

Using the Intel® Distribution of the OpenVINO™ Toolkit for Deploying Accelerated Deep Learning Applications – Part2 [2021.2]

Jan 2021



Agenda

Part 1: OpenVINO Workshop (110mins):

- Demos on DevCloud
 - Post-Training Optimization Tool
 - DL Workbench
 - DL Streamer
-
- Part2: Q & A(10mins)

Notices and Disclaimers

- Performance varies by use, configuration and other factors. Learn more at www.Intel.com/PerformanceIndex.
- Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.
- Your costs and results may vary.
- Intel technologies may require enabled hardware, software or service activation.
- All product plans and roadmaps are subject to change without notice.
- Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.
- © Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.

Post-Training Optimization Tool

Jan. 2021



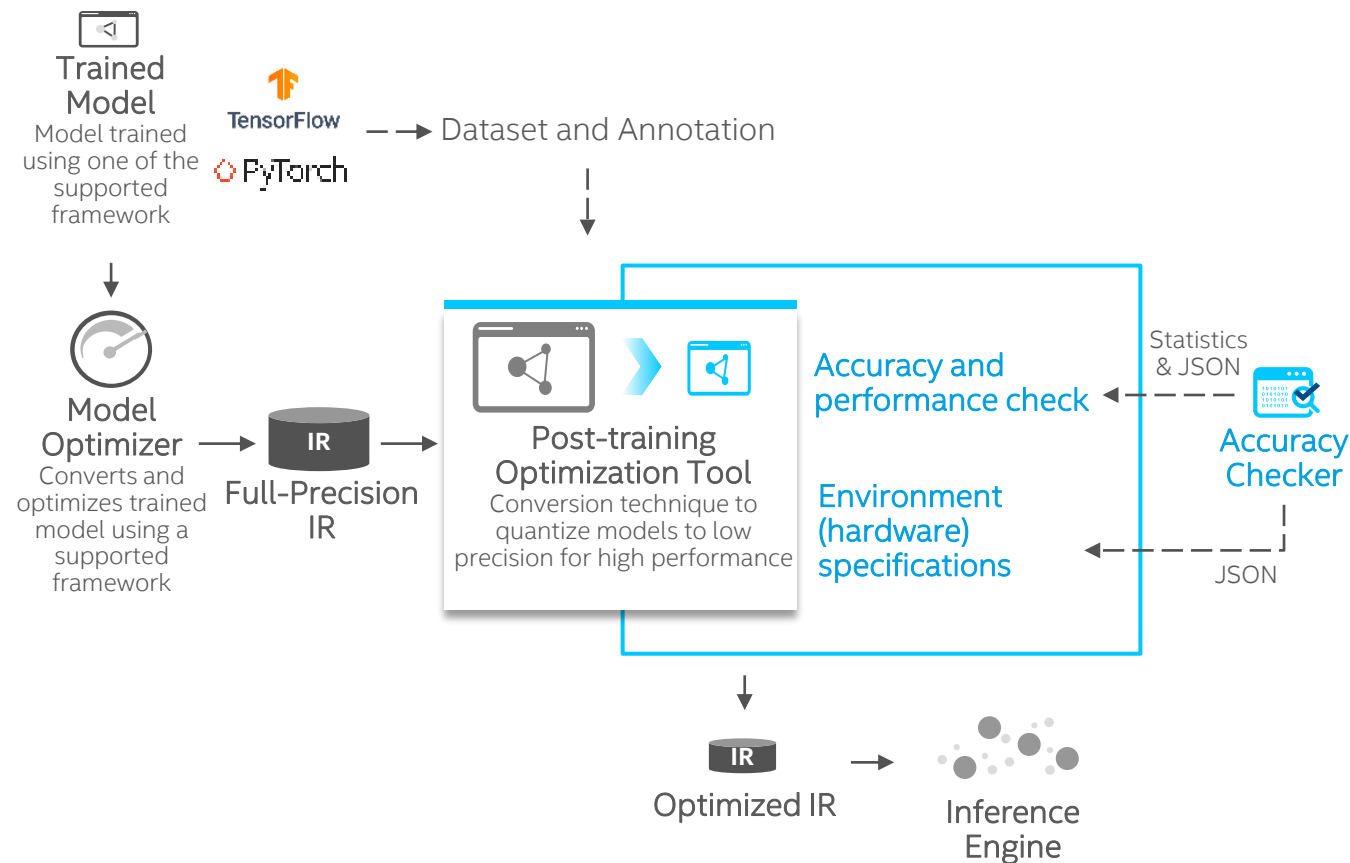
intel®

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.openvino toolkit.org/latest/docs_performance_int8_vs_fp32.html



Post-Training Optimization Tool – features

- Supports quantization of OpenVINO™ toolkit's IR models for various types of Intel® hardware
- *Learn more:* https://docs.openvino toolkit.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

Deep Learning Workbench

Jan 2021



intel®

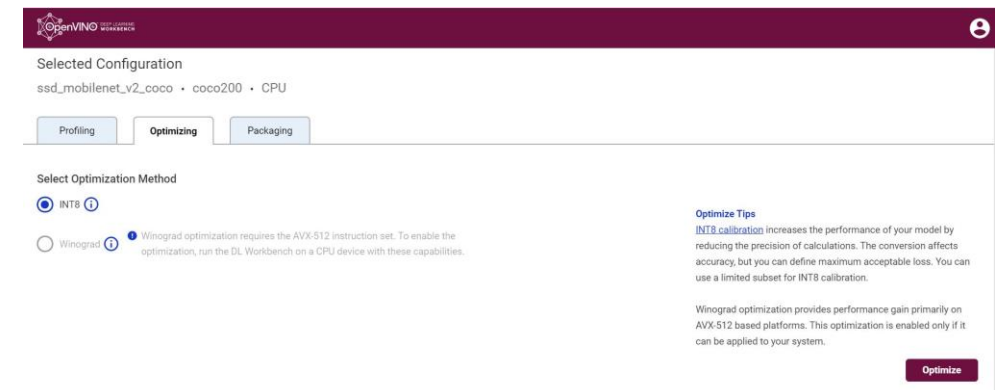
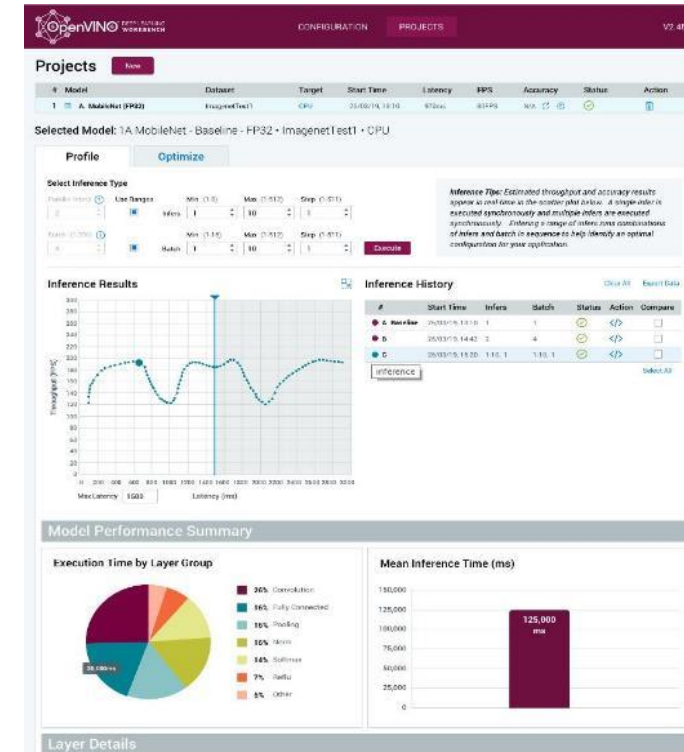
Deep Learning Workbench



- 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.

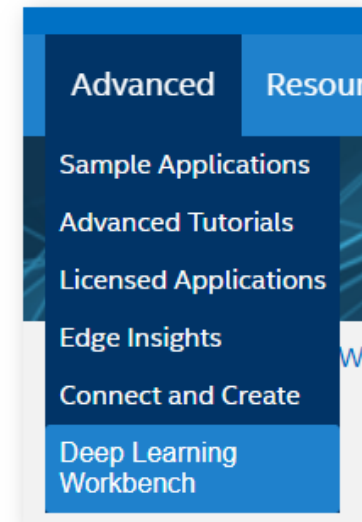
Development Guide ▶

https://docs.openvino toolkit.org/latest/_docs_Workbench_DG_Introduction.html

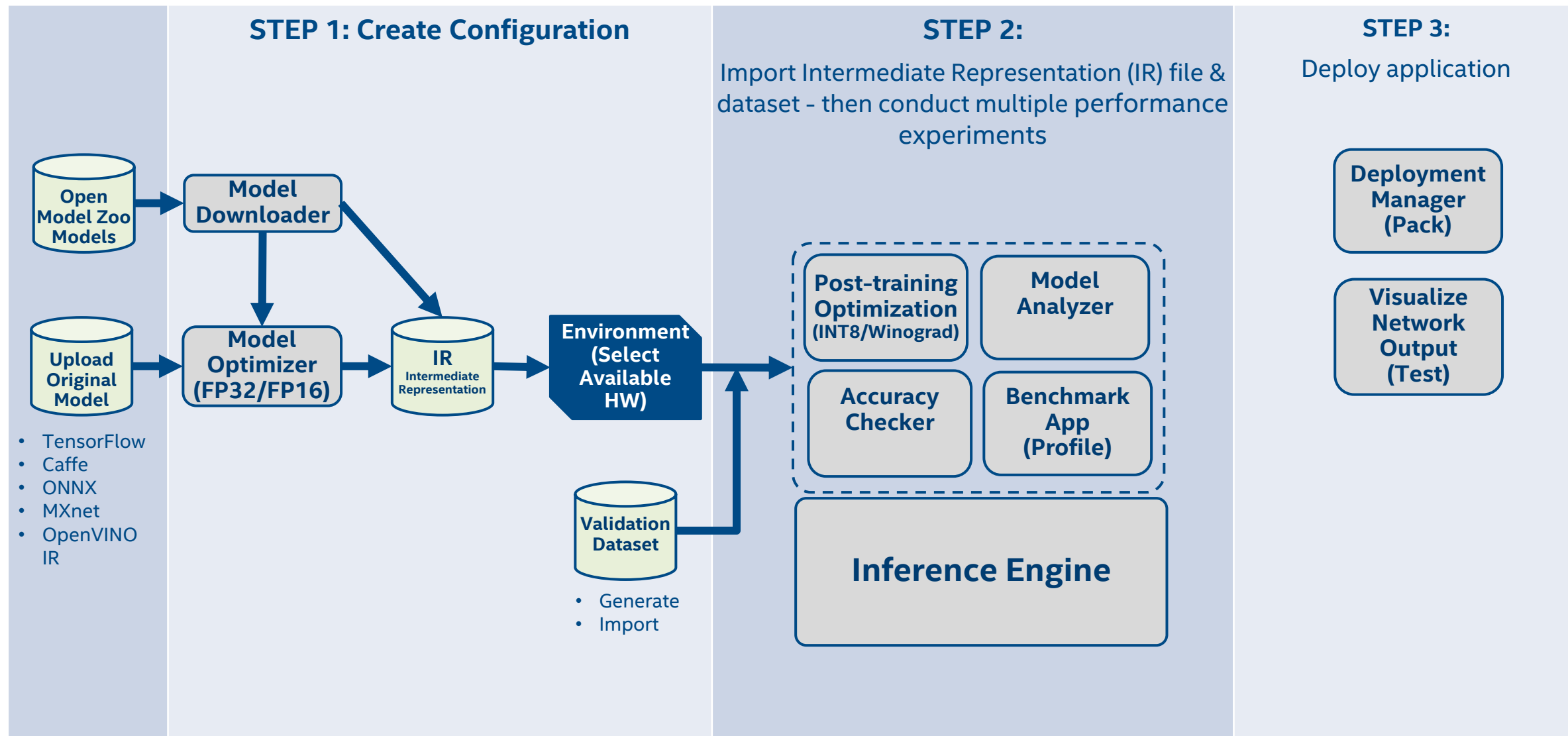


Installation Methods

- Run the DL Workbench on your local system
 - To profile your neural network on your own hardware or targets in your local network
 - Install from Docker Hub (Linux, Windows, macOS):
<https://hub.docker.com/r/openvino/workbench>
 - Install from Intel® Distribution of OpenVINO™ toolkit package: **build_docker.sh**
- Run the DL Workbench in the Intel® DevCloud for the Edge
 - To profile your neural network on various Intel® hardware configurations hosted in the cloud environment without any hardware setup at your end



Deep Learning Workbench Workflow



Work with Models and Sample Datasets

Active Configurations

Create

i No data available. Create a configuration by importing a model and a dataset to profile with.

Create Configuration

i Select a model, dataset, and environment. Then click Create to perform an inference.

Configuration Details

×

 Model: Selection required

×

 Target: Selection required

×

 Environment: Selection required

×

 Dataset: Selection required

Model ^

Import

Configuration Tips

Environment depends on the model you select. Different targets support different model precisions.

Model Name	Date ↓	Usage	Precisions	Size	Status	Actions
i To continue working, import a model.						

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

Select optimization method:

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

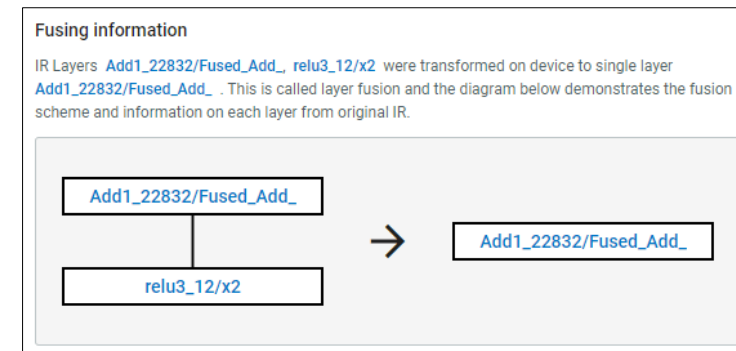
☒ **Optimization method: AccuracyAware**
Optimization method: AccuracyAware
Controllable drop of model accuracy
Increase of model speed

Max Accuracy Drop: %

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

Dataset File:

Dataset Name:



DEEP LEARNING WORKBENCH : FEATURES

Remote profiling support

Add Remote Target

Hostname: ⓘ

Port: ⓘ

Target Name: ⓘ

User: ⓘ

SSH Key: ⓘ

Use Proxy: ⓘ ☐

Support for Segmentation use cases

Configure Accuracy

instance_coco • coco200 • Local Workstation • CPU
Model Framework: OpenVINO IR

Usage: ⓘ Instance Segmentation

Default values are configured here for checking accuracy

Adapter Configuration:	Preprocessing Configuration:	Metric Configuration:	Annotation C
Input Info Layer: ⓘ <input type="text" value="image_info"/>	Resize Type: ⓘ <input type="text" value="Auto"/>	Metric: ⓘ <input type="text" value="COCO DRIO SEGM ..."/>	Separate Bac
Output Layers	<input type="checkbox"/> Use Normalization	Thresholds	
Masks: ⓘ <input type="text" value="masks"/>		Start: ⓘ <input type="text" value="0.5"/>	
Detection: ⓘ <input type="text" value="reshape_do_2d"/>		Step: ⓘ <input type="text" value="0.05"/>	
		End: ⓘ <input type="text" value="0.95"/>	

Deep Learning Streamer

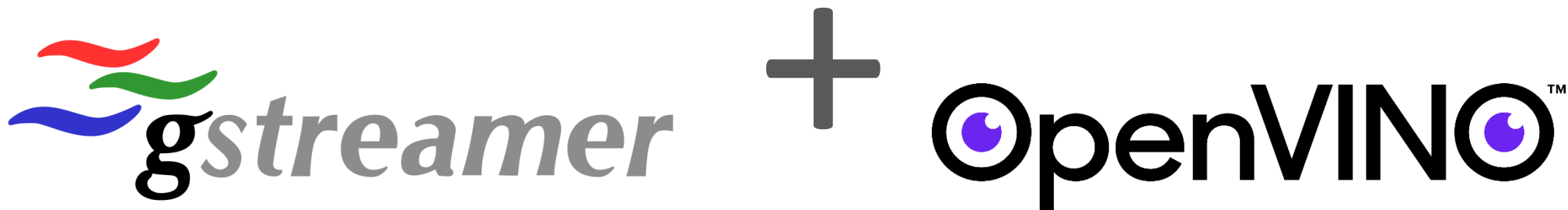
Jan. 2021



intel®

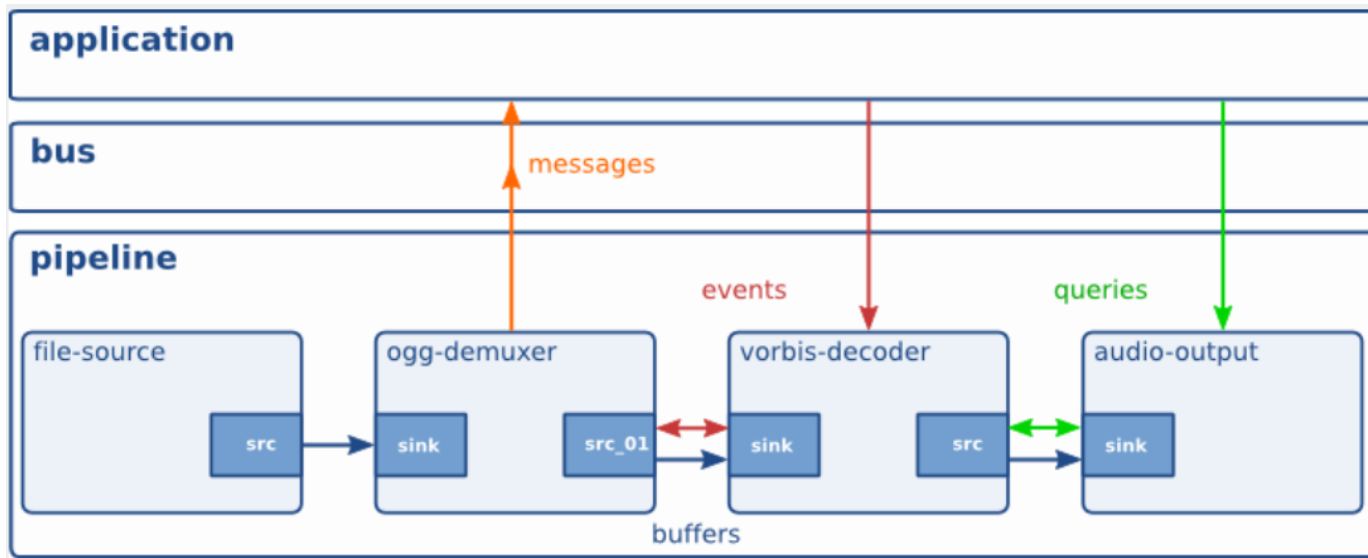
Introducing.. DL streamer

- 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

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

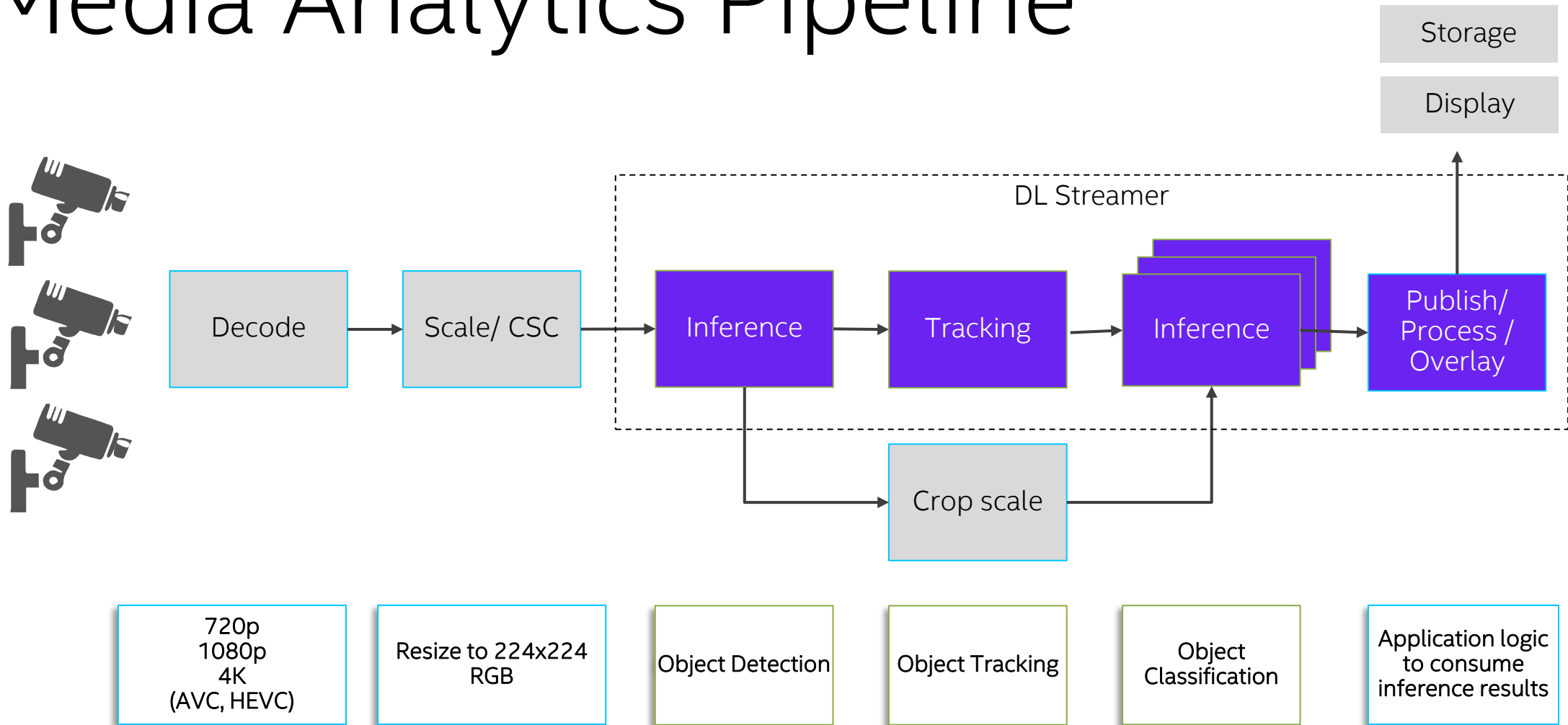


WANT TO KNOW MORE: CHECK OUT THE WEBINAR

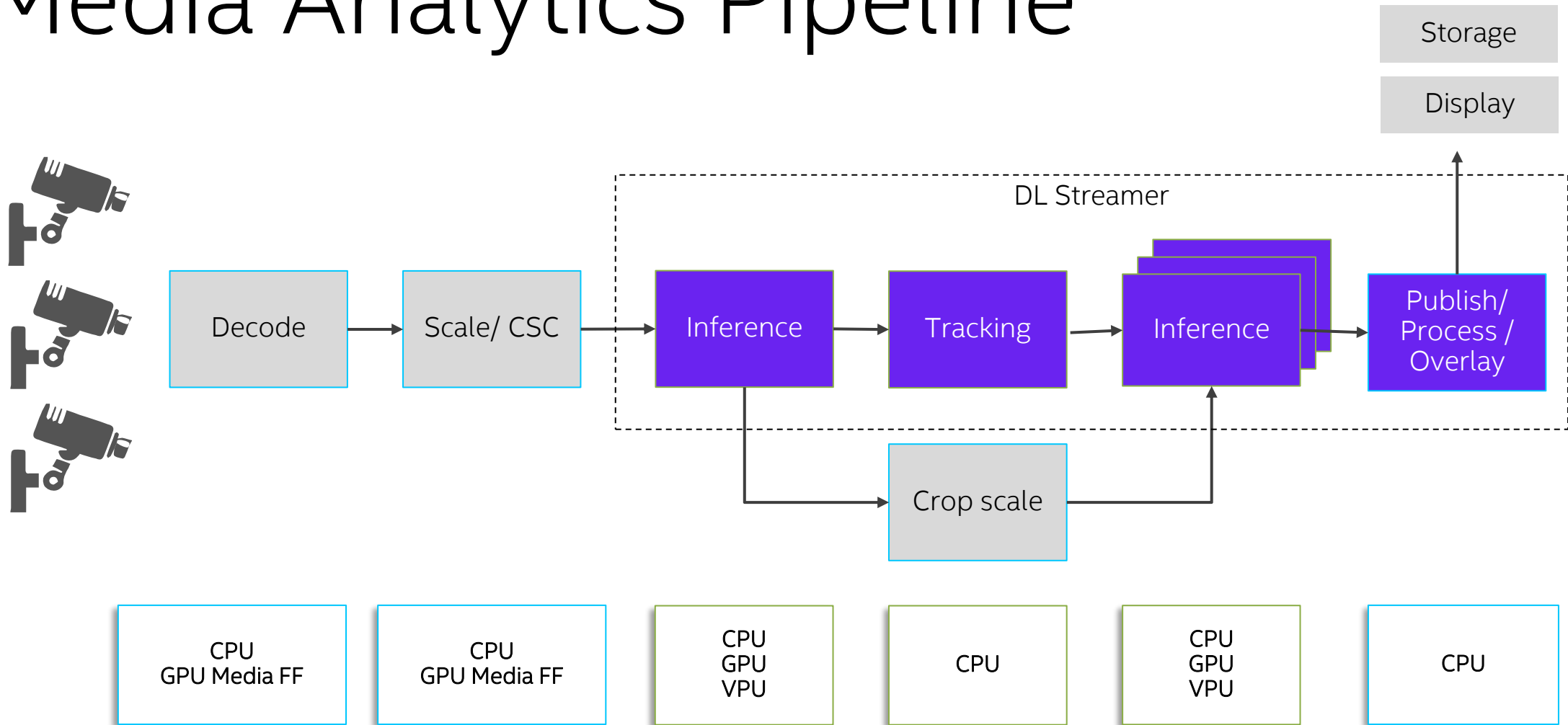
[HTTPS://SOFTWARE.SEEK.INTEL.COM/OPENVINO-WEBINAR-SERIES](https://software.seek.intel.com/openvino-webinar-series)

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

Media Analytics Pipeline

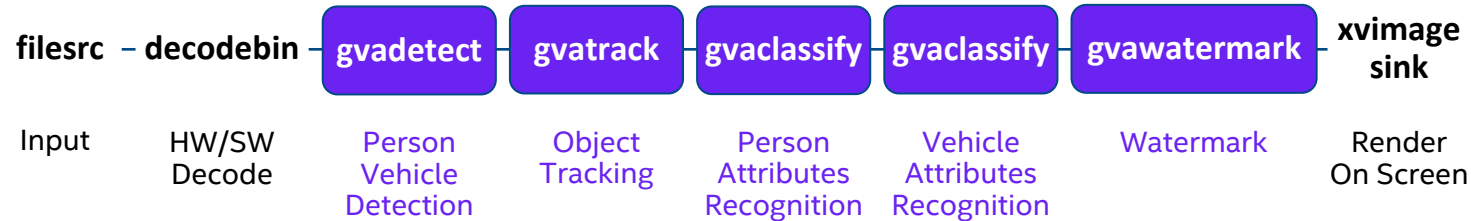


Media Analytics Pipeline



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 gvaudiodetect for audio event detection
 - Can be paired with alcnet public model for end-to-end audio analytics pipeline

DL Streamer Elements:

- [gvaudiodetect](#) 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

Resources to Get Started



Intel® Distribution of OpenVINO™ Toolkit:

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

Intel® Edge Software Hub

Download prevalidated software to learn, develop, and test your solutions for the edge.

Intel® Edge Software Hub:

<https://software.intel.com/content/www/us/en/develop/topics/iot/edge-solutions.html>

Intel® DevCloud
FOR THE EDGE

Intel® DevCloud for the Edge:

<https://devcloud.intel.com/edge/home>

To get access to the full video series, please complete the short form: <http://intel.ly/38B9ix6>

