GSTREAMER VIDEO ANALYTICS: OPTIMIZING INFERENCE ACROSS HW TARGETS

Intel Visual Compute Middleware and Tools

Neelay Shah

Neena Maldikar

Mikhail Nikolskii

Ilya Belyakov

Legal Disclaimer

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document. 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.

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.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting www.intel.com/design/literature.htm.

Intel, the Intel logo, {List the Intel trademarks in your document} are trademarks of Intel Corporation in the U.S. and/or other countries.

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

Agenda

- **Video Analytics Usages**
- Supporting Inference Across Different Hardware
- GStreamer Video Analytics
- Demo
- Video Analytics Serving
- Resources

Growing Demand for Video Intelligence

$2 Billion (2019) -> $11 Billion (2026)

Retail analytics

Industrial inspection

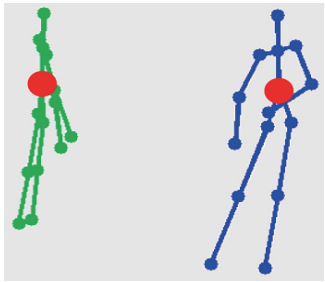
Content filtering

Parking management

Encode Quality Control

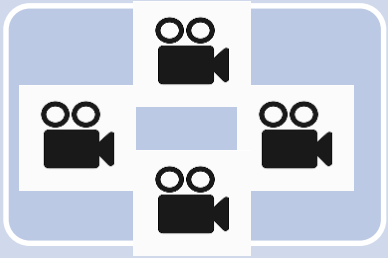
Super Resolution

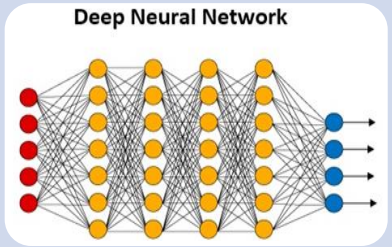
Autonomous driving

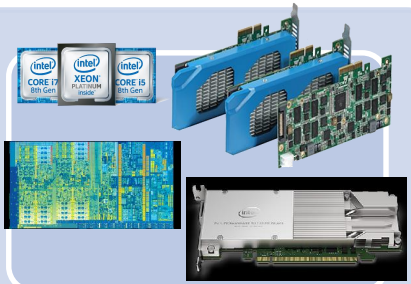


Action recognition

Growing Demand for Video Intelligence







Inexpensive Widely Deployed Cameras

Increasing Edge 2 Cloud Networking capacity with 5G

Deep Learning Networks with Increasing Accuracy

Increasing Compute Capacity and Options

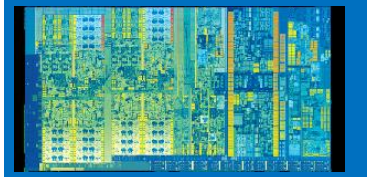
Intel Video Analytics Hardware

CPU

Intel® CPU
Client and server

Scalar

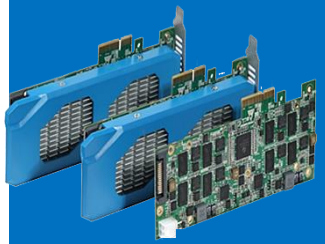
GPU



Intel® GPU
Integrated and discrete

Vector

VPU



Intel® Vision Accelerator Design with Intel® Movidius™ Vision Processing Units (VPU)

Matrix

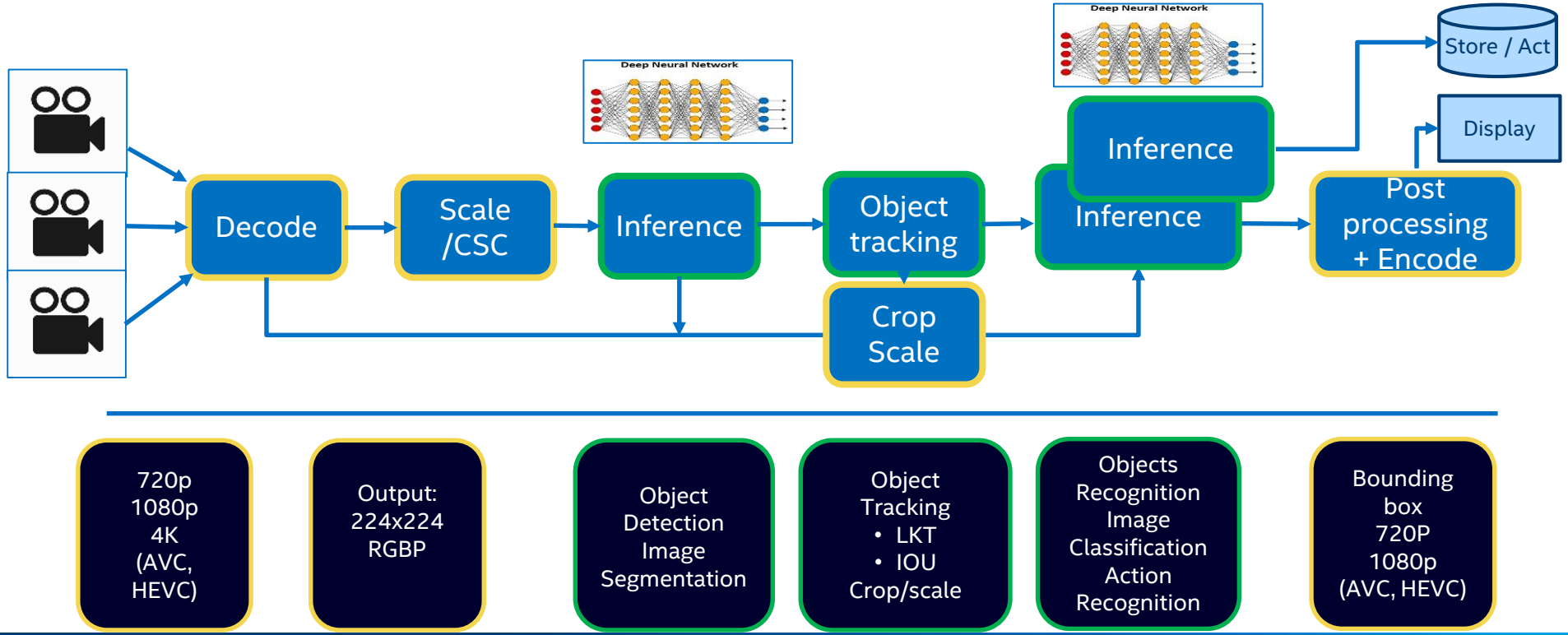
FPGA

Intel® Vision Accelerator Design with an Intel® Arria 10 FPGA (preview)

Spatial

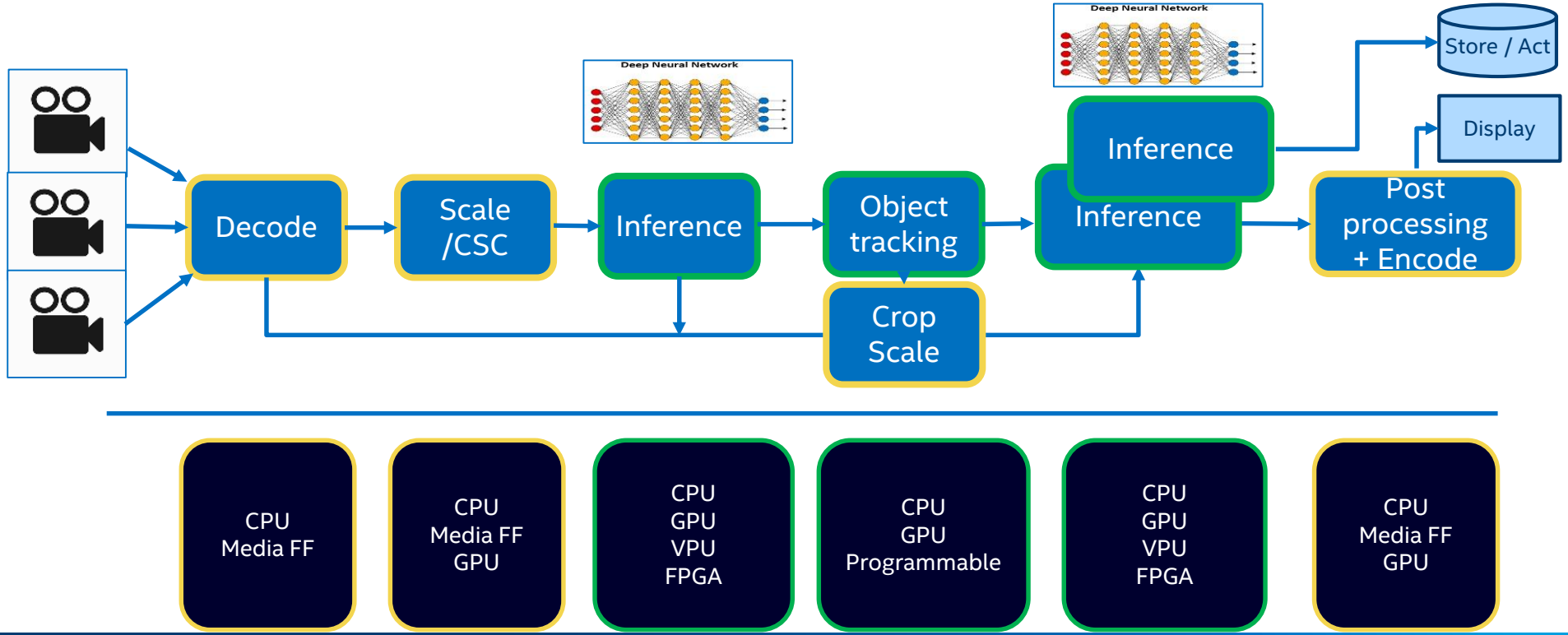
Typical Video Analytics Pipeline

Media + Compute



Typical Video Analytics Pipeline

Media + Compute

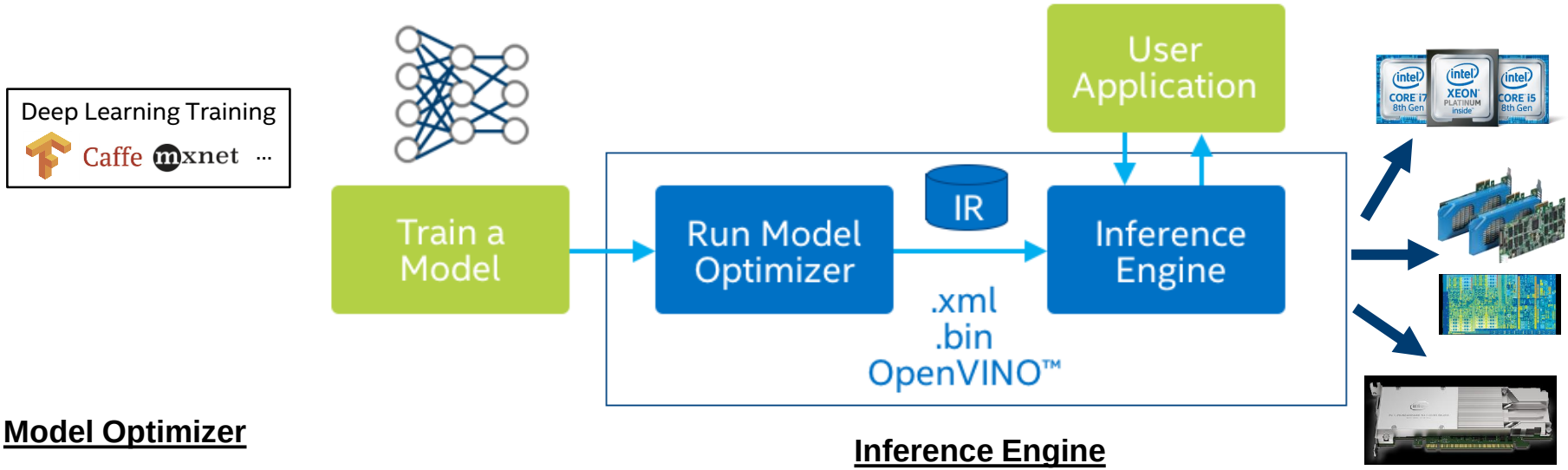


Agenda

- Video Analytics Usages
- **Supporting Inference Across Different Hardware**
- GStreamer Video Analytics
- Demo
- Video Analytics Serving
- Resources

OpenVINO

Optimized Inference Across Different Hardware



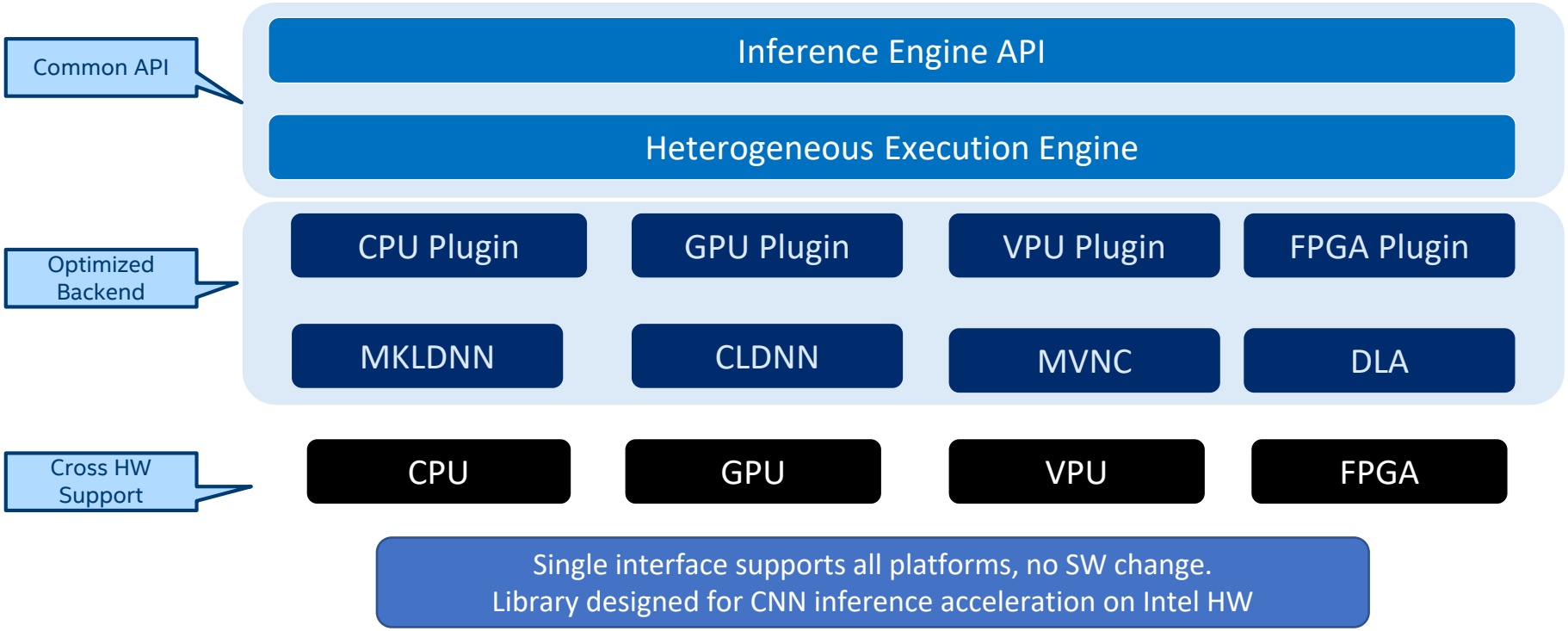
Model Optimizer

This Python*-based command line tool imports trained models from popular deep learning frameworks such as Caffe*, TensorFlow*, and Apache MXNet*, and Open Neural Network Exchange (ONNX).

Inference Engine

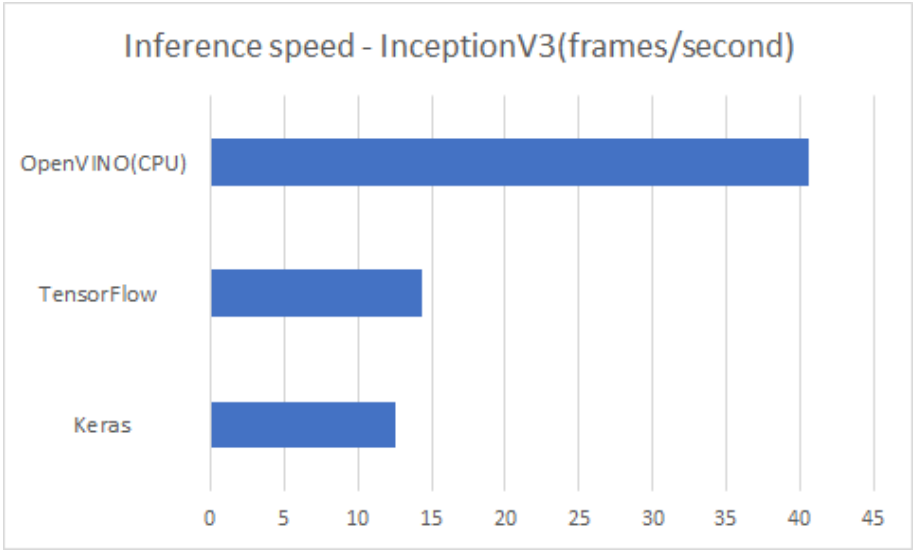
This execution engine uses a common API to deliver inference solutions on the platform of your choice: CPU, GPU, VPU, or FPGA.

OpenVINO High Level Design



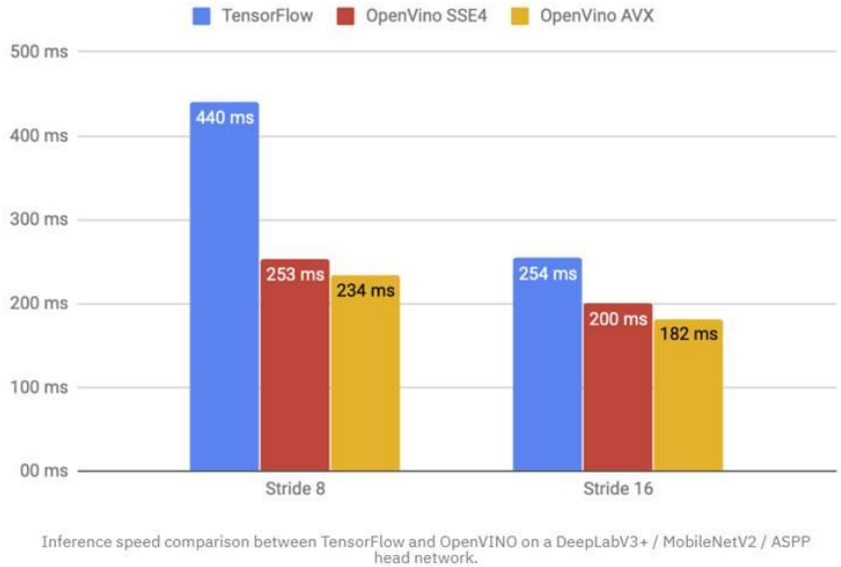
OpenVINO Performance Gains

Increased Throughput



https://www.dlology.com/blog/how-to-run-keras-model-inference-x3-times-faster-with-cpu-and-intel-openvino-1/

Reduced Latency



Inference speed comparison between TensorFlow and OpenVINO on a DeepLabV3+ / MobileNetV2 / ASPP head network.

https://hackernoon.com/optimizing-neural-networks-for-production-with-intels-openvino-a7ee3a6883d

Agenda

- Video Analytics Usages
- Supporting Inference Across Different Hardware
- **GStreamer Video Analytics**
- Demo
- Video Analytics Serving
- Resources

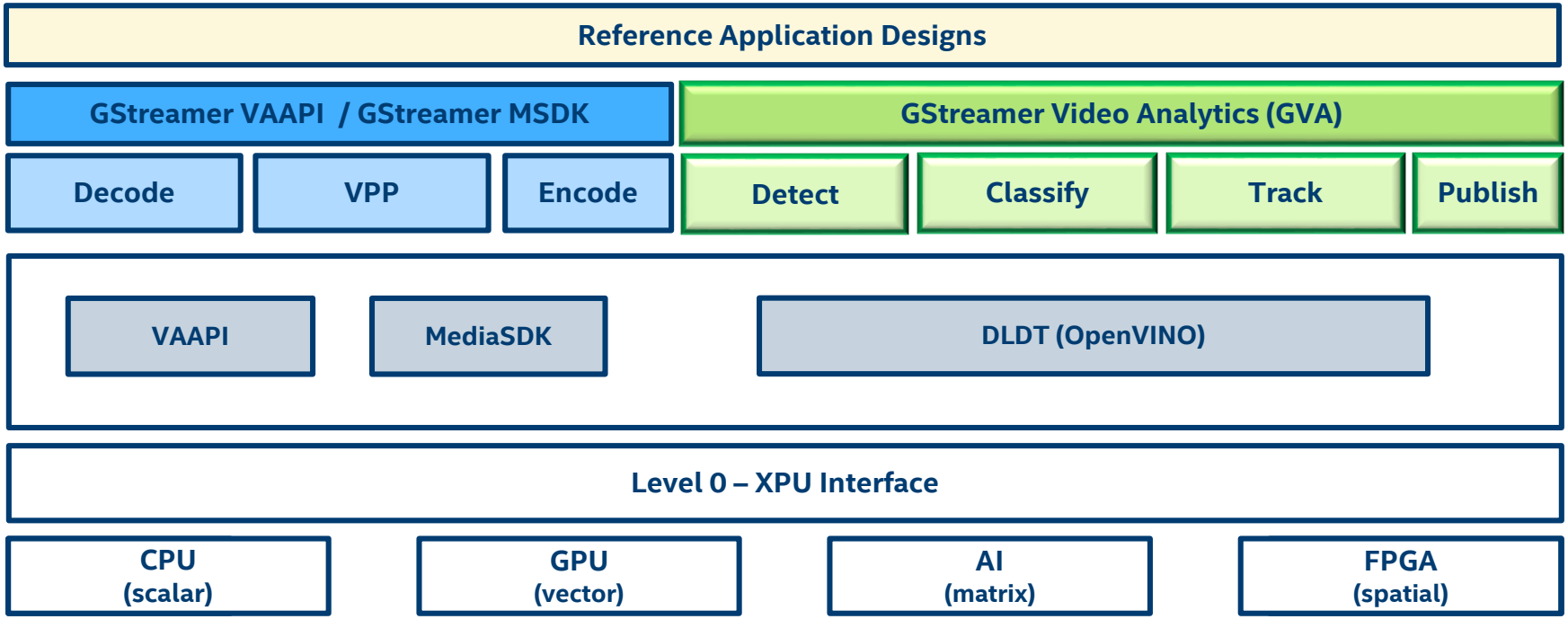
GStreamer Video Analytics

OpenVINO

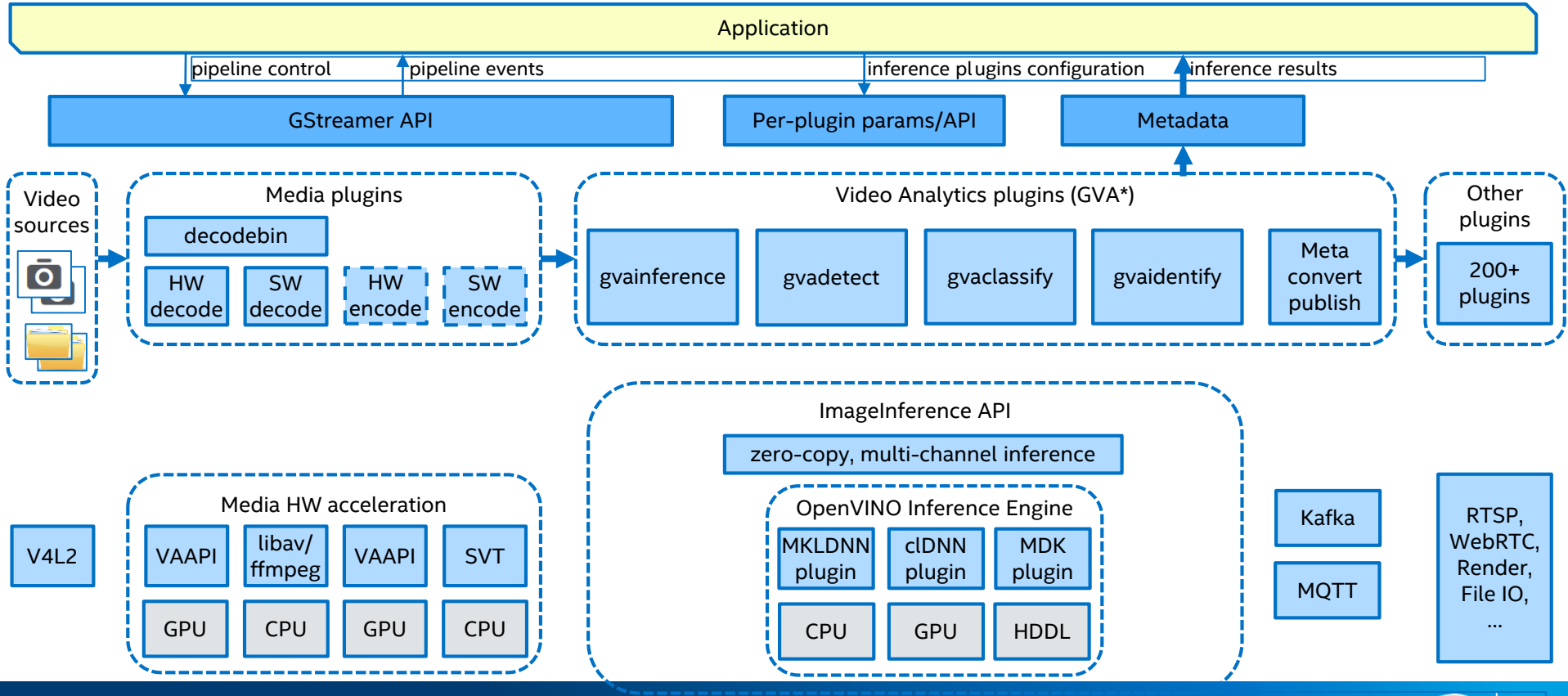
Optimized Media Framework

Optimized Inference Engine

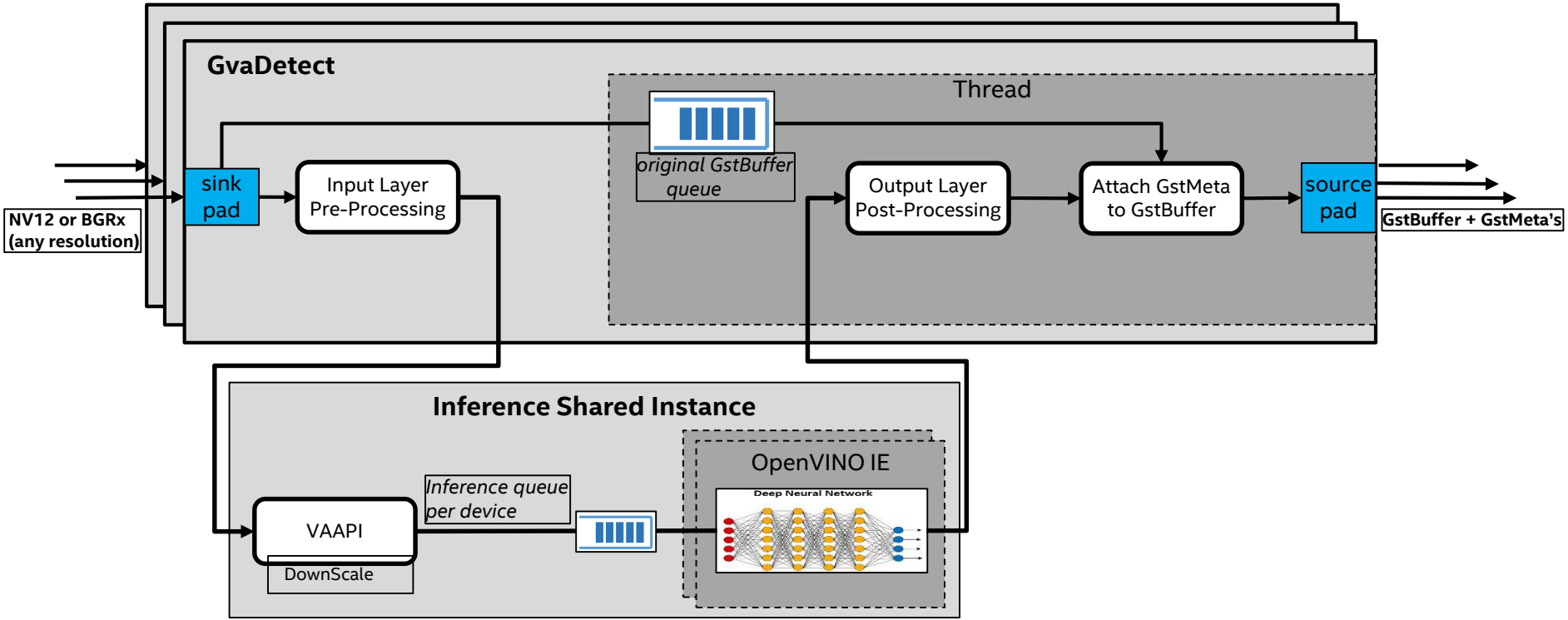
GStreamer Video Analytics



GStreamer Plugins Architecture

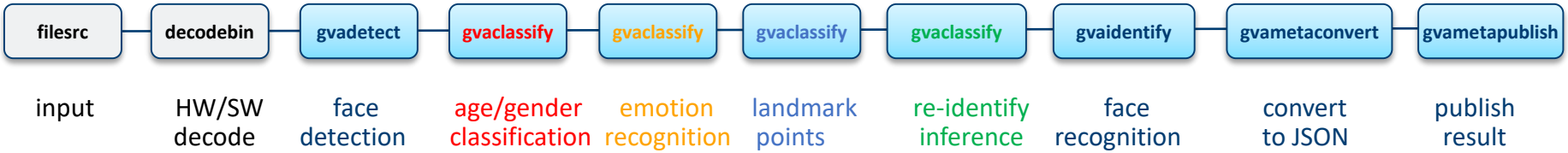


GVA Inference Plugins Architecture



GStreamer Pipeline Example

Video Analytics pipeline – face detection plus age, gender, person recognition




```
filesrc location=${FILE} !
decodebin ! \
gvadetect model=face-detection-adas-0001.xml model-proc=face-detection-adas-0001.json ! queue ! \
gvaclassify device=CPU model=age-gender-recognition.xml model-proc=age-gender-recognition.json ! queue ! \
gvaclassify device=GPU model=emotions-recognition.xml model-proc=emotions-recognition.json ! queue ! \
gvaclassify device=MULTI:CPU,GPU model=landmarks-regression.xml model-proc=landmarks-regression.json ! queue ! \
gvaclassify model=face-reidentification.xml model-proc=face-reidentification.json ! queue ! \
gvaidentify gallery=face_gallery.json ! queue ! \
gvametaconvert method=json ! \
gvametapublish ip=128.45.0.34
```

Running on devices… **DL model**

GStreamer Video Analytics Plugins List

GST element	Description	INPUT	OUTPUT	Properties
gvainference	Generic inference	GstBuffer	INPUT + GvaTensorMeta	Properties
gvadetect	Object detection	GstBuffer	INPUT + GstVideoRegionOfInterestMeta	Properties
gvaclassify	Object classification	GstBuffer + GstVideoRegionOfInterestMeta	INPUT + GstVideoRegionOfInterestMeta	Properties
gvaidentify	Object identification/ recognition	GstBuffer + GstVideoRegionOfInterestMeta	INPUT + GstVideoRegionOfInterestMeta	Properties
gvametaconvert	Metadata conversion	GstBuffer + GvaTensorMeta/ GstVideoRegionOfInterestMeta	INPUT + GvaJSONMeta	Properties
gvawatermark	Overlay	GstBuffer + GvaDetectionMeta + { GvaTensorMeta }	INPUT (with modified image)	Properties
gvametapublish	Message bus (Kafka, MQTT)	GstBuffer + GvaJSONMeta	-	Properties

Metadata

Detection:

GstVideoRegionOfInterestMeta

roi_type – Detection Label (Face, Bottle, …)

x – x component of upper-left corner

y – component of upper-left corner

w – bounding box width

h – bounding box height

params – List of Classification Results

Classification:

Parameter of Region of Interest

model_name – name of model

layer_name – output layer name

label – Classification Label (Age, Gender)

data – Tensor data

Metaconvert:

GvaJSONMeta

message – JSON Object Representing list of regions and classifications per frame

```
{
  "timestamp":0,
  "objects": [
   {
     "h": 85, w": 76, "x": 262, y": 601,
     "roi_type": "face",
     "age": {"label": "24", "model": {
       "name": "age_gender"
      }
     },
     "emotion": {
       "label": "anger",
       "model": {
         "name": "0003_EmoNet_ResNet10"
       }
     },]
}
```

Common Properties

model – path to model (.xml) in IR format

model-proc – path to JSON file with description of input/ output layers for pre/post processing

device – target device for inference (CPU, GPU, CPU-GPU, HDDL, multi device)

inference-id – unique id to enable inference engine instance sharing between OpenVINO plugin instances

batch-size - number of frames to process in one request

nireq - number of inference requests to run in parallel

every-nth-frame – run inference only on each N-th frame

Agenda

- Video Analytics Usages
- Supporting Inference Across Different Hardware
- GStreamer Video Analytics
- **Demo**
- Video Analytics Serving
- Resources

Face Detection And Classification

Models:

Detection:	**face-detection-adas-0001.xml**
Classification:	**age-gender-recognition-retail-0013.xml**
Classification:	**emotions-recognition-retail-0003.xml**
Classification:	**landmarks-regression-retail-0009.xml**

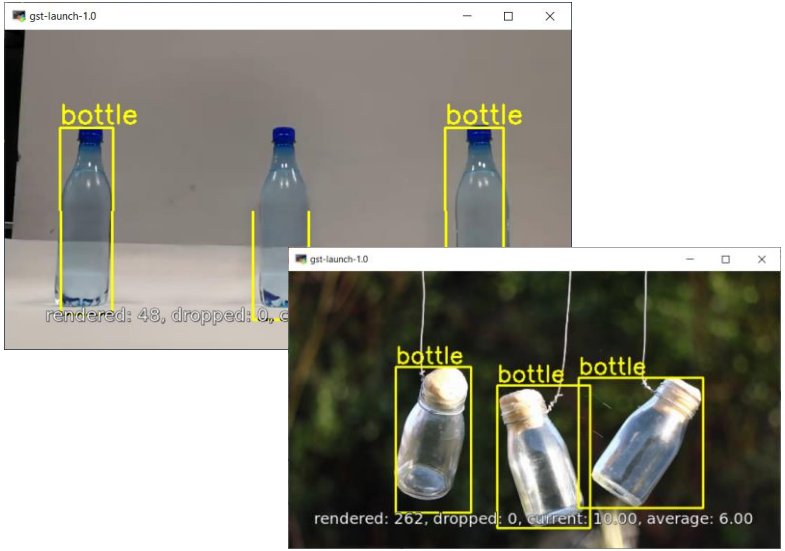
Pipeline:

```
gst-launch-1.0 --gst-plugin-path ${GST_PLUGIN_PATH} \
filesrc location =$INPUT ! ! decodebin ! video/x-raw ! videoconvert ! \
gvadetect model=$DETECT_MODEL_PATH device=$DEVICE pre-proc=$PRE_PROC ! queue ! \
gvaclassify model=$CLASS_MODEL_PATH model-proc=$(PROC_PATH $MODEL2_PROC) device=$DEVICE pre-proc=$PRE_PROC ! queue ! \
gvaclassify model=$CLASS_MODEL_PATH1 model-proc=$(PROC_PATH $MODEL3_PROC) device=$DEVICE pre-proc=$PRE_PROC ! queue ! \
gvaclassify model=$CLASS_MODEL_PATH2 model-proc=$MODEL4_PROC device=$DEVICE pre-proc=$PRE_PROC ! queue ! \
gvawatermark ! videoconvert ! gvametaconvert method=detection !fpsdisplaysink video-sink=ximagesink sync=false
```

Multi-Channel Object Detection

Models:

Detection: mobilenet-ssd.xml



Pipelines:

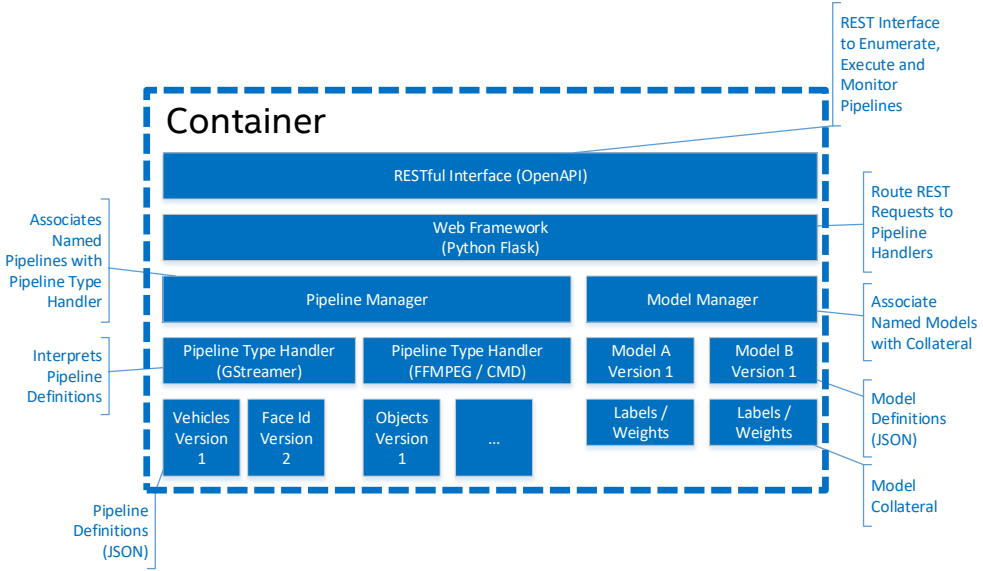
```
gst-launch-1.0 --gst-plugin-path ${GST_PLUGIN_PATH} \

filesrc location=/root/video-examples/bottle_1.mp4 ! \
        decodebin ! video/x-raw ! videoconvert ! \
        gvadetect model=$DETECT_MODEL_PATH device=$DEVICE pre-
proc=$PRE_PROC inference-id=1 model-proc=/root/models/object_detection/1/mobilenet-
ssd.json ! queue ! \
        gvawatermark ! videoconvert ! gvafpscounter ! fpsdisplaysink video-
sink=ximagesink sync=false \
    \
 filesrc location=/root/video-examples/bottle_2.mp4 ! \
        decodebin ! video/x-raw ! videoconvert ! videoscale ! video/x-raw,width=640,height=360 ! \
        gvadetect model=$DETECT_MODEL_PATH device=$DEVICE pre-
proc=$PRE_PROC inference-id=1 model-proc=/root/models/object_detection/1/mobilenet-
ssd.json ! queue ! \
        gvawatermark ! videoconvert ! gvafpscounter ! fpsdisplaysink video-sink=ximagesink sync=false
```

Agenda

- Video Analytics Usages
- Supporting Inference Across Different Hardware
- GStreamer Video Analytics
- Demo
- **Video Analytics Serving**
- Resources

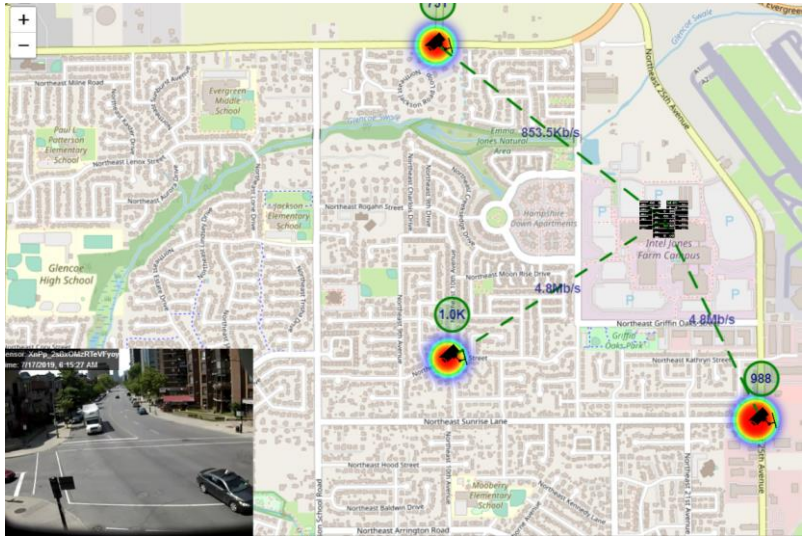
Video Analytics Serving

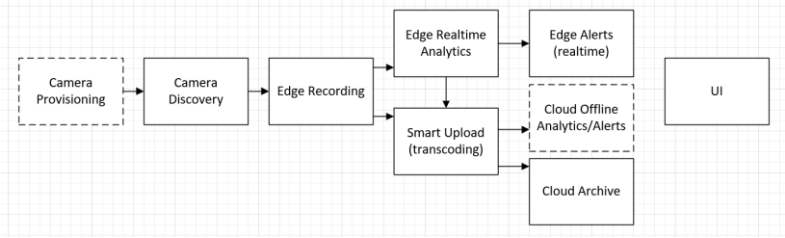
- https://github.com/intel/video-analytics-serving
- Containerized uService for Video Analytics
- RESTful interfaces for executing and monitoring video analytics pipelines
- Interface agnostic to underlying implementation (GStreamer, FFMPEG, Custom backend)
- Support scaling through container deployment and orchestration frameworks
- Simple to integrate and ready to scale



Smart Cities:
Open Visual Cloud

Process video streams and events from multiple cameras in real-time





Video Analytics Serving

GStreamer Elements

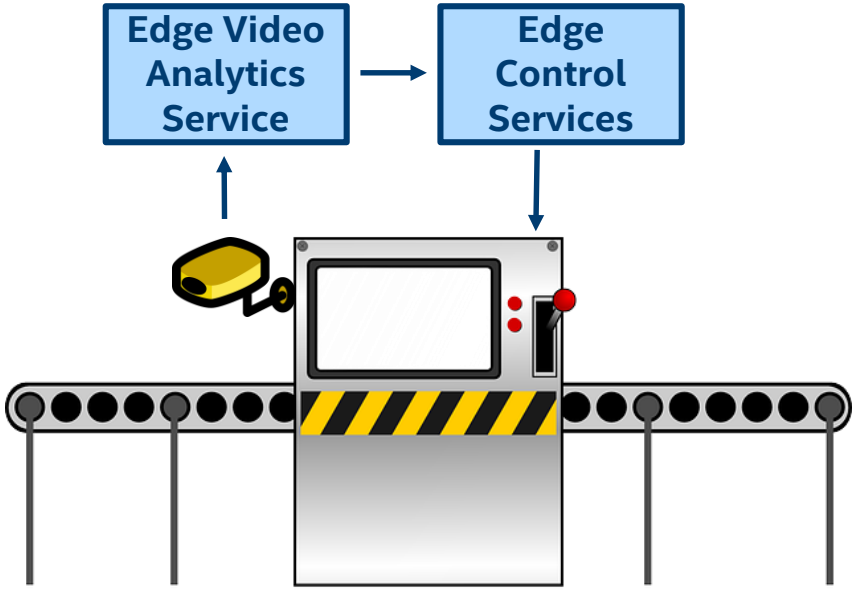
Decode | Scale, CSC | gvadetect | gvaclassify

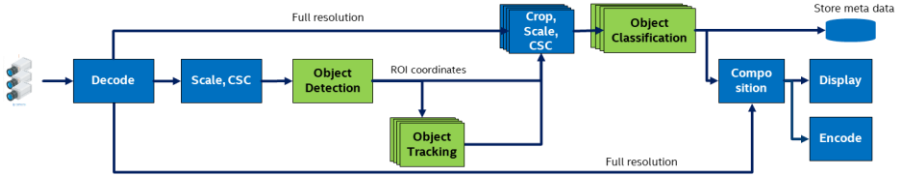
Media | **OpenVINO**

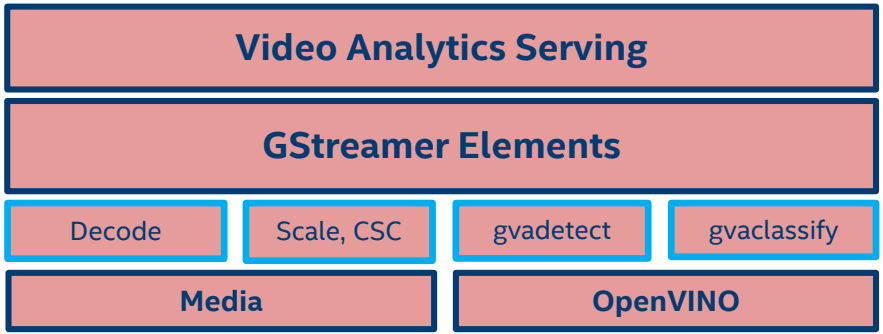
```
gst-launch-1.0 filesrc location=${FILE} !
  decodebin ! \
  gvadetect model=vehicle_detection.xml model-
proc=vehicle_detection.json ! queue ! \
  gvawatermark ! videoconvert ! metapublish method =
mqtt
```

Manufacturing Fault Detection: Edge Insights Platform

Identify defects and respond in real-time








```
gst-launch-1.0 filesrc location=${FILE} !
  decodebin ! \
  gvadetect model=fault_detection.xml model-
proc=fault_detection.json ! queue ! \
  gvawatermark ! videoconvert ! metapublish method =
kafka
```

Resources

OpenVINO

- https://github.com/opencv/dldt

GStreamer Video Analytics

- https://github.com/opencv/gst-video-analytics

Video Analytics Serving

- https://github.com/intel/video-analytics-serving

Open Visual Cloud

- Smart City sample
  https://github.com/OpenVisualCloud/Smart-City-Sample
- Ad Insertion sample
  https://github.com/OpenVisualCloud/Ad-Insertion-Sample
- Docker files including FFMPEG Video Analytics Filters
  https://github.com/OpenVisualCloud/Dockerfiles

THANK YOU!

Intel Visual Compute Middleware and Tools

Neelay Shah
Neena Maldikar
Mikhail Nikolskii
Ilya Belyakov