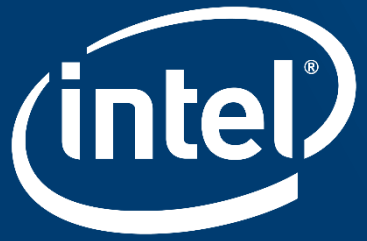


INTRODUCTION TO INTEL® NEURAL COMPUTE STICK 2

LEGAL 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.

A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Code names featured are used internally within Intel to identify products that are in development and not yet publicly announced for release. Customers, licensees and other third parties are not authorized by Intel to use code names in advertising, promotion or marketing of any product or services and any such use of Intel's internal code names is at the sole risk of the user.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. Check with your system manufacturer or retailer or learn more at intel.com.

Intel, the Intel logo, Arria, Intel Core, Intel RealSense, Movidius, Myriad, OpenVINO, and Xeon are trademarks of Intel Corporation in the U.S. and/or other countries.

Microsoft, Windows, and the Windows logo are trademarks, or registered trademarks of Microsoft Corporation in the United States and/or other countries.

OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission by Khronos.

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

Copyright © 2020, Intel Corporation. All rights reserved.

COURSE OBJECTIVE

The objective of this course is to provide a self-paced learning framework to deploy deep learning models on the Intel® Movidius™ Neural Compute Stick

The OpenVINO™ toolkit that facilitates easy deployment on Intel® Neural Compute Stick 2 will be used to implement the course

Chapters 1–5 cover introductory content necessary to understand capabilities of the OpenVINO toolkit to deploy pretrained image classification and object detection models

The final course outcome is to deploy custom models on low power embedded devices like the Raspberry Pi*, which are covered in chapters 6 and 7

AGENDA

- **Chapter Outcome: Learn introductory concepts used throughout the course**
  - Introduction to Computer Vision and Deep Learning
  - Intel® AI Portfolio
  - Introduction to Deep Learning Inference at the Edge
  - Introduction to the Intel® Neural Compute Stick 2 (Intel® NCS2)
  - Real-world use cases using the Intel NCS2
  - How to take advantage of Intel NCS2 for Inference at the Edge using the Intel® Distribution of OpenVINO™ Toolkit

PART 1: INTRODUCTION TO TRADITIONAL COMPUTER VISION AND DEEP LEARNING

TRADITIONAL COMPUTER VISION

TRADITIONAL COMPUTER VISION

$N \times N$

$(f_1, f_2, \ldots, f_K)$

$K \ll N$

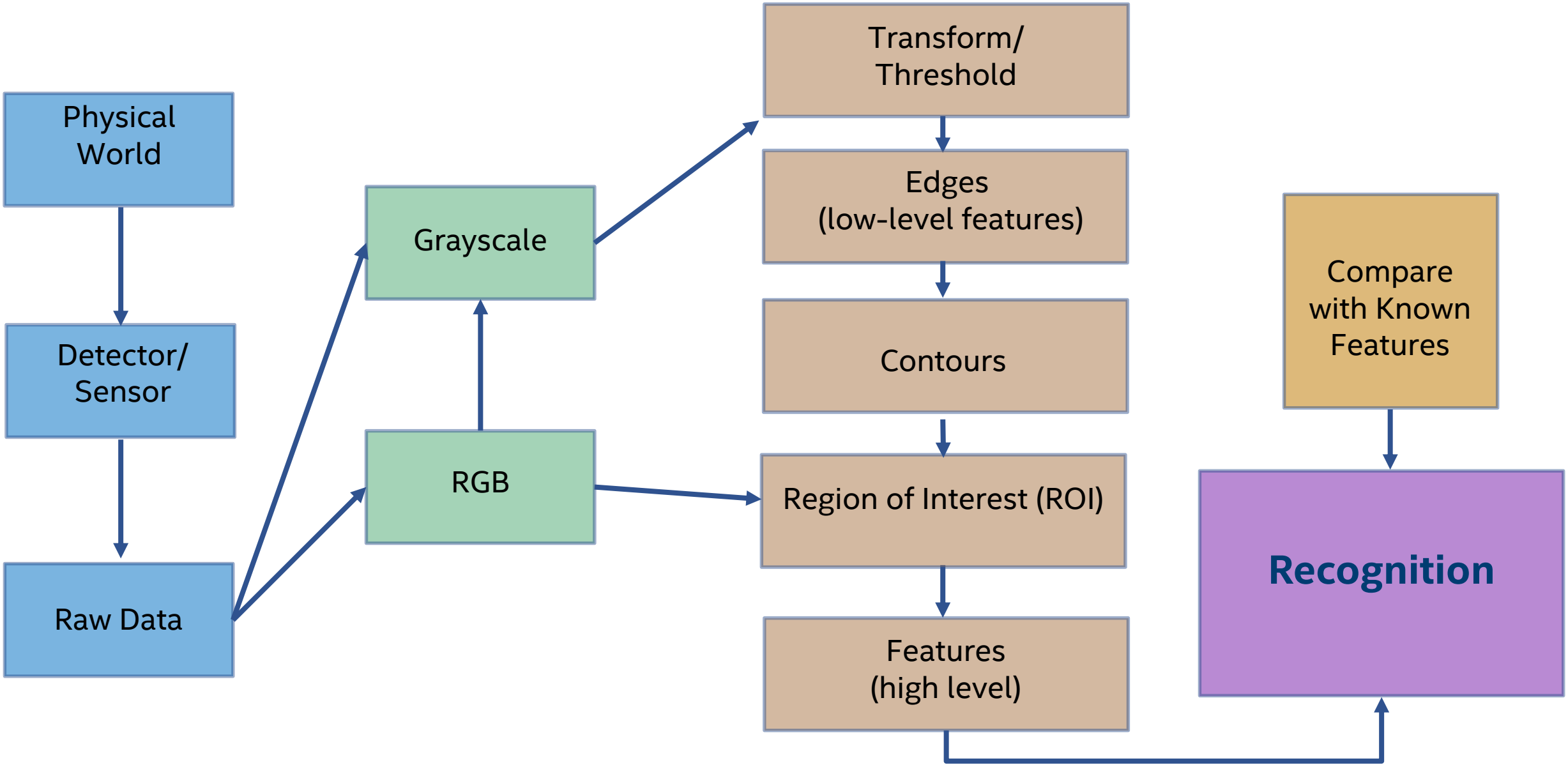
Lines
Edges
Ridge
Corners
Blobs
Points
Texture/motion related

Cat

- Uses an engineered set of features
- Algorithm learns to associate pattern of features to provide an identity
- Requires some domain expertise to extract features

https://commons.wikimedia.org/wiki/File:Cat_Cute.JPG
Attribution: Gaurav Pandit [CC BY (https://creativecommons.org/licenses/by/3.0)]

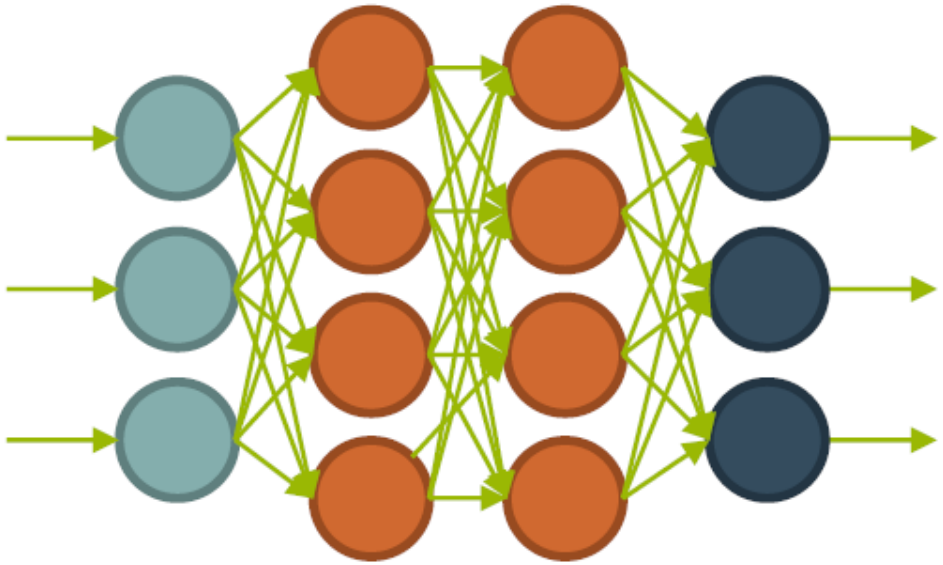
TRADITIONAL VISUAL SYSTEM ARCHITECTURE



HOW IS TRADITIONAL COMPUTER VISION DIFFERENT FROM DEEP NEURAL NETWORKS

MOTIVATION FOR DEEP NEURAL NETWORKS

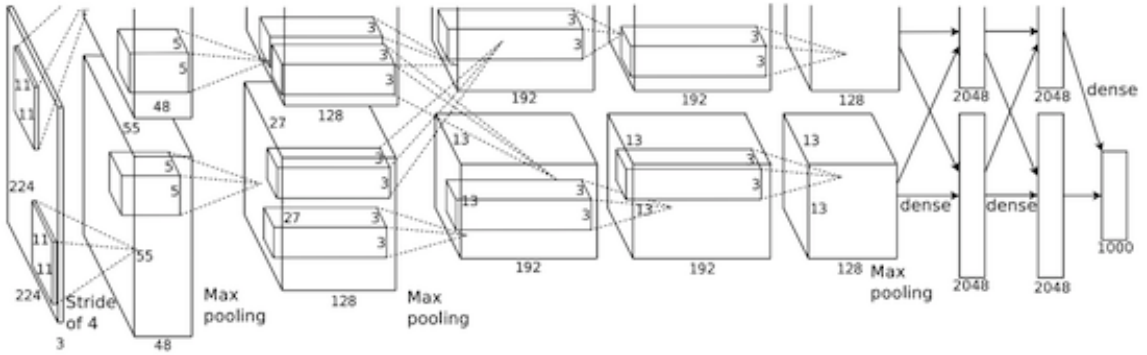
- Use biology as an inspiration for mathematical models
- Get signals from previous neurons
- Generate signals (or not) according to inputs
- Pass signals on to subsequent neurons
- Create complex models by layering multiple neurons



END-TO-END DEEP LEARNING

$N \times N$

Topology: Different layouts of convolution, pooling, and fully connected layers with variable size kernels to learn features during training



Cat

~60 million parameters

Conceptual shift in thinking, from How do you engineer the best features? to How do you guide the model toward finding the best features?

https://commons.wikimedia.org/wiki/File:Cat_Cute.JPG
Attribution: Gaurav Pandit [CC BY (https://creativecommons.org/licenses/by/3.0)]

COMPUTER VISION VERSUS DEEP NEURAL NETWORKS

Computer Vision

- Delivers higher precision since we know the exact features required to form an identity

Deep Neural Networks

- Offers more flexibility in training a model for diverse data sets. Model learns features in real time to adapt.

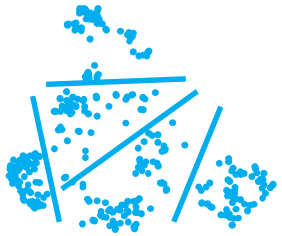
Both of these techniques provide a trained model, which can then be used for inference.

PART 2: INTEL® AI PORTFOLIO

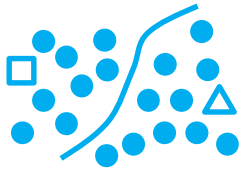
WHAT IS AI?

SUPERVISED LEARNING

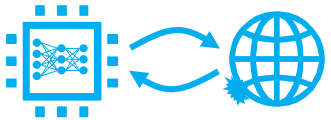
UNSUPERVISED LEARNING

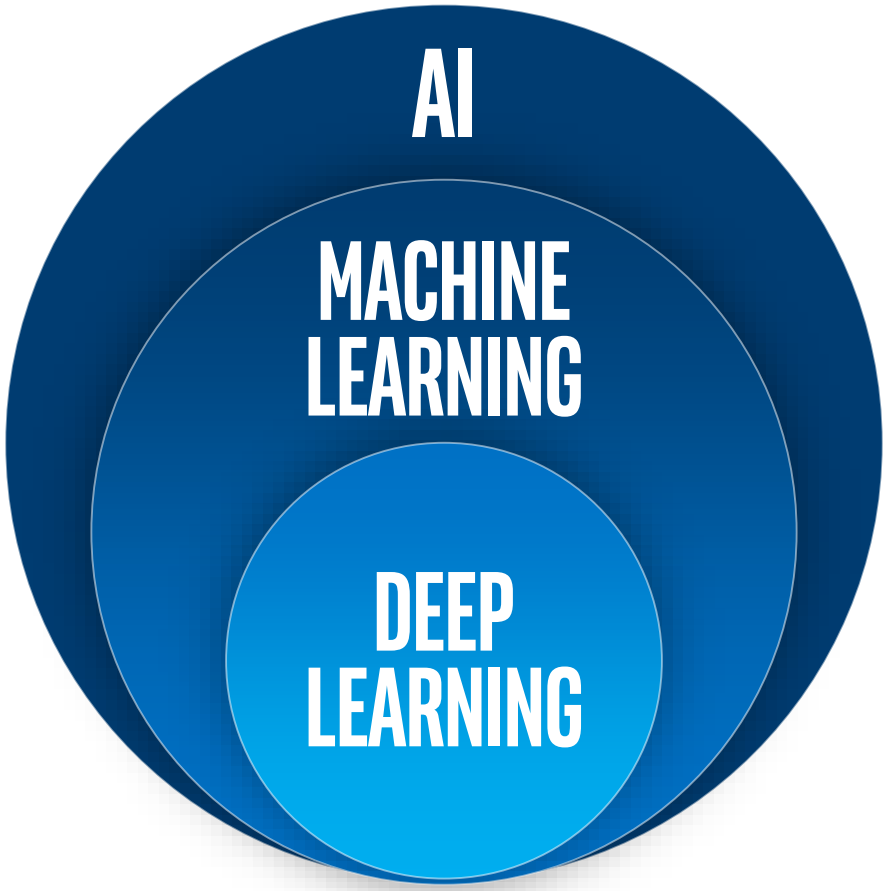


SEMI-SUPERVISED LEARNING



REINFORCEMENT LEARNING





- Regression
- Classification
- Clustering
- Decision Trees
- Data Generation
- Image Processing
- Speech Processing
- Natural Language Processing
- Recommender Systems
- Adversarial Networks

NO ONE-SIZE-FITS-ALL APPROACH TO AI

TRAINING AND INFERENCE WORKFLOW

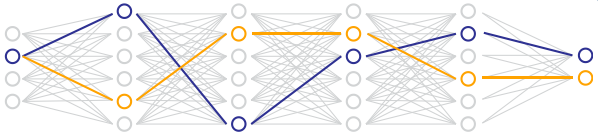
Step 1: Training
(Over Hours/Days/Weeks)

Input data

Create deep network

Output classification

Person



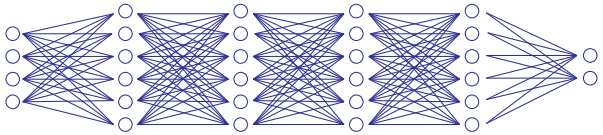
Trained model

90% person
8% traffic light

Step 2: Inference
(Real Time)

New input from camera and sensors

Trained neural network model



97% person

Output classification

INTEL® AI HARDWARE

DEVICE | INTELLIGENT EDGE | MULTI-CLOUD

OPTIMIZED FRAMEWORKS & SOFTWARE

Learn more in this course

CPU	GPU	FPGA	ASIC		

WORKLOAD BREADTH ← → AI SPECIALIZATION

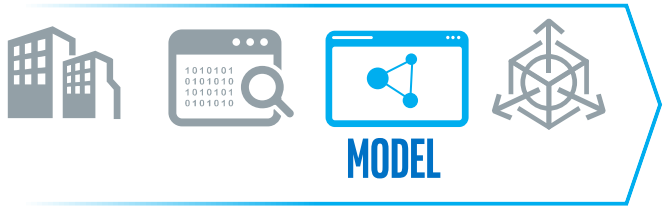
Multi-Purpose Foundation for AI	Data-Parallel Media, Graphics, HPC & AI	Multi-Function & Real-time Deep Learning Inference	Deep Learning Inference	Deep Learning Training	Media & Vision DL Inference at the Edge

Visit: www.intel.ai/technology

All products, computer systems, dates, and figures are preliminary based on current expectations, and are subject to change without notice.
1Unified software stack development in progress DL=Deep Learning

SPEED UP DEVELOPMENT

WITH OPEN AI SOFTWARE



Learn more in this course

	MACHINE LEARNING				DEEP LEARNING	
TOOLKITS App Developers	ANALYTICS ZOO				MODEL ZOO	OpenVINO™
LIBRARIES Data Scientists	**Intel® Data Analytics Acceleration Library** (DAAL)	**Intel® Distribution for Python*** (Sklearn, pandas)	**R** (Cart, Random Forest, e1071)	**Distributed** (MlLib on Apache Spark*, Mahout)	**Frameworks**: TensorFlow, BigDL, Caffe, ONNX, mxnet, PyTorch *More framework optimizations in progress...*	**Intel Tools**: NAUTA, RL Coach, NLP Architect, NN Distiller
KERNELS Library Developers	**Intel® Math Kernel Library** (Intel® MKL)				**Intel® Machine Learning Scaling Library** (Intel® MLSL)	**Intel® Math Kernel Library for Deep Neural Networks** (Intel® MKL-DNN), nGraph
	CPU				CPU ▪ GPU ▪ FPGA ▪ ACCELERATOR	

Visit: www.intel.ai/technology

1 An open source version is available at: 01.org/openvinotoolkit *Other names and brands may be claimed as the property of others.
Developer personas shown above represent the primary user base for each row, but are not mutually-exclusive
All products, computer systems, dates, and figures are preliminary based on current expectations, and are subject to change without notice.

PART 3: WHAT DOES DEEP LEARNING INFERENCE AT THE EDGE MEAN?

LET'S LOOK AT A LARGER SCALE…

20 billion connected devices by 2020[1]

generating **billions of petabytes of data** traffic between devices & the cloud

1 Source: http://www.gartner.com/newsroom/id/3598917

THE NEED FOR INTELLIGENCE AT THE EDGE!

KEY DRIVERS: LOW POWER, LATENCY, BANDWIDTH, STORAGE, SECURITY

PART 4: INTEL® MOVIDIUS™ TECHNOLOGY FOR INFERENCE AT THE EDGE

INTEL® MOVIDIUS™ VISION PROCESSING UNIT (VPU)

SERVICE ROBOTS
- Navigation
- 3D Vol. mapping
- Multimodal sensing

SURVEILLANCE
- Detection/classification
- Identification
- Multinodal systems
- Multimodal sensing
- Video, image capture

WEARABLES
- Detection, tracking
- Recognition
- Video, image, session capture

DRONES
- Sense and avoid
- GPS denied hovering
- Pixel labeling
- Video, image capture

AR-VR HMD
- 6DOF pose, position, mapping
- Gaze, eye tracking
- Gesture tracking, recognition
- See-through camera

SMART HOME
- Detection, tracking
- Perimeter, presence monitoring
- Recognition, classification
- Multinodal systems
- Multimodal sensing
- Video, image capture

POWER-EFFICIENT IMAGE PROCESSING, COMPUTER VISION & DEEP LEARNING FOR DEVICES

All products, computer systems, dates, and figures are preliminary based on current expectations, and are subject to change without notice.

INTEL® NEURAL COMPUTE STICK 2

USB STICK FORM FACTOR
for neural network acceleration

REAL-TIME ON-DEVICE INFERENCE
no cloud connectivity required

NO ADDITIONAL PERIPHERALS
needed to start deploying solutions

ACCELERATE DEVELOPMENT
with Intel® Distribution of OpenVINO™ toolkit

INDUSTRY-LEADING PERFORMANCE
with Intel® Movidius™ Myriad™ X Vision Processing Unit (VPU)

HIGHER PERFORMANCE
on deep neural networks compared to Intel Movidius Myriad 2 VPU

[1]Testing by Intel as of October 12th, 2018
Deep Learning Workload Configuration. Comparing Intel® Movidius™ Neural Compute Stick based on Intel® Movidius™ Myriad™ 2 VPU vs. Intel® Neural Compute Stick 2 based on Intel® Movidius™ Myriad™ X VPU with Asynchronous Plug-in enabled for (2xNCE engines). As measured by images per second across GoogleNetV1. Base System Configuration: Intel® Core™ I7-8700K processor, 95W TDP (6C12T at 3.7 GHz base freq and 4.7 GHz max turbo freq), Graphics: Intel® UHD Graphics 630 Total Memory 65830088 kB Storage: INTEL SSDSC2BB24 (240 GB), Ubuntu* 16.04.5 Linux*-4.15.0-36-generic-x86_64-with-Ubuntu-16.04-xenial, deeplearning_deploymenttoolkit_2018.0.14348.0, API version 1.2, Build 14348, myriadPlugin, FP16, Batch Size = 1

TECHNICAL SPECIFICATIONS - INTEL® MOVIDIUS™ MYRIAD™ X VPU

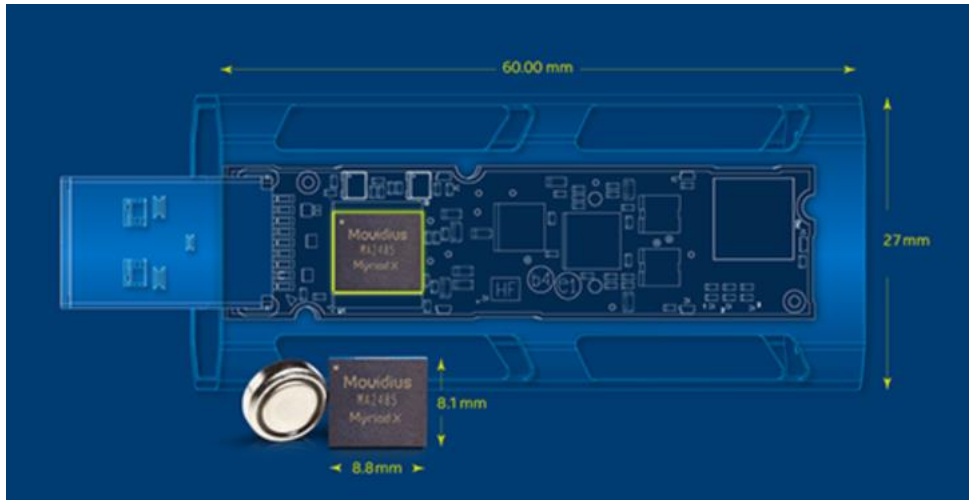
Hardware:

- Processor: Intel Movidius Myriad X Vision Processing Unit (VPU)
- Supported frameworks: TensorFlow*, Caffe*, Apache MXNet*, Open Neural Network Exchange (ONNX*), PyTorch*, and PaddlePaddle via an ONNX conversion
- Connectivity: USB 3.0 Type-A
- Dimensions: 2.85 in. x 1.06 in. x 0.55 in. (72.5 mm x 27 mm x 14 mm)
- Operating temperature: 0° C to 40° C

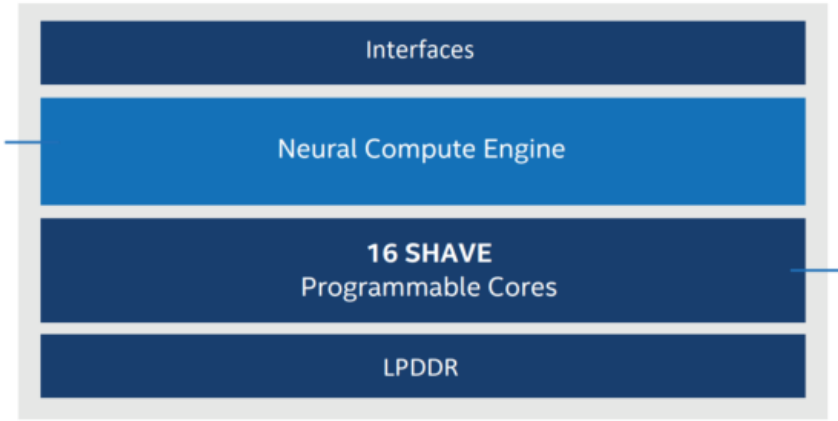
Software:

- Intel® Distribution of OpenVINO™ toolkit
- Supported operating systems:
  - Ubuntu* 16.04.3 LTS (64 bit)
  - CentOS* 7.4 (64 bit)
  - Windows® 10 (64 bit)
  - macOS* 10.14.4 (or later)
  - Raspbian* (target only)
  - Other (via the open source distribution of OpenVINO™ toolkit)

WHAT'S INSIDE INTEL® MOVIDIUS™ MYRIAD™ X VPU



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

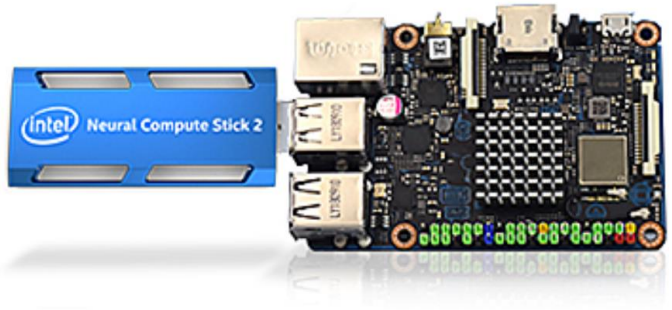


VLIW (DSP) programmable processors are optimized for complex visions and imaging workloads

Homogeneous memory design for low-power, UL latency, sustained High Performance

PLUG AND PLAY AI

USING INTEL® NEURAL COMPUTE STICK 2



Single-Board Computers

Prototype with low-cost edge devices, such as Raspberry Pi* and other Arm* host devices.

AI on the PC

Build cutting-edge AI and machine learning into your PC applications with the Intel® NUC.

Depth-Sensing Cameras

Depth sensing meets plug-and-play edge inferencing with AI using an Intel® RealSense™ Depth Camera D400 series.

GAME CHANGING INTELLIGENT DEVICES

POWERED BY INTEL® MOVIDIUS™ VPU

Hikvision
Intelligent Camera

Hikvision
Industrial Camera

DJI Inspire* 2

DJI
Phantom* 4 Pro

DJI Mavic Pro*

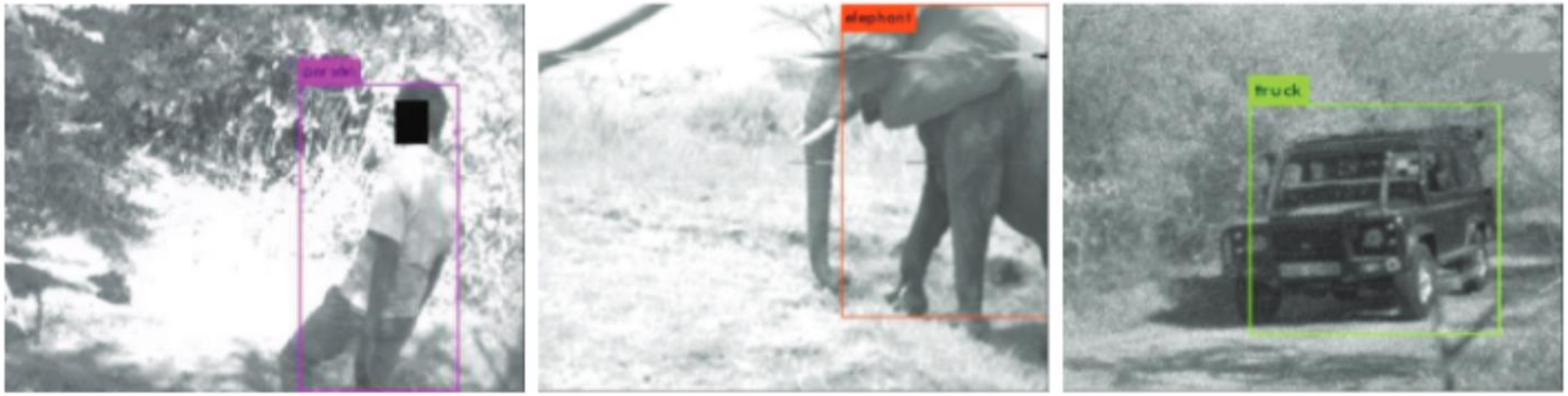
Uniview
IP Camera

Dahua
Industrial Camera

Moto 360º
Camera

PART 5: USE CASES WITH INTEL® MOVIDIUS™ MYRIAD™ VISION PROCESSING UNIT

FIGHTING ILLEGAL POACHING WITH PURPOSE-BUILT AI CAMERA



The TrailGuard AI performs on-camera inference to identify people, animals, and objects.

The Intel® Movidius™ Myriad™ 2 Vision Processing Unit (VPU) powers the TrailGuard AI vision processing and on-camera inference—all while operating at very low power to enable an estimated year-long battery life.

https://www.intel.ai/solutions/fighting-illegal-poaching-with-purpose-built-ai-camera/#gs.wpUQcXJh

AI-DRIVEN TEST SYSTEM TO DETECT BACTERIA IN WATER

Intel AI Academy Innovator applies AI techniques to detect harmful bacteria in water using the Intel® Movidius™ Neural Compute Stick

https://software.intel.com/en-us/articles/ai-driven-test-system-detects-bacteria-in-water

TELLO* – INTEL® MOVIDIUS™ MYRIAD™ VISION PROCESSING UNIT POWERED TOY DRONE FROM RYZE

https://www.movidius.com/news/hello-tello-ryze-announces-intel-myriad-vpu-powered-toy-drone

PART 6: SOFTWARE SUPPORT FOR THE INTEL® MOVIDIUS™ NEURAL COMPUTE STICK

INTRODUCTION TO THE INTEL® DISTRIBUTION OF OPENVINO™ TOOLKIT

BENEFITS OF THE OPENVINO™ TOOLKIT

Harness the Power of Intel® Processors: CPU, CPU with Integrated Graphics, FPGA,VPU

ACCELERATE PERFORMANCE

Access Intel computer vision accelerators.
Speed code performance.
Supports heterogeneous processing
& asynchronous execution.

INTEGRATE DEEP LEARNING

Unleash convolutional neural network (CNN) based deep learning inference across using a common API & 10 trained models.

Up to 10x increase*

SPEED DEVELOPMENT

Reduce time using a library of optimized OpenCV & OpenVX* functions, 15+ samples.

Develop once, deploy for current & future Intel®-based devices.

INNOVATE & CUSTOMIZE

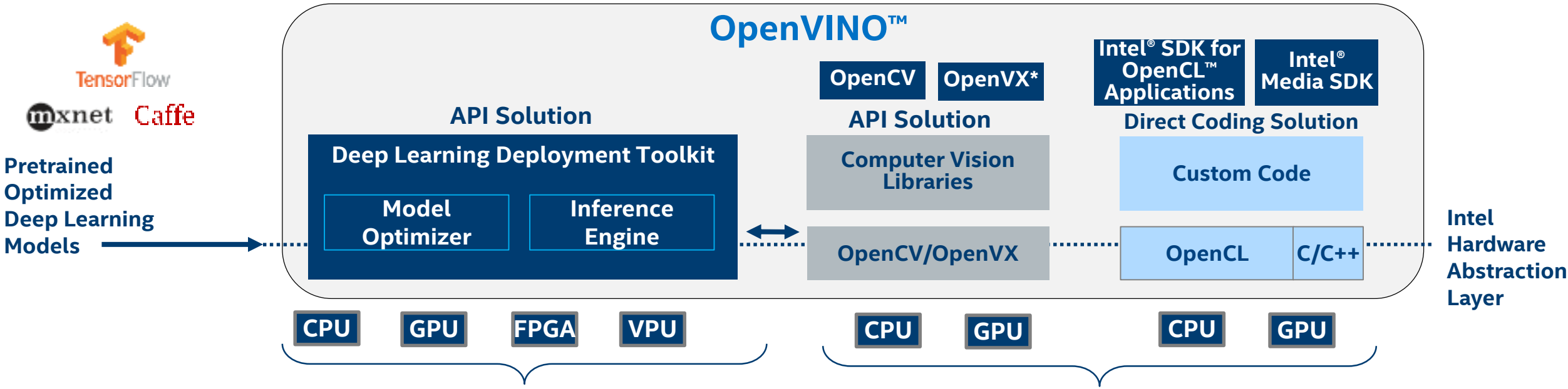
Use the increasing repository of OpenCL™ starting points in OpenCV to add your own unique code.

[1]10x performance increase comparing certain standard framework models vs. Intel-optimized models in the Intel®Deep Learning Deployment Toolkit. See Benchmarks slides.
Benchmark results were obtained prior to implementation of recent software patches and firmware updates intended to address exploits referred to as "Spectre" and "Meltdown". Implementation of these updates may make these results inapplicable to your device or system. For more complete information about performance and benchmark results, visit www.intel.com/benchmarks.

**Certain technical specifications and select processors/skus apply. See product site for more details.
OpenVX and the OpenVX logo are trademarks of the Khronos Group Inc.
OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission by Khronos

DEEP LEARNING VERSUS TRADITIONAL COMPUTER VISION

THE OPENVINO™ TOOLKIT HAS TOOLS FOR AN END-TO-END VISION PIPELINE



DEEP LEARNING COMPUTER VISION

- Based on application of a large number of filters to an image to extract features
- Features in the object(s) are analyzed with the goal of associating each input image with an output node for each type of object
- Values are assigned to output node representing the probability that the image is the object associated with the output node

TRADITIONAL COMPUTER VISION

- Based on selection and connections of computational filters to abstract key features and correlating them to an object
- Works well with well-defined objects and controlled scenes
- Difficult to predict critical features in larger number of objects or varying scenes

OpenVX and the OpenVX logo are trademarks of the Khronos Group Inc.
OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission by Khronos

OPENVINO™ TOOLKIT—USAGE MODEL

Train

Train a DL model.
Currently supports:

- Caffe*
- Apache Mxnet*
- TensorFlow*

TensorFlow

Caffe

mxnet

Prepare Optimize

Model optimizer:

- Converting
- Optimizing
- Preparing to inference

(device agnostic, generic optimization)

.prototxt
.caffemodel

Run Model Optimizer

IR

.xml
.bin

Inference

Inference engine lightweight API to use in applications for inference.

User Application

Inference Engine

Optimize/ Heterogeneous

Inference engine supports multiple devices for heterogeneous flows.

(device-level optimization)

MKL-DNN → CPU: Intel® Xeon®/Intel® Core™/Intel Atom® processors

cl-DNN → GPU

DLA → FPGA

Intel® Movidius™ API → Intel Movidius Myriad™ 2/X

Extend

Inference engine supports extensibility and allows custom kernels for various devices.

- Extensibility C++
- Extensibility OpenCL™
- Extensibility OpenCL/TBD
- Extensibility TBD

A CLOSER LOOK AT THE MODEL OPTIMIZER AND INFERENCE ENGINE

MODEL OPTIMIZER (MO)

PREREQUISITE—TRAIN A MODEL

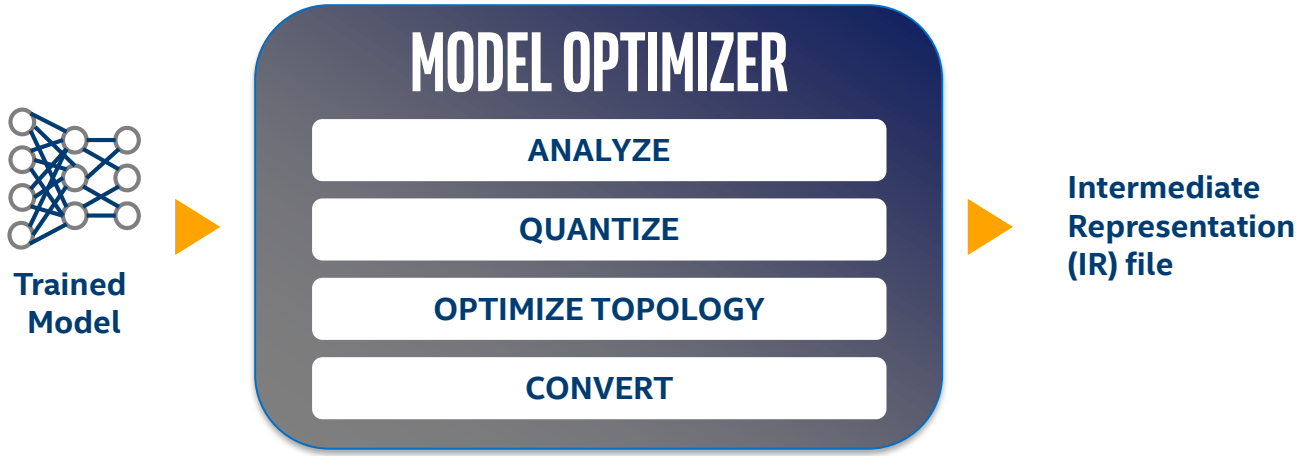
1. A trained model is the input to the Model Optimizer (MO)

2. Use the frozen graph (.pb file) from the Stolen Cars model training as input

3. The MO provides to tools to convert a trained model to a frozen graph in the event it is not already done.

MODEL OPTIMIZER—CORE FUNCTIONS

IMPROVE PERFORMANCE WITH MODEL OPTIMIZER



- Easy to use, Python*-based workflow does not require rebuilding frameworks
- Import Models from various frameworks (Caffe*, TensorFlow*, Apache MXNet*, more are planned…)
- More than 100 models for Caffe, MXNet and TensorFlow validated
- Generate IR files for models using standard layers or user-provided custom layers
- Fallback to original framework is possible in cases of unsupported layers (requires original framework)

MODEL ANALYSIS—PROCESSING STANDARD LAYERS

- **To generate IR files, the MO must recognize the layers in the model**
- **Some layers are standard across frameworks and neural network topologies**
  - Example – Convolution, Pooling, Activation, and so on
- **MO can easily generate the IR representation for these layers**
- **Framework specific instructions to use the MO:**
  - TensorFlow*: https://software.intel.com/en-us/articles/OpenVINO-Using-TensorFlow
  - Apache MXNet*: https://software.intel.com/en-us/articles/OpenVINO-Using-MXNet
  - Caffe*: https://software.intel.com/en-us/articles/OpenVINO-Using-Caffe

MODEL ANALYSIS—PROCESSING CUSTOM LAYERS (OPTIONAL)

- Custom layers are layers not included in the list of layers known to MO
- The OpenVINO™ toolkit provides three ways to process custom layers:
  - Register the custom layers as extensions to the Model Optimizer
  - Sub-graph replacement in the MO
  - Registering definite sub-graphs of the model as those that should be offloaded to TensorFlow* during inference
  - See Chapter 6 for details

QUANTIZATION SUPPORT FOR DIFFERENT HARDWARE CHOICES

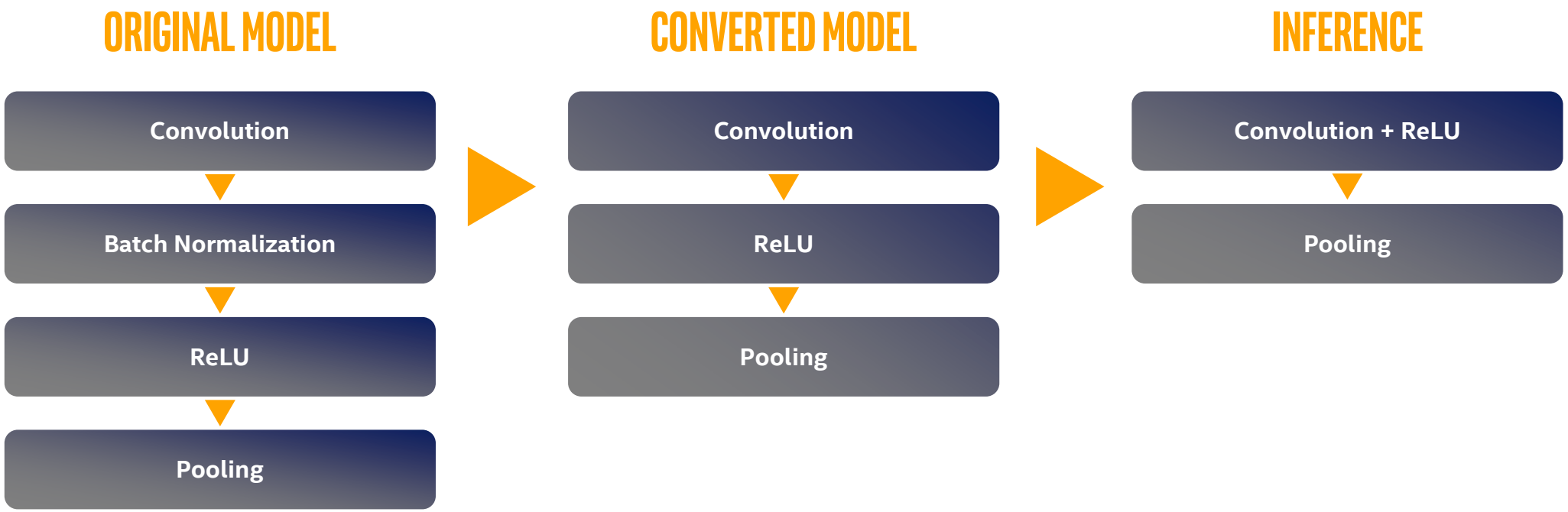
	FP32	FP16
CPU	YES	NO
GPU	YES	RECOMMENDED
MYRIAD	NO	YES
FPGA/DLA	NO	YES

- Model Optimizer creates FP32 and FP16 intermediate representations
- Indicate required precision at the Model Optimizer command line

IMPROVE PERFORMANCE WITH MODEL OPTIMIZER

EXAMPLE

1. Remove Batch Normalization stage.
2. Recalculate the weights to 'include' the operation.
3. Merge Convolution and ReLU into one optimized kernel.



OTHER OPTIMIZATION FUNCTIONS

Model Optimizer performs generic optimization:

- Node merging
- Horizontal fusion
- Batch normalization to scale shift
- Fold scale shift with convolution
- Drop unused layers (dropout)

Model Optimizer can cut out a portion of the network:

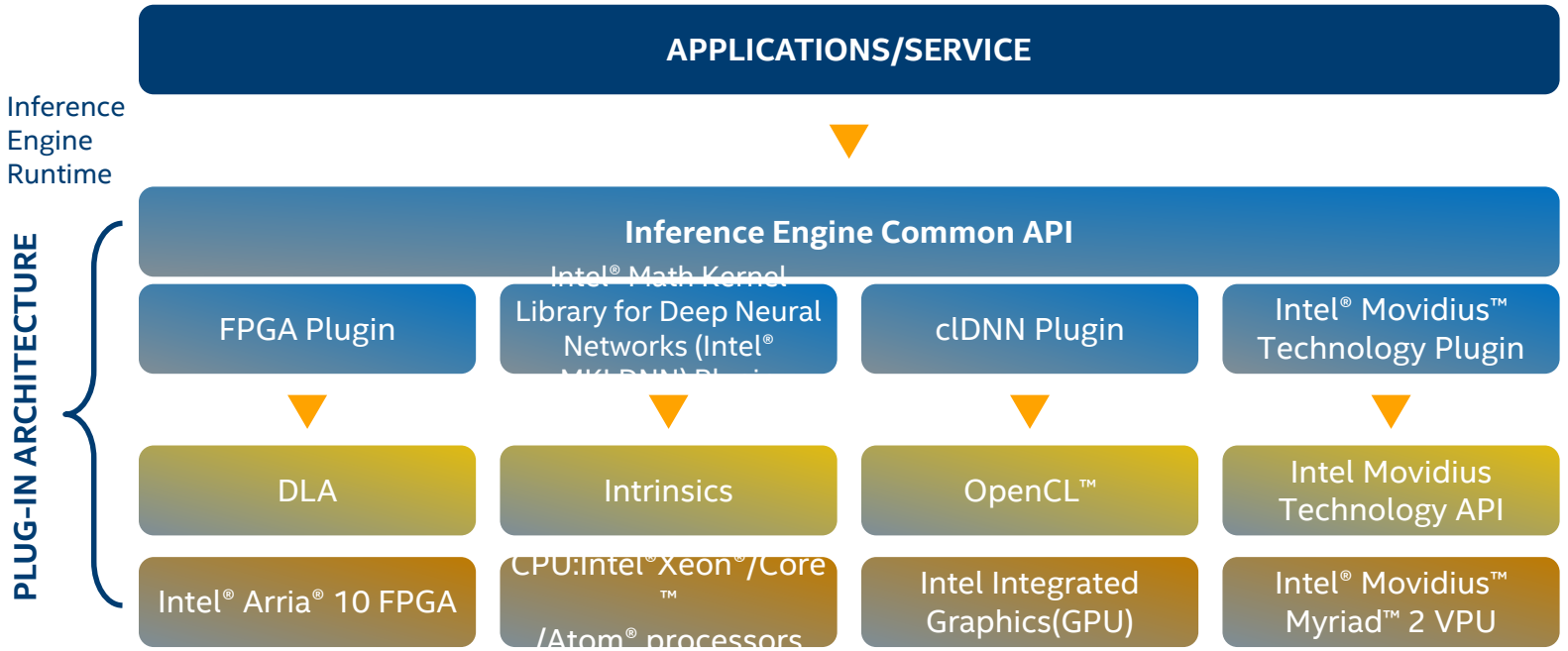
- Model has pre/post-processing parts that cannot be mapped to existing layers
- Model has a training part that is not used during inference
- Model is too complex and cannot be converted in one shot

INFERENCE ENGINE AND SUPPORT FOR CPU/INTEGRATED GRAPHICS/INTEL® MOVIDIUS™ TECHNOLOGY/FPGA

OPTIMAL MODEL PERFORMANCE USING THE INFERENCE ENGINE

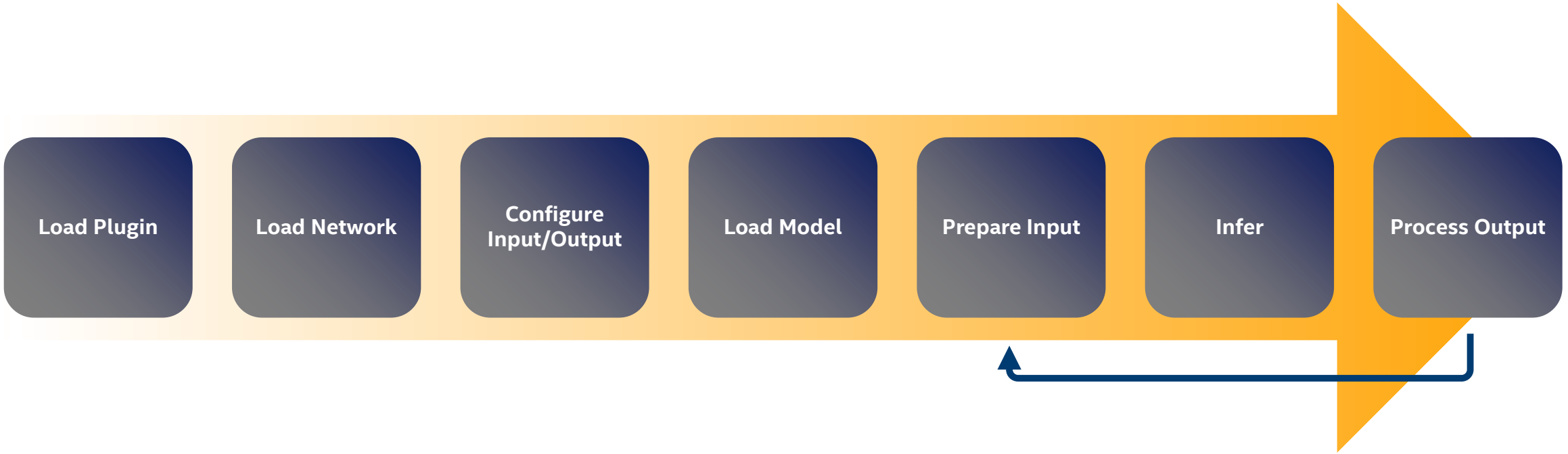
TRANSFORM MODELS & DATA INTO RESULTS & INTELLIGENCE

- Simple and Unified API for Inference across all Intel® architecture (IA)
- Optimized inference on large IA hardware targets (CPU/iGPU/FPGA)
- Heterogeneity support allows execution of layers across hardware types
- Asynchronous execution improves performance
- Futureproof/scale your development for future Intel® processors
- C++/ Python* support



OpenVX and the OpenVX logo are trademarks of the Khronos Group Inc.
OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission by Khronos

OPENVINO™ TOOLKIT APP EXECUTION FLOW



LAYERS SUPPORTED BY INFERENCE ENGINE PLUGINS

- **CPU – Intel® Math Kernel Library for Deep Neural Networks (Intel® MKL-DNN) Plugin**
  - Supports FP32, INT8 (planned)
  - Supports Intel® Xeon®/Intel® Core™/Intel Atom® processor platforms (https://github.com/01org/mkl-dnn)
- **GPU – clDNN Plugin**
  - Supports FP32 and FP16 (recommended for most topologies)
  - Supports Gen9 and later graphics architectures (https://github.com/01org/clDNN)
- **FPGA – DLA Plugin**
  - Supports Intel® Arria® 10
  - FP16 data types, FP11 is coming
- **Intel® Movidius™ Neural Compute Stick– Intel® Movidius™ Myriad™ VPU Plugin**
  - Set of layers are supported on Intel® Movidius™ Myriad™ X (28 layers); non-supported layers must be inferred through other inference engine (IE) plugins. Supports FP16

Layer Type	CPU	FPGA	GPU	MyriadX
Convolution	Yes	Yes	Yes	Yes
Fully Connected	Yes	Yes	Yes	Yes
Deconvolution	Yes	Yes	Yes	Yes
Pooling	Yes	Yes	Yes	Yes
ROI Pooling	Yes		Yes	
ReLU	Yes	Yes	Yes	Yes
PReLU	Yes		Yes	Yes
Sigmoid			Yes	Yes
Tanh			Yes	Yes
Clamp	Yes		Yes	
LRN	Yes	Yes	Yes	Yes
Normalize	Yes		Yes	Yes
Mul & Add	Yes		Yes	Yes
Scale & Bias	Yes	Yes	Yes	Yes
Batch Normalization	Yes		Yes	Yes
SoftMax	Yes		Yes	Yes
Split	Yes		Yes	Yes
Concat	Yes	Yes	Yes	Yes
Flatten	Yes		Yes	Yes
Reshape	Yes		Yes	Yes
Crop	Yes		Yes	Yes
Mul	Yes		Yes	Yes
Add	Yes	Yes	Yes	Yes
Permute	Yes		Yes	Yes
PriorBox	Yes		Yes	Yes
SimplerNMS	Yes		Yes	
Detection Output	Yes		Yes	Yes
Memory / Delay Object	Yes			
Tile	Yes			Yes

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

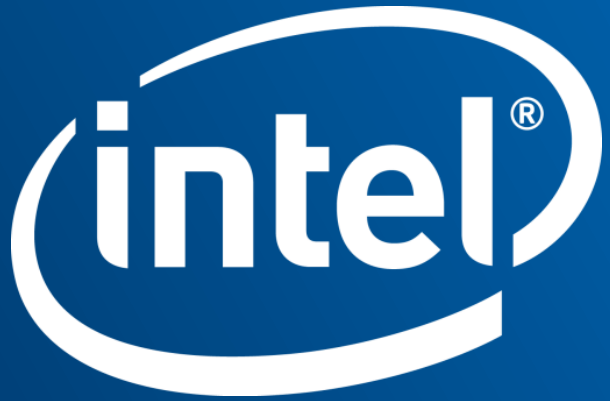
intel

OPENVINO™ TOOLKIT—INSTALLATION GUIDES

The Intel® Distribution of OpenVINO™ toolkit supports:

- Linux* : Installation Guide
- Windows* : Installation Guide
- MacOS* : Installation Guide

This course focuses on the workflow for Linux. For Windows, refer to the AI on the PC course.


intel