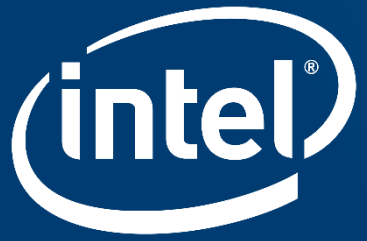


VIDEO ANALYTICS

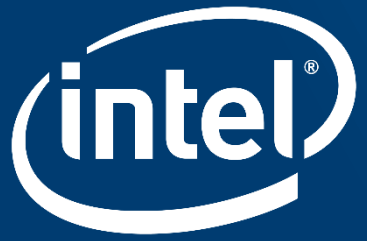
Raghavendra Ural
IoT Developer Evangelist
@ragural

Agenda

- ✓ Video Analytics with OpenCV
- ✓ Advanced Video Analytics
  - ✓ Intel Computer Vision SDK
  - ✓ Intel Deep Learning deployment tool

Video Analytics use cases in Visual Retail?

- Counting faces
- Demographics
- Movement detection
- Expression detection
- Face Recognition
- Security
- Monitor wait time
- And So on



VIDEO ANALYTICS WITH OPENCV

What is OpenCV?

OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage and is now maintained by Itseez.

OpenCV - Wikipedia
https://en.wikipedia.org/wiki/OpenCV

What it contains?

- OpenCV is an open source computer vision and machine learning software library.
- OpenCV was built to provide a common infrastructure for computer vision applications.
- It is under BSD license. Free to use in commercial applications
- The library has more than 2500 optimized algorithms

OpenCV Library functionalities?

Face detection

identify objects

classify human actions in videos

track camera movements

track moving objects

Template matching

Motion detection

extract 3D models of objects

stitch images

find similar images

remove red eyes

follow eye movements

recognize scenery

Face recognition

OpenCV Language support?

Core was built using C/C++

Bindings

- Java
- Python
- NodeJS

OpenCV Library functionalities?

Face detection

identify objects

classify human actions in videos

track camera movements

track moving objects

Template matching

Motion detection

extract 3D models of objects

stitch images

find similar images

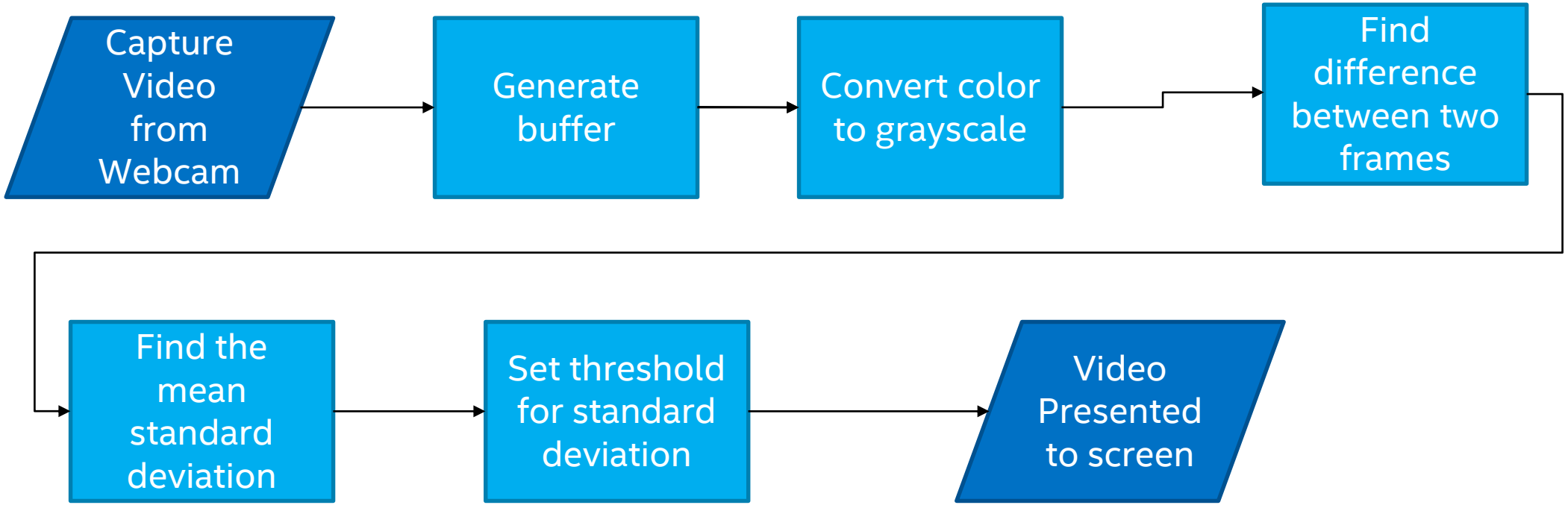
remove red eyes

follow eye movements

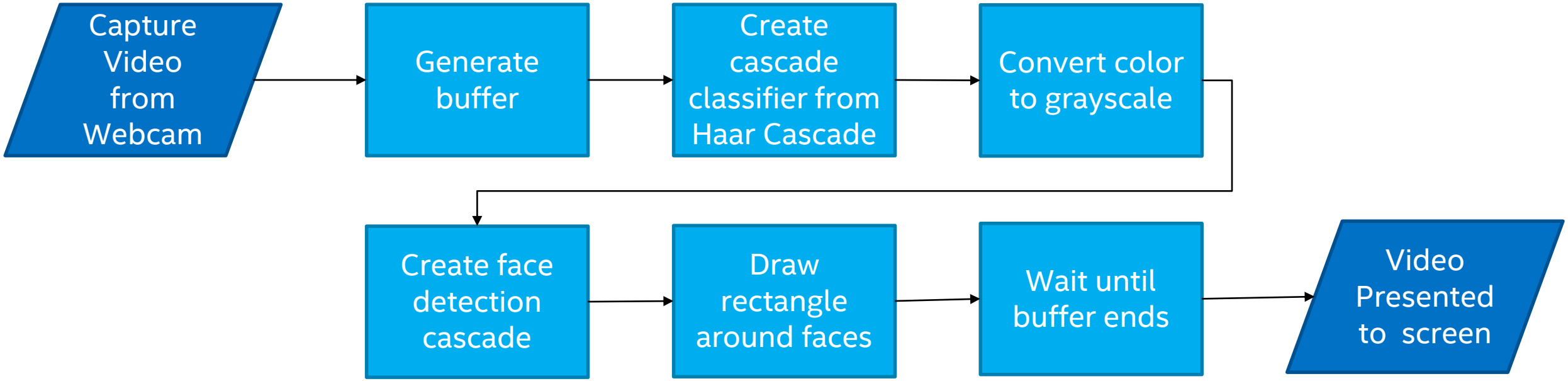
recognize scenery

Face recognition

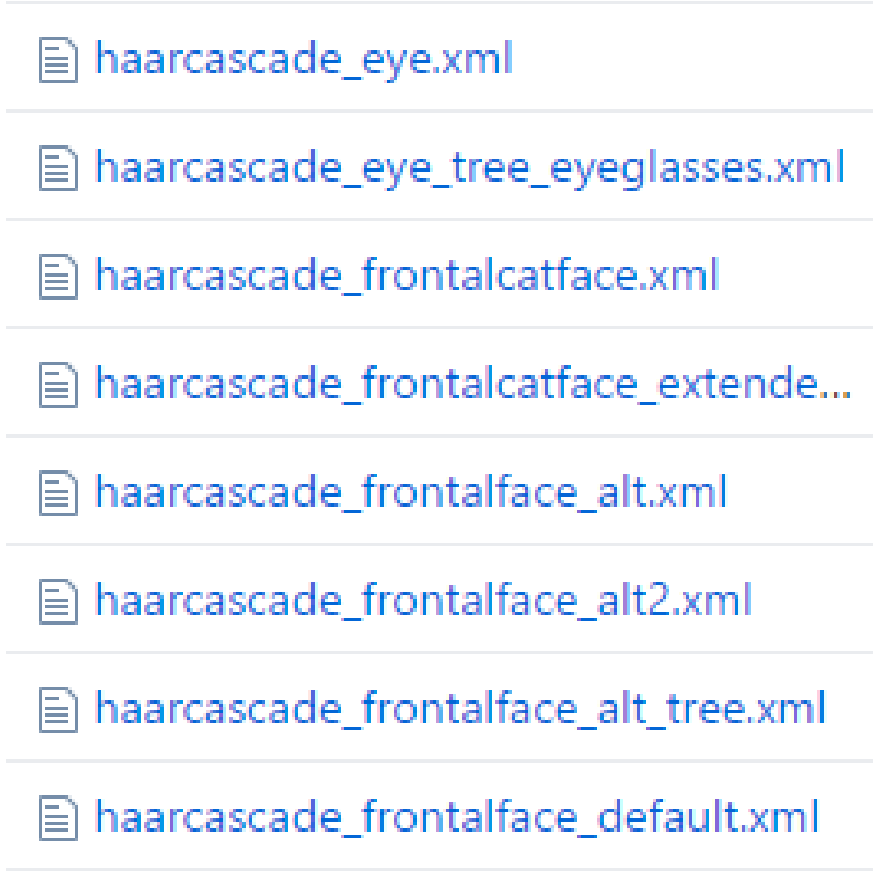
Motion detection

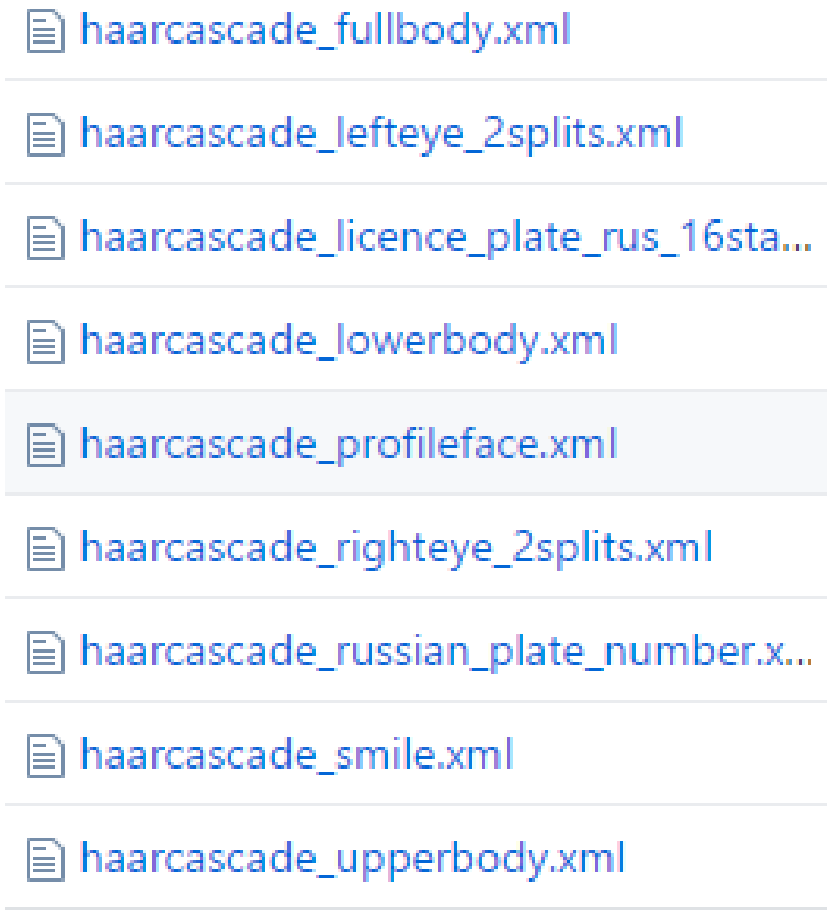


Face detection

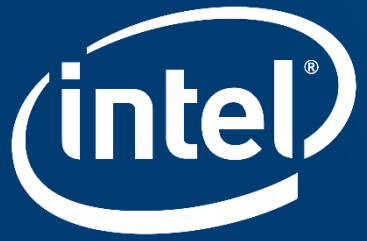


Free haar cascade files





https://github.com/opencv/opencv/tree/master/data/haarcascades



ADVANCED VIDEO ANALYTICS

INTEL
COMPUTER VISION SDK

Intel® Computer Vision SDK

Accelerate Computer Vision Solutions on Intel® Platforms**

What it is

A comprehensive toolkit to **accelerate development of computer vision solutions for autonomous vehicles, smart cameras, robotics, & image processing.**

Why important

Demand is growing for **intelligent, computer vision** & **visual understanding** solutions, **media analytics**, & **artificial intelligence** from edge to cloud.

Users

- **Software developers**
- **Data scientists** interested in neural network inference & deep learning deployment capabilities.

Learn more & download at: software.intel.com/computer-vision-sdk

**Certain technical specifications and select processors/skus apply. See product site for more details.

What's Inside the Intel® Computer Vision SDK

(Linux version components shown)*

OPTIMIZED LIBRARIES

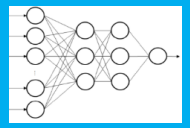
OpenVX™
Intel® Atom™ Runtime, Computer Vision Engine Runtime & Emulator, Kernels, Graphs, Workload samples

INNOVATE & CUSTOMIZE

OpenCV
Quick start to create own custom kernels or use library of functions

MACHINE LEARNING

Model Optimizer



Intel Deep Leaning Deployment Tool

EXPLORE

Vision Algorithm Designer
Prototype, optimize algorithm hardware affinity; debug

OpenCL™ Driver for Intel® Architecture

Deep Learning Frameworks

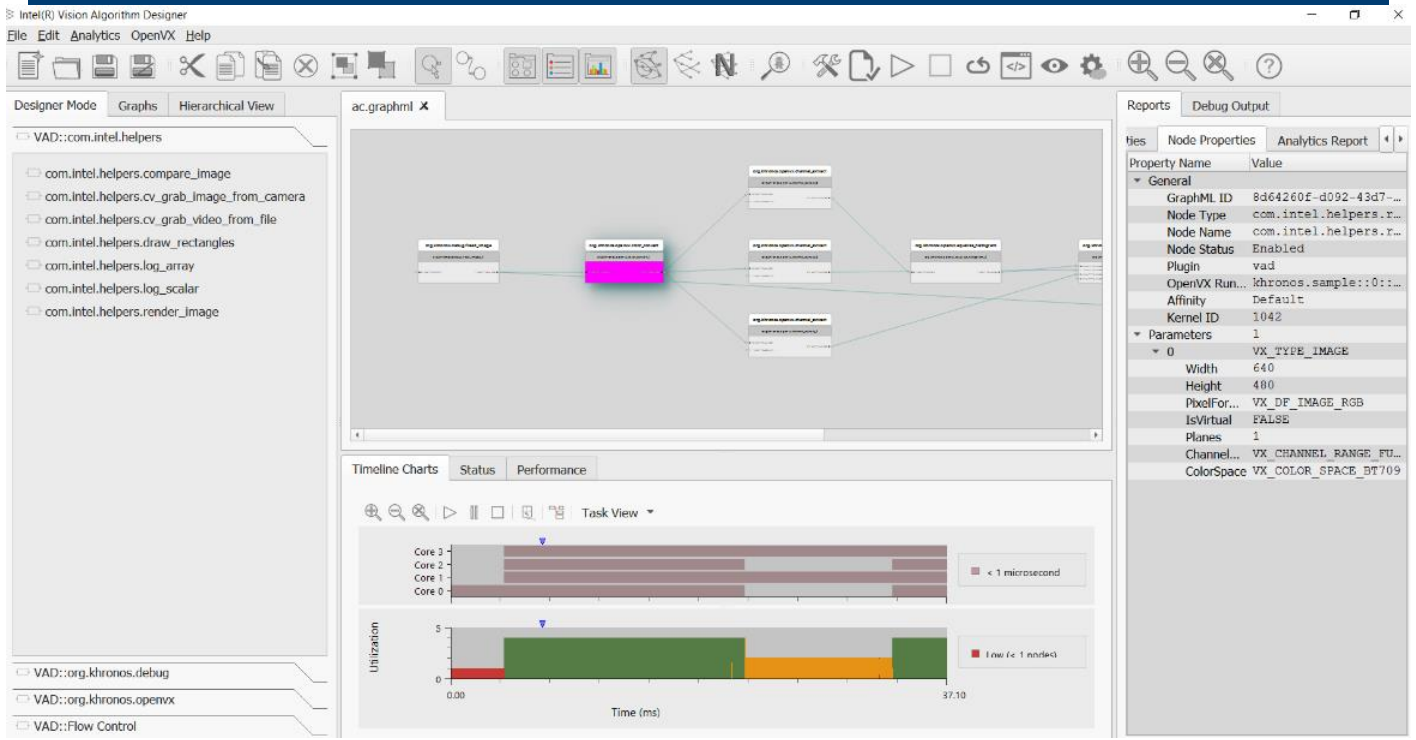
DEEP LEARNING

Intel Deep Learning Tools

Development Flow Options

Vision Algorithm Designer

- Visual environment for computer vision algorithm development
- Produces optimized code
- Unique debug capabilities to root-cause algorithm issues
- Performance profiling, analysis & visualization capabilities



OpenVX™ C/C++ API

- Use with familiar IDEs
- Interoperable with other libraries, SDKs & programming models

```
vx_context context = vxCreateContext();
vx_image input = vxCreateImage( context, 640, 480, VX_DF_IMAGE_U8 );
vx_image output = vxCreateImage( context, 640, 480, VX_DF_IMAGE_U8 );

vx_graph graph = vxCreateGraph( context );
vx_image intermediate = vxCreateVirtualImage( graph, 640, 480, VX_DF_IMAGE_U8 );
vx_node F1 = vxF1Node( graph, input, intermediate );
vx_node F2 = vxF2Node( graph, intermediate, output );

vxVerifyGraph( graph );
vxProcessGraph( graph ); // run in a loop
```

Intel® Computer Vision SDK

Optimize performance of Intel computer vision accelerators

Accelerate Performance

CPU | Intel® Graphics processor (GPU) | IPU | Future Intel® Processors | FPGA

Intel Computer Vision SDK

OpenVX

Image Processing	Core Computer Vision	Feature Extraction and Tracking

INTEL EXTENSIONS

Deep Learning	Library of Common Functions	Pre-processing / Feature Extraction

OpenCV functions	High-level Video/ Image Processing

VAD UTILITIES

Flow Operations	Debug/Validation	Profiling

open source functions ⇩ implemented in

OpenCL™

Intel® SDK for OpenCL™ Applications

Innovate and accelerate beyond the common functions to customize your solutions

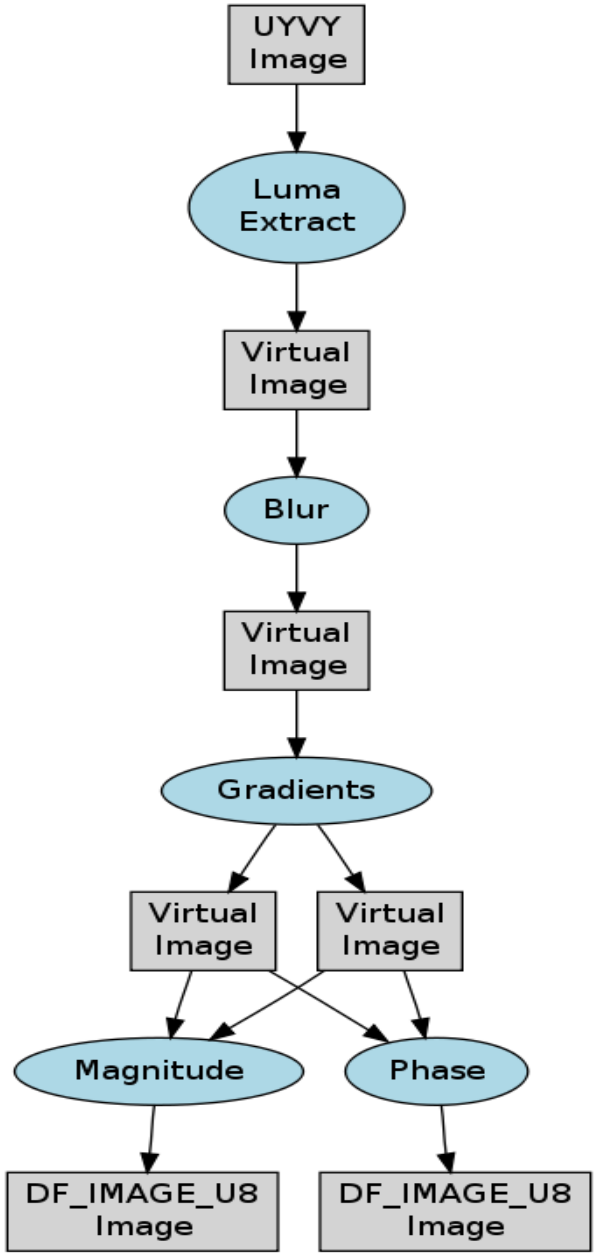
A Graph-Based Approach

Computer vision tasks are well described by graphs

- Each node in the graph is an “operation”
- Performed by a “Kernel”

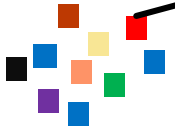
The fact that we see the whole flow allows us to provide performance acceleration and graph manipulations

A “graph compiler”, aware of the specific HW can do a better scheduling work..



Kernels

OpenVX standard,
Intel extensions, utilities



```
 vx_imagepatch_addressing_t src_addr;
 src_addr.dim_x    = src_width;
 src_addr.dim_y    = src_height;
 src_addr.stride_x = 3*sizeof( vx_uint8 );
 src_addr.stride_y = cv_src_rgb.step;

 void *src_ptrs[]  = { cv_src_rgb.data };

 vx_image img = vxCreateImageFromHandle(
context, VX_DF_IMAGE_RGB, &src_addr,


src_ptrs, VX_IMPORT_TYPE_HOST );
….
…
…
```

Kernels
OpenVX standard,
Intel extensions, utilities

Custom Kernels
C, C++, OpenCL, OpenCV, DSP

Kernels

OpenVX standard,
Intel extensions, utilities

Custom Kernels

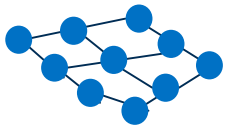
C, C++, OpenCL, OpenCV, DSP

A Sample

Supplied with the SDK

Caffe

DL Graph
OpenVX graph

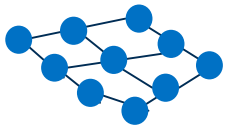


Kernels
OpenVX standard,
Intel extensions, utilities

Custom Kernels
C, C++, OpenCL, OpenCV, DSP

A Sample
Supplied with the SDK

DL Graph
OpenVX graph

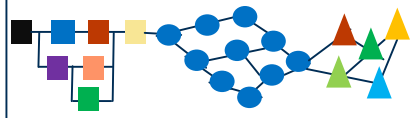


Kernels
OpenVX standard,
Intel extensions, utilities

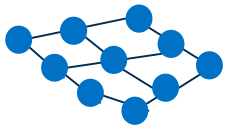
Custom Kernels
C, C++, OpenCL, OpenCV, DSP

A Sample
Supplied with the SDK

Graph
OpenVX graph



DL Graph
OpenVX graph

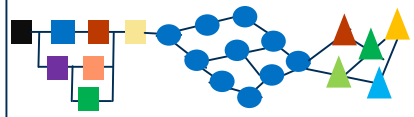


Kernels
OpenVX standard,
Intel extensions, utilities

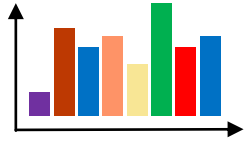
Custom Kernels
C, C++, OpenCL, OpenCV, DSP

A Sample
Supplied with the SDK

Graph
OpenVX graph



Optimize
Performance,
Heterogeneous



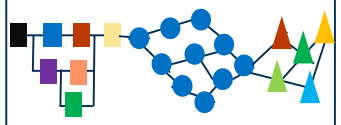
DL Graph
OpenVX graph

Kernels
OpenVX standard,
Intel extensions, utilities

Custom Kernels
C, C++, OpenCL, OpenCV, DSP

A Sample
Supplied with the SDK

Graph
OpenVX graph



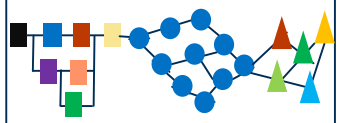
DL Graph
OpenVX graph

Kernels
OpenVX standard,
Intel extensions, utilities

Custom Kernels
C, C++, OpenCL, OpenCV, DSP

A Sample
Supplied with the SDK

Graph
OpenVX graph



Optimize
Performance,
Heterogeneous

Optimize
Performance,
Heterogeneous

Optimize
Performance,
Heterogeneous

CPU GPU — Sky-Lake

CPU GPU IPU — Apollo-Lake

CPU GPU IPU FPGA

OpenVX 1.0.1 Base Kernels

AbsDiff	Color Convert	Magnitude	Threshold
Accumulate	Convert Image Depth	MeanStdDev	Xor
Accumulate Squared	Convolve	Median3x3	Phase
Accumulate Weighted	Dilate3x3	MinMaxLoc	Remap
Add	Erode3x3	Multiply	Optical Flow Pyramid (LK)
And	Gaussian3x3	Not	Equalize Histogram
Box3x3	HalfScale Gaussian3x3	Or	Warp Affine
Canny Edge Detector	Histogram	Pyramid	Warp Perspective
Channel Combine	Integral Image	Scale Image	FAST Corners
Channel Extract	Table Lookup	Sobel3x3	Harris Corners

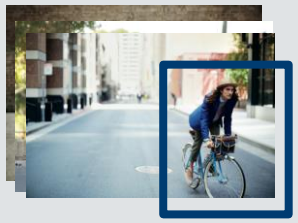

intel
Software
Deep
Learning
Deployment
Toolkit
From Intel

DEEP LEARNING STEPS

STEP 1: TRAINING

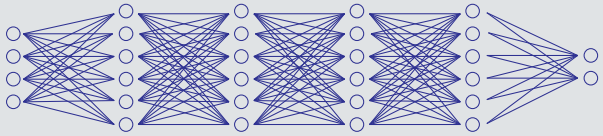
(In Data Center – Over Hours/Days/Weeks)

Lots of labeled input data



Person

Create “Deep neural net” math model



Output: Trained Model

Trained Model

STEP 2: INFERENCE

(End point or Data Center - Instantaneous)

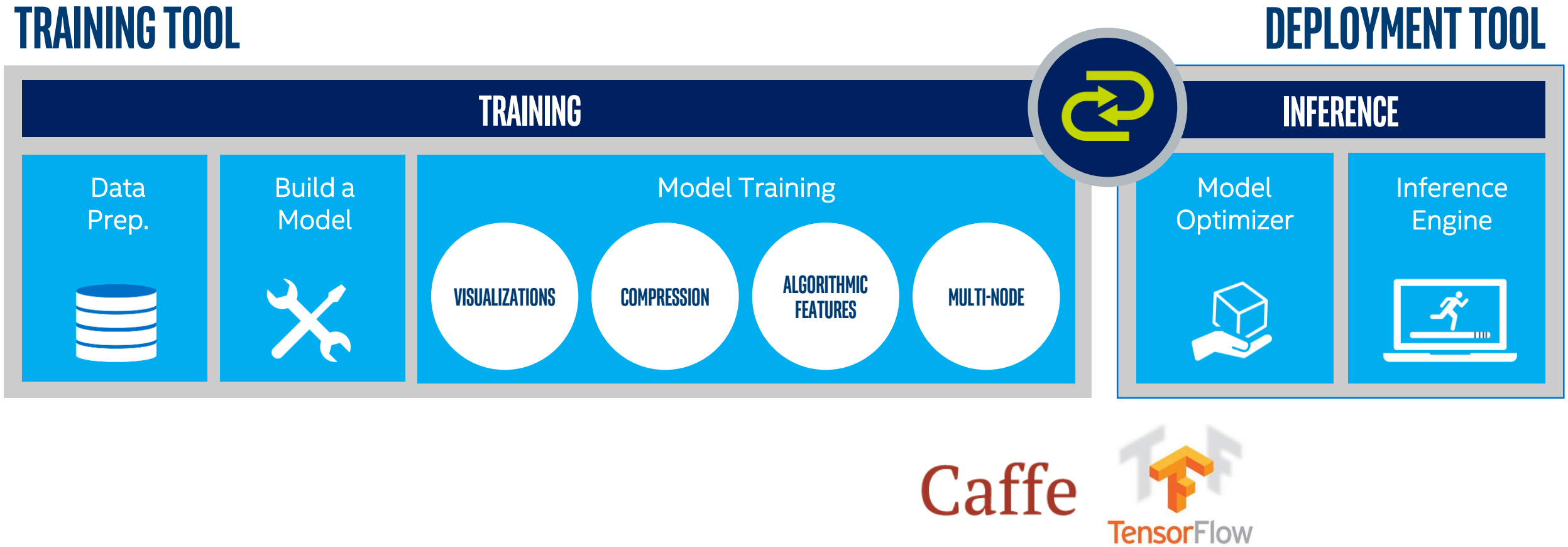
New input from camera and sensors

Trained neural network model

97% person
2% traffic light

Output: Classification

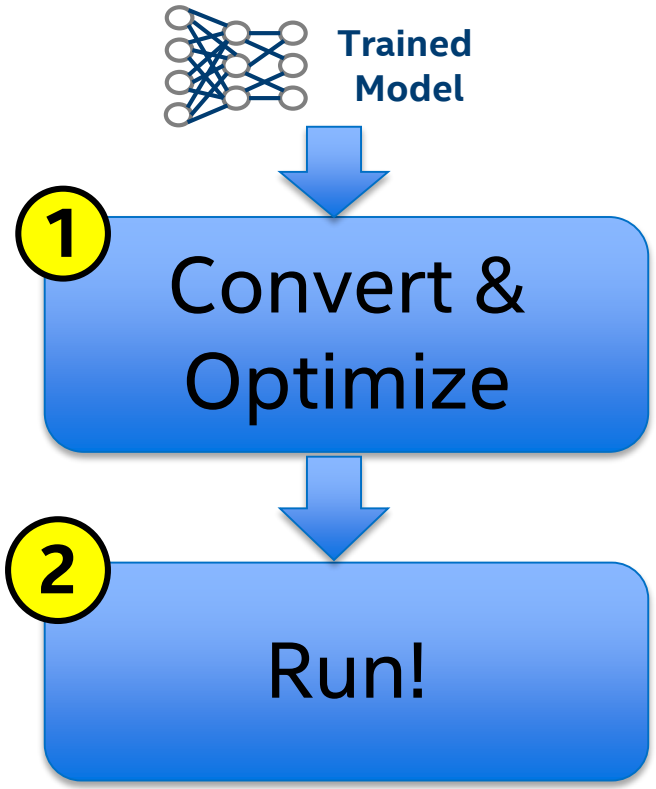
The Deep Learning Workflow



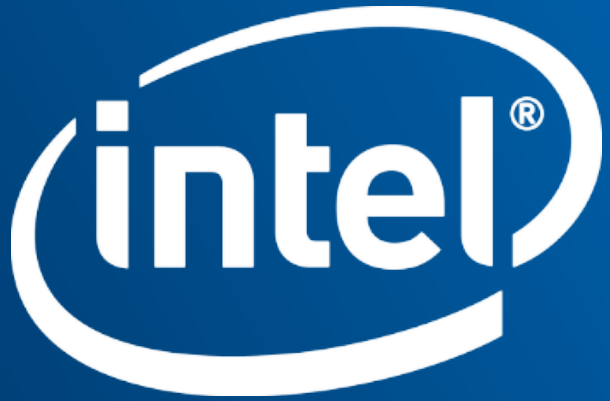
INTEL'S DEEP LEARNING DEPLOYMENT TOOLKIT

Enable full utilization of Intel® architecture Inference while abstracting HW from developers

1. Imports trained models from popular DL framework regardless of training HW
   - Enhances model for improved execution, storage & transmission
2. Optimizes Inference execution for target hardware (computational graph analysis, scheduling, model compression, quantization)
   - Enables seamless integration with application logic
   - Delivers embedded friendly Inference solution



Ease of use + Embedded friendly + Extra performance boost


intel