Using the Intel® Distribution of the OpenVINO™ Toolkit for Deploying Accelerated Deep Learning Applications – Part2 [2021.3]

April 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® DevCloud for the Edge Demo

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

April. 2021

intel®

Post-Training Optimization Tool

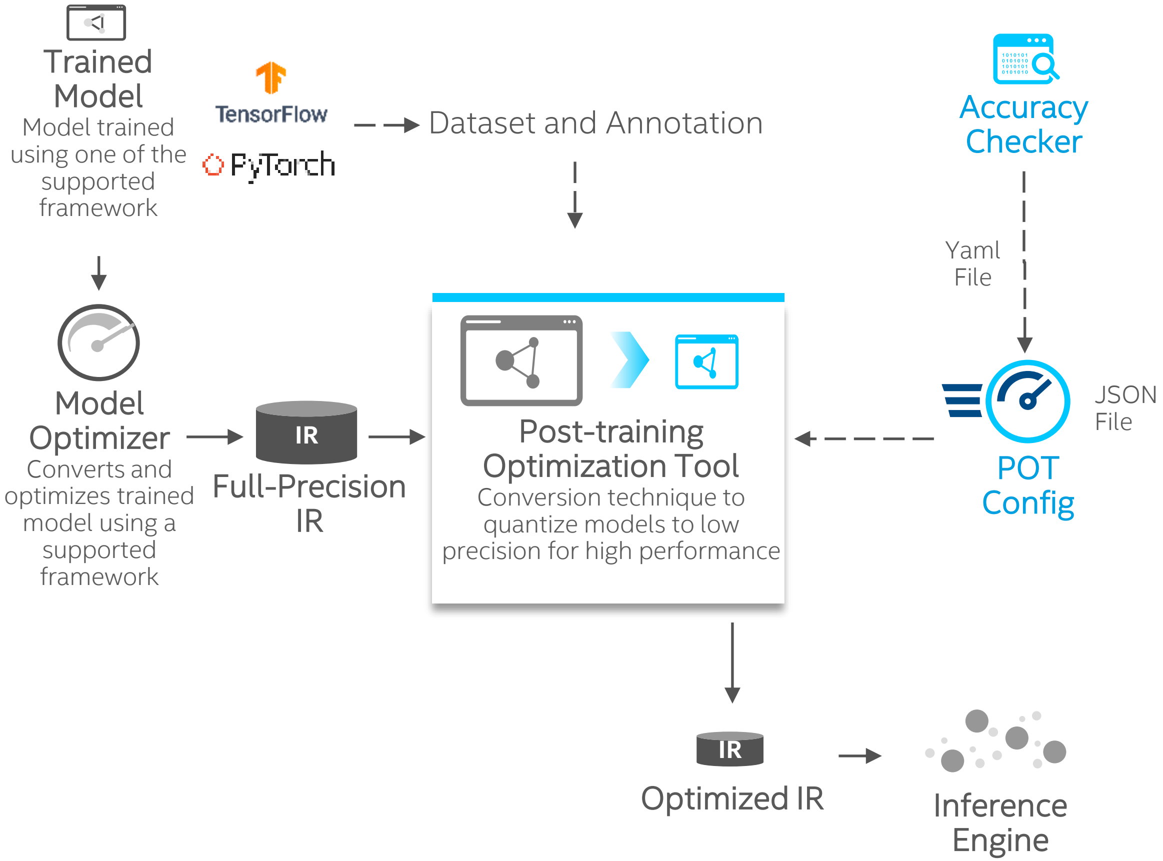
April. 2021

Post-Training Optimization Tool

https://docs.openvinotoolkit.org/latest/pot_README.html

- 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

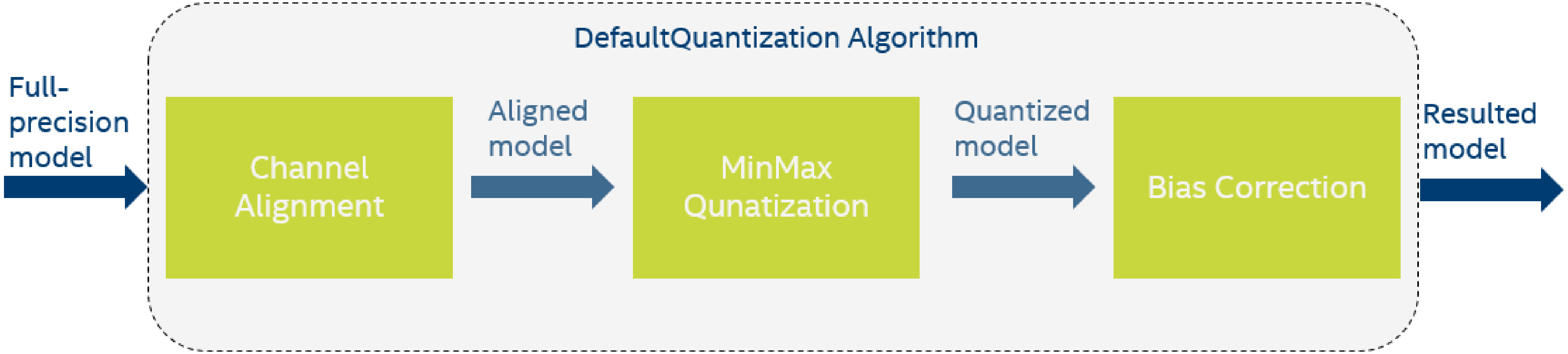


Trained Model
Model trained using one of the supported framework
TensorFlow
PyTorch
Dataset and Annotation
Accuracy Checker
Yaml File
Model Optimizer
Converts and optimizes trained model using a supported framework
IR
Full-Precision IR
Post-training Optimization Tool
Conversion technique to quantize models to low precision for high performance
JSON File
POT Config
IR
Optimized IR
Inference Engine

Post-Training Optimization Tool – DefaultQuantization

Designed to perform a fast and, in many cases, accurate 8-bits quantization of NNs

- Mandatory parameters (refer to the configuration file *default_quantization_template.json*)
  - "preset" - preset which controls the quantization mode (symmetric and asymmetric).
  - "stat_subset_size" - size of subset to calculate activations statistics used for quantization.
- Optional parameters (refer to the configuration file *default_quantization_spec.json*)
  - All other options can be considered as an advanced mode and require deep knowledge of the quantization process.


DefaultQuantization Algorithm

Full-precision model

Channel Alignment

Aligned model

MinMax Qunatization

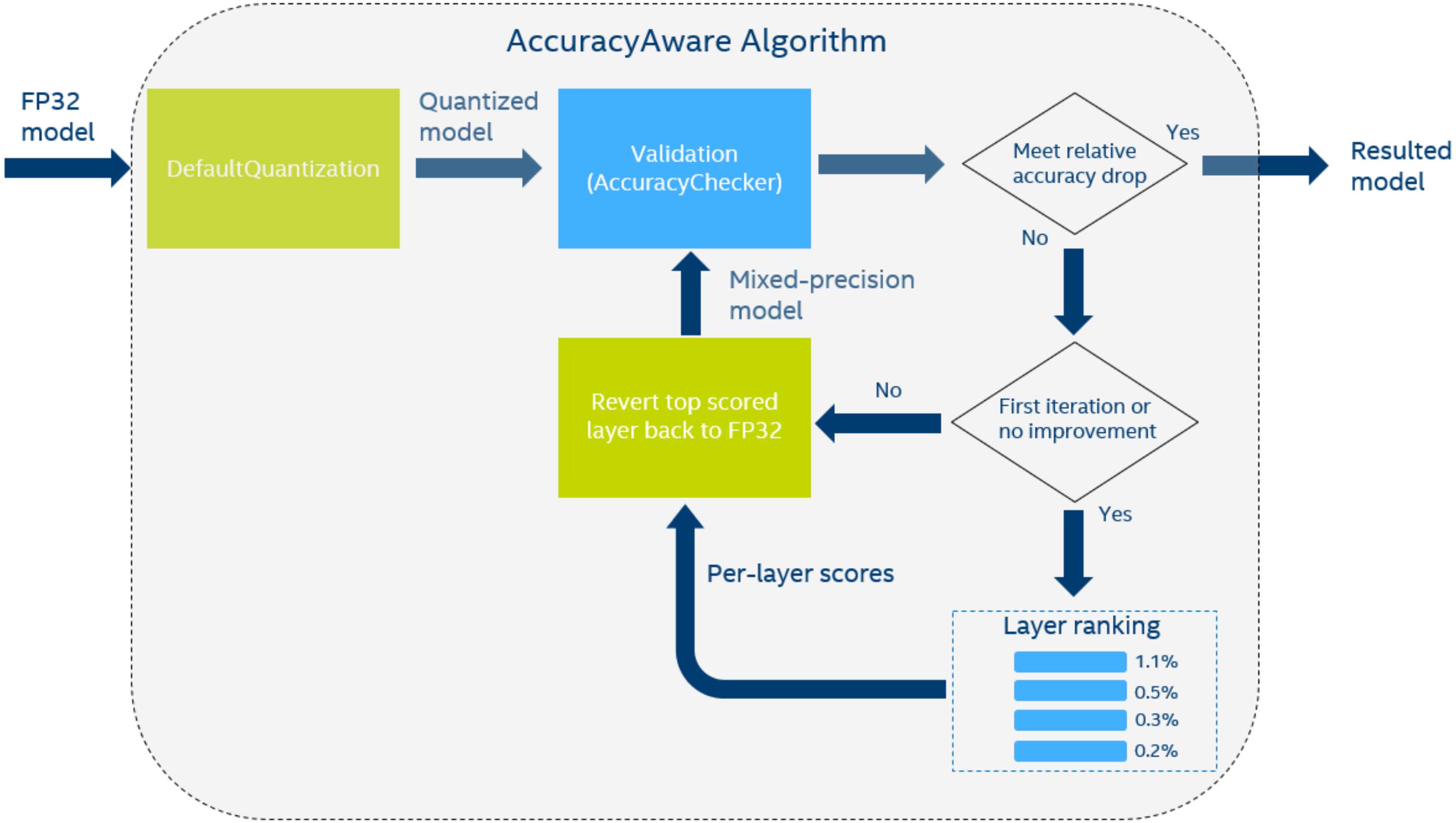
Quantized model

Bias Correction

Resulted model

Post-Training Optimization Tool – AccuracyAwareQuantization

Designed to perform accurate 8-bit quantization and allows the model to stay in the pre-defined range of accuracy drop, for example 1%

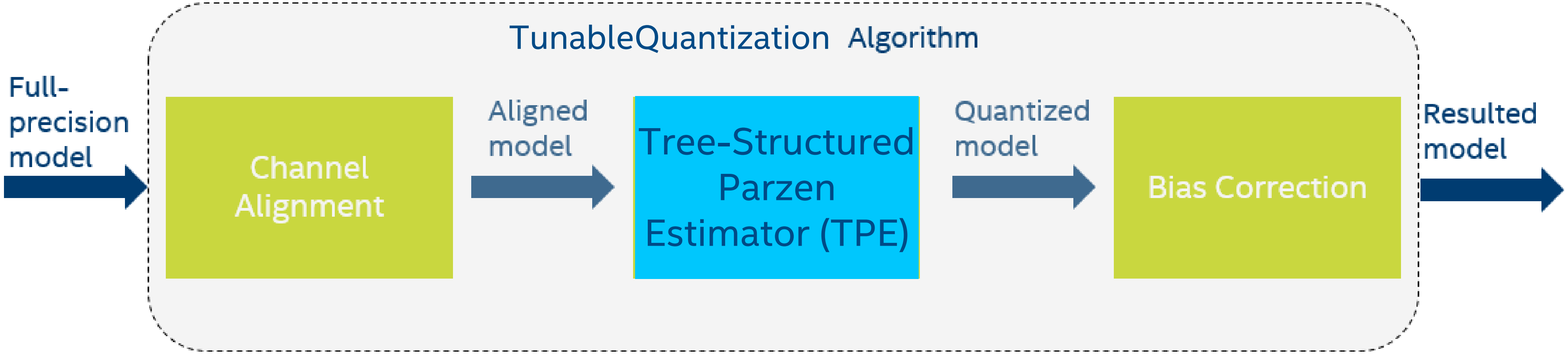


- To learn more about available parameters for AccuracyAwareQuantization, refer to the documentation and the *accuracy_aware_quantization_spec.json* file

Post-Training Optimization Tool - TunableQuantization

Layer-Wise Hyperparameters Tuning Using TPE

- TunableQuantization algorithm is a modified version (to support hyperparameters setting by **Tree-Structured Parzen Estimator (TPE)**) of the vanilla MinMaxQuantization quantization method that automatically inserts FakeQuantize operations into the model graph based on the specified target hardware and initializes them using statistics collected on the calibration dataset.
- Parameters for TunableQuantization, refer to the documentation and the *tpe_spec.json* file



Deep Learning Workbench

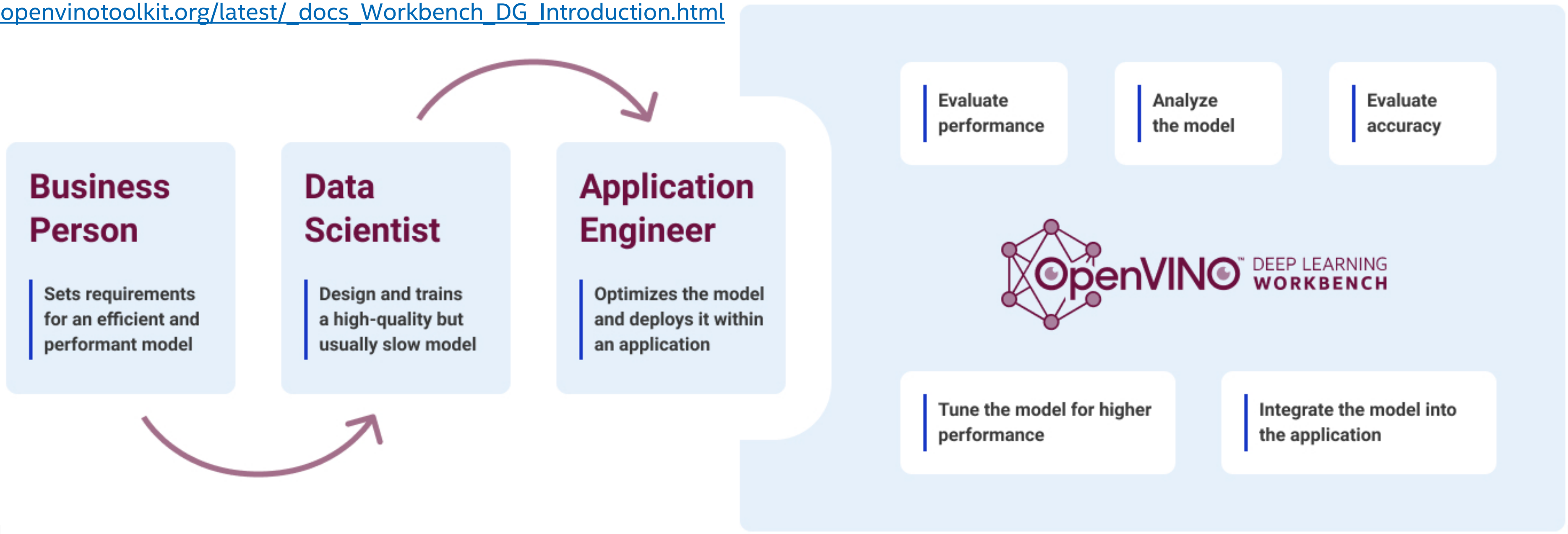
April 2021

Deep Learning Workbench

https://docs.openvinotoolkit.org/latest/workbench_docs_Workbench_DG_Introduction.html

- 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



Business Person

Sets requirements for an efficient and performant model

Data Scientist

Design and trains a high-quality but usually slow model

Application Engineer

Optimizes the model and deploys it within an application

Evaluate performance

Analyze the model

Evaluate accuracy

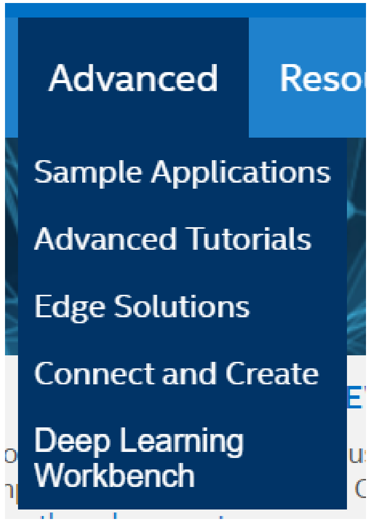
Tune the model for higher performance

Integrate the model into the application

Installation Methods

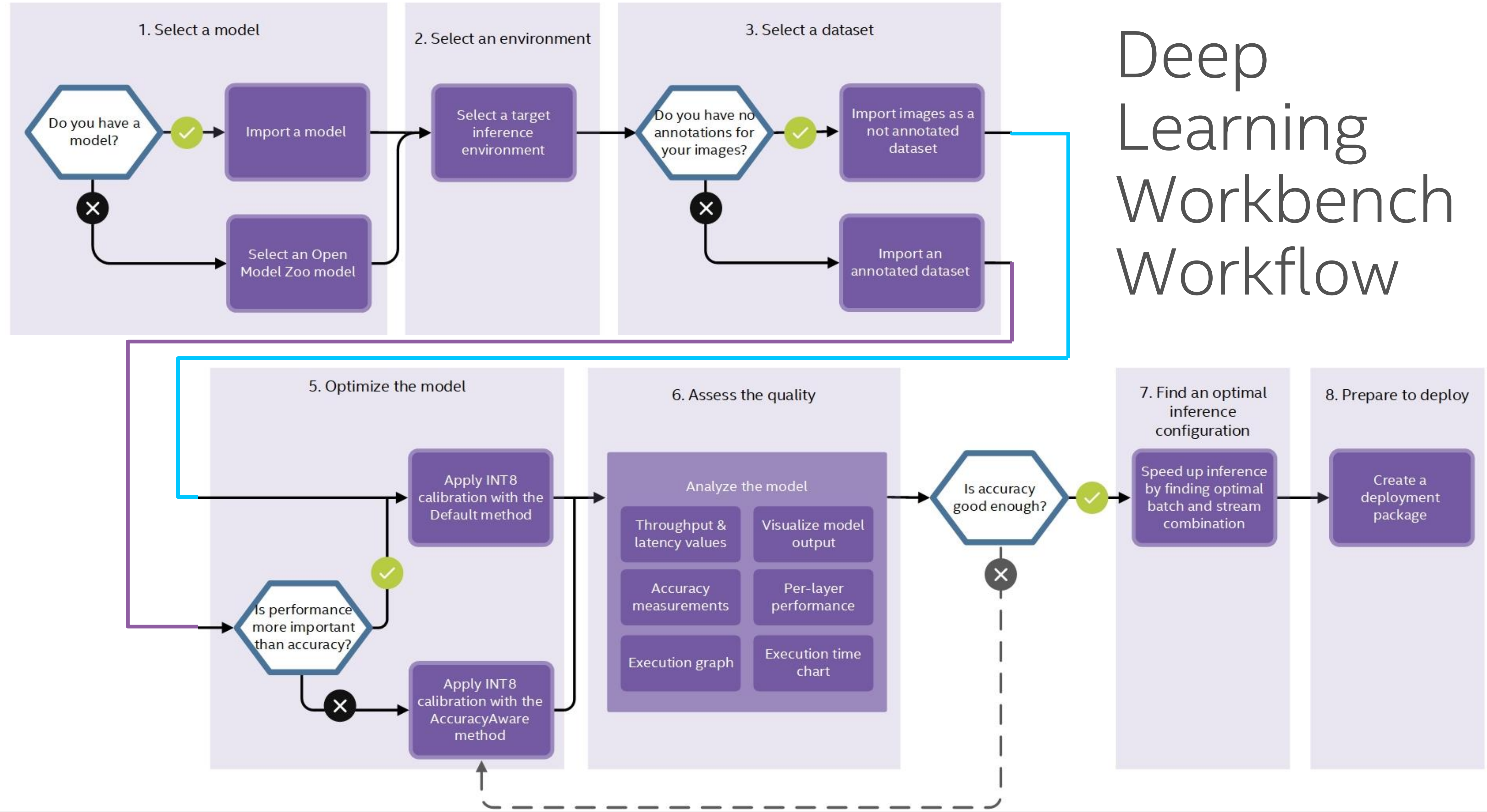
- Run the DL Workbench on your local system
  - To profile your neural network on your own hardware or targets in your local network
    - Install from Docker Hub (Linux, Windows, macOS):https://hub.docker.com/r/openvino/workbench
      - **start_workbench.sh**
      - **docker run** Command line
    - 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



Note: To get full features of DL Workbench, please run it on local system

Deep Learning Workbench Workflow


1. Select a model
Do you have a model?
Import a model
Select an Open Model Zoo model
2. Select an environment
Select a target inference environment
3. Select a dataset
Do you have no annotations for your images?
Import images as a not annotated dataset
Import an annotated dataset
5. Optimize the model
Is performance more important than accuracy?
Apply INT8 calibration with the Default method
Apply INT8 calibration with the AccuracyAware method
6. Assess the quality
Analyze the model
Throughput & latency values
Visualize model output
Accuracy measurements
Per-layer performance
Execution graph
Execution time chart
Is accuracy good enough?
7. Find an optimal inference configuration
Speed up inference by finding optimal batch and stream combination
8. Prepare to deploy
Create a deployment package

DL Workbench Demo

Deep Learning Streamer

April 2021

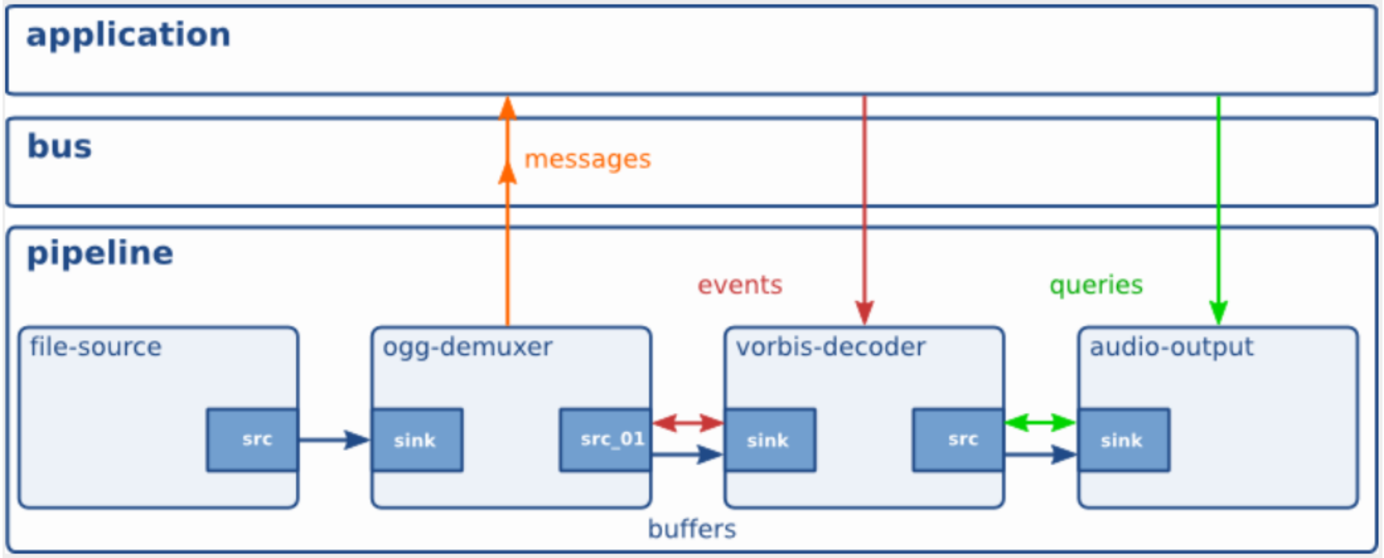
Introducing.. Dl streamer

- 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

gstreamer + OpenVINO™

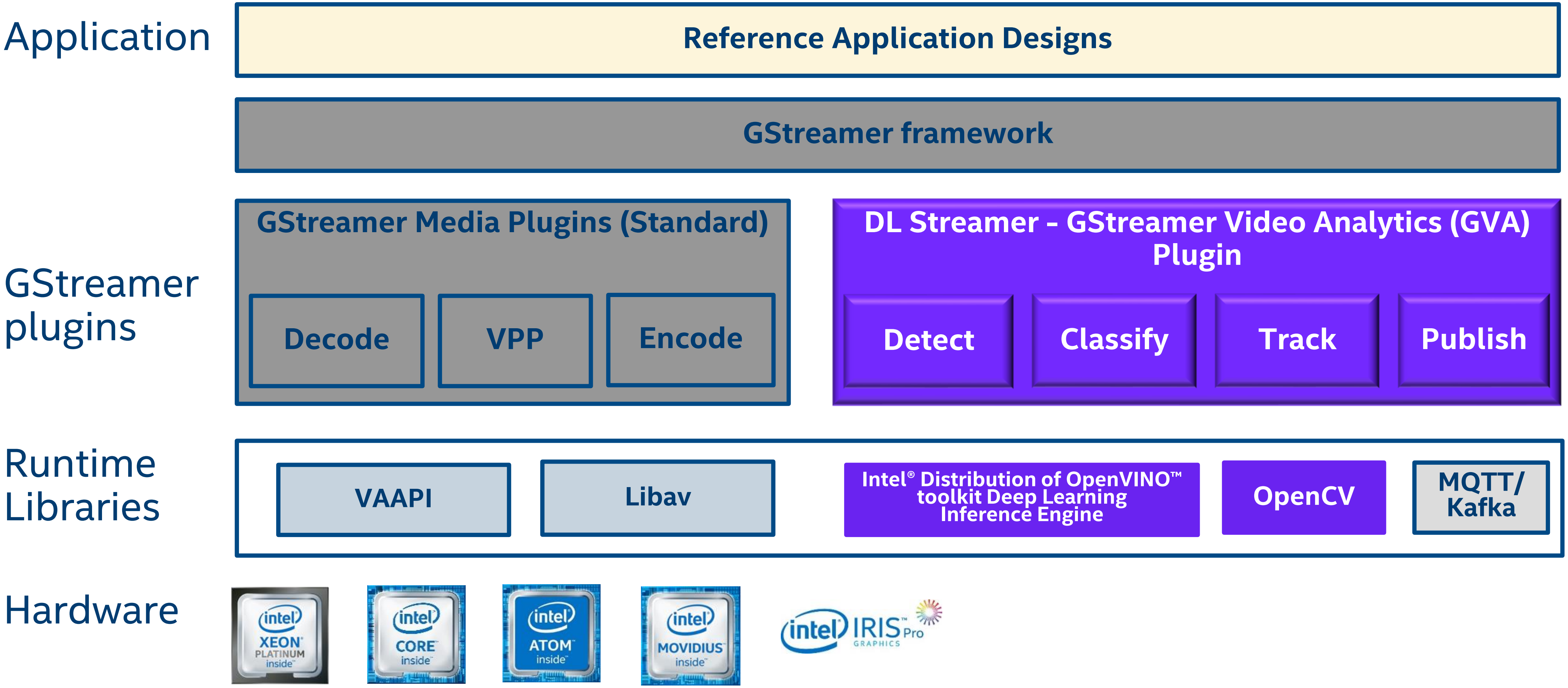
What is GStreamer?

- 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

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

Media Processing Pipeline

Video Pipeline – decode, convert, render

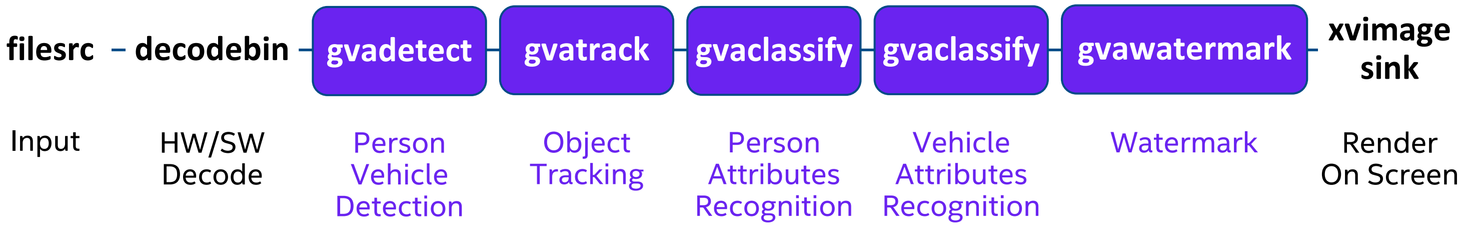
filesrc — decodebin — videoconvert — xvimagesink

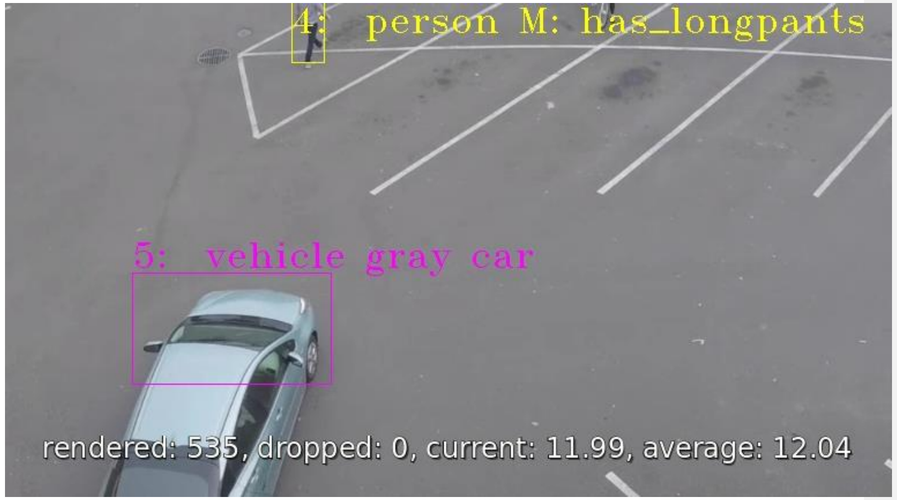
filesrc	decodebin	videoconvert	xvimagesink
input	HW/SW decode	convert	render on screen

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

Using the DL Streamer

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

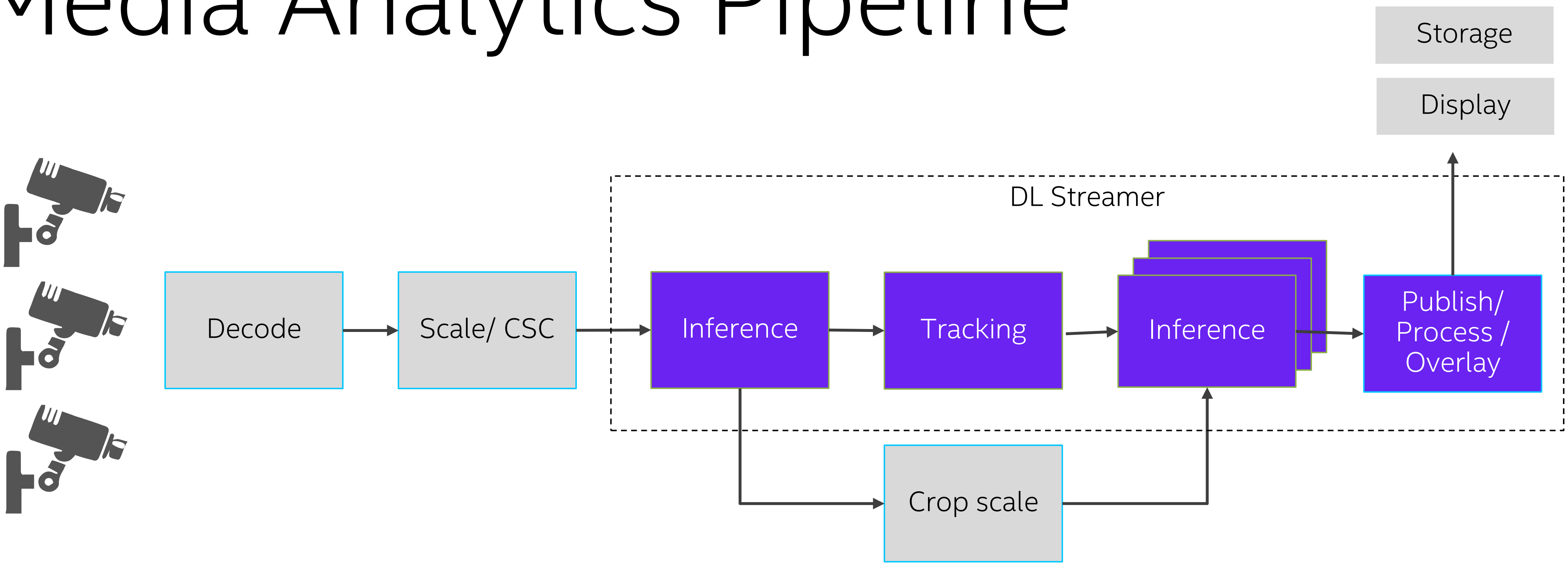




rendered: 535, dropped: 0, current: 11.99, average: 12.04

```
 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
```

Media Analytics Pipeline



720p
1080p
4K
(AVC, HEVC)

Resize to 224x224
RGB

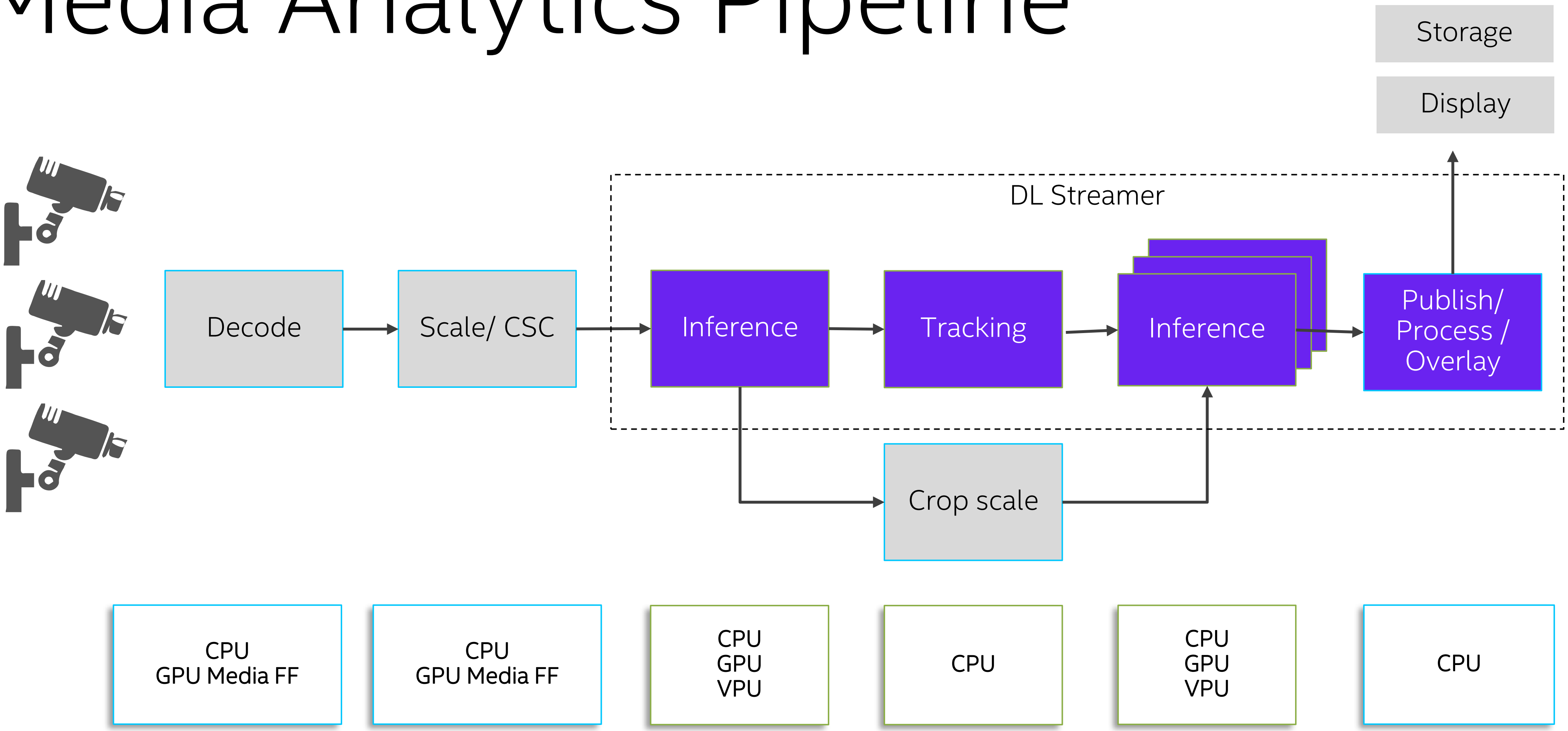
Object Detection

Object Tracking

Object Classification

Application logic to consume inference results

Media Analytics Pipeline



Storage

Display

DL Streamer

Decode

Scale/ CSC

Inference

Tracking

Inference

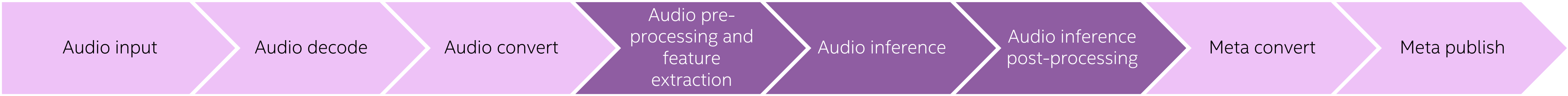
Publish/ Process / Overlay

Crop scale

CPU GPU Media FF	CPU GPU Media FF	CPU GPU VPU	CPU	CPU GPU VPU	CPU

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

Resources to Get Started

OpenVINO™

Intel® Distribution of OpenVINO™ Toolkit:

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