INNOVATE VISION SOLUTIONS

WITH INTEL® DISTRIBUTION OF OPENVINO™ TOOLKIT

(OPEN VISUAL INFERENCE & NEURAL NETWORK OPTIMIZATION)

Intel AI Workshop – CERN – May, 8th 2019
francisco.perez@intel.com

INTEL® DISTRIBUTION OF OPENVINO™ TOOLKIT

Take your computer vision solutions to a new level with deep learning inference intelligence.

What it is

A toolkit to accelerate development of **high performance computer vision** & **deep learning inference into vision/AI applications** used from device to cloud. It enables deep learning on hardware accelerators and easy deployment across multiple types of Intel® platforms.

Who needs this product?

- Computer vision, deep learning software developers
- Data scientists
- OEMs, ISVs, System Integrators

Usages

Security surveillance, robotics, retail, healthcare, AI, office automation, transportation, non-vision use cases (speech, text) & more.

HIGH PERFORMANCE, PERFORM AI AT THE EDGE

STREAMLINED & OPTIMIZED DEEP LEARNING INFERENCE

HETEROGENEOUS, CROSS-PLATFORM FLEXIBILITY

Free Download ▸ software.intel.com/openvino-toolkit

Open Source version ▸ 01.org/openvinotoolkit

Optimization Notice

Latest version is 2019 R1

Benefits of Intel® Distribution of OpenVINO™ toolkit

Maximize the Power of Intel® Processors: CPU, GPU/Intel® Processor Graphics, FPGA,VPU

ACCELERATE PERFORMANCE

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

INTEGRATE DEEP LEARNING

Unleash CNN-based deep learning inference using a common API, 30+ pre-trained models, & computer vision algorithms. Validated on more than 100 public/custom models.

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 OpenCL™ kernels/tools to add your own unique code. Customize layers without the overhead of frameworks.

[1] Tractica 2Q 2017

What's Inside Intel® Distribution of OpenVINO™ toolkit

Intel® Deep Learning Deployment Toolkit

Model Optimizer
Convert & Optimize

→ IR →

Inference Engine
Optimized Inference

Open Model Zoo
(30+ Pre-trained Models)

Samples

IR = Intermediate Representation file

Traditional Computer Vision

Optimized Libraries & Code Samples

OpenCV* | **OpenVX*** | **Samples**

For Intel® CPU & GPU/Intel® Processor Graphics

Tools & Libraries

Increase Media/Video/Graphics Performance

Intel® Media SDK
Open Source version

OpenCL™
Drivers & Runtimes

For GPU/Intel® Processor Graphics

Optimize Intel® FPGA (Linux* only)

FPGA RunTime Environment
(from Intel® FPGA SDK for OpenCL™)

Bitstreams

OS Support: CentOS* 7.4 (64 bit), Ubuntu* 16.04.3 LTS (64 bit), Microsoft Windows* 10 (64 bit), Yocto Project* version Poky Jethro v2.0.3 (64 bit), macOS* 10.13 & 10.14 (64 bit)

Intel® Architecture-Based
Platforms Support

intel CELERON inside | intel ATOM inside | intel CORE inside | intel XEON inside | intel ARRIA 10 inside | intel MOVIDIUS inside | intel IRIS Pro GRAPHICS

Intel® Vision Accelerator
Design Products &
AI in Production/
Developer Kits

An open source version is available at 01.org/openvinotoolkit (some deep learning functions support Intel CPU/GPU only).

Optimization Notice

Intel® Deep Learning Deployment Toolkit

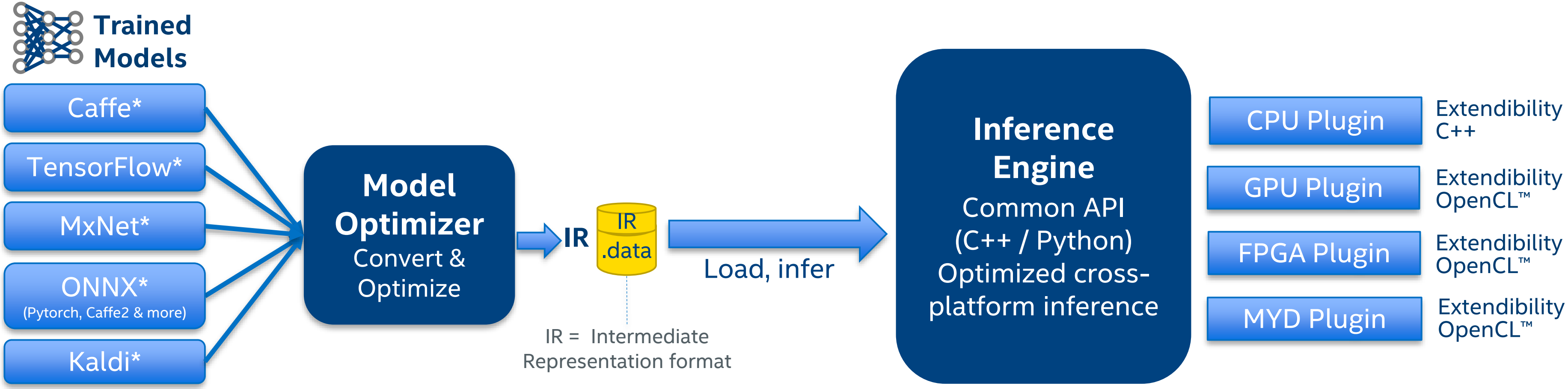
For Deep Learning Inference

Model Optimizer

- **What it is**: A Python*-based tool to import trained models and convert them to Intermediate representation.
- **Why important**: Optimizes for performance/space with conservative topology transformations; biggest boost is from conversion to data types matching hardware.

Inference Engine

- **What it is**: High-level inference API
- **Why important**: Interface is implemented as dynamically loaded plugins for each hardware type. Delivers best performance for each type without requiring users to implement and maintain multiple code pathways.

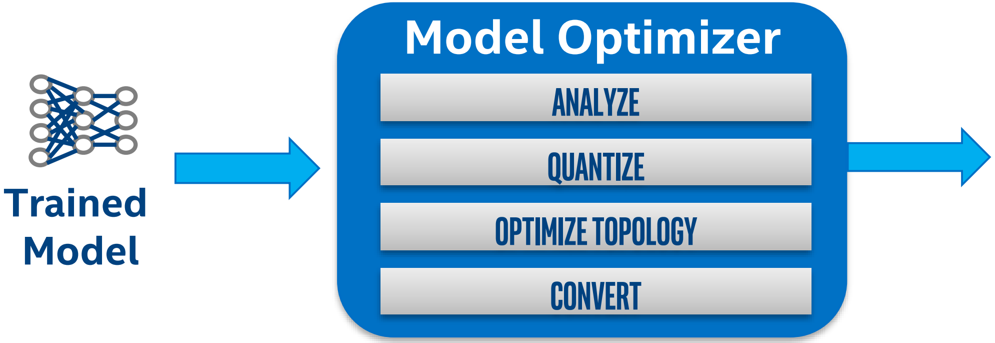


IR = Intermediate Representation format

GPU = Intel CPU with integrated graphics processing unit/Intel® Processor Graphics

Optimization Notice

Improve Performance with Model Optimizer

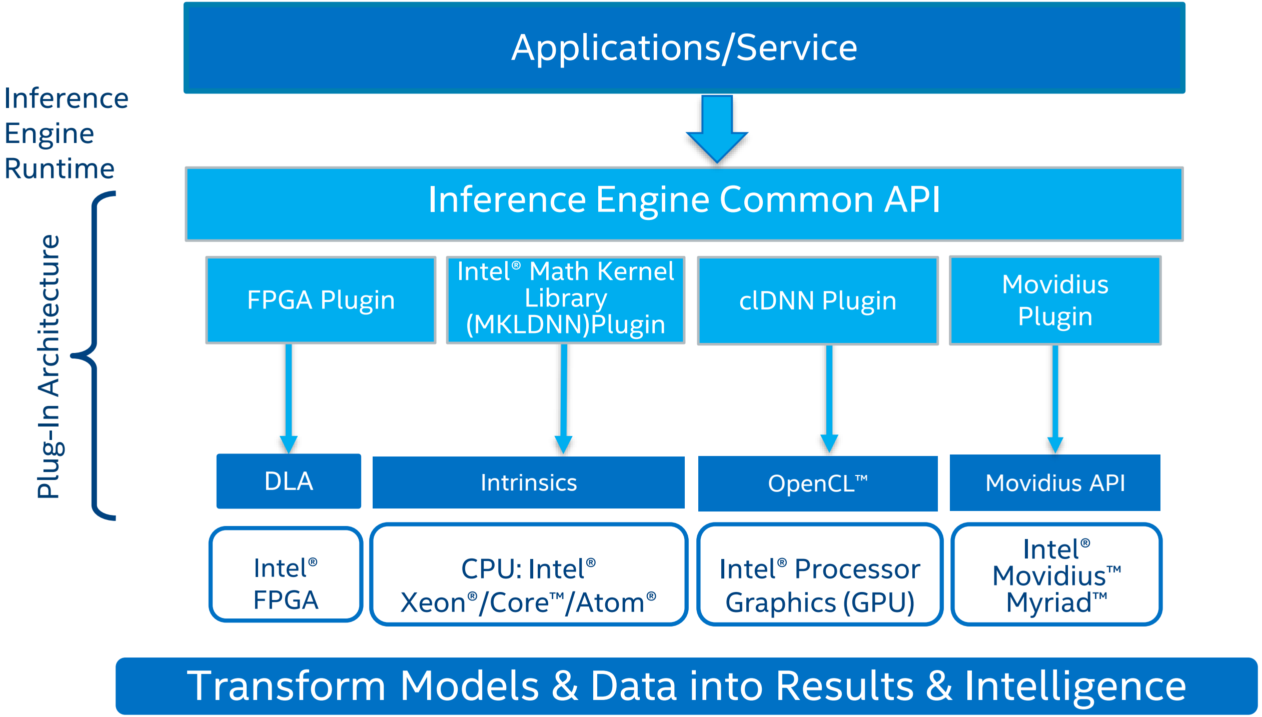


Intermediate Representation (IR) file

- Easy to use, Python*-based workflow
- Import Models from many supported frameworks: Caffe*, TensorFlow*, MXNet*, Kaldi*, exchange formats like ONNX* (Pytorch*, Caffe2* and others through ONNX).
- 100+ models for Caffe, MXNet, TensorFlow validated. Supports all ONNX* model zoo public models.
- Extends inferencing for non-vision networks with support of LSTM and 3D Convolutional based networks and Kaldi framework/Kaldi Nnet2*.

Optimal Model Performance Using the Inference Engine

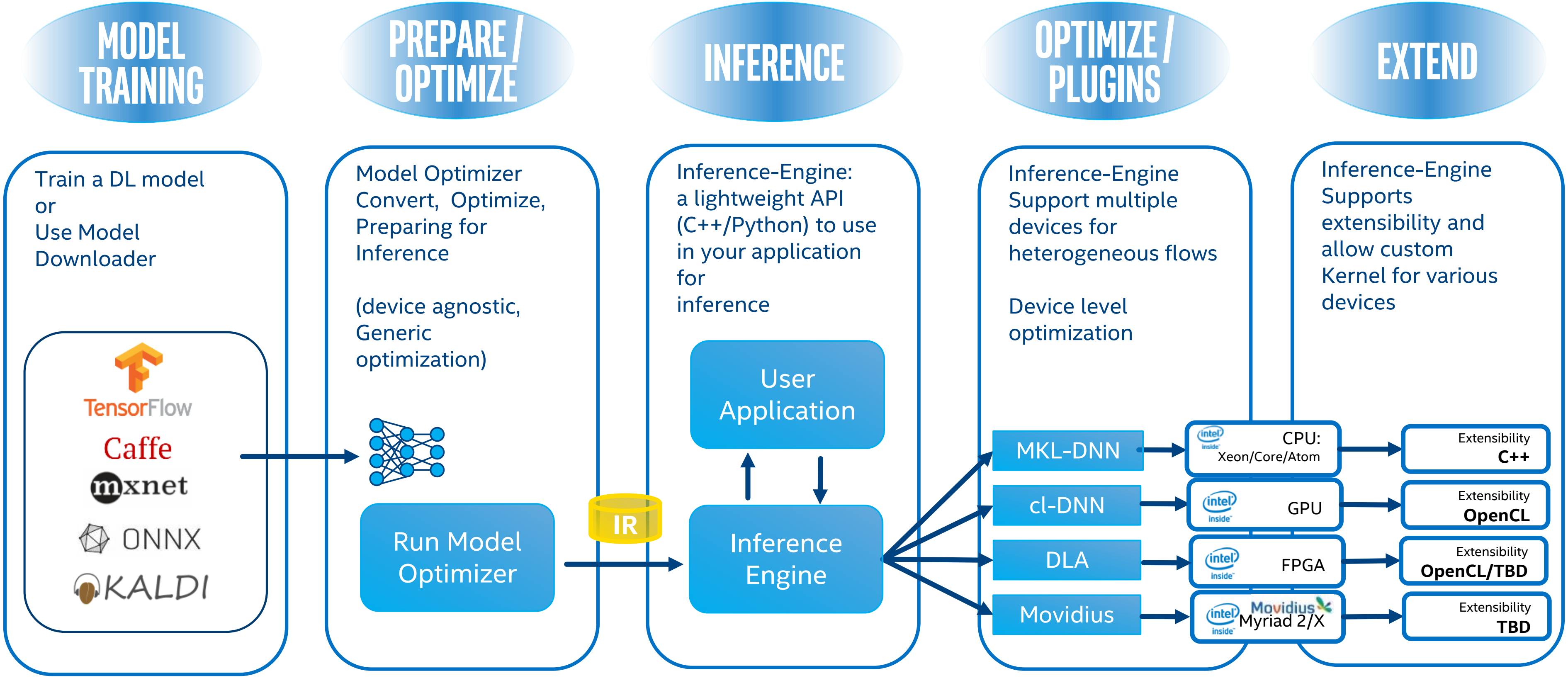
- Simple & unified API for inference across all Intel® architecture
- Optimized inference on large IA hardware targets (CPU/GEN/FPGA/MYD)
- Heterogeneity support allows execution of layers across hardware types
- Asynchronous execution improves performance
- Futureproof/scale your development for future Intel® processors


Applications/Service
Inference Engine Runtime
Inference Engine Common API
Plug-In Architecture
FPGA Plugin
Intel® Math Kernel Library (MKLDNN)Plugin
clDNN Plugin
Movidius Plugin
DLA
Intrinsics
OpenCL™
Movidius API
Intel® FPGA
CPU: Intel® Xeon®/Core™/Atom®
Intel® Processor Graphics (GPU)
Intel® Movidius™ Myriad™
Transform Models & Data into Results & Intelligence

GPU = Intel CPU with integrated graphics/Intel® Processor Graphics/GEN

Optimization Notice

Computer Vision Application Pipeline



MODEL TRAINING

Train a DL model
or
Use Model Downloader

PREPARE / OPTIMIZE

Model Optimizer
Convert, Optimize, Preparing for Inference

(device agnostic, Generic optimization)

Run Model Optimizer

INFERENCE

Inference-Engine:
a lightweight API (C++/Python) to use in your application for
inference

User Application

Inference Engine

OPTIMIZE / PLUGINS

Inference-Engine
Support multiple devices for heterogeneous flows

Device level optimization

MKL-DNN

cl-DNN

DLA

Movidius

CPU: Xeon/Core/Atom

GPU

FPGA

Movidius Myriad 2/X

EXTEND

Inference-Engine
Supports extensibility and allow custom Kernel for various devices

Extensibility
C++

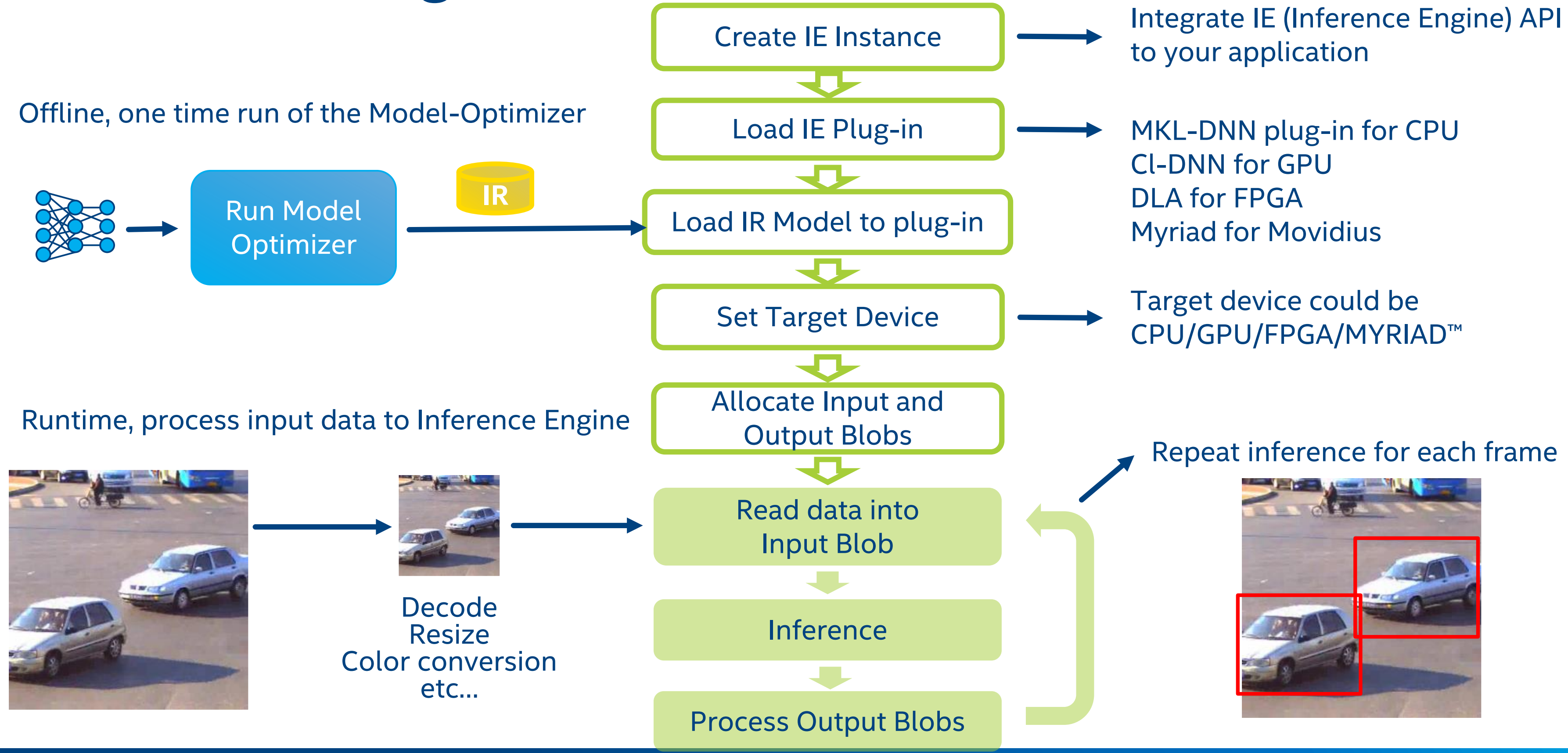
Extensibility
OpenCL

Extensibility
OpenCL/TBD

Extensibility
TBD

Optimization Notice

Inference engine – workflow



Offline, one time run of the Model-Optimizer

Run Model Optimizer

IR

Create IE Instance → Integrate IE (Inference Engine) API to your application

Load IE Plug-in → MKL-DNN plug-in for CPU
Cl-DNN for GPU
DLA for FPGA
Myriad for Movidius

Load IR Model to plug-in

Set Target Device → Target device could be CPU/GPU/FPGA/MYRIAD™

Allocate Input and Output Blobs

Runtime, process input data to Inference Engine

Decode
Resize
Color conversion
etc…

Read data into Input Blob

Inference

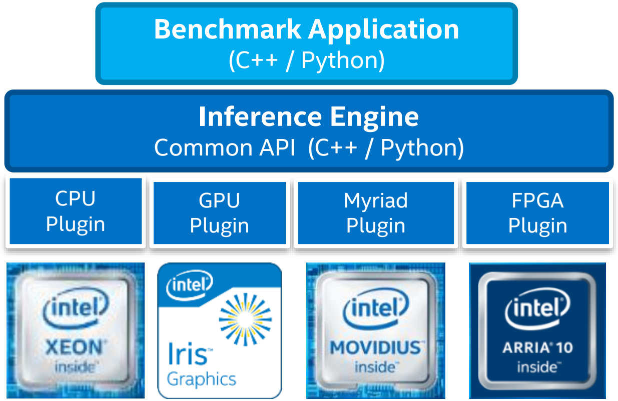
Process Output Blobs

Repeat inference for each frame

Enabling multiple accelerators with openVINO

```
#define MKLDNN  "MKLDNNPlugin.dll"
#define CLDNN   "clDNNPlugin.dll"
#define HDDLDNN "HDDLPlugin.dll"
#define MYXDNN  "myraidPlugin.dll"
#define FPGADNN "dliaPlugin.dll"
#else
#define MKLDNN  "libMKLDNNPlugin.so"
#define CLDNN   "libclDNNPlugin.so"
#define HDDLDNN "libHDDLPlugin.so"
#define MYXDNN  "libmyraidplugin.so"
#define FPGADNN "libdliaplugin.so"
#endif

if (dev == "cpu" )
  {
   plugin    = InferenceEngine::InferenceEnginePluginPtr(MKLDNN );
   CPUplugin = InferenceEngine::InferencePlugin(plugin);
   CPUplugin.AddExtension(std::make_shared<Extensions::Cpu::CpuExtensions>());
  }
else if (dev == "gpu" )
  plugin = InferenceEngine::InferenceEnginePluginPtr(CLDNN  );
else if (dev == "myx" )
  plugin = InferenceEngine::InferenceEnginePluginPtr(MYXDNN);
else if (dev == "fpga" )
  plugin = InferenceEngine::InferenceEnginePluginPtr(FPGADNN);
else
  {
   std::cout << "Unrecognized device : " << dev << std::endl;
   std::cout << "This is very unlikely to end well." << std::endl;
  }
```



Benchmark Application
(C++ / Python)

Inference Engine
Common API (C++ / Python)

CPU Plugin

GPU Plugin

Myriad Plugin

FPGA Plugin

Speed Deployment with Pre-trained Models & Samples

Pretrained Models in Intel® Distribution of OpenVINO™ toolkit

- Age & Gender
- Face Detection–standard & enhanced
- Head Position
- Human Detection–eye-level & high-angle detection
- Detect People, Vehicles & Bikes
- License Plate Detection: small & front facing
- Vehicle Metadata
- Human Pose Estimation
- Action recognition – encoder & decoder

- Text Detection & Recognition
- Vehicle Detection
- Retail Environment
- Pedestrian Detection
- Pedestrian & Vehicle Detection
- Person Attributes Recognition Crossroad
- Emotion Recognition
- Identify Someone from Different Videos–standard & enhanced
- Facial Landmarks
- Gaze estimation

- Identify Roadside objects
- Advanced Roadside Identification
- Person Detection & Action Recognition
- Person Re-identification–ultra small/ultra fast
- Face Re-identification
- Landmarks Regression
- Smart Classroom Use Cases
- Single image Super Resolution (3 models)
- Instance segmentation
- and more…

Binary Models

- Face Detection Binary
- Pedestrian Detection Binary

- Vehicle Detection Binary

- ResNet50 Binary

Optimization Notice

Save Time with Deep Learning Samples

Use Model Optimizer & Inference Engine for public models & Intel pretrained models

- Object Detection
- Standard & Pipelined Image Classification
- Security Barrier
- Object Detection SSD
- Neural Style Transfer
- Object Detection for Single Shot Multibox Detector using Asynch API+

- Hello Infer Classification
- Interactive Face Detection
- Image Segmentation
- Validation Application
- Multi-channel Face Detection

OpenVINO™ Toolkit

Open Source Version

- Provides flexibility and availability to the developer community to extend OpenVINO™ toolkit for custom needs
- Components that are open sourced
  - **Deep Learning Deployment Toolkit** with **CPU, GPU & Heterogeneous** plugins github.com/opencv/dldt
  - **Open Model Zoo** - Includes pre-trained models, model downloader, demos and samples: github.com/opencv/open_model_zoo
- See [FAQ]() and next slide for key differences between the open source and Intel distribution

DESCRIPTION

OpenVINO™ toolkit, short for Open Visual Inference and Neural network Optimization toolkit, provides developers with improved neural network performance on a variety of Intel® processors and helps them further unlock cost-effective, real-time vision applications.The toolkit enables deep learning inference and easy heterogeneous execution across multiple Intel® platforms (CPU, Intel® Processor Graphics)—providing implementations across cloud architectures to edge devices. This open source distribution provides flexibility and availability to the developer community to innovate deep learning and AI solutions.

OpenVINO™ toolkit contains:

- Deep Learning Deployment Toolkit
- Open Model Zoo

The **Intel® Distribution of OpenVINO™ toolkit** is also available with additional, proprietary support for Intel® FPGAs, Intel®

Learn More ▶ 01.org/openvinotoolkit

Quick Guide: What’s Inside the Intel Distribution vs Open Source version of OpenVINO™ toolkit

Tool/Component	Intel® Distribution of OpenVINO™ toolkit	OpenVINO™ toolkit (open source)	Open Source Directory https://github.com
Installer (including necessary drivers)	✓		
Intel® Deep Learning Deployment toolkit			
Model Optimizer	✓	✓	/opencv/dldt/tree/2018/model-optimizer
Inference Engine	✓	✓	/opencv/dldt/tree/2018/inference-engine
Intel CPU plug-in	✓ Intel® Math Kernel Library (Intel® MKL) only[1]	✓ BLAS, Intel® MKL[1], jit (Intel MKL)	/opencv/dldt/tree/2018/inference-engine
Intel GPU (Intel® Processor Graphics) plug-in	✓	✓	/opencv/dldt/tree/2018/inference-engine
Heterogeneous plug-in	✓	✓	/opencv/dldt/tree/2018/inference-engine
Intel GNA plug-in	✓		
Intel® FPGA plug-in	✓		
Intel® Neural Compute Stick (1 & 2) VPU plug-in	✓		
Intel® Vision Accelerator based on Movidius plug-in	✓		
30+ Pretrained Models - incl. Model Zoo (IR models that run in IE + open sources models)	✓	✓	/opencv/open_model_zoo
Samples (APIs)	✓	✓	/opencv/dldt/tree/2018/inference-engine
Demos	✓	✓	/opencv/open_model_zoo
Traditional Computer Vision			
OpenCV*	✓	✓	/opencv/opencv
OpenVX (with samples)	✓		
Intel® Media SDK	✓	✓[2]	/Intel-Media-SDK/MediaSDK
OpenCL™ Drivers & Runtimes	✓	✓[2]	/intel/compute-runtime
FPGA RunTime Environment, Deep Learning Acceleration & Bitstreams (Linux* only)	✓		

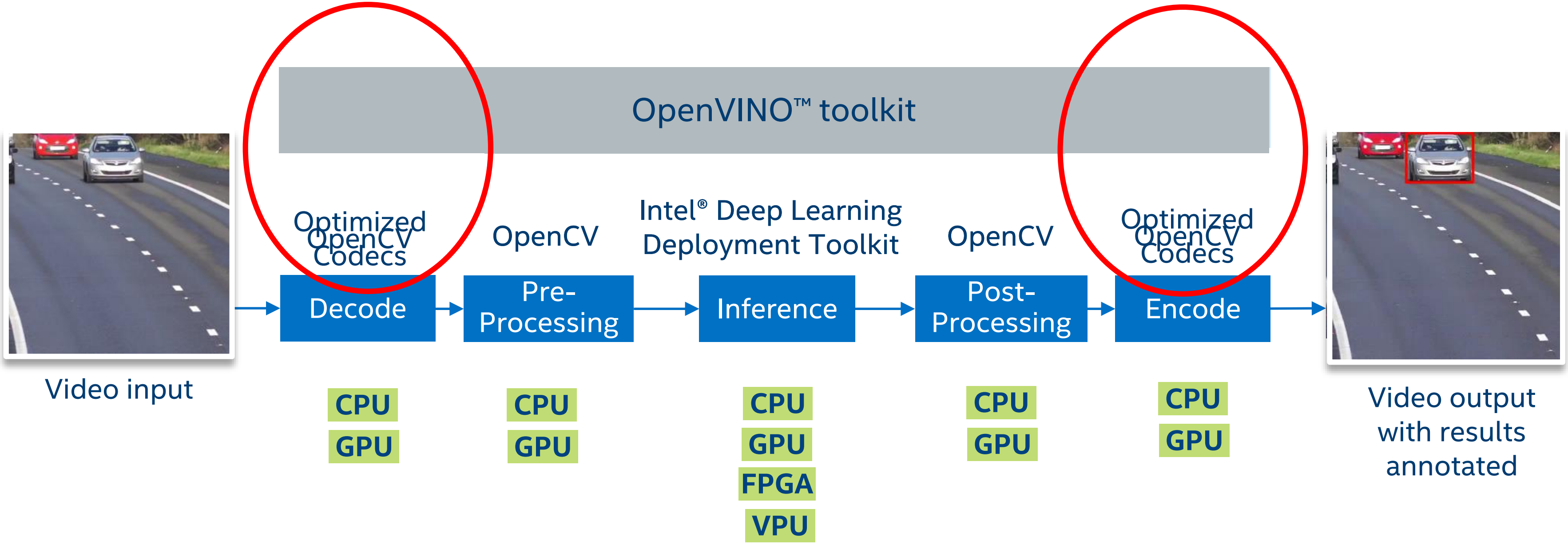
Optimization Notice

[1] Intel MKL is not open source but does provide the best performance
[2] Refer to readme file for validated versions

END TO END VIDEO PIPELINE

Media SDK
HW Accelerators

End-to-End Vision Workflow



OpenVINO™ toolkit

Video input

Optimized OpenCV Codecs

Decode

CPU
GPU

OpenCV

Pre-Processing

CPU
GPU

Intel® Deep Learning Deployment Toolkit

Inference

CPU
GPU
FPGA
VPU

OpenCV

Post-Processing

CPU
GPU

Optimized OpenCV Codecs

Encode

CPU
GPU

Video output with results annotated

Intel® Media SDK for Linux Overview

Included in OpenVINO installation. Available as standalone tool FREE Download

What it is:

An API to access Intel® Quick Sync Video hardware-accelerated encode/decode & processing

Optimized Industry Standard Video Codecs

- H.265 (HEVC), H.264 (AVC), MJPEG
- MPEG-2, VP9, VP8, VC1 & more

Video Pre & Post Processing

- Resize, Scale, Deinterlace
- Color Conversion, Composition, Alpha Blending
- Denoise, Sharpen & more

Benefits:

Boost media and video application performance with hardware-accelerated codecs & programmable graphics on Intel® processors.**

Improve video quality, innovate cloud graphics & media analytics.

Reduce infrastructure & development costs.

Hardware Support

Select Intel® Xeon®, Celeron®, Pentium®, and Intel Atom® processors that support Intel® Quick Sync Video

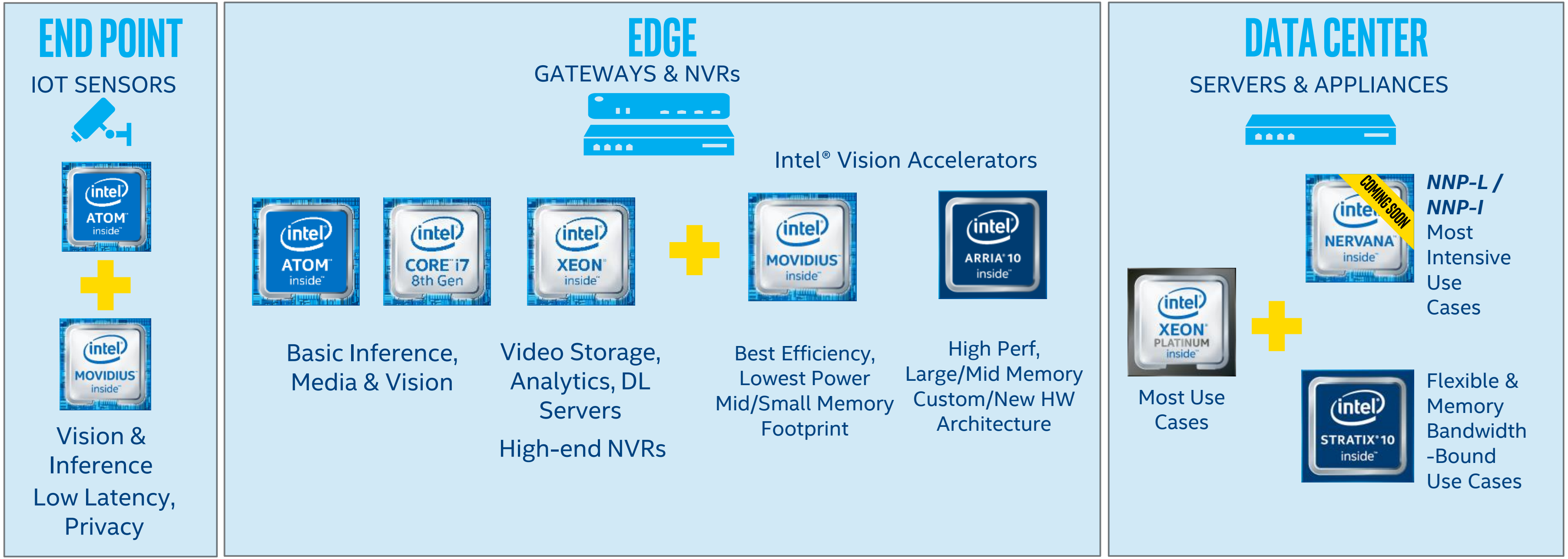
Use Cases

Media Creation & Delivery for Embedded Applications

Deliver fast, high quality video decoding / encoding / transcoding from camera to cloud

Intel Vision supports AI across endpoint, edge & cloud

typical devices by application



END POINT

IOT SENSORS

Vision & Inference

Low Latency, Privacy

EDGE

GATEWAYS & NVRs

Intel® Vision Accelerators

Basic Inference, Media & Vision

Video Storage, Analytics, DL Servers

High-end NVRs

Best Efficiency, Lowest Power Mid/Small Memory Footprint

High Perf, Large/Mid Memory Custom/New HW Architecture

DATA CENTER

SERVERS & APPLIANCES

Most Use Cases

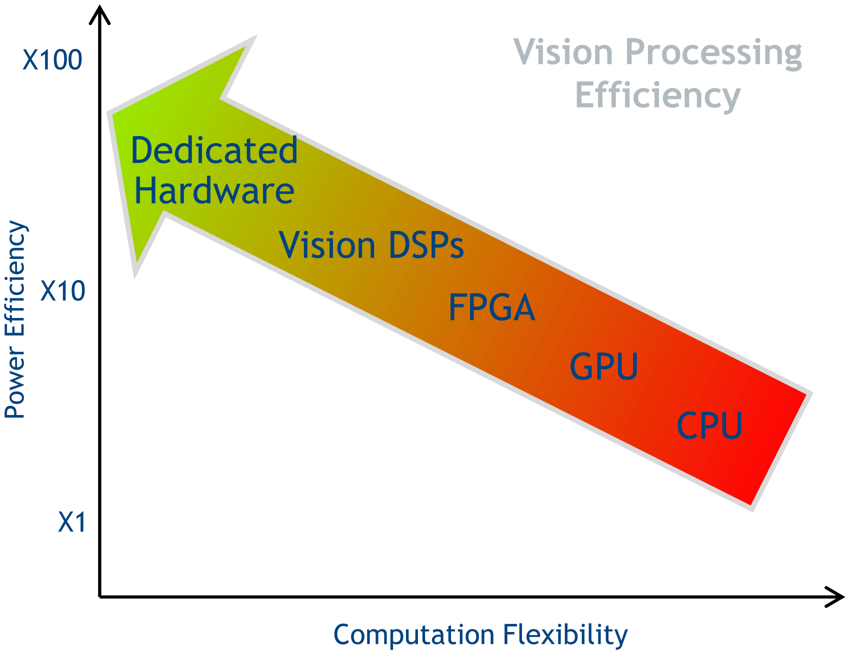
NNP-L / NNP-I Most Intensive Use Cases

Flexible & Memory Bandwidth -Bound Use Cases

Optimization Notice

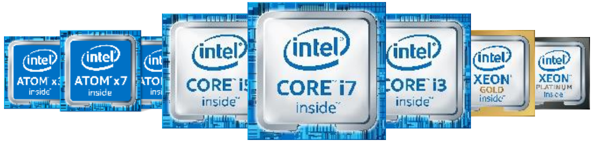
Choosing the “right” hardware

- Consider in each device
  - Compute efficiency
  - Compute parallelism (# of EU/Cores)
  - Power consumption
  - Memory hierarchy, size, communication
  - Programming model, APIs
- Trade offs
  - Power/ performance
  - Price
  - Software flexibility, portability



Optimization Notice

Intel® Vision Products Comparison



HOST IA PLATFORMS: APPLICATION PROCESSING, MEDIA, "FREE" CV/DL

Use the Intel® Media SDK to achieve en/de/trans-code performance

Maximize CV/DL performance on the host platform with the Open Visual Inference & Neural Network Optimization (OpenVINO™) toolkit

INTEL® MOVIDIUS™ VPUS

OVERVIEW

Intel Movidius VPUs offer high performance per watt per dollar.
Easily add AI-based visual intelligence by plugging in one or more cards.

VALUE PROP

Intel Movidius VPUs enable deep neural network inferencing workloads with high compute efficiency, low power and form factor constraints (e.g., cameras), and excellent performance/W/$, for well-defined workloads.

KEY USE CASES

Intel Movidius VPUs work well with networks that have:

- A small memory footprint (less than 250 MParameters)
- Lower performance requirements (<3 GMACs)
- Accelerator Power Budget: 2-25W

INTEL® FPGAS

OVERVIEW

Intel FPGAs offer exceptional performance, flexibility, and scalability for NVRs, edge deep learning inference appliances, and on-premise servers or cloud.

VALUE PROP

Intel FPGAs achieve TOPS performance required on a single chip, support compute intensive networks (VGG*, ResNet* 101).

KEY USE CASES

The Intel Arria 10 FPGAs work well with networks that have:

- Larger memory footprint (more than 250 MParameters)
- Larger performance requirements (>3 GMACs)
- Accelerator Power Budget: <50W
- # of streams: 3-15

Optimization Notice

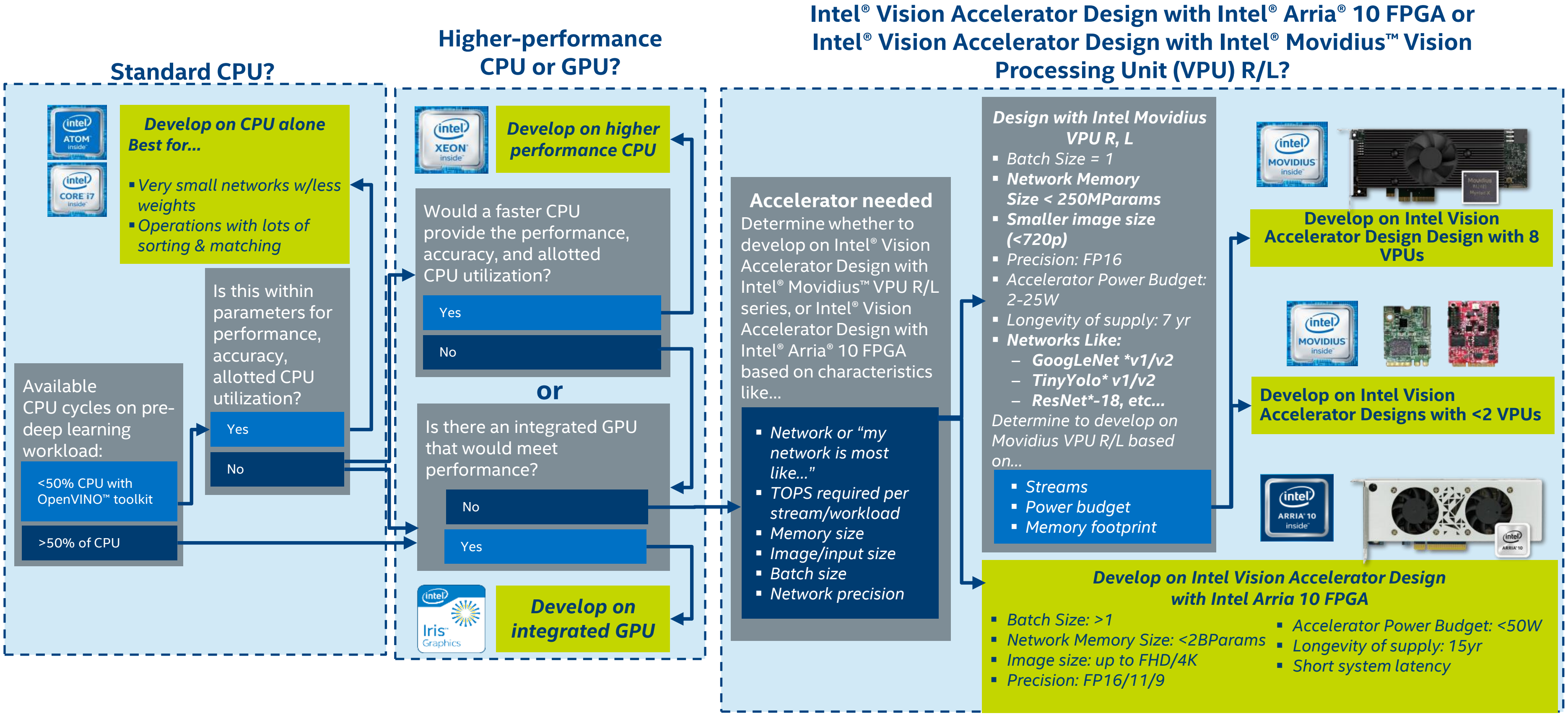
Examples of Intel® Vision Accelerator Products

	INTEL® VISION ACCELERATOR DESIGN WITH INTEL® MOVIDIUS™ VPU			INTEL® VISION ACCELERATOR DESIGN WITH INTEL® ARRIA® 10 FPGA
EXAMPLE CARD BASED ON VISION ACCELERATOR DESIGNS	1 Movidius MA2485 VPU	2 Movidius MA2485 VPUs	8 Movidius MA2485 VPUs	Intel® Arria® 10 FPGA 1150GX
INTERFACE	M.2, Key E	miniPCIe*	PCIe x4	PCIe x8
CURRENTLY MANUFACTURED BY*	iEi ADVANTECH AAEON ADLINK NEXCOM	iEi ADVANTECH ADLINK NEXCOM	iEi NEXCOM	iEi
SOFTWARE TOOLS	INTEL® DISTRIBUTION OF OPENVINO™ TOOLKIT Develop NN Model; Deploy across Intel® CPU, GPU, VPU, FPGA; Leverage common algorithms			

FUTURE

Optimization Notice

Deep Learning Inference Engine Decision Tree



Standard CPU?

Develop on CPU alone
Best for…

- *Very small networks w/less weights*
- *Operations with lots of sorting & matching*

Available CPU cycles on pre-deep learning workload:

- <50% CPU with OpenVINO™ toolkit
- >50% of CPU

Is this within parameters for performance, accuracy, allotted CPU utilization?

- Yes
- No

Higher-performance CPU or GPU?

Develop on higher performance CPU

Would a faster CPU provide the performance, accuracy, and allotted CPU utilization?

- Yes
- No

or

Is there an integrated GPU that would meet performance?

- No
- Yes

Develop on integrated GPU

Intel® Vision Accelerator Design with Intel® Arria® 10 FPGA or Intel® Vision Accelerator Design with Intel® Movidius™ Vision Processing Unit (VPU) R/L?

Accelerator needed

Determine whether to develop on Intel® Vision Accelerator Design with Intel® Movidius™ VPU R/L series, or Intel® Vision Accelerator Design with Intel® Arria® 10 FPGA based on characteristics like…

- *Network or "my network is most like…"*
- *TOPS required per stream/workload*
- *Memory size*
- *Image/input size*
- *Batch size*
- *Network precision*

Design with Intel Movidius VPU R, L

- *Batch Size = 1*
- ***Network Memory Size < 250MParams***
- ***Smaller image size (<720p)***
- *Precision: FP16*
- *Accelerator Power Budget: 2-25W*
- *Longevity of supply: 7 yr*
- ***Networks Like:***
  - ***GoogLeNet *v1/v2***
  - ***TinyYolo* v1/v2***
  - ***ResNet*-18, etc…***

Determine to develop on Movidius VPU R/L based on…

- *Streams*
- *Power budget*
- *Memory footprint*

Develop on Intel Vision Accelerator Design Design with 8 VPUs

Develop on Intel Vision Accelerator Designs with <2 VPUs

Develop on Intel Vision Accelerator Design with Intel Arria 10 FPGA

- *Batch Size: >1*
- *Network Memory Size: <2BParams*
- *Image size: up to FHD/4K*
- *Precision: FP16/11/9*
- *Accelerator Power Budget: <50W*
- *Longevity of supply: 15yr*
- *Short system latency*

INTEL® NEURAL COMPUTE STICK 2

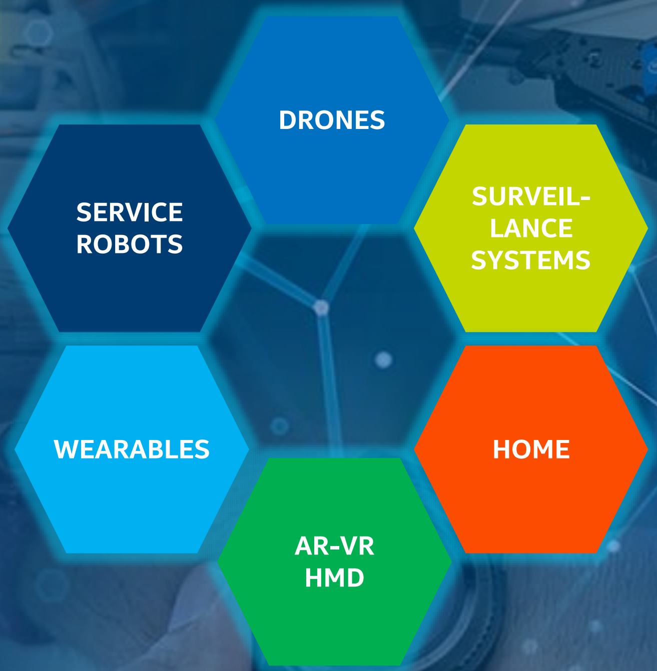
ACCELERATE DEEP LEARNING DEVELOPMENT FOR EDGE DEVICES

COMPUTER VISION AND ARTIFICIAL INTELLIGENCE ARE TRANSFORMING IOT DEVICES AT THE NETWORK EDGE

- Navigation
- 3D Vol. Mapping
- Multi-Modal Sensing

- Detection, Tracking
- Recognition
- Video, Image, Session Capture

- Position, Mapping
- Gaze, Eye Tracking
- Gesture Tracking, Recognition
- See through Camera



- Sense & Avoid
- GPS Denied Hovering
- Pixel Labeling
- Video, Image Capture

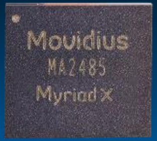
- Detection, Tracking
- Identification
- Classification
- Multi-Nodal Systems
- Multi-Modal Sensing
- Video, Image Capture

- Detection, Tracking
- Perimeter, Presence Monitoring
- Recognition, Classification
- Multi-Nodal Systems
- Multi-Modal Sensing
- Video, Image Capture

INTRODUCING
INTEL® NEURAL COMPUTE STICK 2

A Plug-and-Play Deep Learning Development Kit

POWERED BY



Intel® Movidius™ Myriad™ X VPU delivers industry leading performance

+

OpenVINO™

Intel® Distribution of OpenVINO™ toolkit accelerates solution development and streamlines deployment

DELIVERING UP TO 8X[1] HIGHER PERFORMANCE

On deep neural networks compared to Intel® Movidius™ Neural Compute Stick

INTEL® DISTRIBUTION OF OPENVINO™ TOOLKIT
SUPPORTED NETWORKS

View Documentation ▸ https://docs.openvinotoolkit.org/latest/_docs_IE_DG_supported_plugins_MYRIAD.html

Caffe

- AlexNet
- CaffeNet
- GoogleNet (Inception) v1, v2, v4
- VGG family (VGG16, VGG19)
- SqueezeNet v1.0, v1.1
- ResNet v1 family (18** ***, 50, 101, 152)
- MobileNet (mobilenet-v1-1.0-224, mobilenet-v2)
- Inception ResNet v2
- DenseNet family** (121,161,169,201)
- SSD-300, SSD-512, SSD-MobileNet, SSD-GoogleNet, SSD-SqueezeNet

TensorFlow

- AlexNet
- Inception v1, v2, v3, v4
- Inception ResNet v2
- MobileNet v1, v2
- ResNet v1 family (50, 101, 152)
- ResNet v2 family (50, 101, 152)
- SqueezeNet v1.0, v1.1
- VGG family (VGG16, VGG19)
- Yolo family (yolo-v2, yolo-v3, tiny-yolo-v1, tiny-yolo-v2, tiny-yolo-v3)
- faster_rcnn_inception_v2, faster_rcnn_resnet101
- ssd_mobilenet_v1
- DeepLab-v3+

mxnet

- AlexNet and CaffeNet
- DenseNet family** (121,161,169,201)
- SqueezeNet v1.1
- MobileNet v1, v2
- NiN
- ResNet v1 (101, 152)
- ResNet v2 (101)
- SqueezeNet v1.1
- VGG family (VGG16, VGG19)
- SSD-Inception-v3, SSD-MobileNet, SSD-ResNet-50, SSD-300

NOTE: Not an exhaustive list – only includes popular networks.

** Network is tested on Intel® Movidius™ Neural Compute Stick with BatchNormalization fusion optimization disabled during Model Optimizer import

*** Network is tested on Intel® Neural Compute Stick 2 with BatchNormalization fusion optimization disabled during Model Optimizer import

INTEL® DISTRIBUTION OF OPENVINO™ TOOLKIT
SUPPORTED LAYERS

View Documentation ▸ https://docs.openvinotoolkit.org/latest/_docs_IE_DG_supported_plugins_Supported_Devices.html

- Activation-Clamp
- Activation-ELU
- Activation-Leaky ReLU
- Active-PReLU
- Activation-ReLU
- Activation-ReLU6
- Activation-Sigmoid/Logistic
- Activation-TanH
- ArgMax
- BatchNormalization
- Concat
- Const
- Convolution-Dilated
- Convolution-Grouped
- Convolution-Ordinary
- Crop
- CTCGreedyDecoder*
- Deconvolution
- DetectionOutput*
- Eltwise-Max
- Eltwise-Mul
- Eltwise-Sum
- Flatten
- FullyConnected (Inner Product)
- GRN
- Interp
- LRN (Norm)
- MVN*
- Normalize*
- Pad*
- Permute
- Pooling(AVG,MAX)*
- Power
- PriorBox
- PriorBoxClustered
- Proposal
- PSROIPooling
- RegionYolo
- ReorgYolo
- Resample
- Reshape
- RNN
- ROIPooling
- ScaleShift*
- Slice
- SoftMax
- Split
- Tile

* Support is limited to the specific parameters. Refer to "Known Layers Limitation" section for the device from the list of supported.

Changed since last update

EXPEDITE DEVELOPMENT AND DEPLOYMENT
PRE-TRAINED MODELS

Description	Pre-trained Model	Supported Samples
Face detection for driver monitoring	face-detection-adas-0001	Interactive face detection
Age and gender recognition	age-gender-recognition-retail-0013	Interactive face detection
Emotion recognition for retail	emotions-recognition-retail-0003	Interactive face detection
License plate detector	vehicle-license-plate-detection-barrier-0106	Security barrier camera
Vehicle attributes recognition	vehicle-attributes-recognition-barrier-0039	Security barrier camera
License plate recognition	license-plate-recognition-barrier-0001	Security barrier camera
Person, vehicle and bike detection	person-vehicle-bike-detection-crossroad-0078	Crossroad camera
Person re-identification	person-reidentification-retail-0076	Crossroad camera
	person-reidentification-retail-0031	Crossroad camera pedestrian tracker
	person-reidentification-retail-0079	Crossroad camera
Person detection	person-detection-retail-0013	Any SSD-based sample
Face detection for retail	face-detection-retail-0004	Any SSD-based sample
Face and person detection for retail	face-person-detection-retail-0002	Any SSD-based sample
Vehicle detection	vehicle-detection-adas-0002	Any SSD-based sample
Landmarks regression fro retail	landmarks-regression-retail-0009	Smart classroom

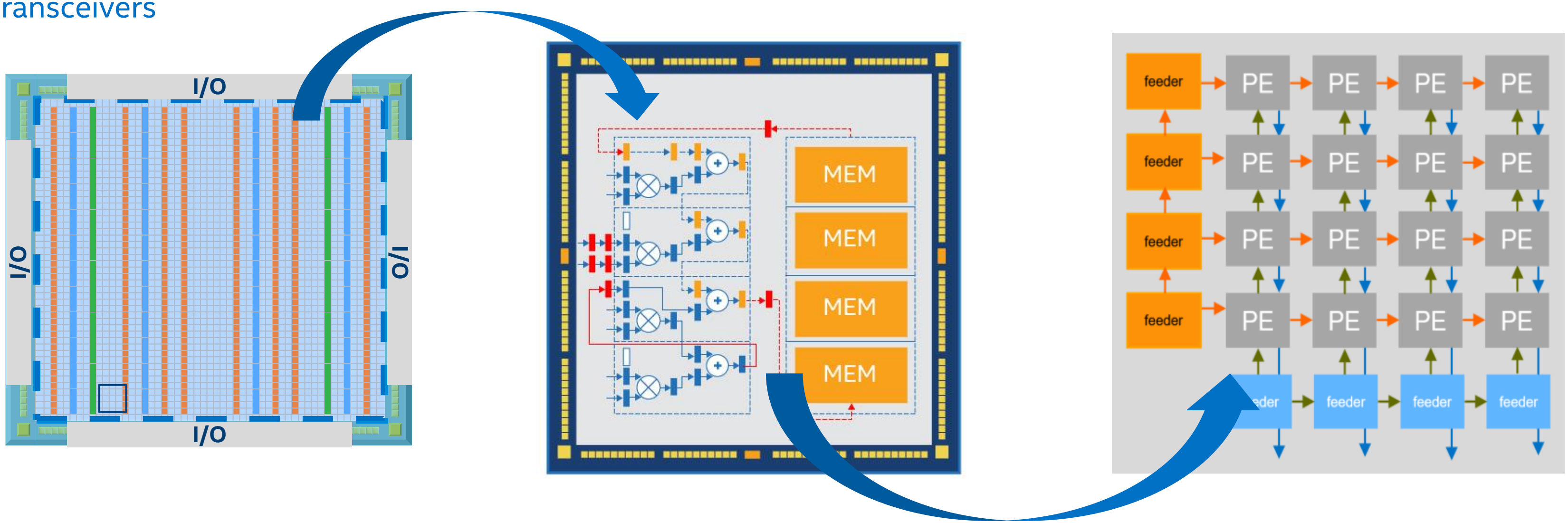
View Documentation ▸ http://docs.openvinotoolkit.org/latest/_docs_Pre_Trained_Models.html

INTEL FPGAS FOR AI

How Intel® FPGAs enable DEEP Learning

- Millions of reconfigurable logic elements & routing fabric
- Thousands of 20Kb memory blocks & MLABs
- Thousands of variable precision digital signal processing (DSP) blocks
- Hundreds of configurable I/O & high-speed transceivers

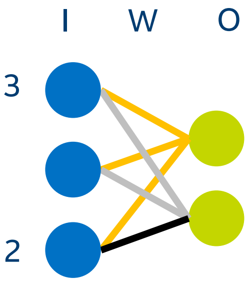
- Programmable Datapath
- Customized Memory structure
- Configurable compute

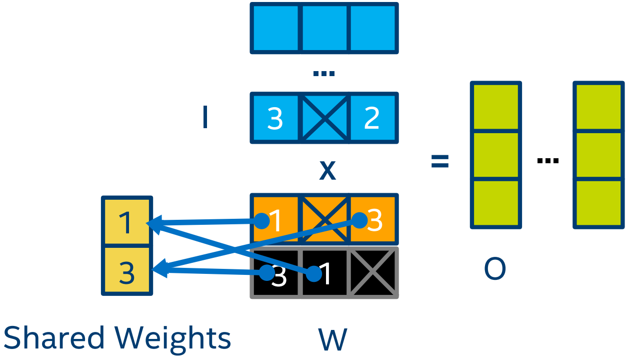


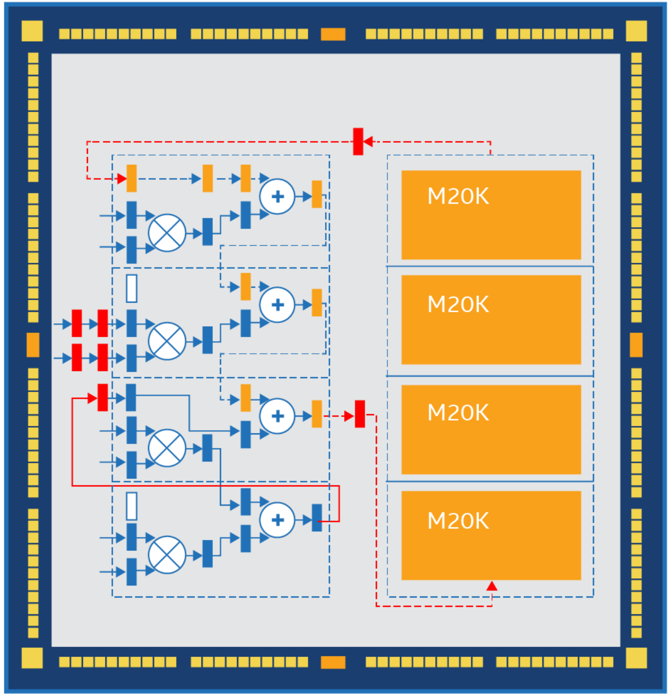
Adapting to innovation

Many efforts to improve efficiency

- Batching
- Reduce bit width
- Sparse weights
- Sparse activations
- Weight sharing
- Compact network

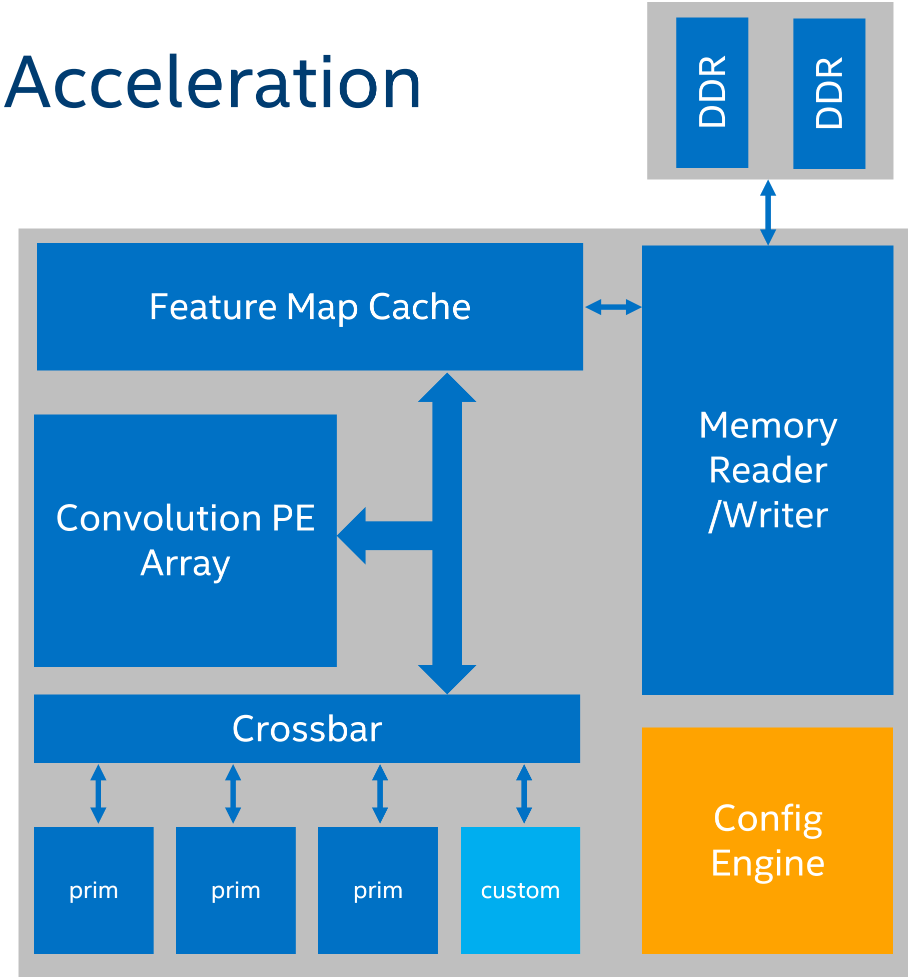






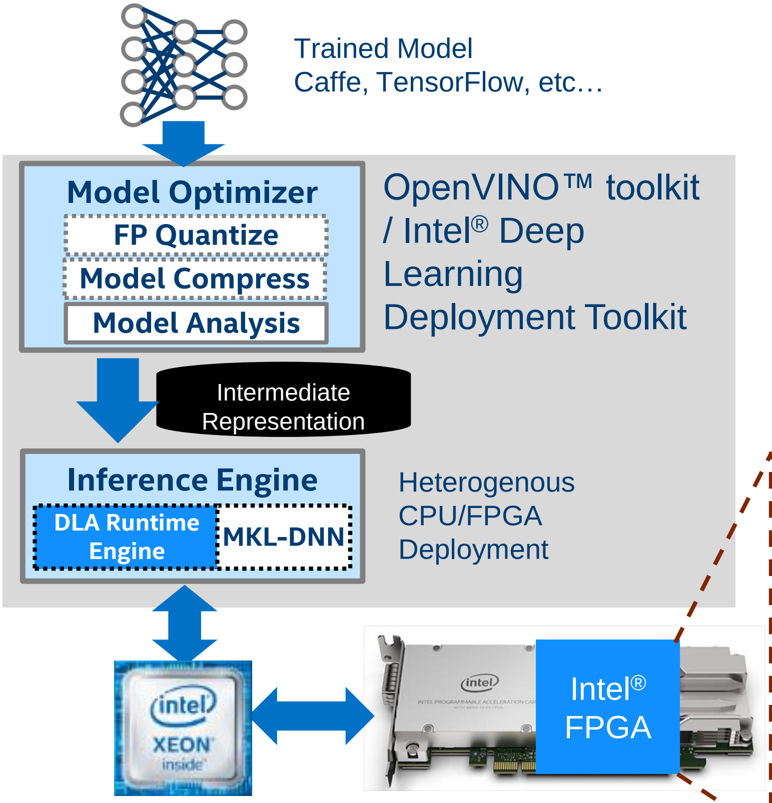
Intel® FPGA Deep Learning Acceleration Suite Features

- CNN acceleration engine for common topologies executed in a graph loop architecture
  - AlexNet, GoogleNet, SqueezeNet, VGG16, ResNet, Yolo, SSD…
- Software Deployment
  - No FPGA compile required
  - Run-time reconfigurable
- Customized Hardware Development
  - Custom architecture creation w/ parameters
  - Custom primitives using OpenCL™ flow



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

FPGA Usage with OpenVINO™ toolkit

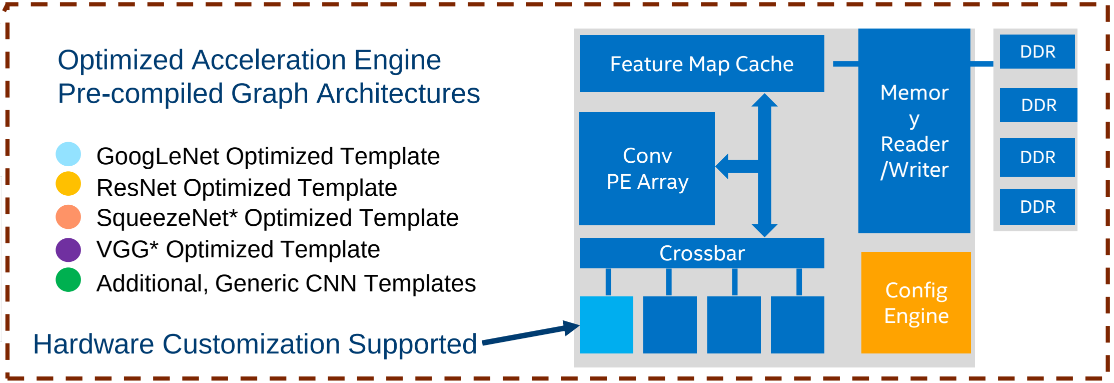


Trained Model
Caffe, TensorFlow, etc…

OpenVINO™ toolkit / Intel® Deep Learning Deployment Toolkit

Heterogenous CPU/FPGA Deployment

- Supports common software frameworks (Caffe, TensorFlow)
- Model Optimizer enhances model for improved execution, storage, and transmission
- Inference Engine optimizes inference execution across Intel® hardware solutions using unified deployment API
- Intel FPGA DLA Suite provides turn-key or customized CNN acceleration for common topologies



Optimized Acceleration Engine
Pre-compiled Graph Architectures

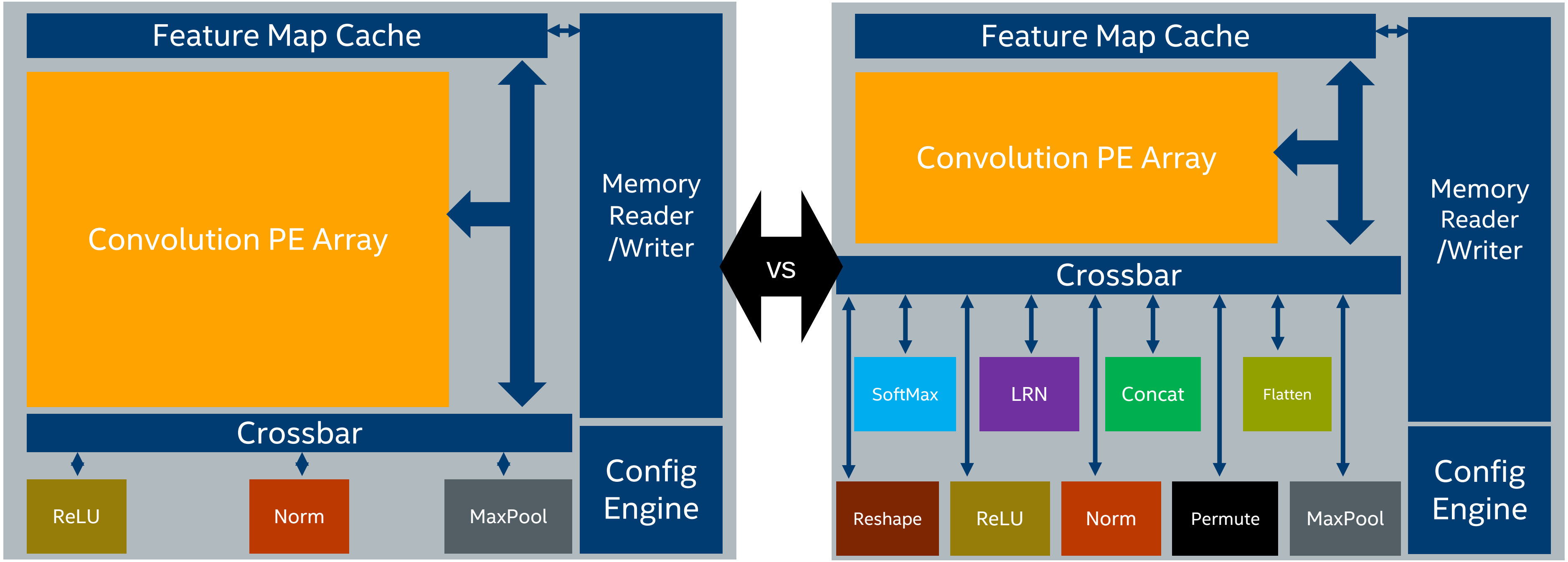
- ● GoogLeNet Optimized Template
- ● ResNet Optimized Template
- ● SqueezeNet* Optimized Template
- ● VGG* Optimized Template
- ● Additional, Generic CNN Templates

Hardware Customization Supported

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

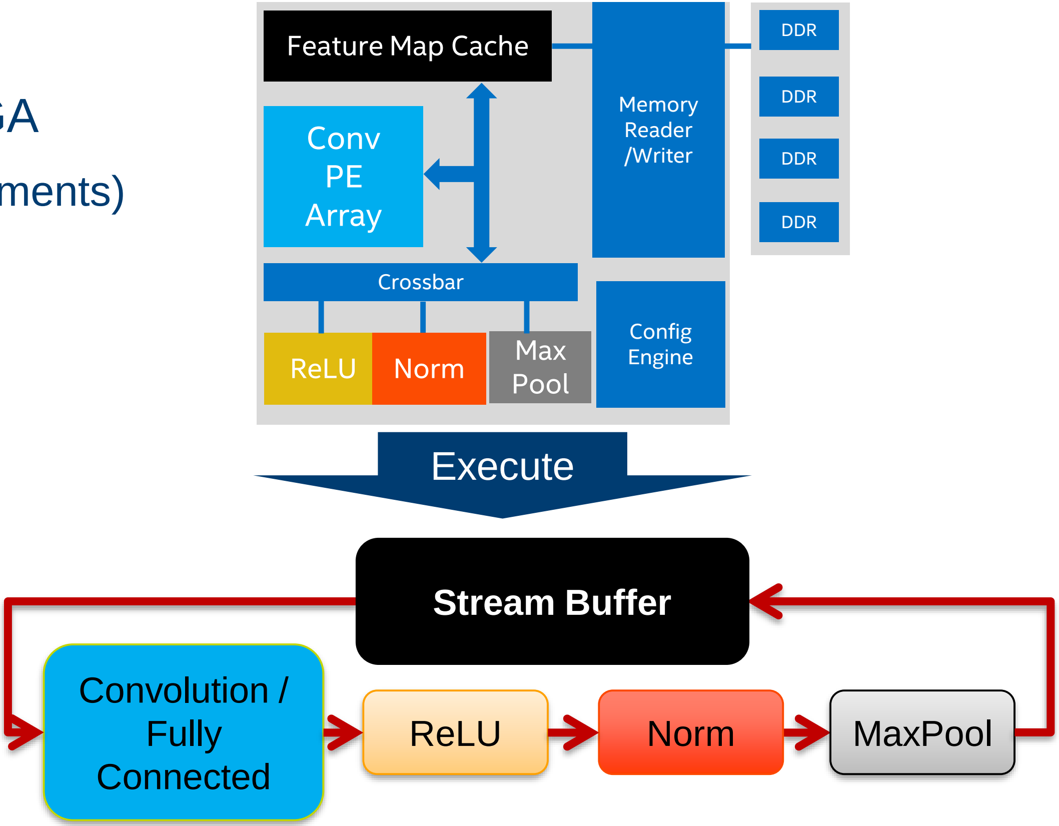
Support for Different Topologies

Tradeoff between features and performance

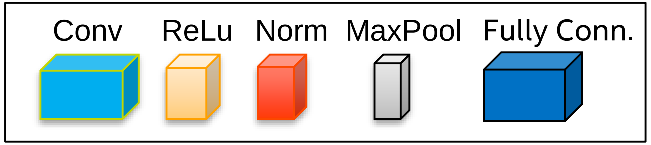


DLA Architecture: Built for Performance

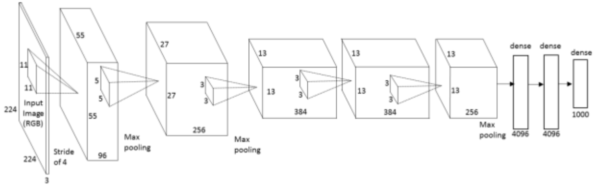
- Maximize Parallelism on the FPGA
  - Filter Parallelism (Processing Elements)
  - Input-Depth Parallelism
  - Batching
  - Feature Stream Buffer
  - Filter Cache
- Choosing FPGA Bitstream
  - Data Type / Design Exploration
  - Primitive Support

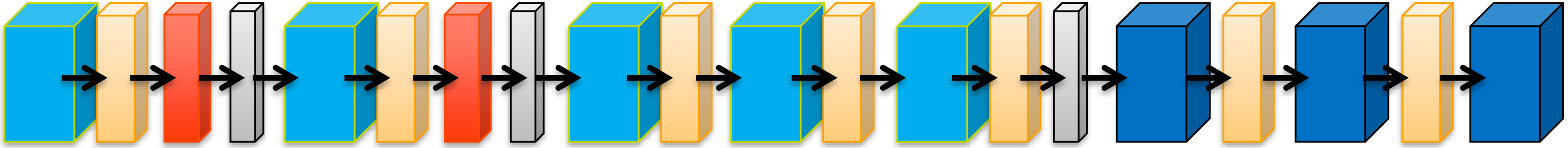


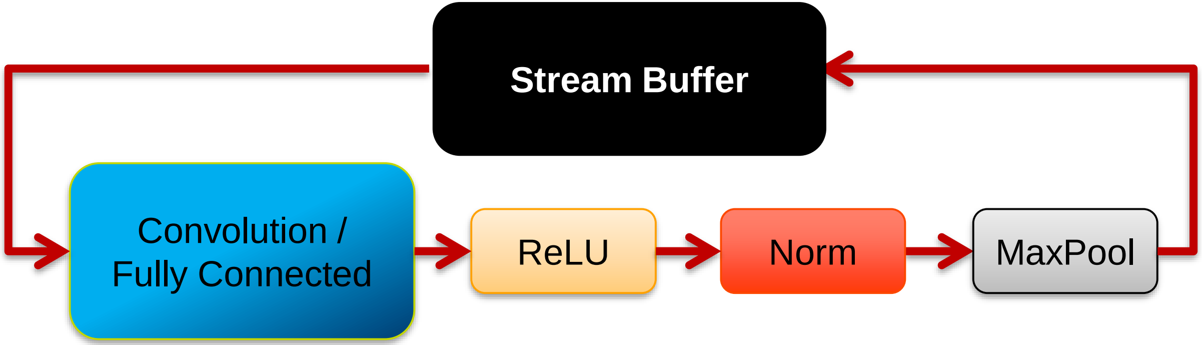
Mapping Graphs in DLA



AlexNet Graph

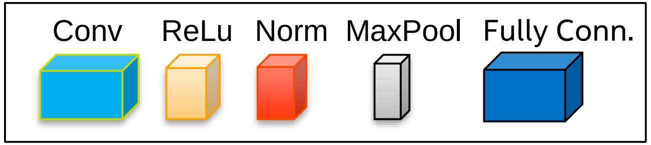




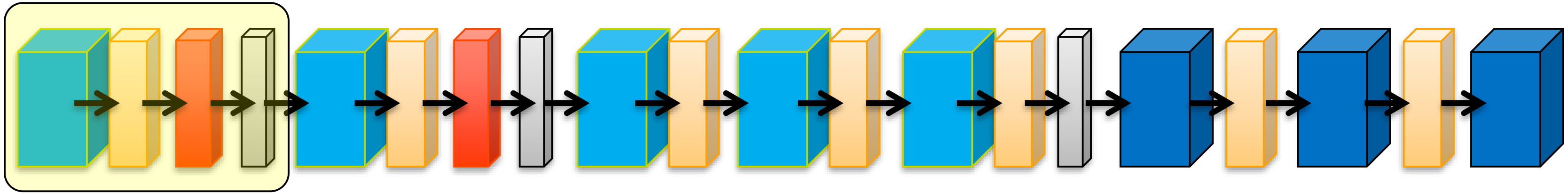


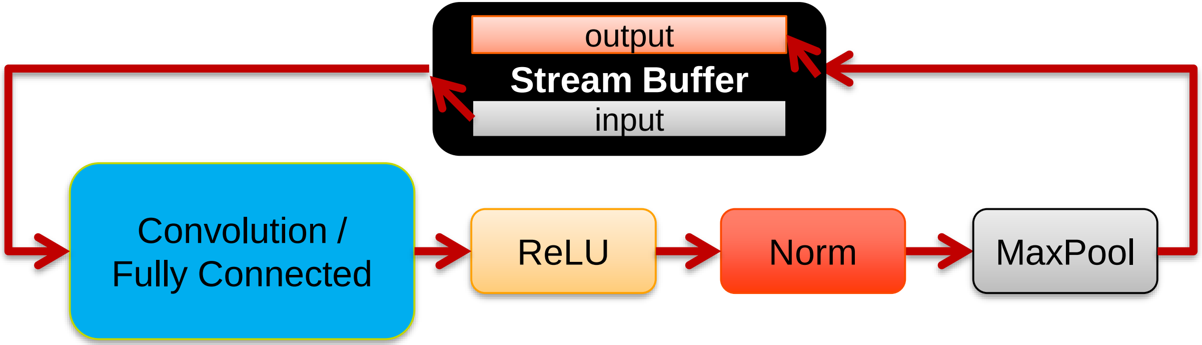
Blocks are run-time reconfigurable and bypassable

Mapping Graphs in DLA



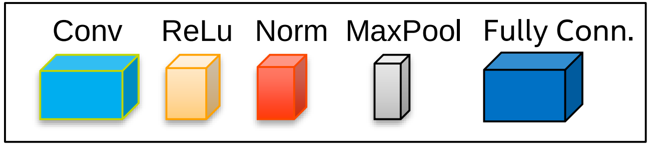
AlexNet Graph



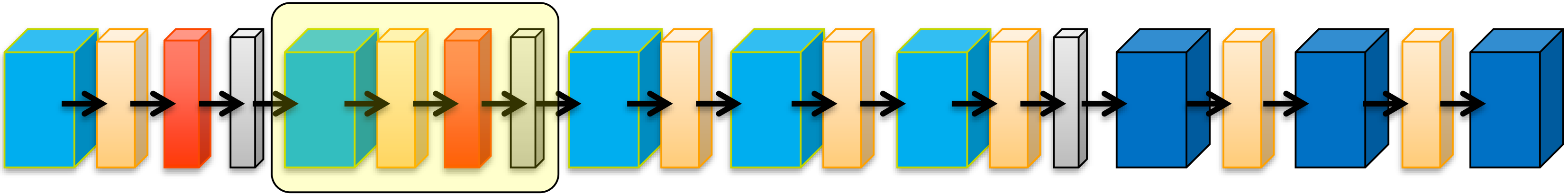


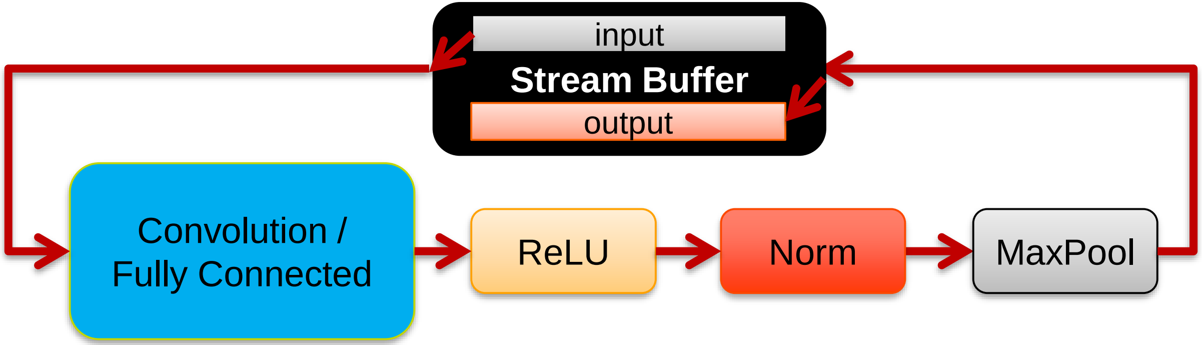
Blocks are run-time reconfigurable and bypassable

Mapping Graphs in DLA



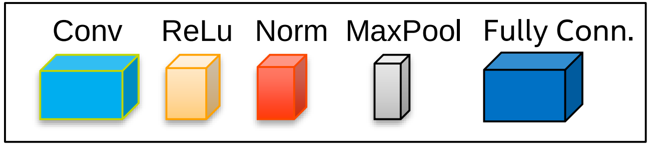
AlexNet Graph



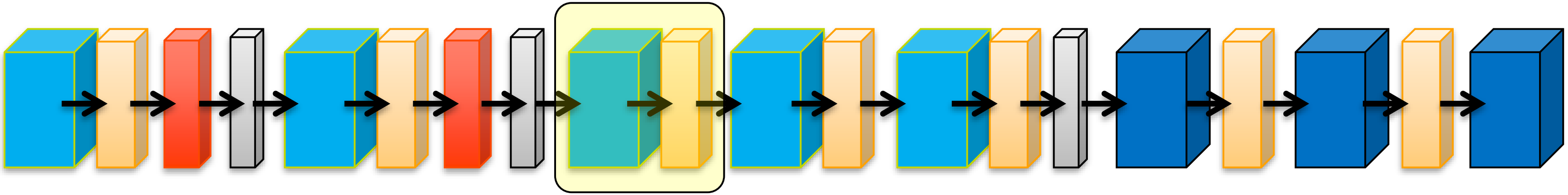


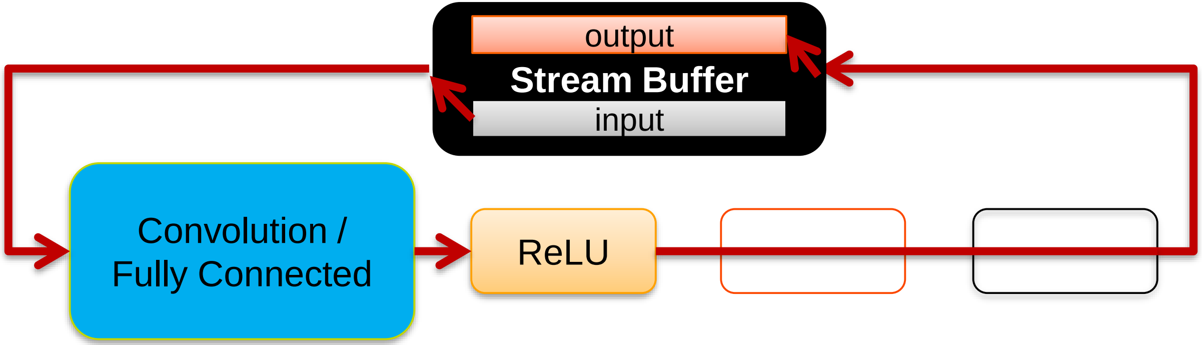
Blocks are run-time reconfigurable and bypassable

Mapping Graphs in DLA



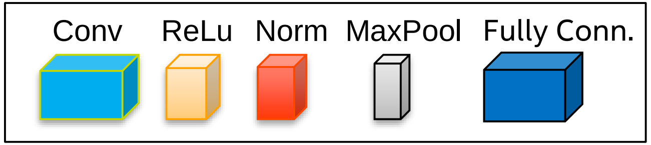
AlexNet Graph



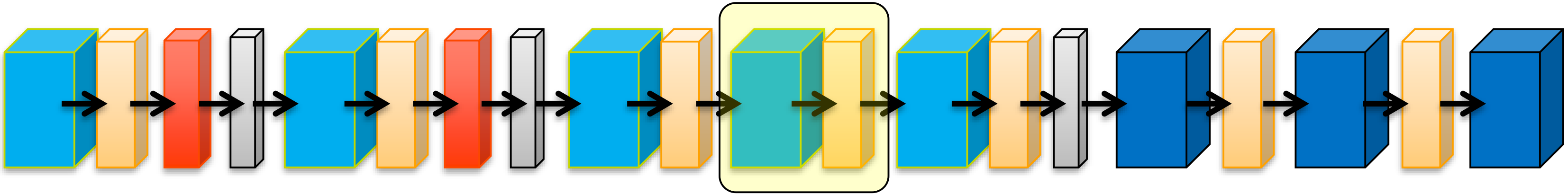


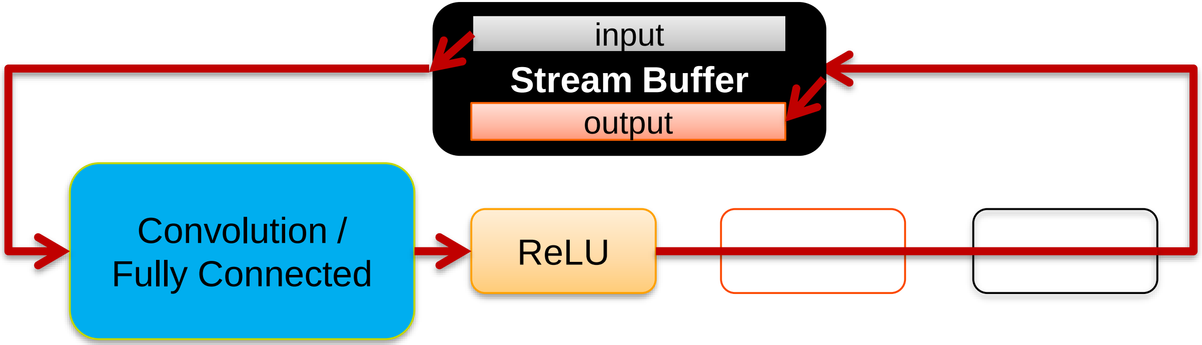
Blocks are run-time reconfigurable and bypassable

Mapping Graphs in DLA



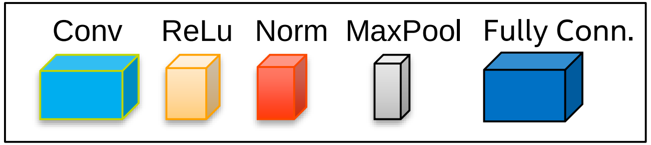
AlexNet Graph



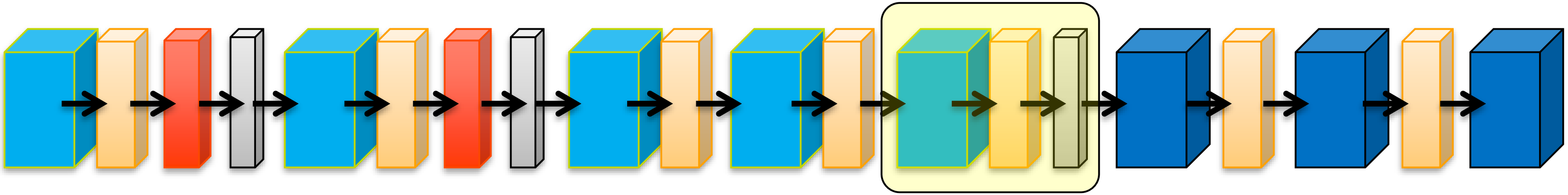


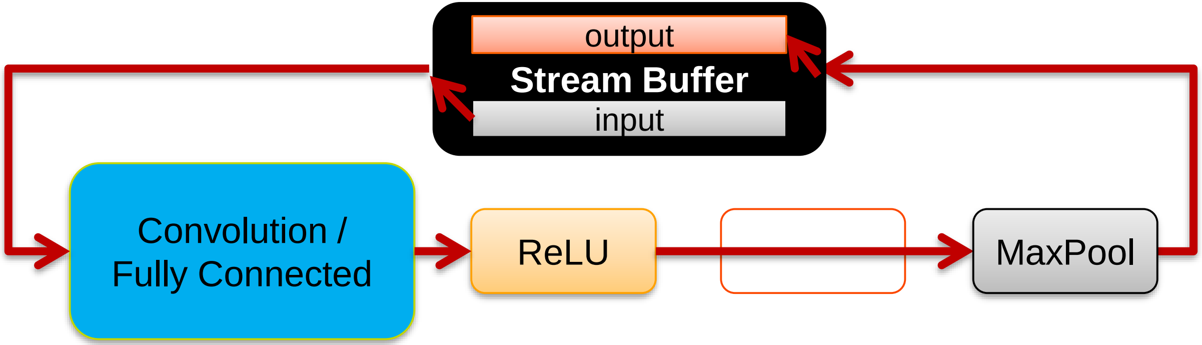
Blocks are run-time reconfigurable and bypassable

Mapping Graphs in DLA



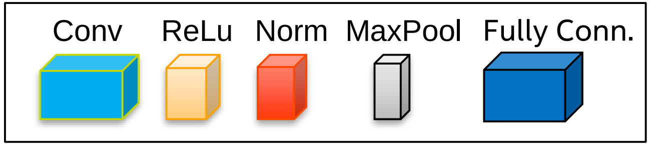
AlexNet Graph



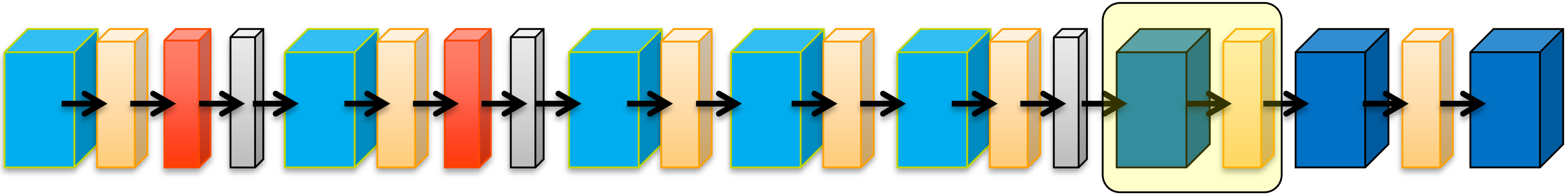


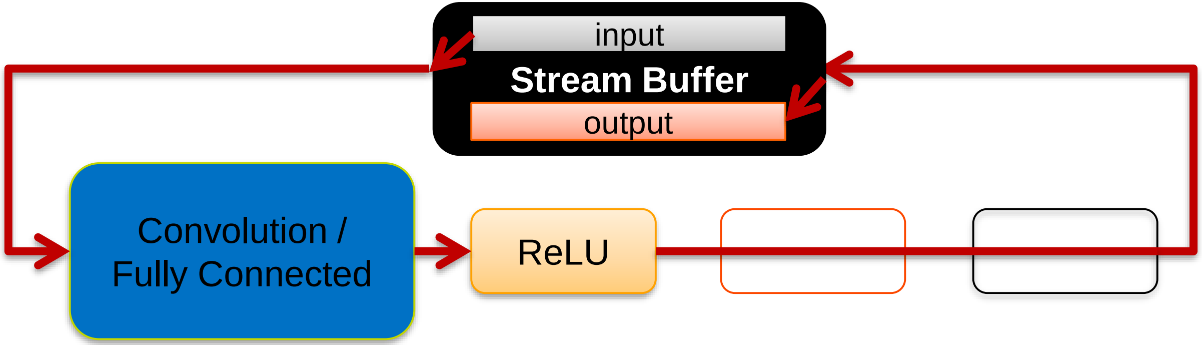
Blocks are run-time reconfigurable and bypassable

Mapping Graphs in DLA



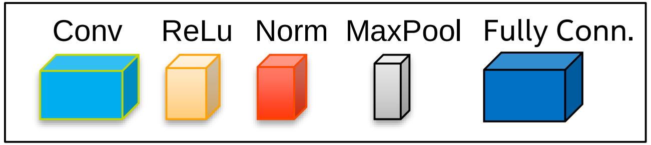
AlexNet Graph



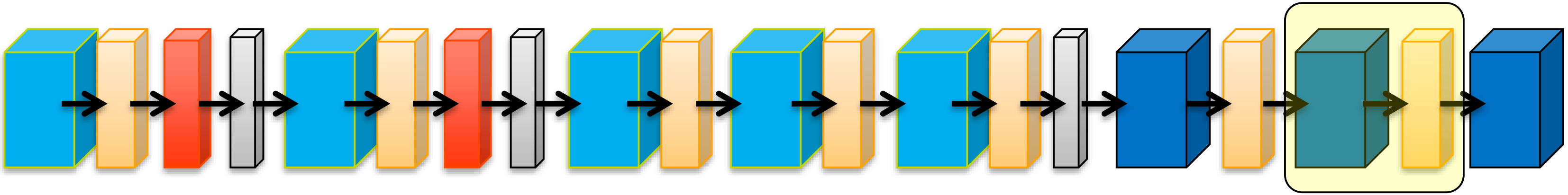


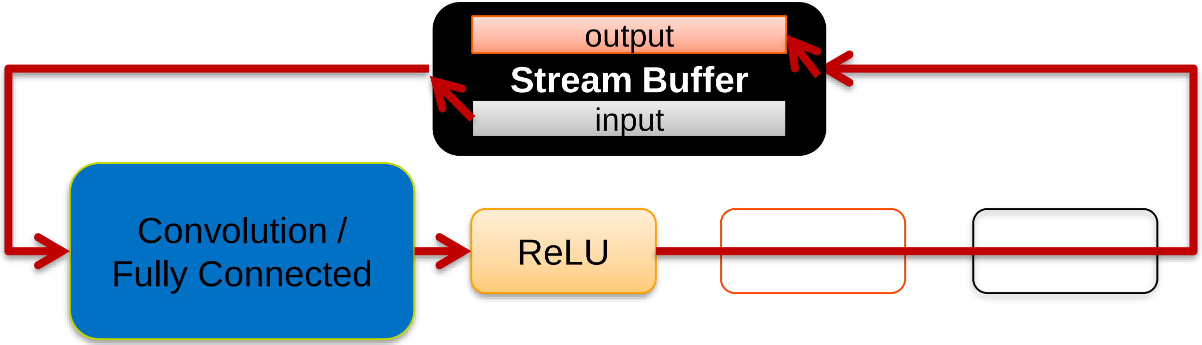
Blocks are run-time reconfigurable and bypassable

Mapping Graphs in DLA



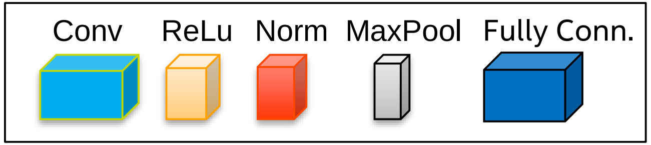
AlexNet Graph



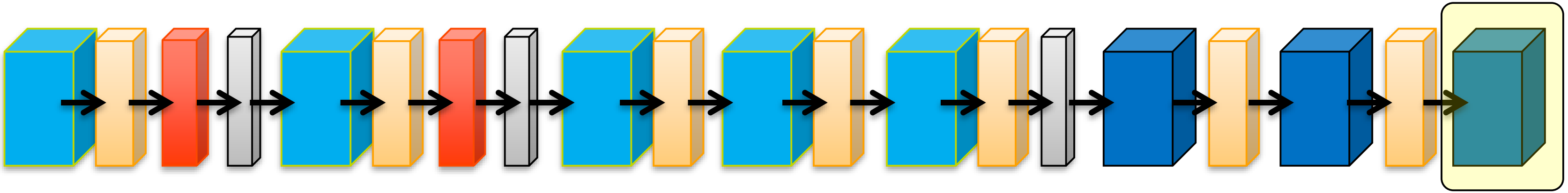


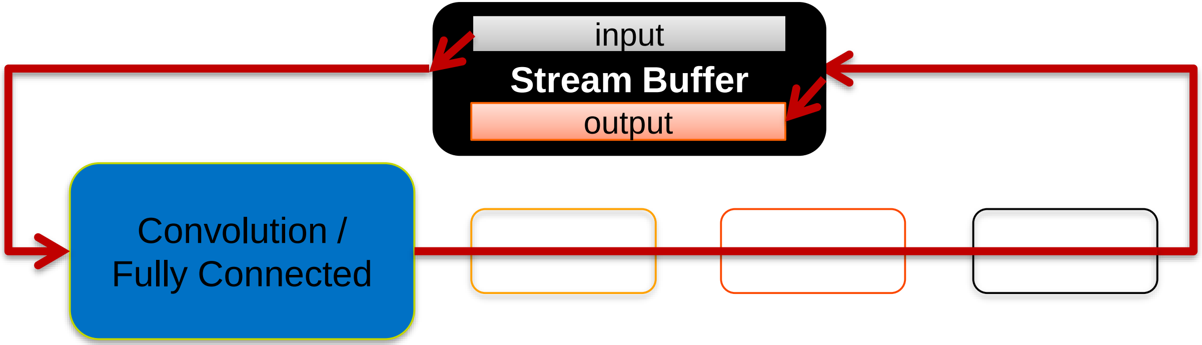
Blocks are run-time reconfigurable and bypassable

Mapping Graphs in DLA



AlexNet Graph





Blocks are run-time reconfigurable and bypassable

Demos with OpenVINO

Application	Supported samples
Face detection	ADAS Interactive face detection
Age/gender recognition	Retail Interactive face detection
Head pose estimation	ADAS Interactive face detection
Emotion recognition	Retail Interactive face detection
Vehicle License plate detection	Security barrier camera
Vehicle attribute recognition	Security barrier camera
License plate recognition	Security barrier camera
Person, vehicle, bike detection	Object detection
Landmarks regression	Smart classroom

Application	Supported samples
Person Reidentification	Crossroad camera
Person Reidentification	Crossroad camera pedestrian tracker
Person Reidentification	Retail Crossroad camera
Person detection	Retail SSD based
Face detection	Retail SSD based
Face person detection	Retail SSD based
Pedestrian detection	ADAS SSD based
Vehicle detection	ADAS SSD based
Person and vehicle detector	ADAS SSD based

https://software.intel.com/en-us/openvino-toolkit/documentation/pretrained-models

FPGA performance evolves over time

PLATFORM	TOPOLOGY	CVSDK MAR.'18[1]	OpenVINO R2[2]	OpenVINO R5[3]	OpenVINO 2019 R1		OpenVINO 2019 R2*
INTEL® ARRIA® 10 FPGA	RESNET-50	31.7 INFERENCE/SEC	76.2	144	314	9.9X	410
	RESNET-101	27	49.4	100	175	6.5X	200

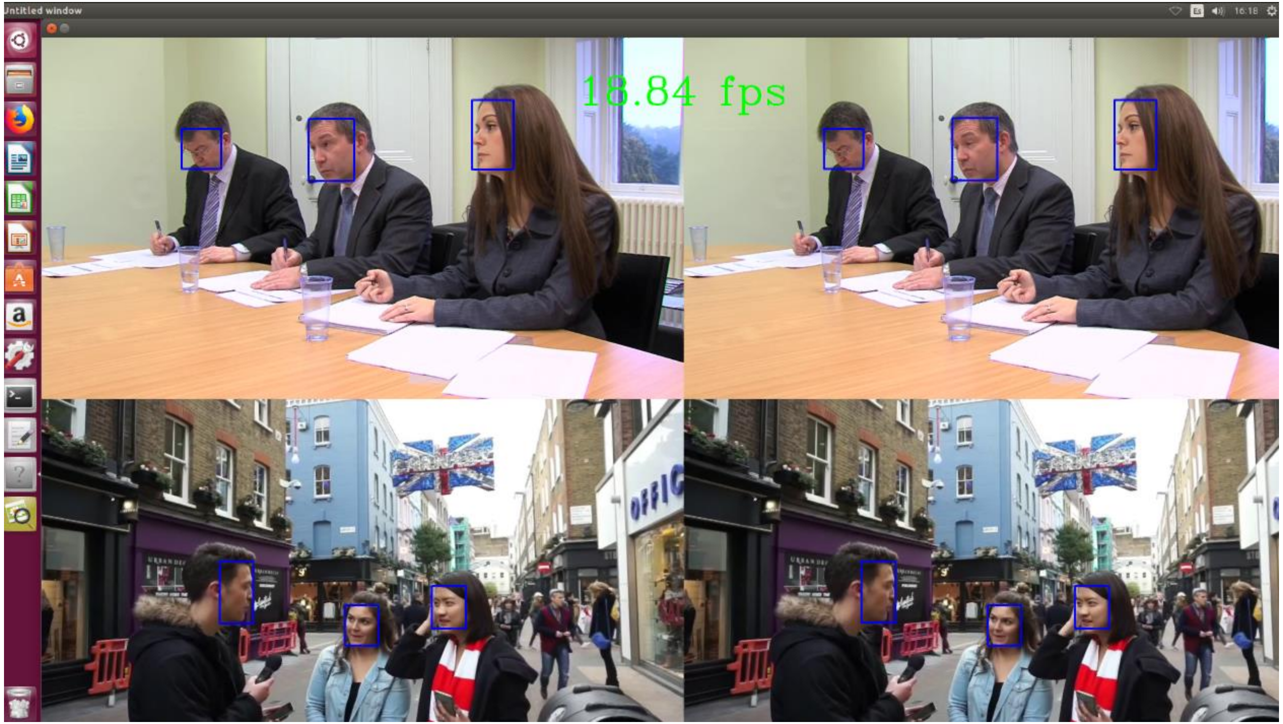
SAME HARDWARE, BETTER BITSTREAMS

INTEL® VISION ACCELERATOR DESIGN WITH INTEL® ARRIA® 10 FPGA

2019 R1

Optimization Notice

OpenVINO demo – Multiple Channel Face Detection



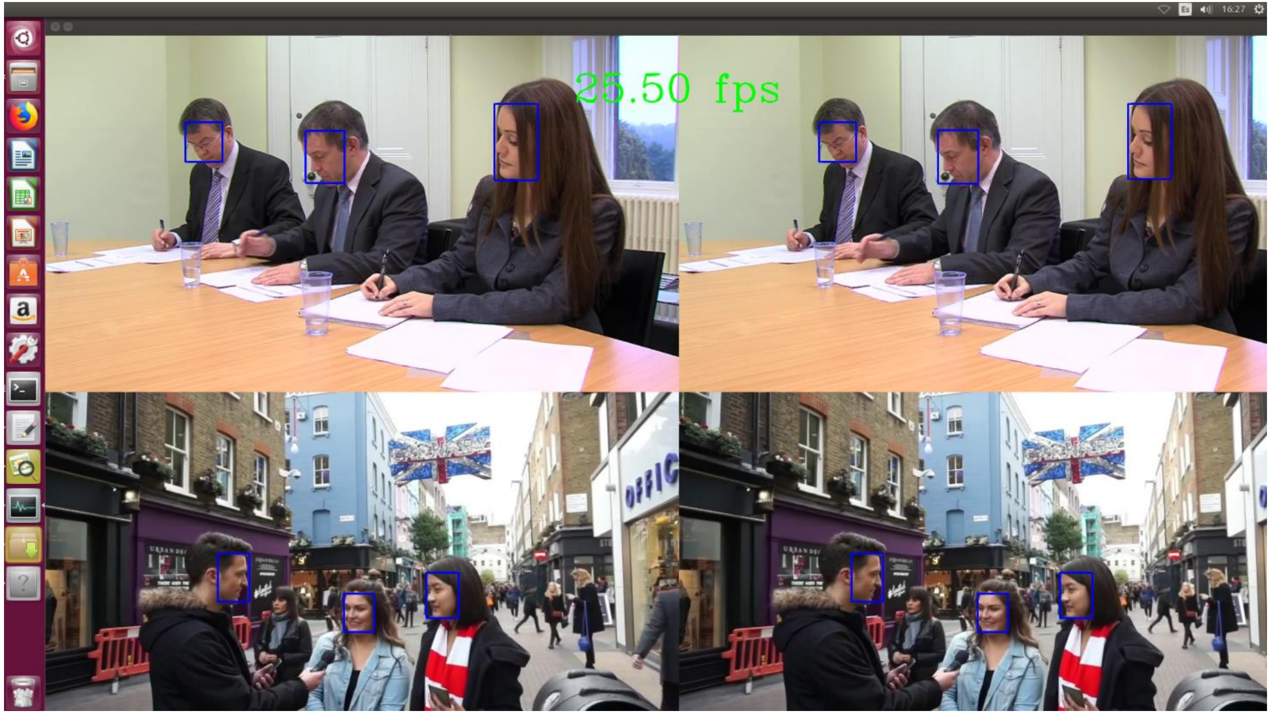
CPU only mode

- 4 channels
- 19fps @channel
- 90% CPU used

System Configuration

CPU: i7-6820EQ CPU @ 2.80GHz
4 physical cores
HD 530 iGPU – Gen 9
24 ex units @350MHz
FPGA card: Mustang F-100
Arria® 10 GX1150 FPGA
PCIe Gen3x8
8G on-board DDR4

OpenVINO demo – Multiple Channel Face Detection



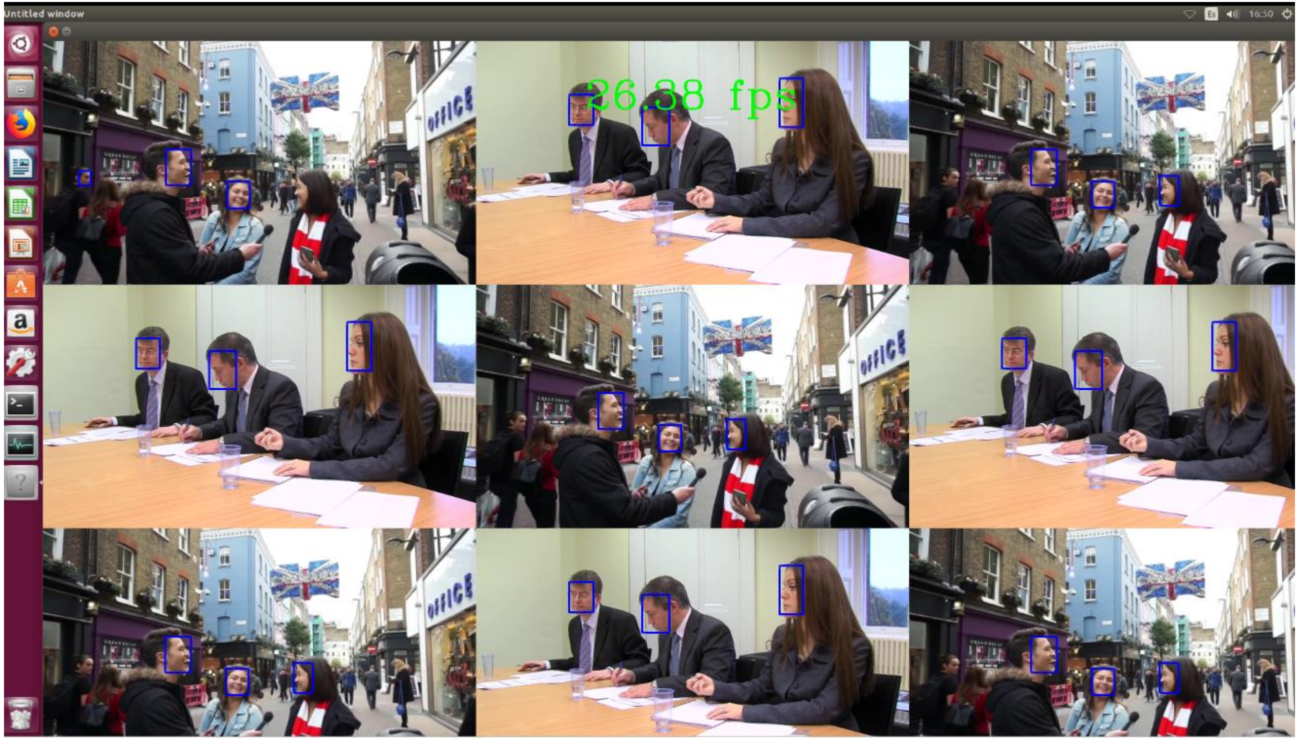
HETERO: GPU, CPU

- 4 channels
- 26fps @channel
- 75% CPU used

System Configuration

CPU: i7-6820EQ CPU @ 2.80GHz
4 physical cores
HD 530 iGPU – Gen 9
24 ex units @350MHz
FPGA card: Mustang F-100
Arria® 10 GX1150 FPGA
PCIe Gen3x8
8G on-board DDR4

OpenVINO demo – Multiple Channel Face Detection



HETERO: FPGA, CPU

- 9 channels
- 26fps @channel
- 55% CPU used

System Configuration

CPU: i7-6820EQ CPU @ 2.80GHz
4 physical cores
HD 530 iGPU – Gen 9
24 ex units @350MHz

FPGA card: Mustang F-100
Arria® 10 GX1150 FPGA
PCIe Gen3x8
8G on-board DDR4

experience
what's inside™