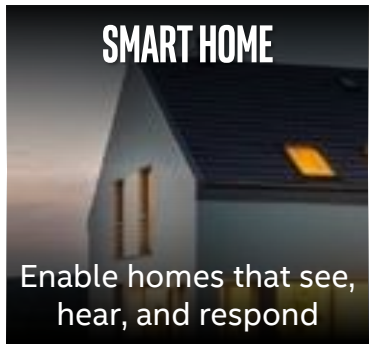
Global deep learning chip market is expected to reach USD 29.4 billion by 2025[3]

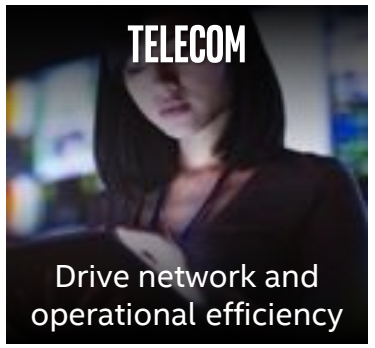


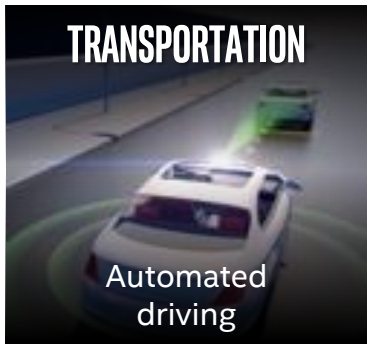












1. Tractica, Artificial Intelligence Software Market, 2020

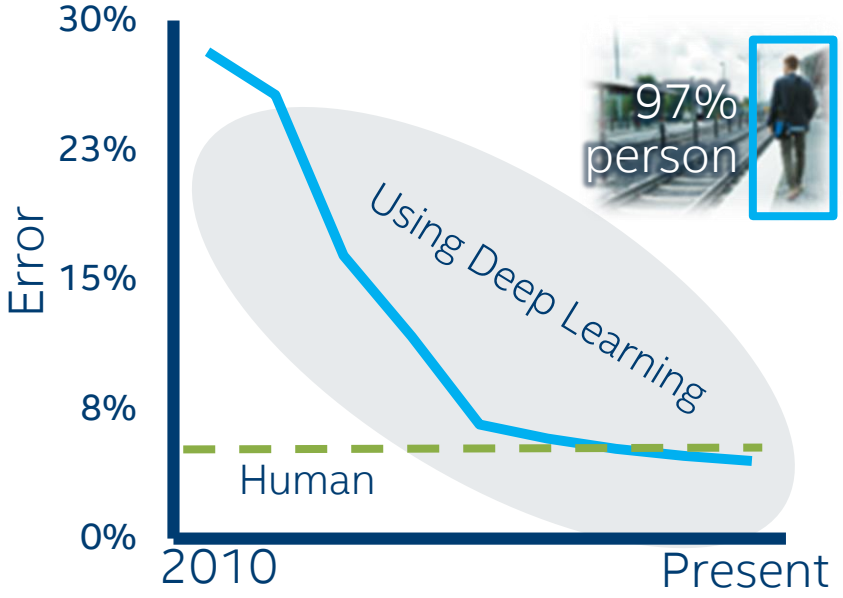
2. Tractica, deep learning research, 2018

3. AlliedMarketResearch, Deep Learning Chip Market, 2018

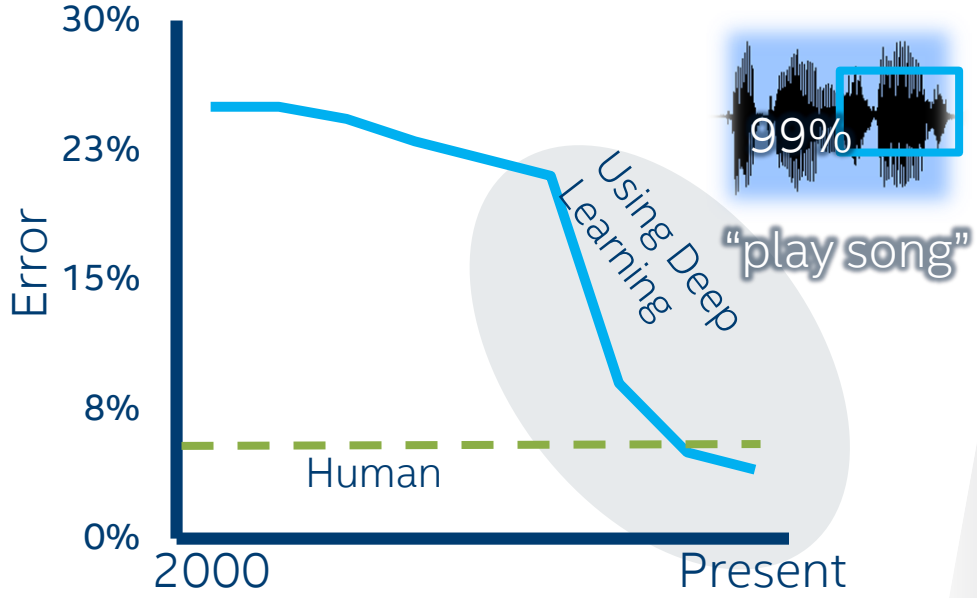
Deep learning breakthroughs and opportunities

Machines able to meet or exceed human image and speech recognition

Image Recognition



Speech Recognition



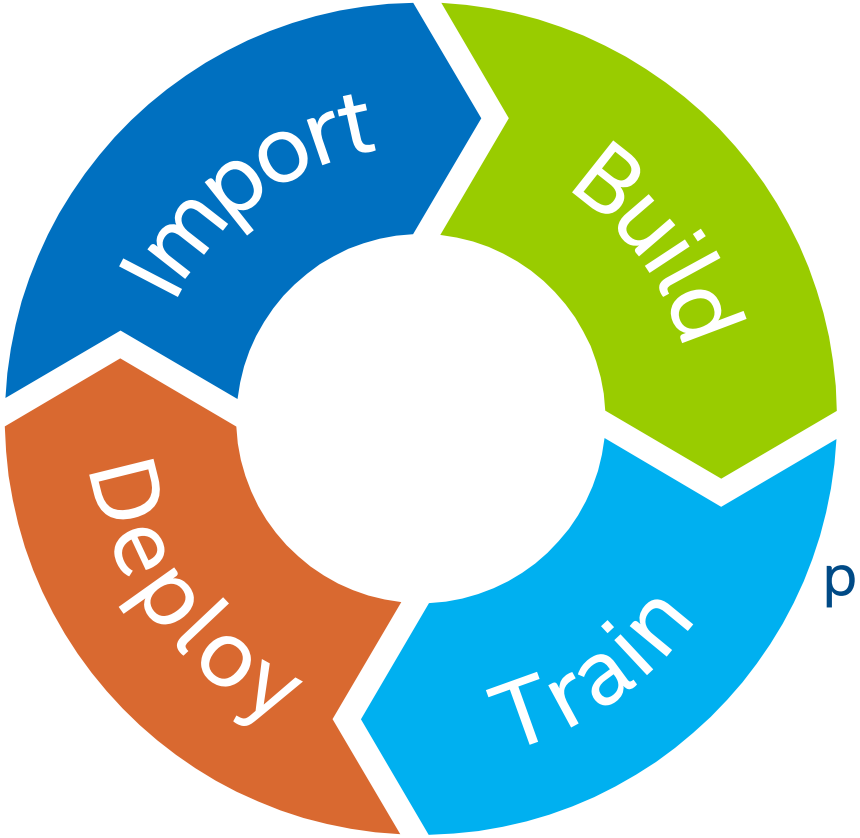
ADDITIONAL ECONOMIC IMPACT DRIVEN BY AI $13 TRILLION IN 2030

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



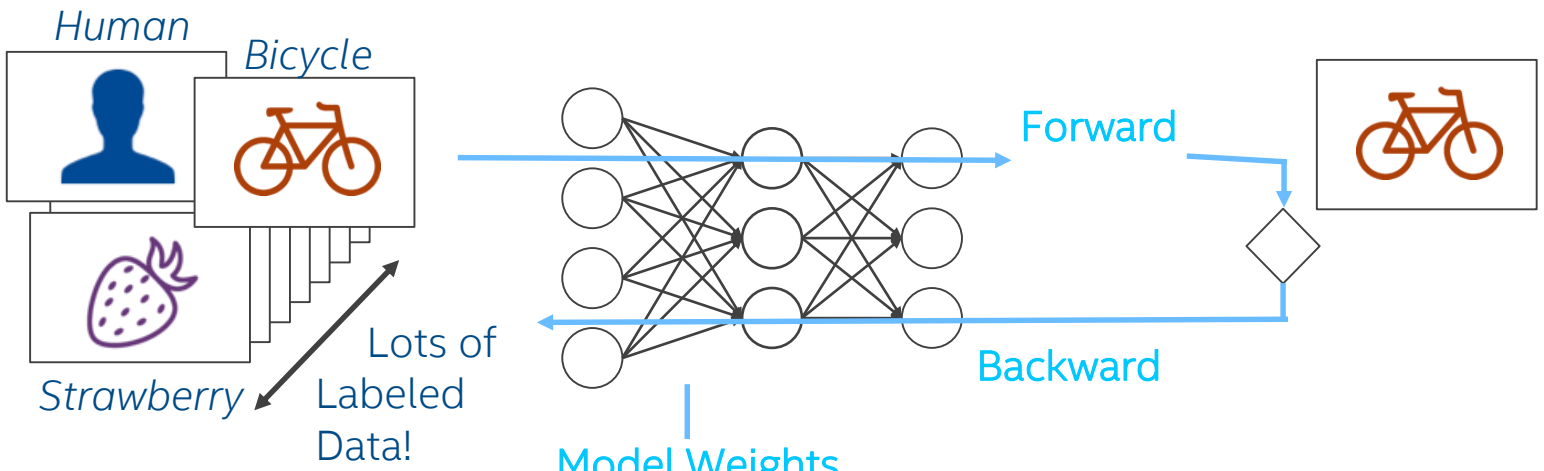
Integrate trained models with application code

Adjust models to meet performance and accuracy objectives

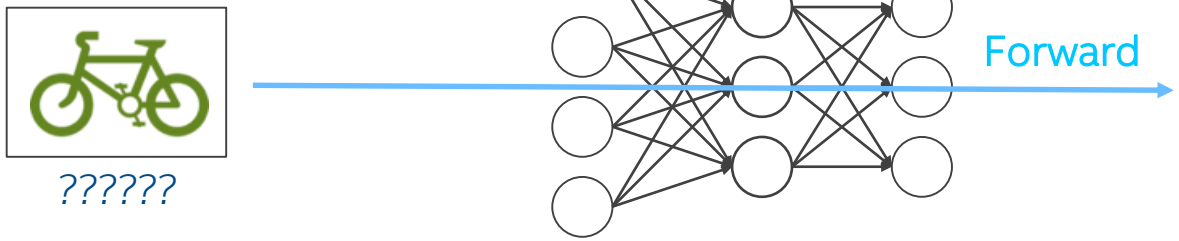
Intel® Distribution OpenVINO™ Toolkit Provides Deployment from Intel® Edge to Cloud

Deep Learning: Training vs. Inference

Training

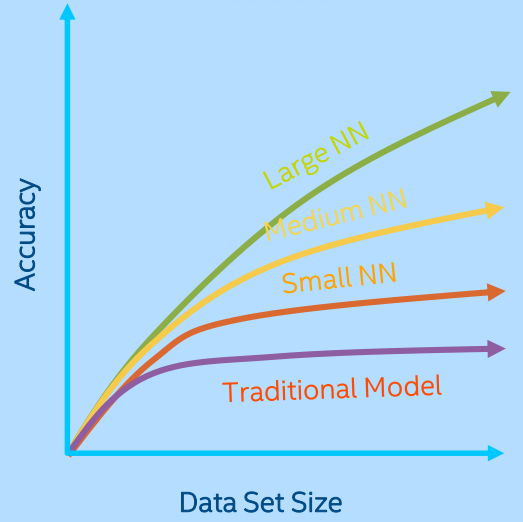


Inference



Did You Know?

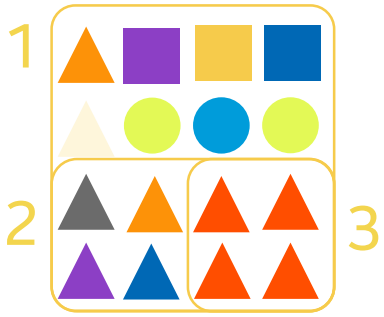
Training requires a very large data set and deep neural network (many layers) to achieve the highest accuracy in most cases



AI COMPUTE CONSIDERATIONS

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

WORKLOADS

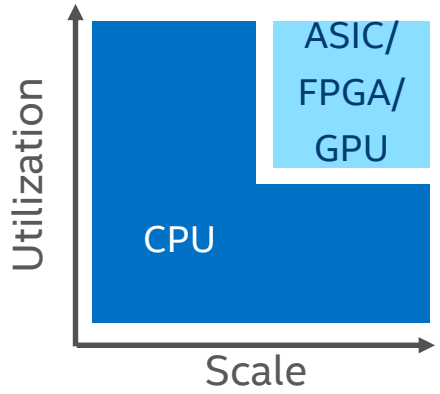


What is my workload profile?

REQUIREMENTS

What are my use case requirements?

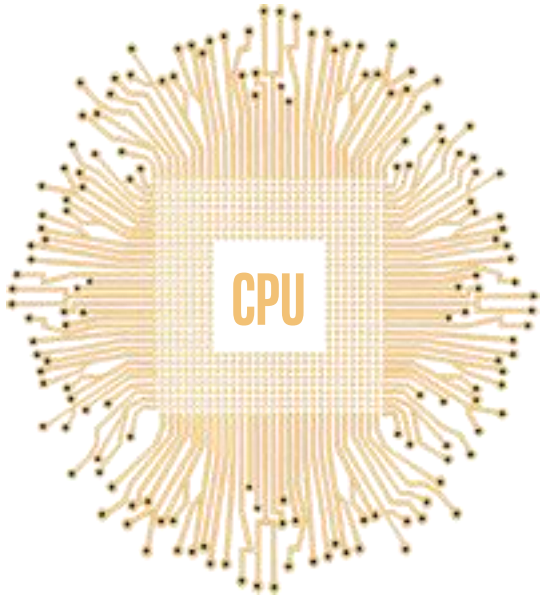
DEMAND



How prevalent is AI in my environment?

WHY INTEL AI COMPUTE?

MAXIMIZE

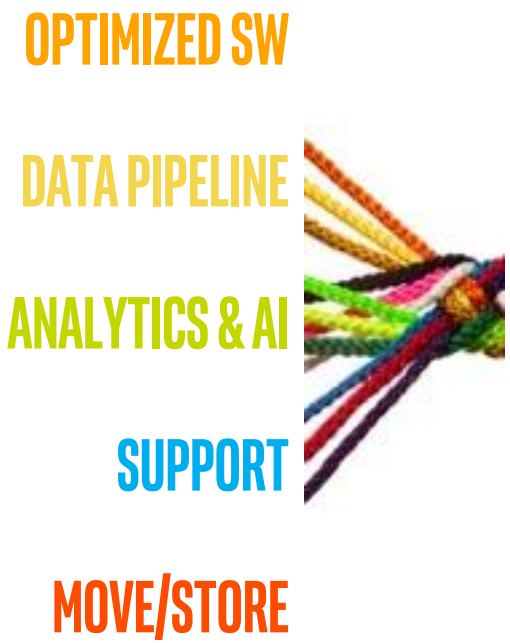


Get the most out of the foundation for AI from the CPU leader

OPTIMIZE

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

SIMPLIFY



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

New Features from OpenVINO Toolkit 2021.1

- Support for Tiger Lake (11th generation Intel® Core™ processors)
- New capabilities in OpenVINO™ Model Server
- Support for TensorFlow 2.x
- Support for non-computer vision workloads
- (Coming in Q4) Beta Release: Integration of OpenVINO™ toolkit DL Workbench and Intel® DevCloud for the Edge
- Support for GNA 2.0

Three steps for the Intel® Distribution of OpenVINO™ toolkit

1 Build

Trained Model

TensorFlow · Caffe · mxnet · KALDI · ONNX · PyTorch · PaddlePaddle

Open Model Zoo
100+ open sourced & optimized pre-trained models available

2 Optimize

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
Reduces model size into low-precision without re-training

Deep Learning Workbench
Visually analyze and fine-tune

Deep Learning Streamer

OpenCV | **OpenCL™**

Code Samples & Demos (e.g. Benchmark app, Accuracy Checker, Model Downloader)

3 Deploy

Model Server
gRPC/ REST Server with C++ backend

Deployment Manager

intel ATOM · intel CORE i7 · intel Xeon · intel IRIS Pro Graphics · intel MOVIDIUS · Intel GNA (IP) · intel ARRIA 10

Additional Tools and Add-ons from the OpenVINO GitHub Repo

Computer Vision Annotation Tool
This web-based tool helps annotate videos and images before training a model

Dataset Management Framework
Use this add-on to build, transform and analyze datasets

Neural Network Compression Framework
Training framework based on PyTorch* for quantization-aware training

[*NEW*] **OpenVINO™ Model Server**
Scalable inference server for serving optimized models and applications

Training Extensions
Trainable deep learning models for training with custom data

Speed up development with open source resources

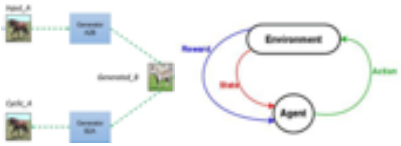
Open source resources with pre-trained models, samples and demos

Computer Vision

Object detection
Object recognition
Reidentification
Volumetric segmentation
Semantic segmentation
Instance segmentation
3D reconstruction
Human pose estimation
Image processing
Action recognition
Image super resolution

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

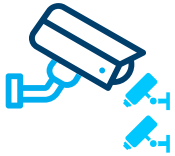
Speed up development with open source resources

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

Choose between Release Types

Standard Releases vs Long-Term Support Releases

Standard Release (3-4 releases a year): Users looking to take advantage of new features, tools and support in order to keep current with the advancements in deep learning technologies

Long-Term Support Release: Users looking for a stable and reliable version that is maintained for a longer period of time, and are looking for little to no new feature changes

Supported OS and Install Options [2021.1]

https://software.intel.com/content/www/us/en/develop/tools/openvino-toolkit.html

- **Operating Systems**
  - Ubuntu 18.04.x long-term support (LTS), 64-bit
  - CentOS 7.6, 64-bit (for target only)
  - Yocto Project v3.0, 64-bit (for target only and requires modifications)
  - Microsoft Windows* 10 64-bit
  - macOS* 10.15
  - Raspbian* Buster, Stretch

- **Install From Images and Repositories**
  - GitHub
    - https://github.com/openvinotoolkit/openvino.git
  - Anaconda Cloud
    - https://anaconda.org/intel/openvino-ie4py
  - Python* Package Installer (PIP)
    - https://pypi.org/project/openvino-python/
  - Docker
    - Install from Image file
    - Download from DockerHub
  - APT
    - sudo apt-cache search install-openvino-runtime-ubuntu18
  - YUM
    - sudo install intel-openvino-runtime-centos7

- **Intel® Edge Software Hub**
  - Edge Insights for Vision

Agenda

Part 1: Deploying Deep Learning-based Computer Vision Applications

- Intel® Smart Video/Computer vision Tools Overview
- Model Optimizer
- Post-Training Optimization Tool
- Inference Engine
- Accelerators based on Intel® Movidius™ Vision Processing Unit
- Multiple Models in One Application
- DL Workbench + Demo
- DL Streamer

15 Minutes Break

Part 2: DevCloud and Demos

- Intel® DevCloud for the Edge
- Demo - DevCloud Sample Application: Accelerated Object Detection

Part 3: Get a DevCloud Account

- Register for access to Intel® DevCloud for the Edge

Model Optimizer

Intel® Deep Learning Deployment Toolkit

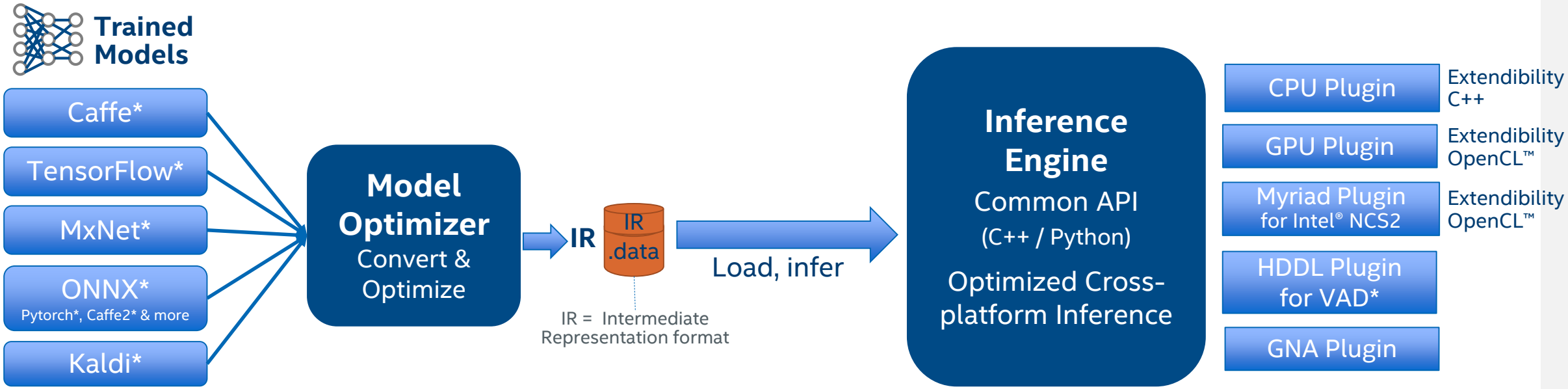
For Deep Learning Inference

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

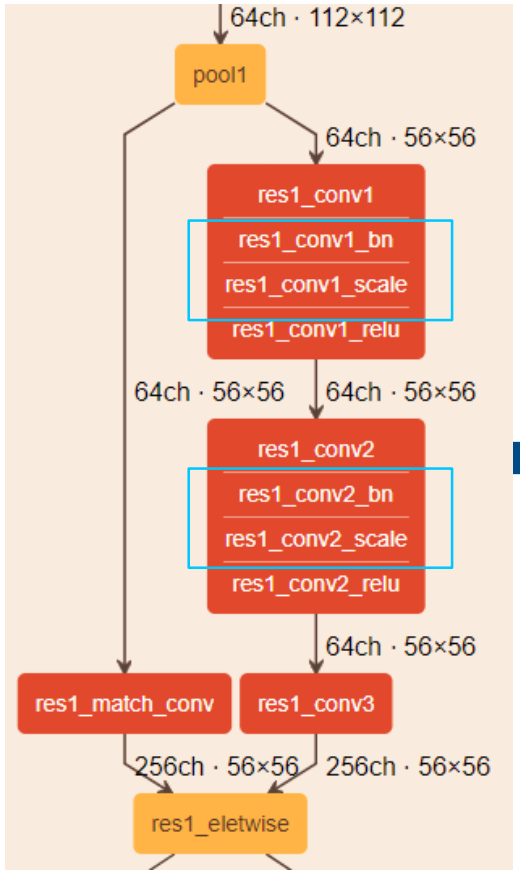
- By default, generic optimization will be automatically applied, unless manually set disable

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

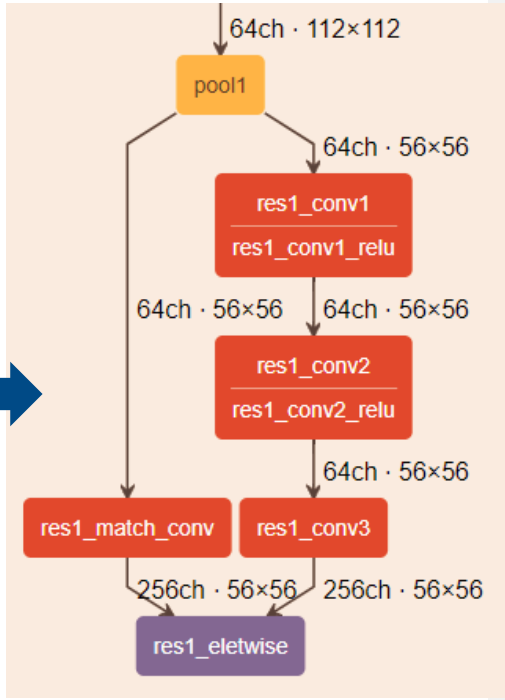
Model Optimization Techniques

- 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 FullyConnected layers.



Caffe* Resnet269 block (from Netscope)

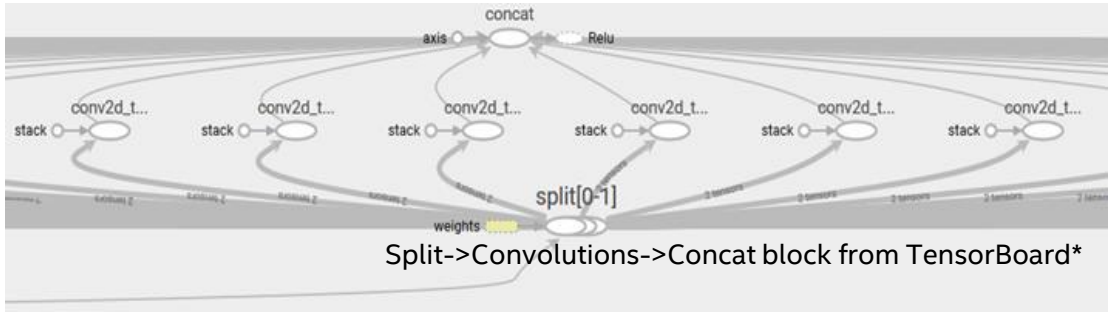


Merged Caffe* Resnet269 block (from Netscope*)

Model Optimizer: Framework or topology specific optimization

Grouped Convolutions Fusing

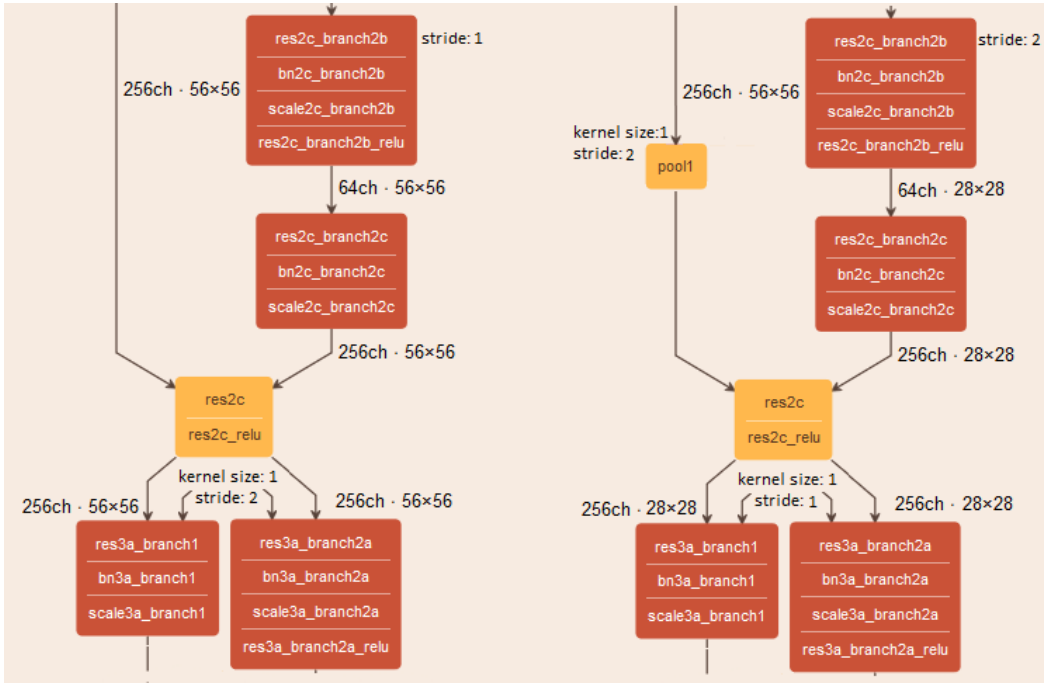
- Grouped convolution fusing is a specific optimization that applies for TensorFlow* topologies. The main idea of this optimization is to combine convolutions results for the Split outputs and then recombine them using **Concat** operation in the same order as they were out from **Split**.



Split->Convolutions->Concat block from TensorBoard*

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.



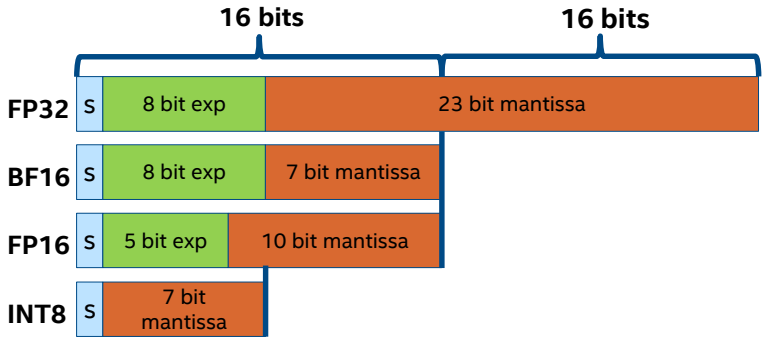
Model Optimizer: Quantization

--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

Model Optimizer: Other Common Parameters

- --scale, --scale_values, --mean_values, --mean_file
  - Usually neural network models are trained with the normalized input data. This means that the input data values are converted to be in a specific range, for example, [0, 1] or [-1, 1]. Sometimes the mean values (mean images) are subtracted from the input data values as part of the pre-processing
- --input_shape
  - when the input data shape for the model is not fixed, like for the fully-convolutional neural networks. In this case, for example, TensorFlow* models contain -1 values in the shape attribute of the Placeholder operation. Inference Engine does not support input layers with undefined size, so if the input shapes are not defined in the model, the Model Optimizer fails to convert the model.
- --reverse_input_channels
  - Inference Engine samples load input images in the BGR channels order. However, the model may be trained on images loaded with the opposite order

Agenda

Part 1: Deploying Deep Learning-based Computer Vision Applications

- Intel® Smart Video/Computer vision Tools Overview
- Model Optimizer
- Post-Training Optimization Tool
- Inference Engine
- Accelerators based on Intel® Movidius™ Vision Processing Unit
- Multiple Models in One Application
- DL Workbench + Demo
- DL Streamer

15 Minutes Break

Part 2: DevCloud and Demos

- Intel® DevCloud for the Edge
- Demo - DevCloud Sample Application: Accelerated Object Detection

Part 3: Get a DevCloud Account

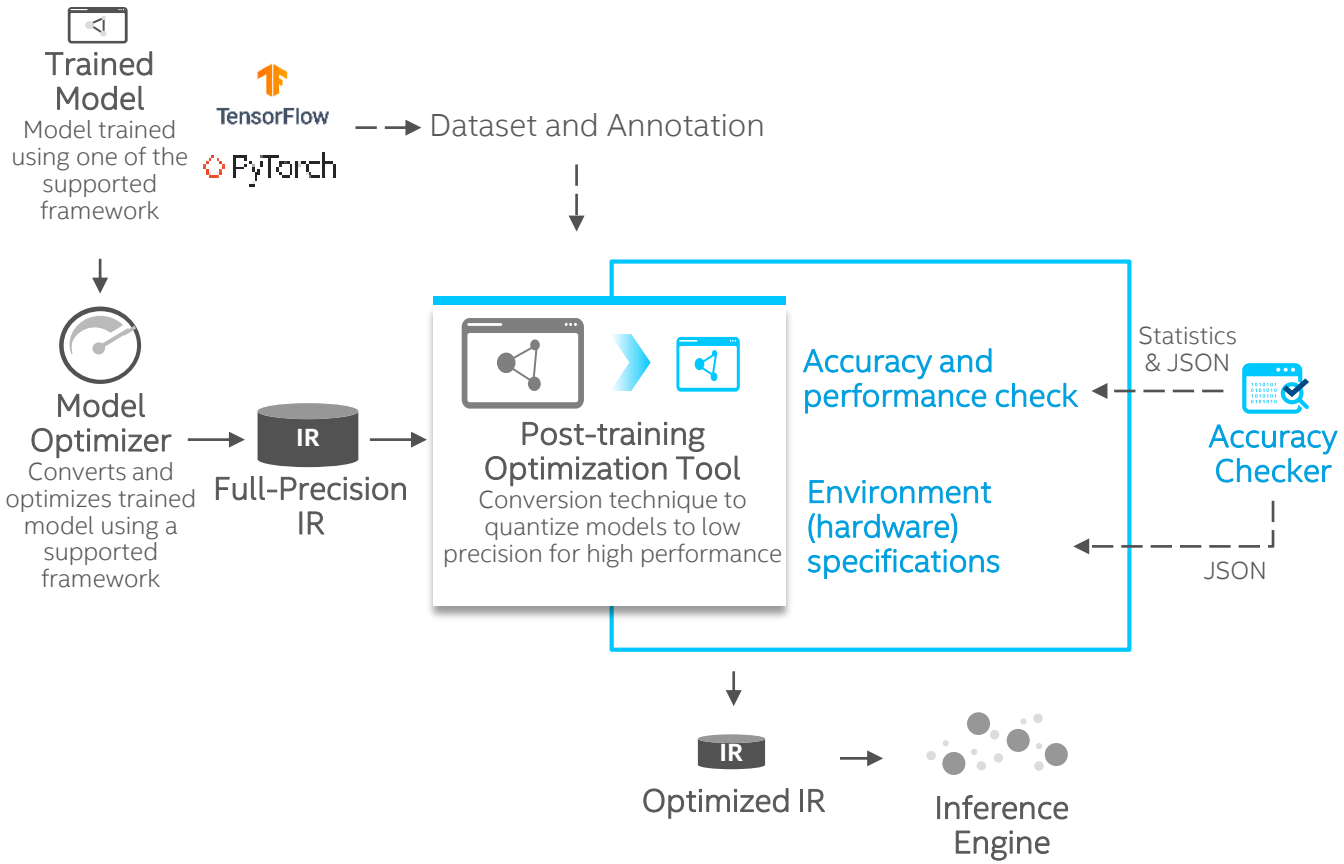
- Register for access to Intel® DevCloud for the Edge

Post-Training Optimization Tool

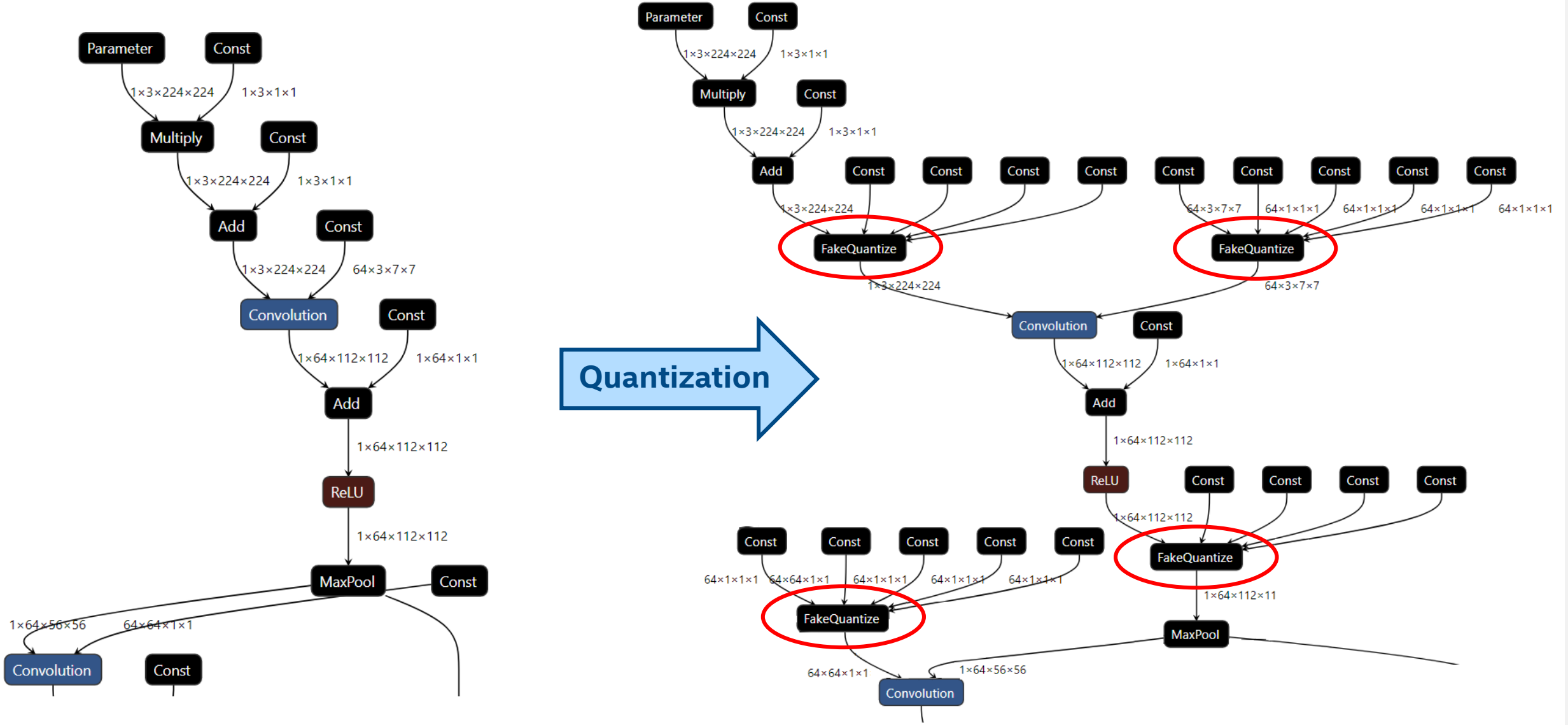
Post-Training Optimization Tool

- 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.

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



IR transformation

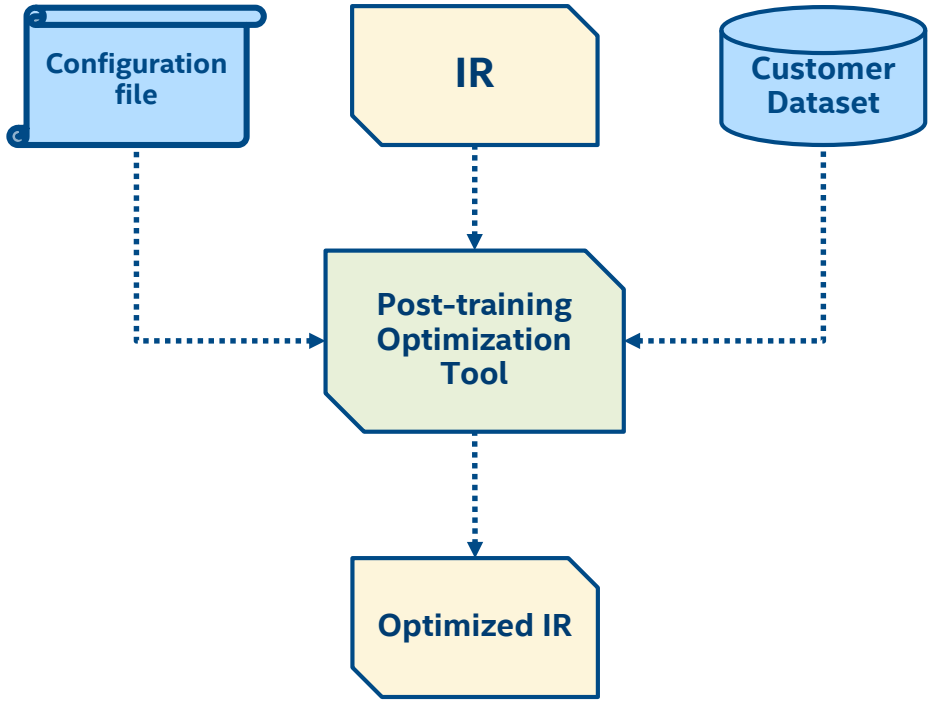


Post-Training Optimization Tool – features

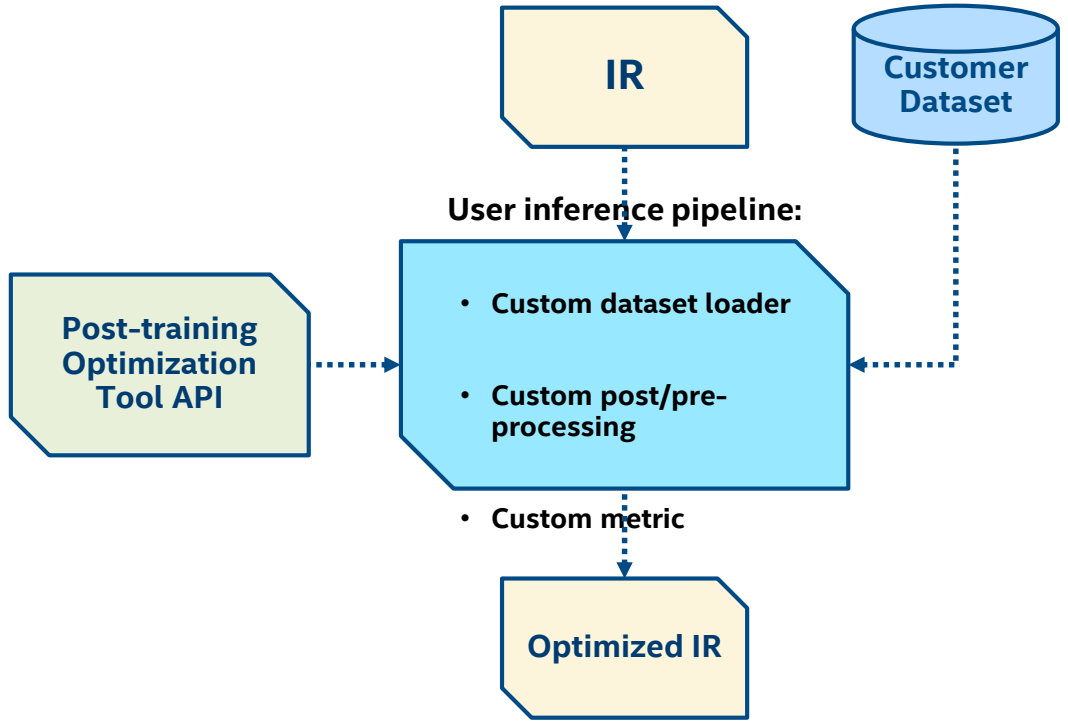
- Supports quantization of OpenVINO™ toolkit's IR models for various types of Intel® hardware
- *Learn more:* https://docs.openvinotoolkit.org/latest/_compression_algorithms_quantization_README.html
  - Two main algorithms supported and exposed through Deep Learning Workbench:
    - Default algorithm: essentially a pipeline running three base algorithms:
      i. Activation Channel Alignment (applied to align activation ranges)
      ii. MinMax
      iii. Bias Correction (runs atop naive algorithm; based on minimization of per-channel quantization error)
    - Accuracy-Aware algorithm: preserves accuracy of the resulting model, keeping accuracy drop below threshold
  - Provides hardware-specific configurations
  - Features per-channel/per-tensor quantization granularity
  - Supports symmetric/asymmetric quantization through presets mechanism

Usage scenarios

1 Used as-is. Command line/Workbench scenarios.



2 Integration in user pipeline.



Agenda

Part 1: Deploying Deep Learning-based Computer Vision Applications

- Intel® Smart Video/Computer vision Tools Overview
- Model Optimizer
- Post-Training Optimization Tool
- Inference Engine
- Accelerators based on Intel® Movidius™ Vision Processing Unit
- Accelerators based on Intel® Arria® FPGA
- Multiple Models in One Application
- DL Workbench + Demo
- DL Streamer

15 Minutes Break

Part 2: DevCloud and Demos

- Intel® DevCloud for the Edge
- Demo - DevCloud Sample Application: Accelerated Object Detection

Part 3: Get a DevCloud Account

- Register for access to Intel® DevCloud for the Edge

Inference Engine

Intel® Deep Learning Deployment Toolkit

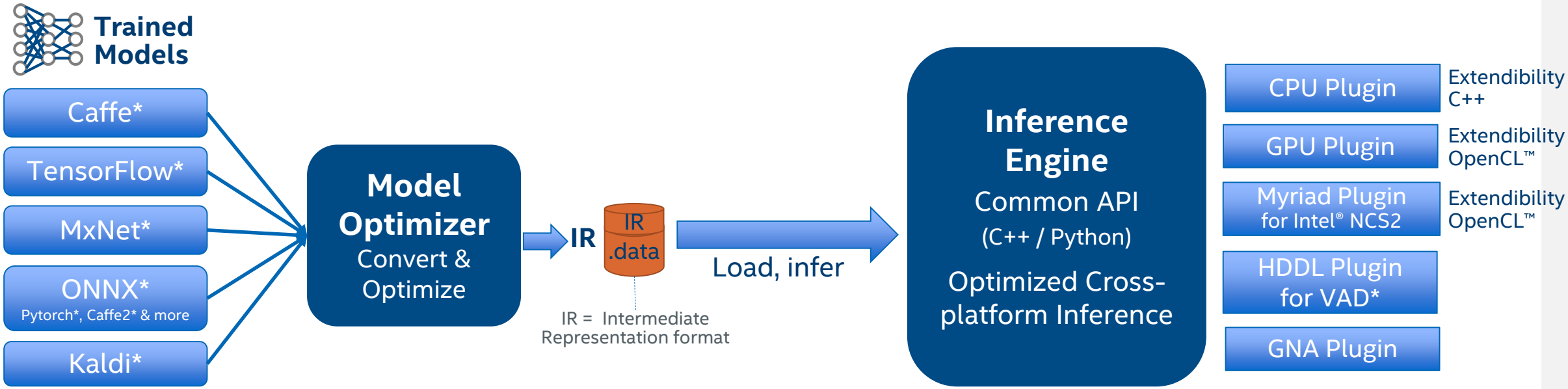
For Deep Learning Inference

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)

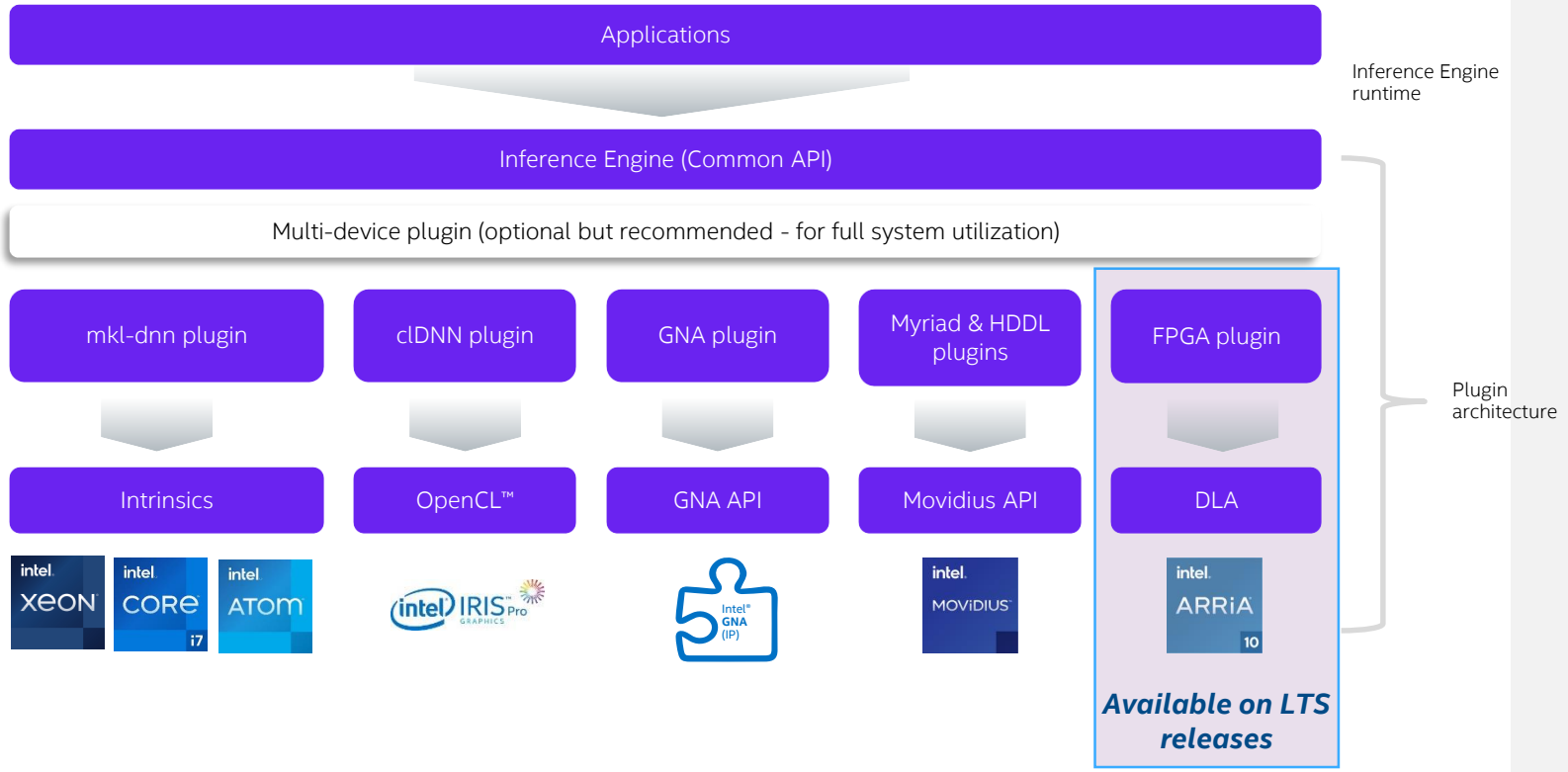
Optimal Model Performance Using the Inference Engine

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

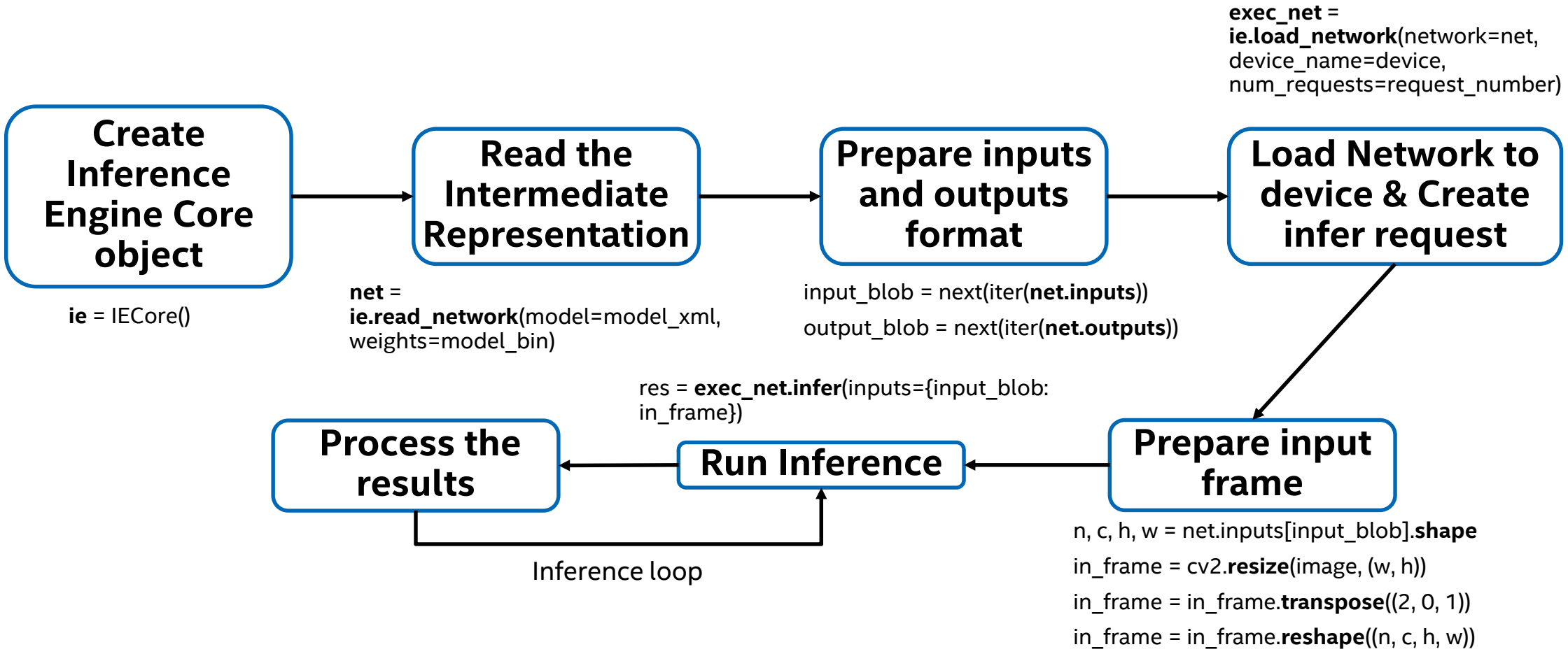
- For each supported target device, Inference Engine provides a plugin — a DLL/shared library that contains complete implementation for inference on this device.



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

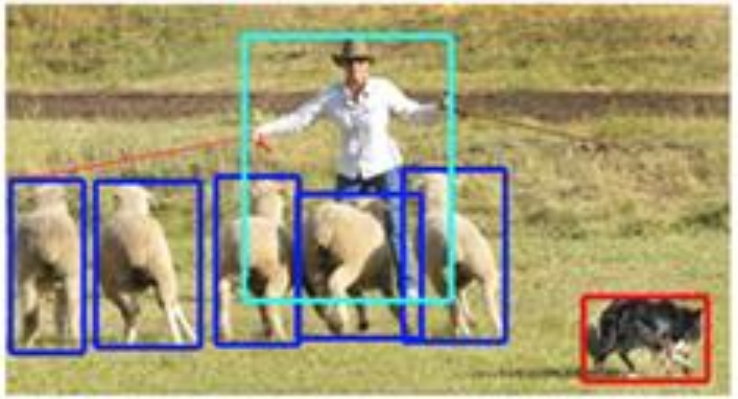


http://docs.openvinotoolkit.org/latest/_docs_IE_DG_Integrate_with_customer_application_new_API.html

Inference on an Intel® Edge System

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

(a) classification


(b) detection


(c) segmentation

- 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])

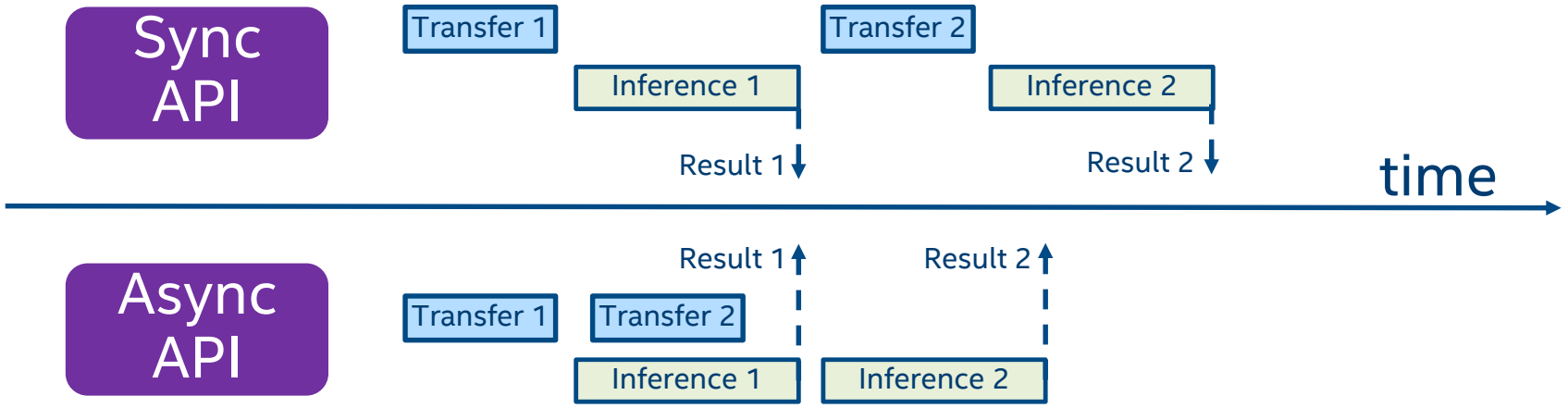
```
res = res[out_blob]
boxes, classes = {}, {}
data = res[0][0]
for number, proposal in enumerate(data):
    if proposal[2] > 0:
        imid = np.int(proposal[0])
        ih, iw = images_hw[imid]
        label = np.int(proposal[1])
        confidence = proposal[2]
        xmin = np.int(iw * proposal[3])
        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

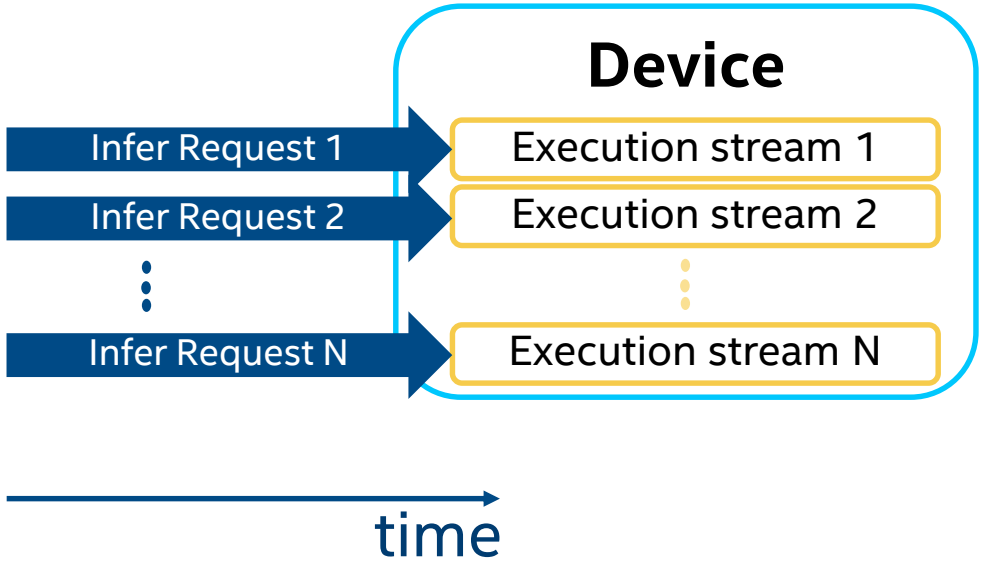
    do something



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



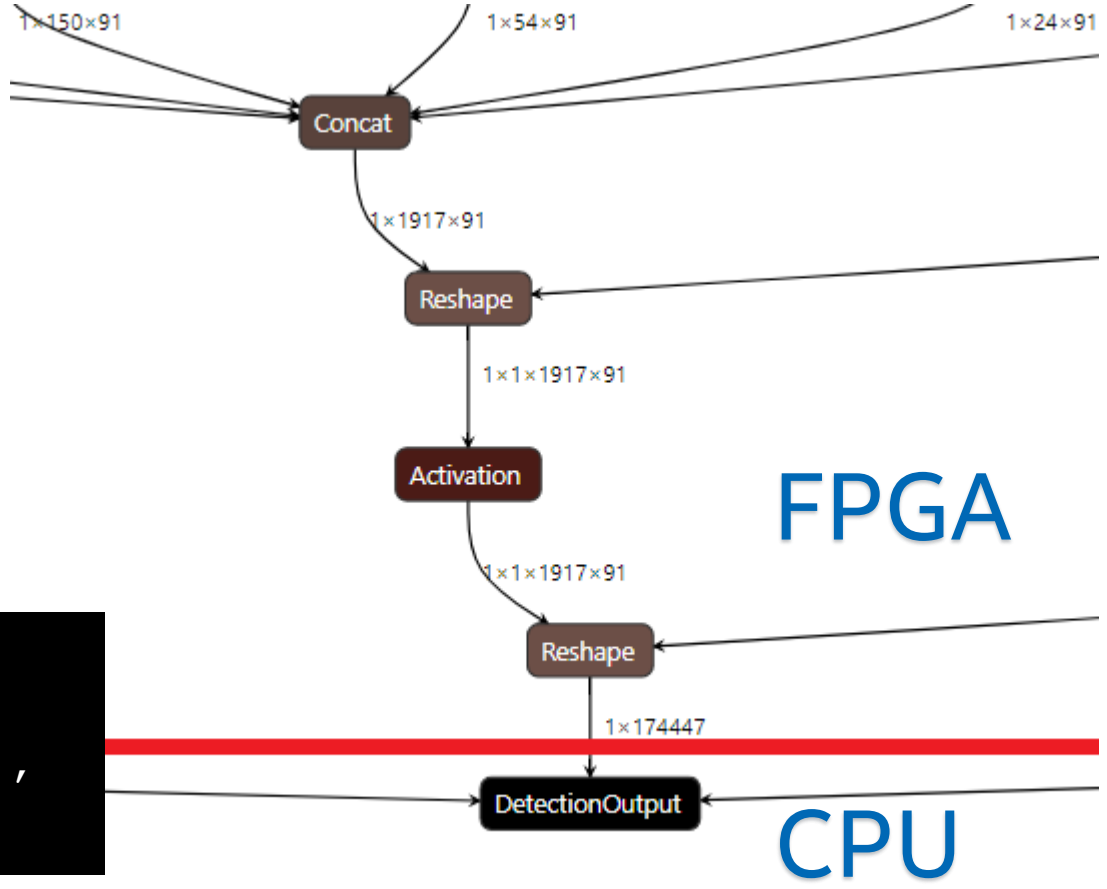
CPU Example:

```
ie = IECore()
ie.GetConfig(CPU, KEY_CPU_THROUGHPUT_STREAMS)
```

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");
```

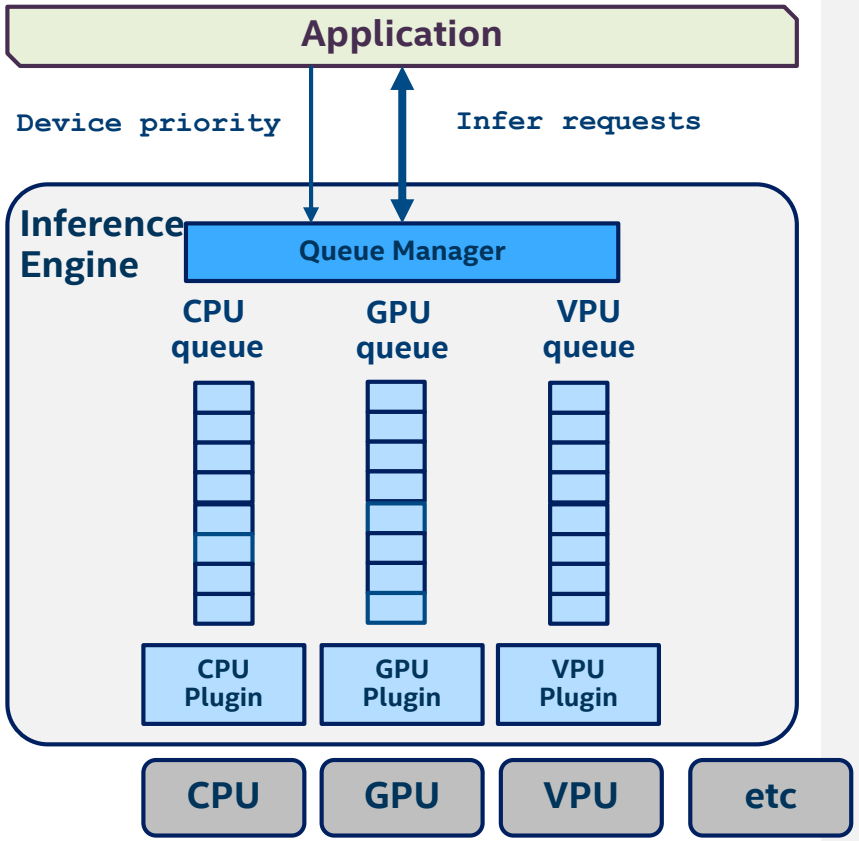
Inference Engine

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”)
```



Agenda

Part 1: Deploying Deep Learning-based Computer Vision Applications

- Intel® Smart Video/Computer vision Tools Overview
- Model Optimizer
- Post-Training Optimization Tool
- Inference Engine
- Accelerators based on Intel® Movidius™ Vision Processing Unit
- Multiple Models in One Application
- DL Workbench + Demo
- DL Streamer

15 Minutes Break

Part 2: DevCloud and Demos

- Intel® DevCloud for the Edge
- Demo - DevCloud Sample Application: Accelerated Object Detection

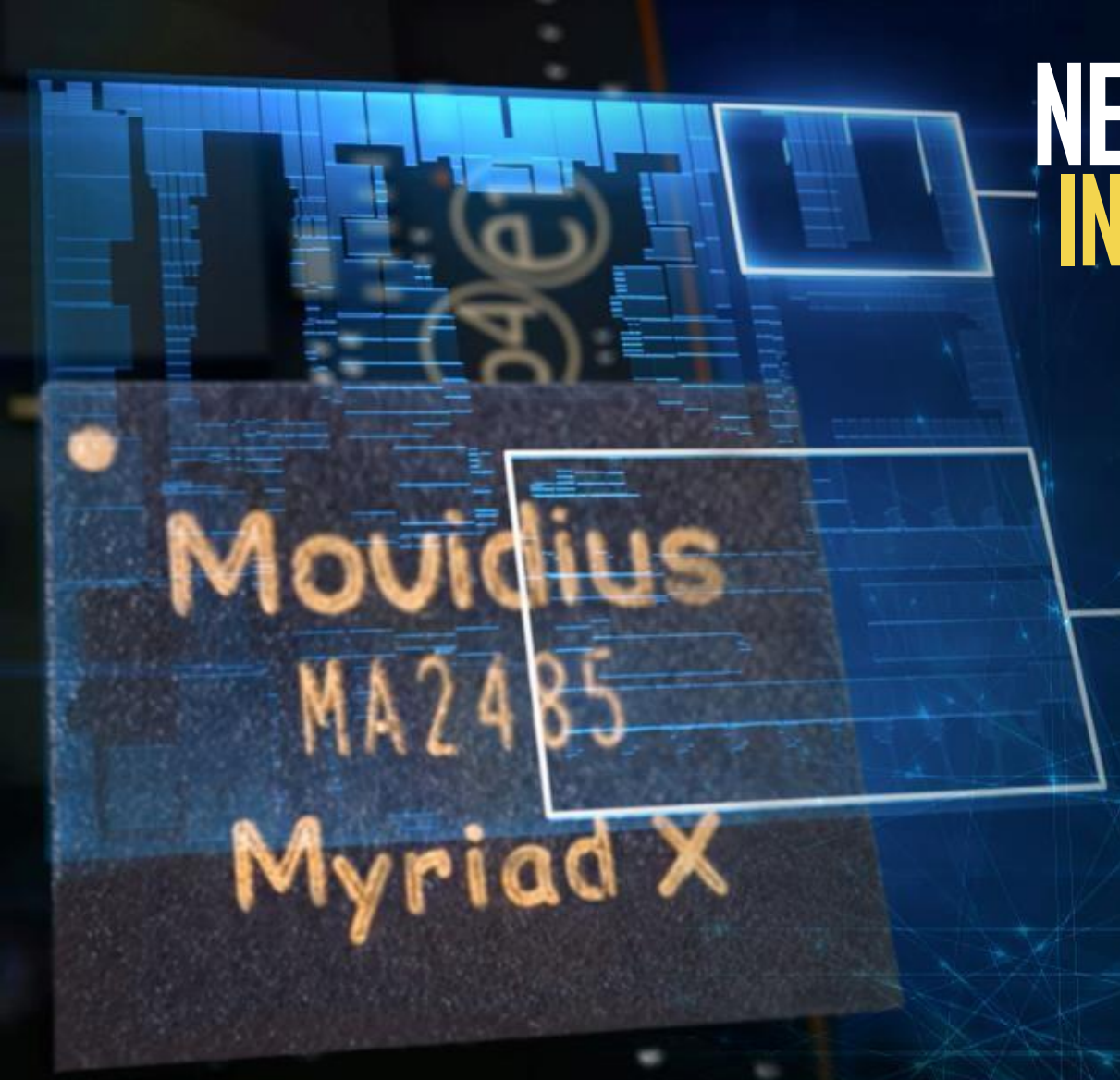
Part 3: Get a DevCloud Account

- Register for access to Intel® DevCloud for the Edge

Accelerators based on Intel® Movidius™ Vision Processing Unit

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



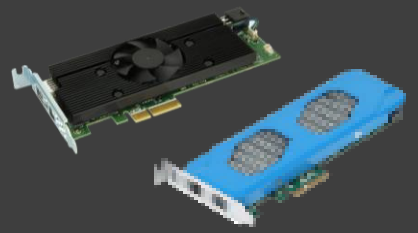
NEXT GENERATION AI INFERENCE
INTEL® MOVIDIUS™ MYRIAD™ X VPU

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

Hardware-based encoder
for up to 4K video resolution and includes a new stereo depth block that is capable of processing dual 720p feeds at up to 180Hz.

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, ADLINK, ADVANTECH, iEi, JW IPC, NEXCOM, tinyGO, uzei		
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

Part 1: Deploying Deep Learning-based Computer Vision Applications

- Intel® Smart Video/Computer vision Tools Overview
- Model Optimizer
- Post-Training Optimization Tool
- Inference Engine
- Accelerators based on Intel® Movidius™ Vision Processing Unit
- Multiple Models in One Application
- DL Workbench + Demo
- DL Streamer

15 Minutes Break

Part 2: DevCloud and Demos

- Intel® DevCloud for the Edge
- Demo - DevCloud Sample Application: Accelerated Object Detection

Part 3: Get a DevCloud Account

- Register for access to Intel® DevCloud for the Edge

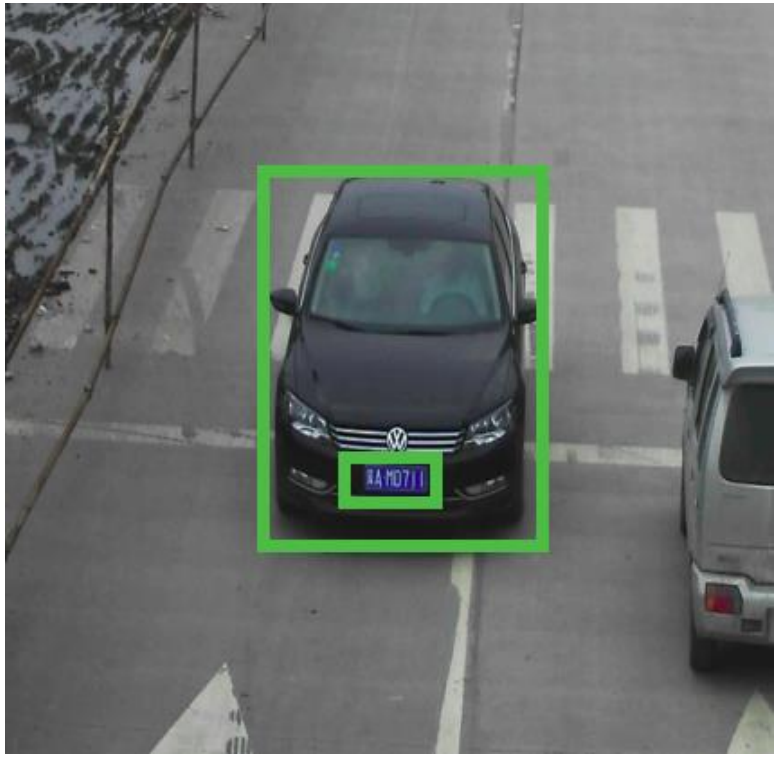
Multiple Models in One Application Security barrier demo

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.

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.

Security Barrier Demo

Agenda

Part 1: Deploying Deep Learning-based Computer Vision Applications

- Intel® Smart Video/Computer vision Tools Overview
- Model Optimizer
- Post-Training Optimization Tool
- Inference Engine
- Accelerators based on Intel® Movidius™ Vision Processing Unit
- Multiple Models in One Application
- DL Workbench + Demo
- DL Streamer

15 Minutes Break

Part 2: DevCloud and Demos

- Intel® DevCloud for the Edge
- Demo - DevCloud Sample Application: Accelerated Object Detection

Part 3: Get a DevCloud Account

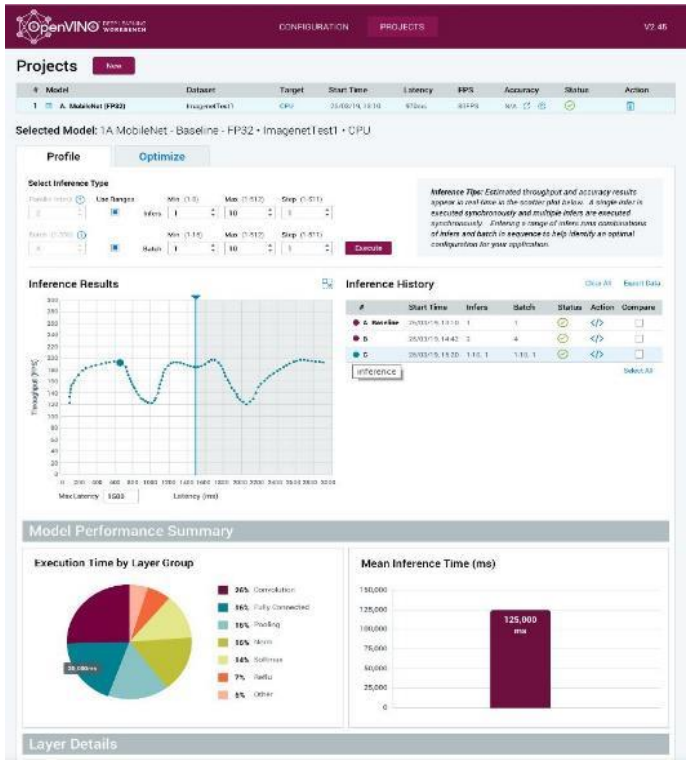
- Register for access to Intel® DevCloud for the Edge

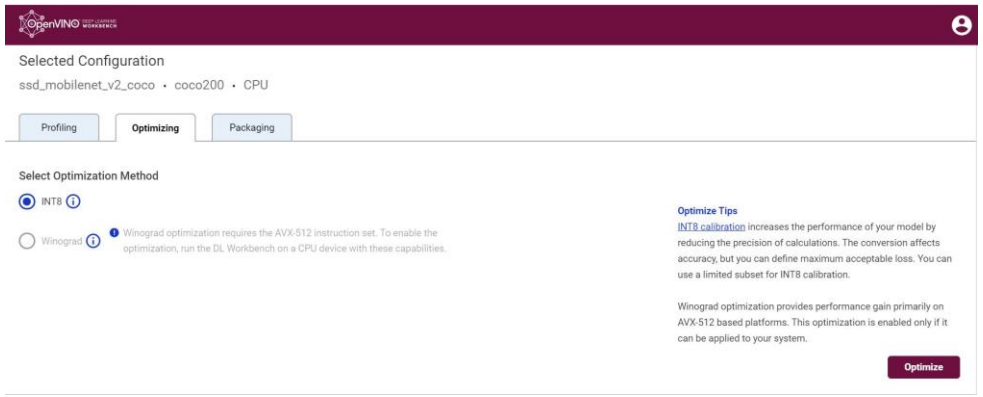
Deep Learning Workbench

Deep Learning Workbench

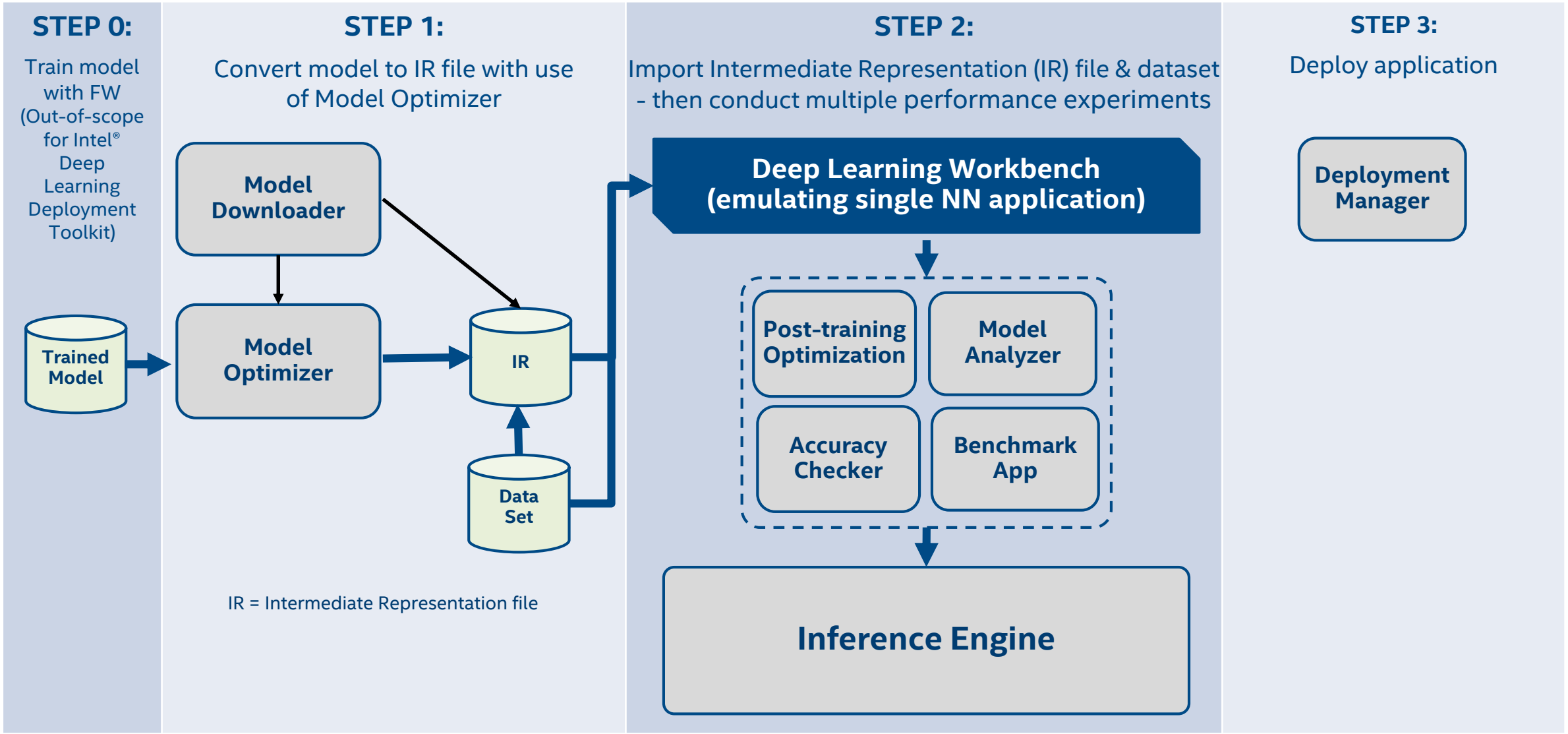
- Web-based, **UI extension tool** of the Intel® Distribution of OpenVINO™ toolkit
- **Visualizes performance data for** topologies and layers to aid in model analysis
- **Automates analysis** for optimal performance configuration (streams, batches, latency)
- **Experiment with INT8 or Winograd calibration** for optimal tuning using the Post Training Optimization Tool
- Provide **accuracy information** 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





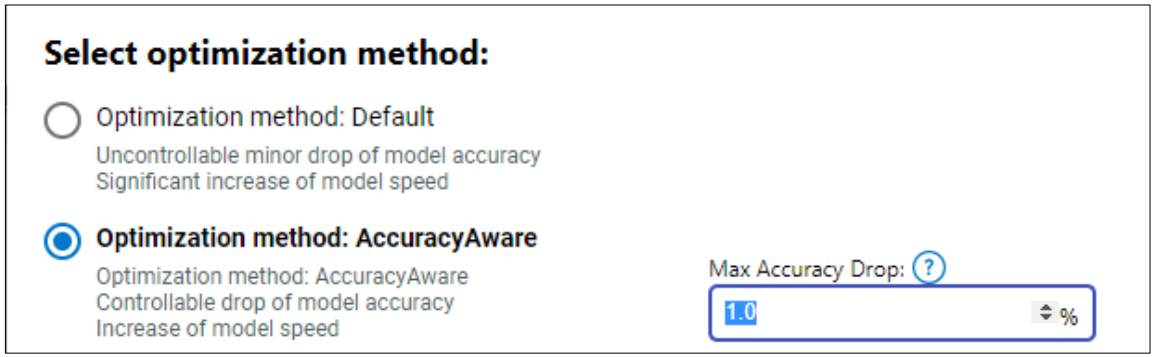
Deep Learning Workbench Data Flow

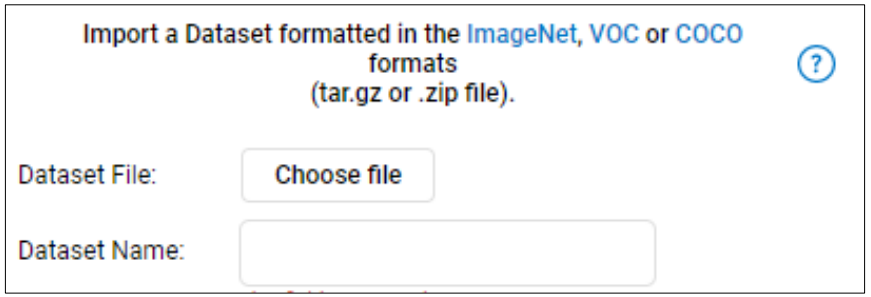


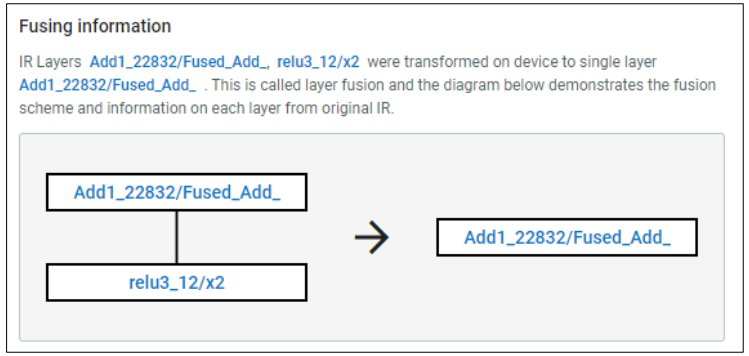
Work with Models and Sample Datasets

DEEP LEARNING WORKBENCH : FEATURES

- Convert model to Int8 using 2 new calibration algorithms
- Import dataset in COCO format to use with model
- Improved per-layer data visualization and comparison mode

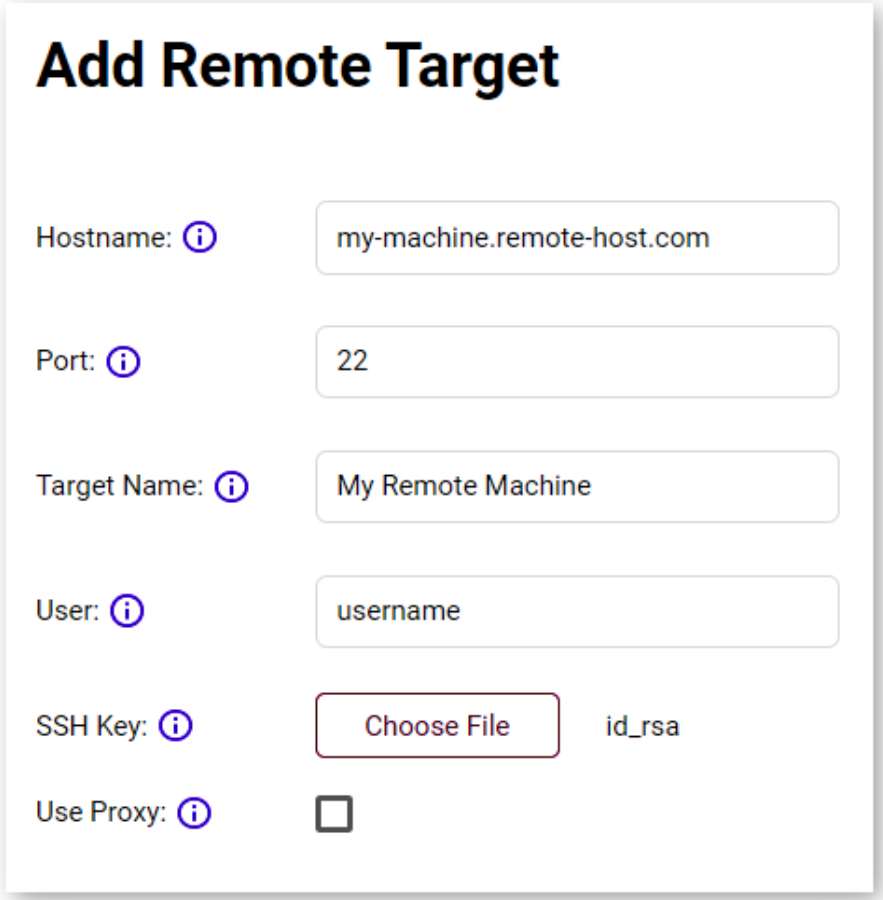




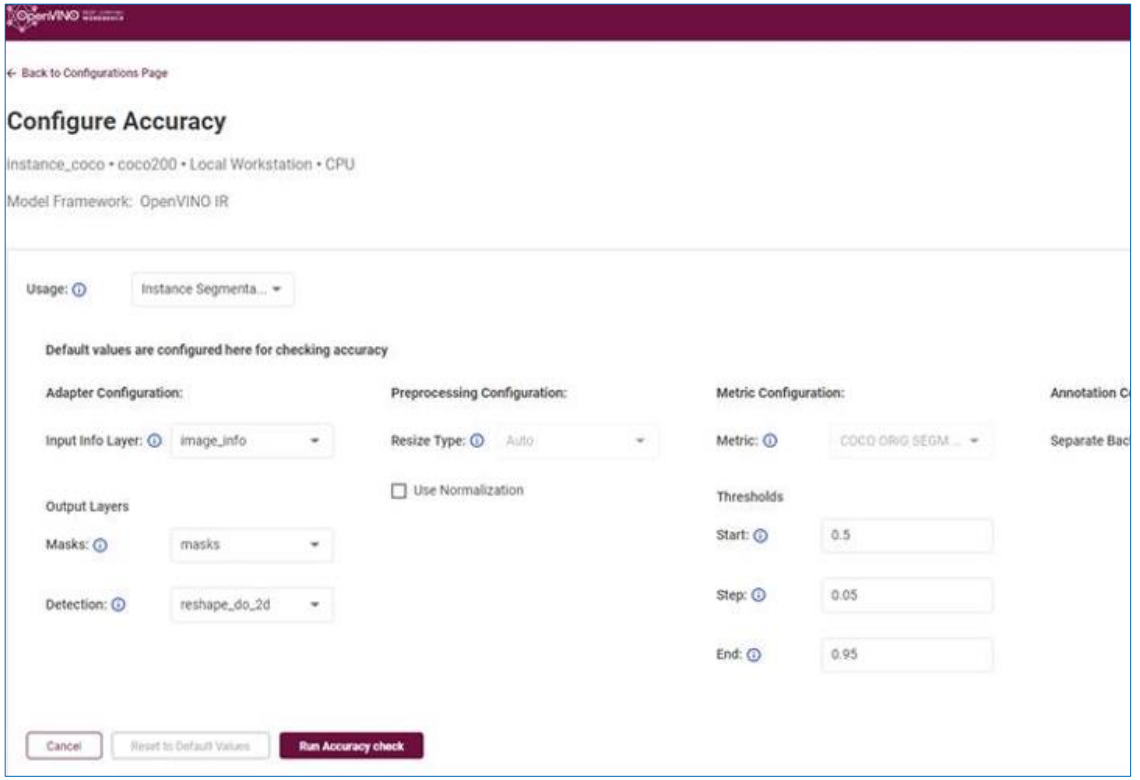


DEEP LEARNING WORKBENCH : FEATURES

Remote profiling support



Support for Segmentation use cases



Demo - DL Workbench Walkthrough

Agenda

Part 1: Deploying Deep Learning-based Computer Vision Applications

- Intel® Smart Video/Computer vision Tools Overview
- Model Optimizer
- Post-Training Optimization Tool
- Inference Engine
- Accelerators based on Intel® Movidius™ Vision Processing Unit
- Multiple Models in One Application
- DL Workbench + Demo
- DL Streamer

15 Minutes Break

Part 2: DevCloud and Demos

- Intel® DevCloud for the Edge
- Demo - DevCloud Sample Application: Accelerated Object Detection

Part 3: Get a DevCloud Account

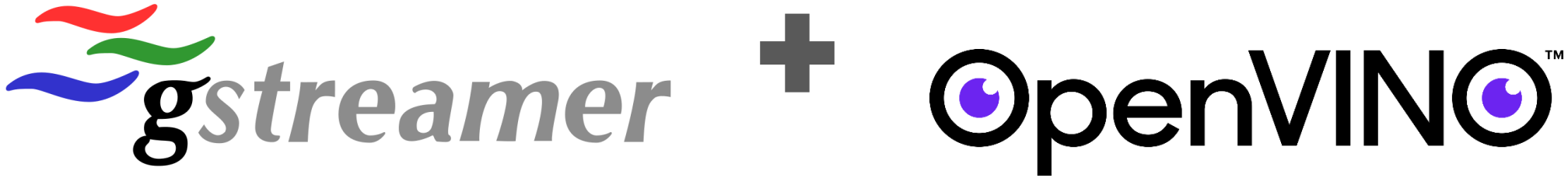
- Register for access to Intel® DevCloud for the Edge

Deep Learning Streamer

intel®

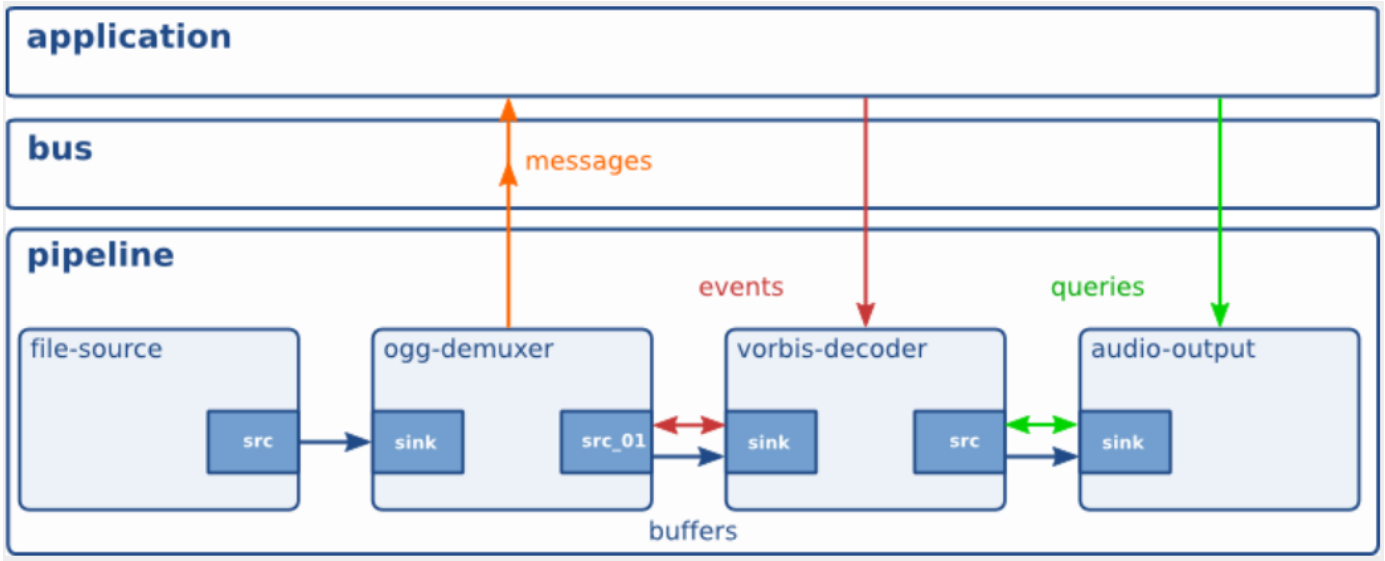
Introducing.. Dl streamer

- 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

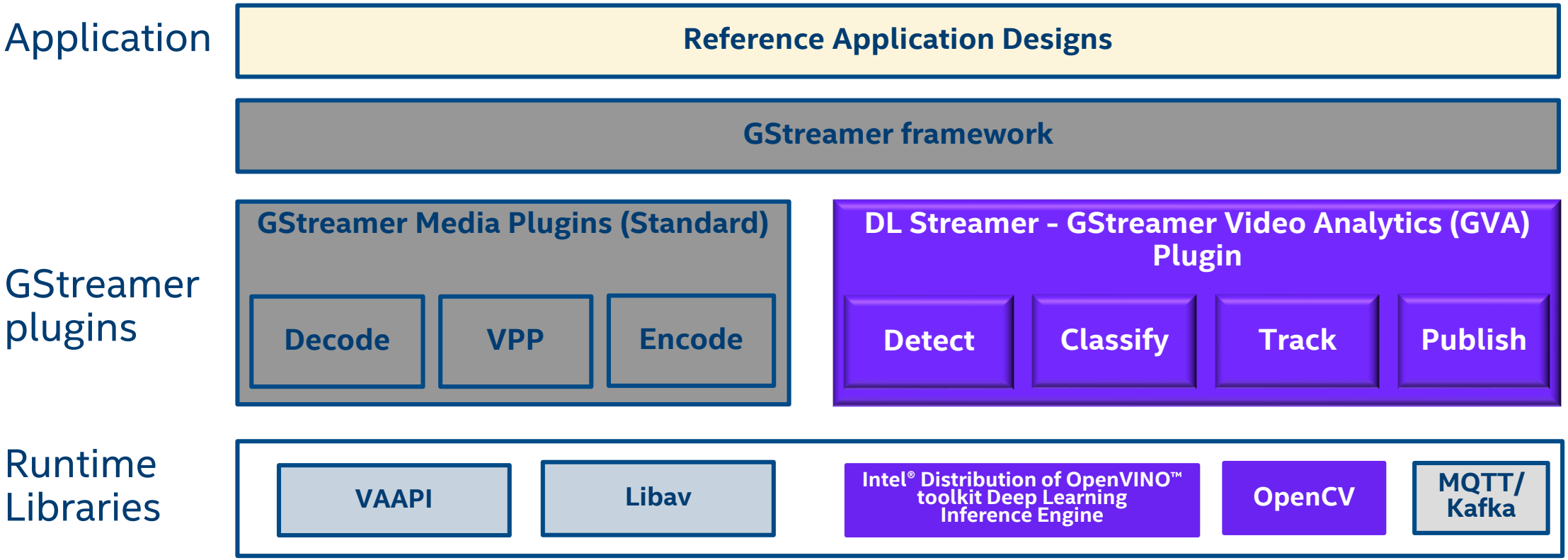
Video Pipeline – decode, convert, render

filesrc — decodebin — videoconvert — xvimagesink

input	HW/SW decode	convert	render on screen

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

Under the hood: DL Streamer

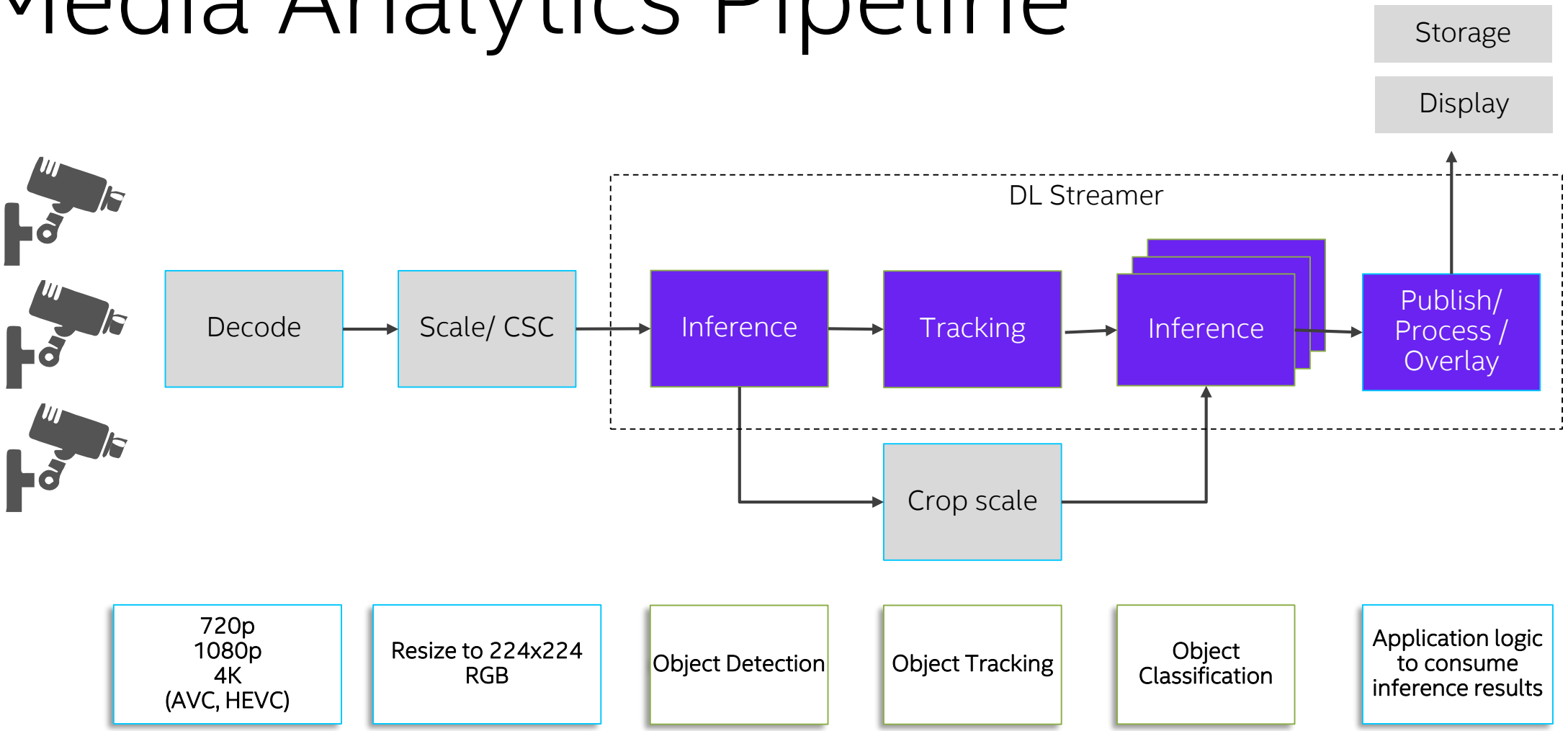


WANT TO KNOW MORE: CHECK OUT THE WEBINAR

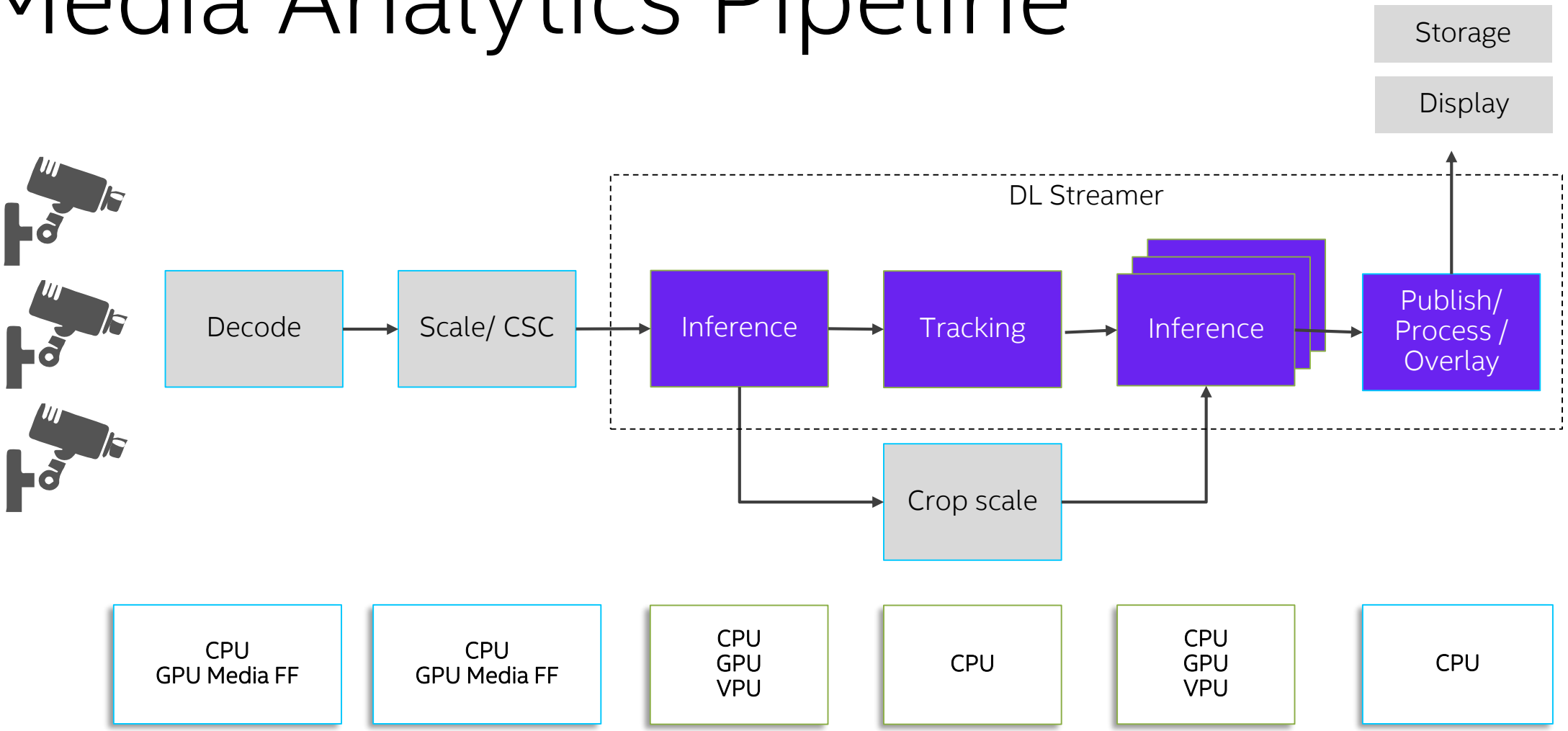
HTTPS://SOFTWARE.SEEK.INTEL.COM/OPENVINO-WEBINAR-SERIES

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

Media Analytics Pipeline

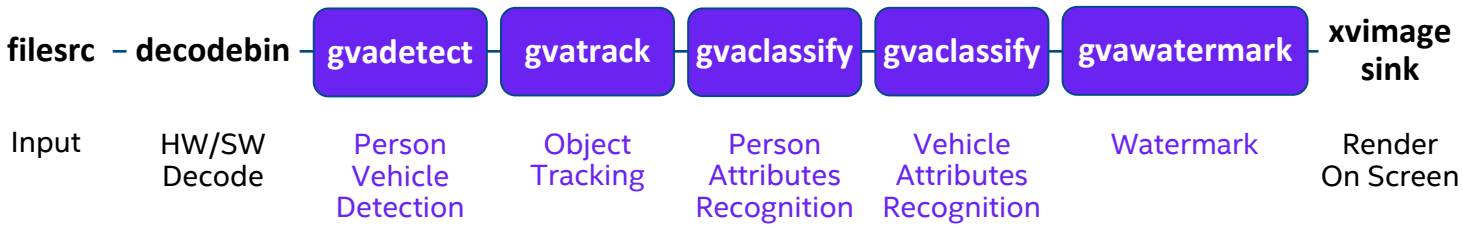


Media Analytics Pipeline



Using the DL Streamer

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




```
 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
```

Audio Processing

DL Streamer for end-to-end audio analytics pipeline

- Intel® Distribution of OpenVINO™ toolkit Deep Learning (DL) Streamer, 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
- Introduces gvaaudiodetect for audio event detection
  - Can be paired with alcnet public model for end-to-end audio analytics pipeline

DL Streamer Elements:

- gvaaudiodetect for audio event detection using ACLNet
- gvametaconvert for converting ACLNet detection results into JSON for further processing and display
- gvametapublish for printing detection results to stdout

15 mins break

- 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

Part 1: Deploying Deep Learning-based Computer Vision Applications

- Intel® Smart Video/Computer vision Tools Overview
- Model Optimizer
- Post-Training Optimization Tool
- Inference Engine
- Accelerators based on Intel® Movidius™ Vision Processing Unit
- Multiple Models in One Application
- DL Workbench + Demo
- DL Streamer

15 Minutes Break

Part 2: DevCloud and Demos

- Intel® DevCloud for the Edge
- Demo - DevCloud Sample Application: Accelerated Object Detection

Part 3: Get a DevCloud Account

- Register for access to Intel® DevCloud for the Edge

Intel® DevCloud for the Edge

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

Accelerate Test Cycles with the Intel® DevCloud for the Edge

A Development Sandbox for Developers, Researchers, and Startups to Test AI and Vision Workloads Remotely before Deployment.

With the Intel® DevCloud for the Edge users can:

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

[New]DL Workbench + Intel® DevCloud for the Edge

Developers can now graphically analyze models using the DL Workbench on Intel® DevCloud for the Edge (instead of local machine only) to compare, visualize and fine-tune a solution against multiple remote hardware configurations

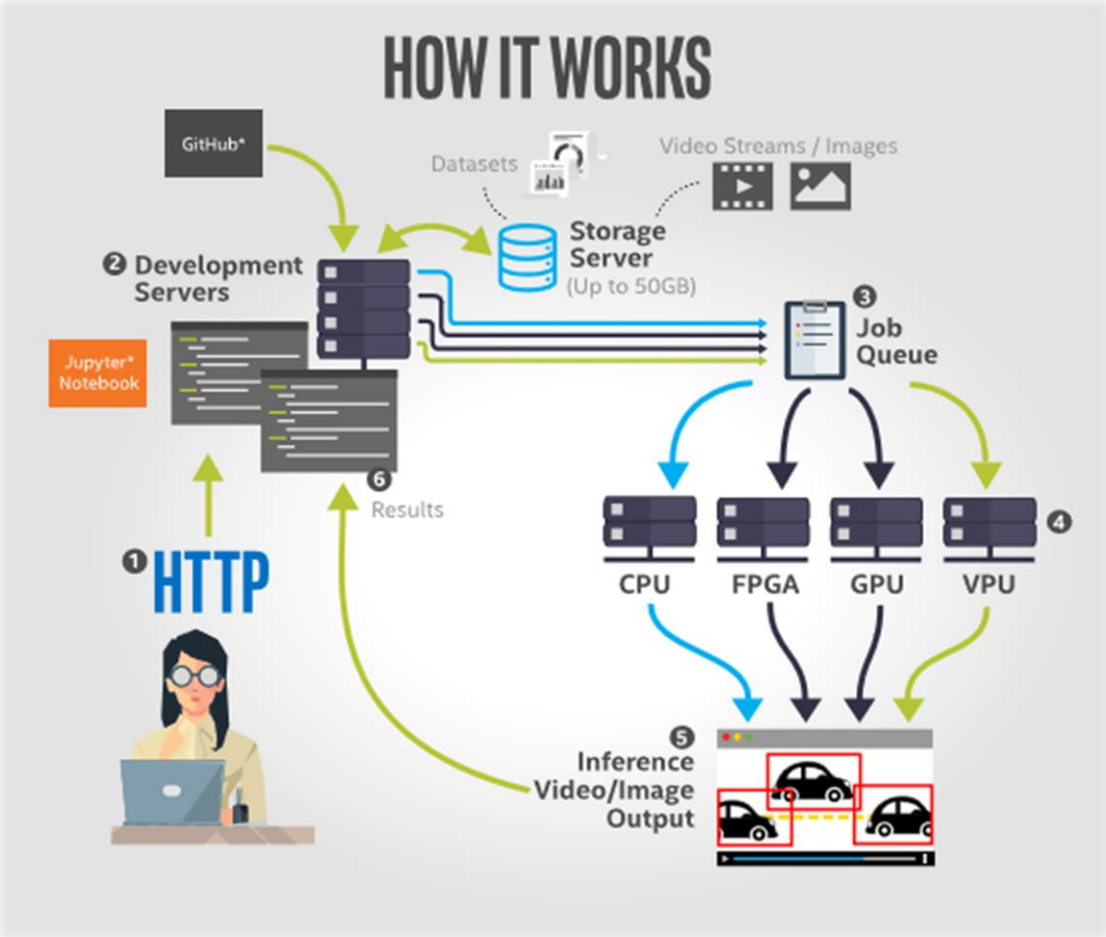
Intel® DevCloud for the Edge

Deploy and scale

For more information visit ▸ https://devcloud.intel.com/edge/

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

Demo: 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.

Agenda

Part 1: Deploying Deep Learning-based Computer Vision Applications

- Intel® Smart Video/Computer vision Tools Overview
- Model Optimizer
- Inference Engine
- Accelerators based on Intel® Movidius™ Vision Processing Unit
- Accelerators based on Intel® Arria® FPGA
- Multiple Models in One Application
- DL Workbench + Demo
- DL Streamer

15 Minutes Break

Part 2: DevCloud and Demos

- Intel® DevCloud for the Edge
- Demo - DevCloud Sample Application: Accelerated Object Detection

Part 3: Get a DevCloud Account

- Register for access to Intel® DevCloud for the Edge

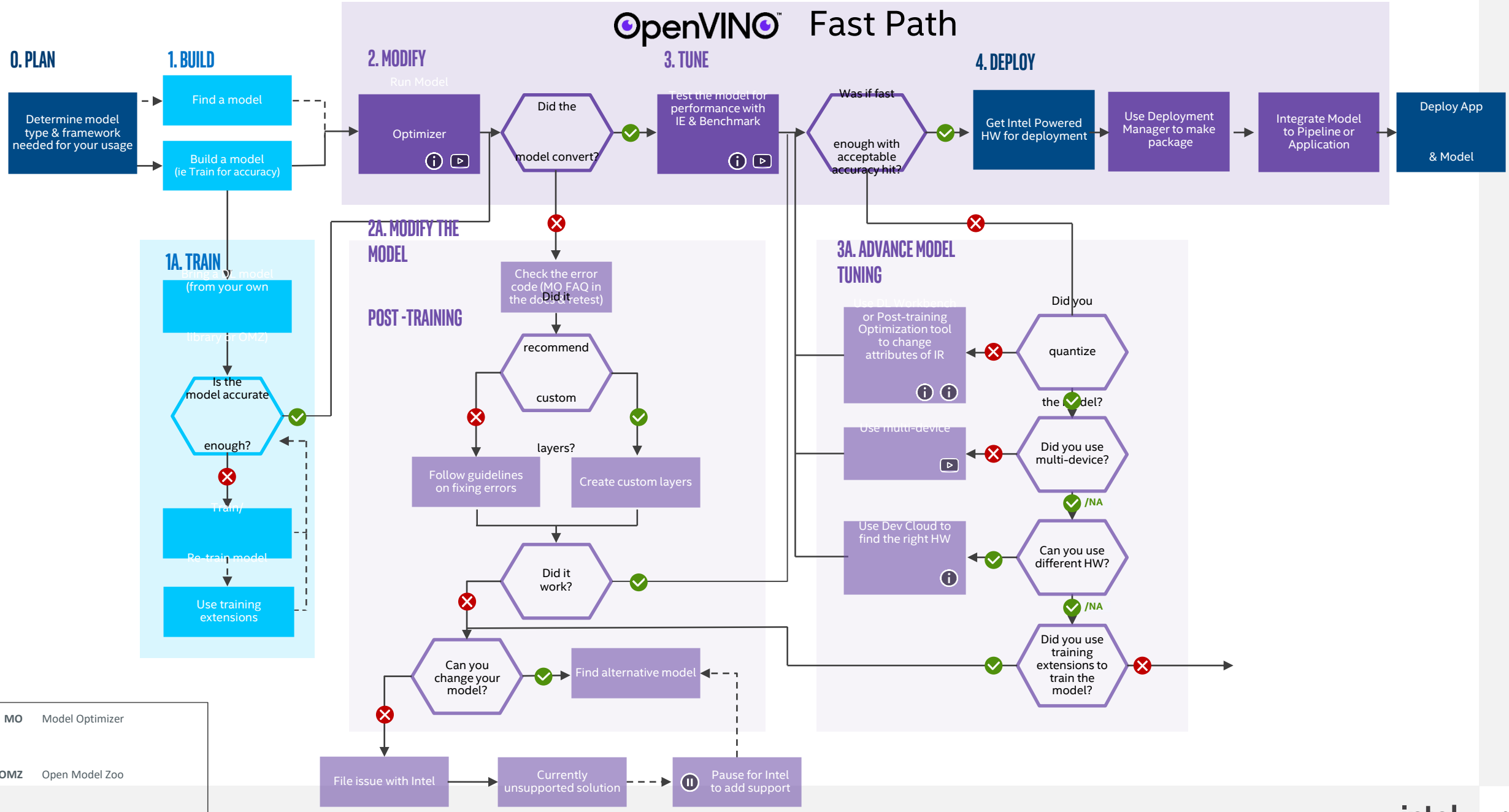
Signup for Access to the Intel® DevCloud for Edge

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

Intel's Registration Passcode:	
Code Valid From:	
Code Valid To:	
Access Duration in Days:	Valid for 30 days

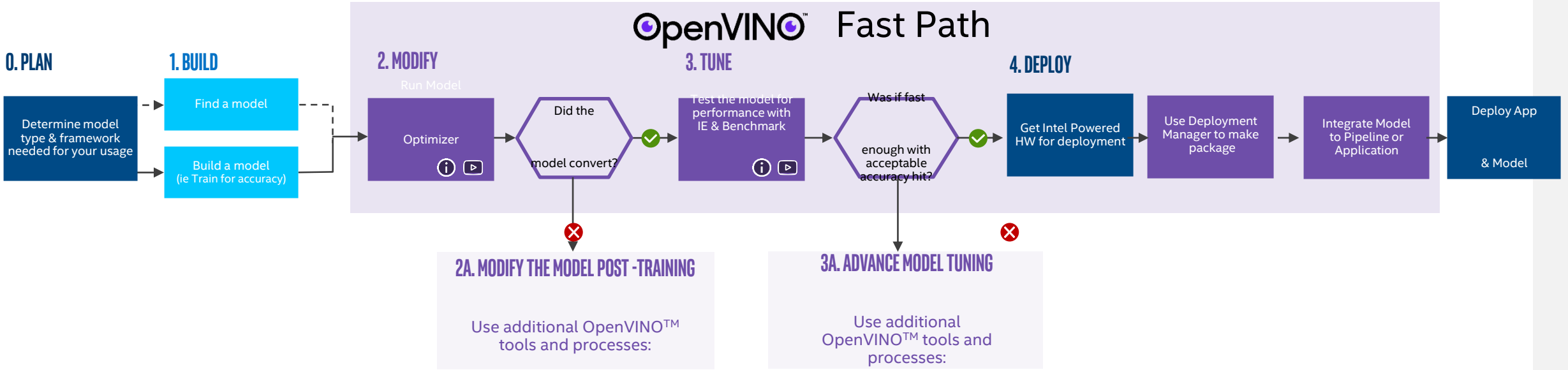

intel

A comprehensive workflow to optimize your DL model for the Intel Hardware that will be used for running inference



MO	Model Optimizer
OMZ	Open Model Zoo
DL	Deep Learning

A comprehensive workflow to optimize your DL model for the Intel Hardware that will be used for running inference



Introducing Add-ons

OpenVINO™ Model Server

- Enables customers to deploy Intel® Distribution of OpenVINO™ toolkit as a **containerized microservice**
- Reduced footprint: serve models with a <500MB container
- Higher throughput, lower latency: at parity or better than TFServing and Triton Inference Server*
- Similar performance to Inference Engine: users can expect similar performance to OpenVINO benchmarks when serving models*

https://github.com/openvinotoolkit/model_server