# INTEL® DISTRIBUTION OF OPENVINO™

# OPTIMIZATION NOTICE

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness or any optimization on microprocessors not manufactured by Intel. Microprocessor dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice Revision #20110804.

# LEGAL NOTICES AND DISCLAIMERS (1 OF 2)

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software, or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at www.intel.com.

Performance estimates 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.

Cost reduction scenarios described are intended as examples of how a given Intel-based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

This document contains information on products, services, and 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.

Any forecasts of goods and services needed for Intel's operations are provided for discussion purposes only. Intel will have no liability to make any purchase in connection with forecasts published in this document.

Arduino* 101 and the Arduino infinity logo are trademarks or registered trademarks of Arduino, LLC.

Altera, Arria, the Arria logo, Intel, the Intel logo, Intel Atom, Intel Core, Intel Nervana, Intel Xeon Phi, Movidius, Saffron, and Xeon are trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

# LEGAL NOTICES AND DISCLAIMERS (2 OF 2)

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. Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software, or service activation. Learn more at intel.com, or from the OEM or retailer.

No computer system can be absolutely secure.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit www.intel.com/performance.

Cost-reduction scenarios described are intended as examples of how a given Intel-based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

Statements in this document that refer to Intel's plans and expectations for the quarter, the year, and the future are forward-looking statements that involve a number of risks and uncertainties.

A detailed discussion of the factors that could affect Intel's results and plans is included in Intel's SEC filings, including the annual report on Form 10-K.

The products described 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.

Performance estimates 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.

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

Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

Intel, the Intel logo, Pentium, Celeron, Atom, Core, Xeon, Movidius, Saffron, and others are trademarks of Intel Corporation in the United States and other countries.

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

# AGENDA

EDGE COMPUTING

INTEL® DISTRIBUTION OF OPENVINO™ TOOLKIT

MODEL DOWNLOADER

MODEL OPTIMIZER

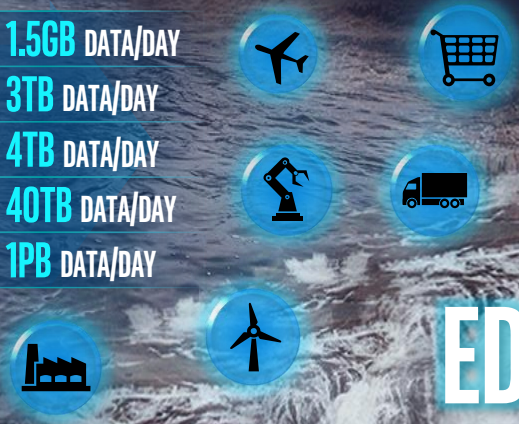
INFERENCE ENGINE API

HANDS-ON LAB

# DRIVERS FOR EDGE: LATENCY, BANDWIDTH, SECURITY, CONNECTIVITY

## BY 2020

| | |
|---|---|
| AVERAGE INTERNET USER | 1.5GB DATA/DAY |
| SMART HOSPITAL | 3TB DATA/DAY |
| AUTONOMOUS AUTOMOBILE | 4TB DATA/DAY |
| CONNECTED AIRPLANE | 40TB DATA/DAY |
| SMART FACTORY | 1PB DATA/DAY |



CLOUD

EDGE

**BY 2019, 45% OF DATA WILL BE STORED, ANALYZED, AND ACTED ON AT THE EDGE**

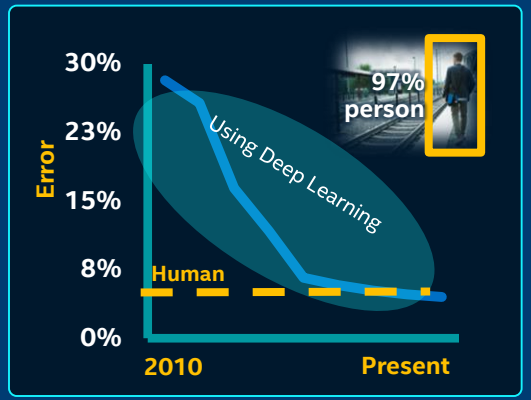
1. Amalgamation of analyst data and Intel analysis.
2. IDC FutureScape: Worldwide Internet of Things 2017 Predictions (link)
3. IDC FutureScape: Worldwide Internet of Things 2015 Predictions (link)
4. Source: IDC FutureScape: Worldwide Internet of Things 2017 Predictions

# CAMERA IS THE ULTIMATE SENSOR. DATA = VIDEO

## WHY EDGE HAS NOT BECOME SMART BEFORE ?

### TRADITIONAL COMPUTER VISION

### DEEP LEARNING


Source: ILSVRC ImageNet winning entry classification error rate each year 2010-2016

# INTEL® COMPUTER VISION PORTFOLIO

**EXPERIENCES**

**TOOLS**

Intel® Media SDK/Media Server Studio - OpenVINO™ toolkit - Intel® System Studio Intel® SDK for OpenCL™ Applications - Intel® Parallel Studio XE

**FRAMEWORKS**

Spark MLlib BigDL TensorFlow mxnet theano neon torch Caffe

**LIBRARIES**

Intel® Data Analytics Acceleration Library

Intel® Distribution for python

Intel® Math Kernel Library

Intel® Nervana™ Graph

saffron TECHNOLOGY an intel company

**HARDWARE**

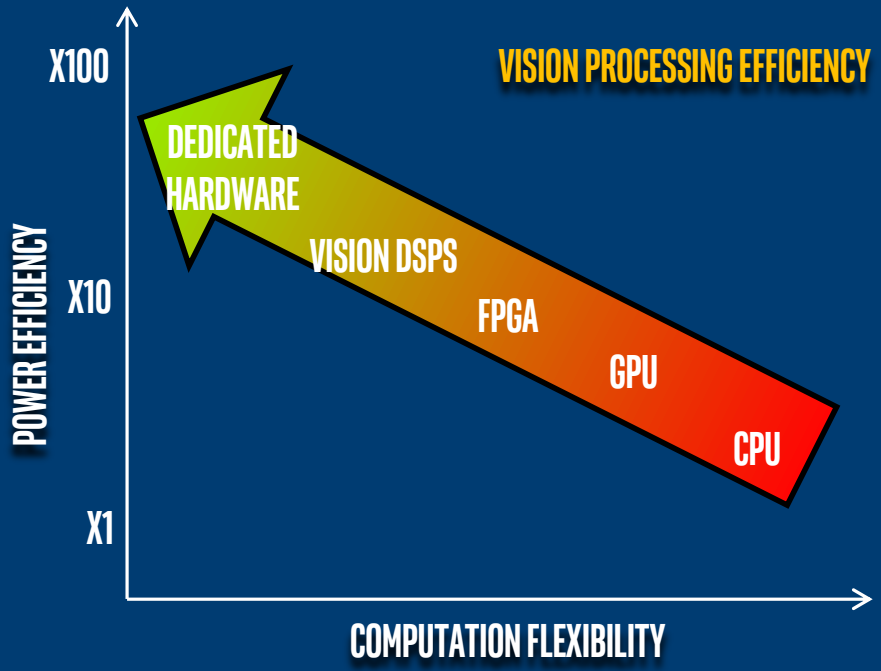
Compute: Intel Atom, Quark, Core i7, Xeon, Arria 10

Memory & Storage

Networking

Visual Intelligence: RealSense Technology, Movidius

# CHOOSING THE "RIGHT" HARDWARE




**Consider each device with**

- Compute efficiency, parallelism
- Power consumption, Memory hierarchy, size, communication, Programming model, APIs

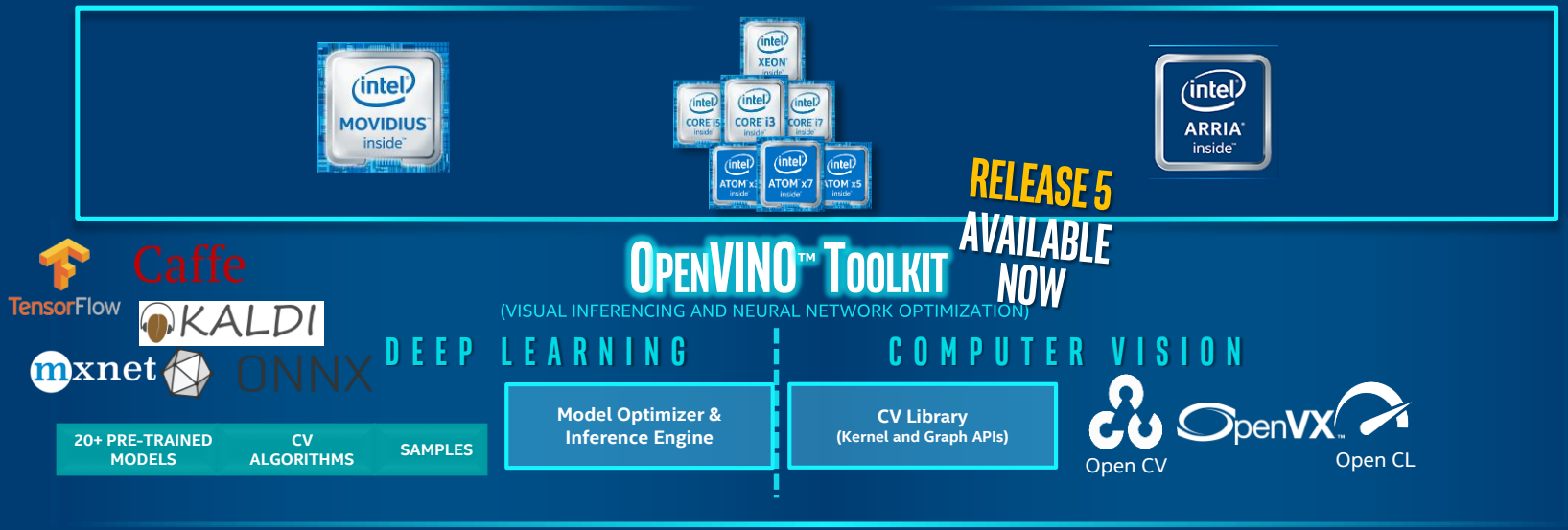
**Power/performance efficiency vary**

- Running the right workload on the right piece of HW → higher efficiency

**Trade offs**

- Power/ performance
- Price
- Software flexibility, portability

# INTEL® VISION PRODUCTS: UNLEASHING AI AT THE EDGE




**Download today:** software.intel.com/openvino-toolkit

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

# WHAT'S INSIDE THE OPENVINO™ TOOLKIT

## Intel® Deep Learning Deployment Toolkit

**Model Optimizer**
Convert & Optimize

IR

**Inference Engine**
Optimized Inference

20+ Pre-trained Models

Computer Vision Algorithms

Samples

IR = Intermediate Representation file

## Traditional Computer Vision Tools & Libraries

### Optimized Libraries

OpenCV*

OpenVX*

Photography Vision

Code Samples

For Intel® CPU & CPU with integrated graphics

### Increase Media/Video/Graphics Performance

Intel® Media SDK
Open Source version

OpenCL™ Drivers & Runtimes

For CPU with integrated graphics

### Optimize Intel® FPGA

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

Bitstreams

FPGA – Linux* only

**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)

Intel® Architecture-Based Platforms Support

CELERON · ATOM · CORE · XEON · ARRIA 10 · Movidius · IRIS Pro Graphics

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

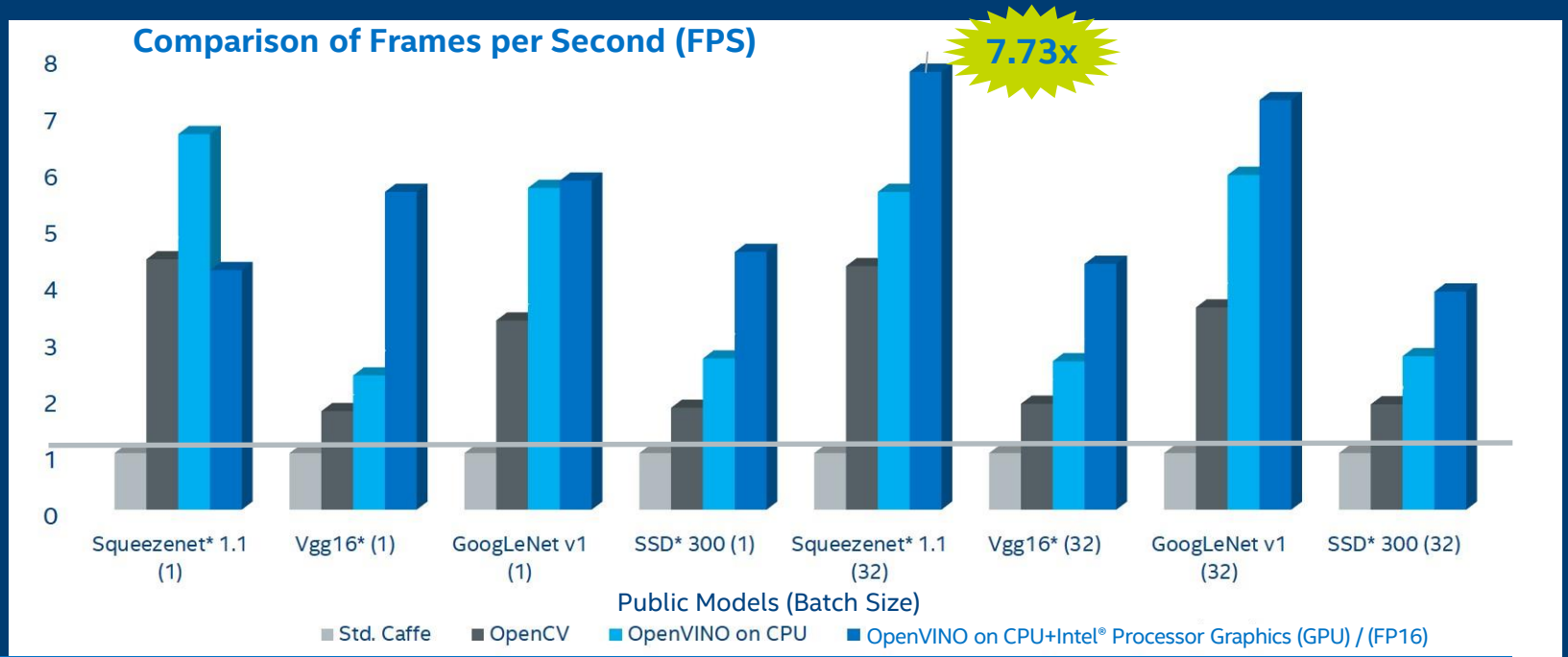
# TECHNICAL REQUIREMENTS & SPECS

## OPENVINO™ TOOLKIT Technical Specifications

| | Intel® Platforms | Compatible Operating Systems |
|---|---|---|
| **Target Solution Platforms** | **CPU**<br>▪ 6th-8th generation Intel® Xeon® & Core™ processors | ▪ Ubuntu* 16.04.3 LTS (64 bit)<br>▪ Microsoft Windows* 10 (64 bit)<br>▪ CentOS* 7.4 (64 bit) |
| | ▪ Intel® Pentium® processor N4200/5, N3350/5, N3450/5 with Intel® HD Graphics | ▪ Yocto Project* Poky Jethro v2.0.3 (64 bit) |
| | **Iris® Pro & Intel® HD Graphics**<br>▪ 6th-8th generation Intel® Core™ processor with Intel® Iris™ Pro graphics & Intel® HD Graphics<br>▪ 6th-8th generation Intel® Xeon® processor with Intel® Iris™ Pro Graphics & Intel® HD Graphics (excluding e5 product family, which does not have graphics[1]) | ▪ Ubuntu 16.04.3 LTS (64 bit)<br>▪ Windows 10 (64 bit)<br>▪ CentOS 7.4 (64 bit) |
| | **FPGA**<br>▪ Intel® Arria® FPGA 10 GX development kit<br>▪ Intel® Programmable Acceleration Card with Intel® Arria® 10 GX FPGA operating systems<br>▪ OpenCV* & OpenVX* functions must be run against the CPU or Intel® Processor Graphics (GPU) | ▪ Ubuntu 16.04.3 LTS (64 bit)<br>▪ CentOS 7.4 (64 bit) |
| | **VPU**<br>▪ Intel Movidius™ Neural Compute Stick | ▪ Ubuntu 16.04.3 LTS (64 bit)<br>▪ CentOS 7.4 (64 bit)<br>▪ Windows 10 (64 bit) |
| **Development Platforms** | 6th-8th generation Intel® Core™ and Intel® Xeon® processors | ▪ Ubuntu* 16.04.3 LTS (64 bit)<br>▪ Windows* 10 (64 bit)<br>▪ CentOS* 7.4 (64 bit) |
| **Additional Software Requirements** | Linux* build environment required components<br>▪ OpenCV 3.4 or higher<br>▪ CMake* 2.8 or higher<br>▪ GNU Compiler Collection (GCC) 3.4 or higher<br>▪ Python* 3.4 or higher | |
| | Microsoft Windows* build environment required components<br>▪ Intel® HD Graphics Driver (latest version)†<br>▪ Intel® C++ Compiler 2017 Update 4<br>▪ Python 3.4 or higher<br>▪ OpenCV 3.4 or higher<br>▪ CMake 2.8 or higher<br>▪ Microsoft Visual Studio* 2015 | |
| **External Dependencies/Additional Software** | | View Product Site, detailed System Requirements |

[1]Graphics drivers are required only if you use Intel® Processor Graphics (GPU).

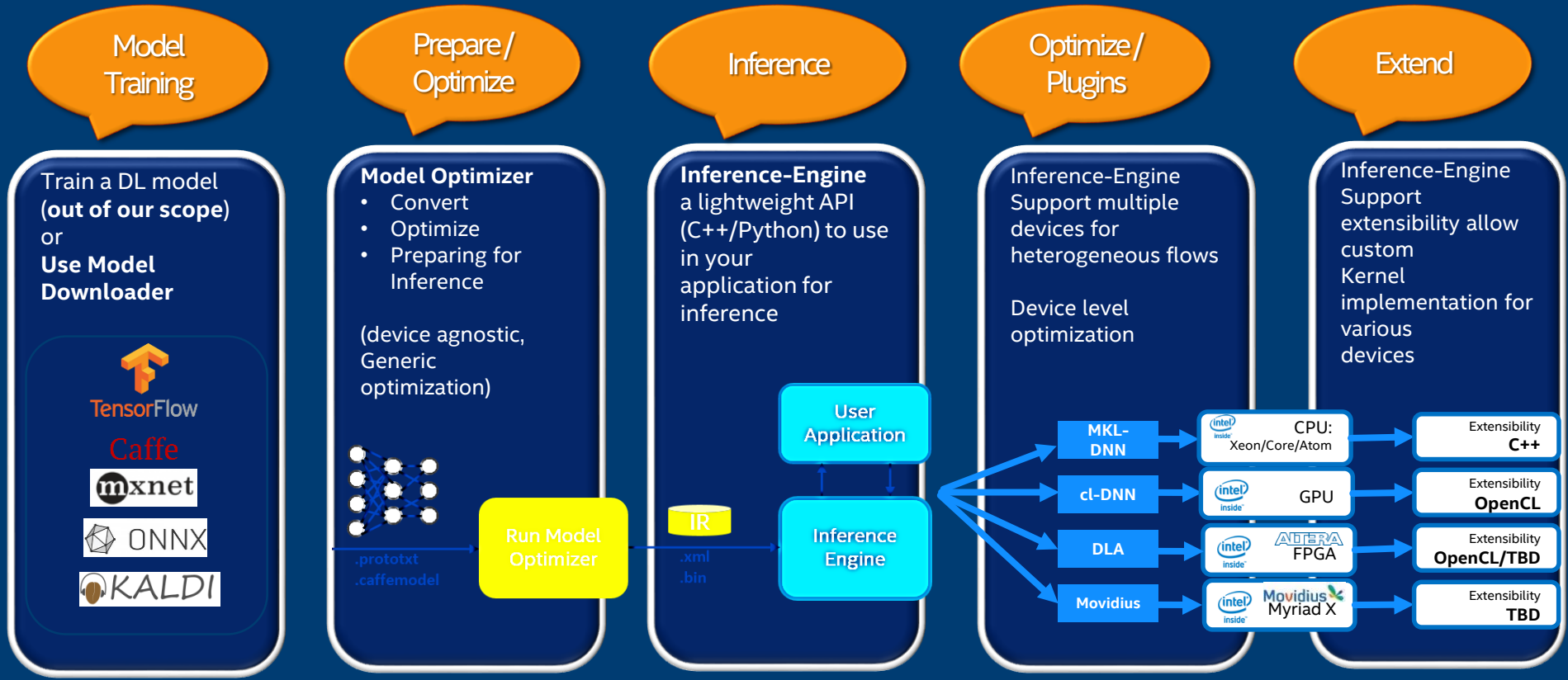
# DEEP LEARNING WORKLOAD PERFORMANCE ON PUBLIC MODELS USING OPENVINO™ TOOLKIT & INTEL® ARCHITECTURE




## Fast Results on Intel Hardware, even before using Accelerators

[1]Depending on workload, quality/resolution for FP16 may be marginally impacted. A performance/quality tradeoff from FP32 to FP16 can affect accuracy; customers are encouraged to experiment to find what works best for their situation. The benchmark results reported in this deck may need to be revised as additional testing is conducted. Performance results are based on testing as of April 10, 2018 and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure. For more complete information about performance and benchmark results, visit www.intel.com/benchmarks.
**Configuration:** Testing by Intel as of April 10, 2018. Intel® Core™ i7-6700K CPU @ 2.90GHz fixed, GPU GT2 @ 1.00GHz fixed Internal ONLY testing, Test v312.30 – Ubuntu* 16.04, OpenVINO™ 2018 RC4. Tests were based on various parameters such as model used (these are public), batch size, and other factors. Different models can be accelerated with different Intel hardware solutions, yet use the same Intel software tools.

# COMPUTER VISION APPLICATION PIPELINE



# MODEL DOWNLOADER & INTEL MODELS

- For fast development and deployment of Computer Vision applications Intel® Distribution of OpenVINO™ provides **Model Downloader** and **pre-trained Intel Models**
- Model Downloader can be used to download many popular open source/public pre-trained Caffe/Tensorflow/MxNet models.
- Maintained under **Open Model Zoo**: https://github.com/opencv/open_model_zoo/tree/2018/model_downloader

```
$ cd /opt/intel/computer_vision_sdk/deployment_tools/model_downloader
$ python3 downloader.py --print_all
densenet-121
densenet-161
densenet-169
densenet-201
squeezenet1.0
squeezenet1.1
mtcnn-p
mtcnn-r
mtcnn-o
mobilenet-ssd
vgg19
vgg16
ssd512
ssd300
inception-resnet-v2
dilation
```

# PUBLIC MODELS

| Network family | Model | Problem/ Dataset | URL |
|---|---|---|---|
| DenseNet | densenet-121<br>densenet-161<br>densenet-169<br>densenet-201 | ImageNet | https://github.com/liuzhuang13/DenseNet |
| SqueezeNet | squeezenet1.0<br>squeezenet1.1 | ImageNet | https://github.com/DeepScale/SqueezeNet |
| MTCNN | mtcnn-p<br>mtcnn-r<br>mtcnn-o | FDDB, AFLW | https://github.com/kpzhang93/MTCNN_face_detection_alignment |
| MobileNet-SSD | mobilenet-ssd | VOC0712 | https://github.com/chuanqi305/MobileNet-SSD |
| VGG | vgg19<br>vgg16 | ImageNet | https://gist.github.com/ksimonyan/3785162f95cd2d5fee77<br>https://gist.github.com/ksimonyan/211839e770f7b538e2d8 |
| SSD | ssd512<br>ssd300 | VOC0712 | https://github.com/weiliu89/caffe |

# PRE-TRAINED INTEL MODELS

### Supported Samples

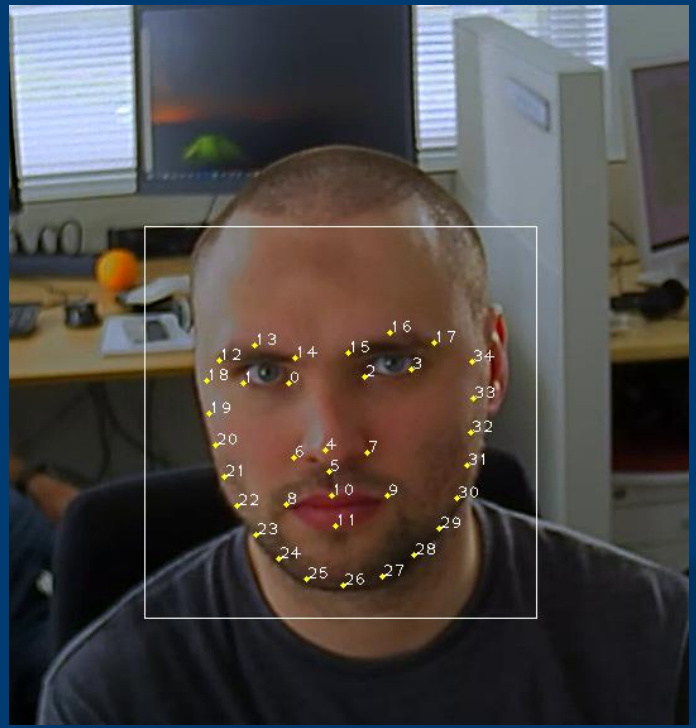
Reference this table for components that support the pretrained models.

| Pretrained Model | Supported Samples | CPU | Integrated Graphics | FPGA | VPU |
|---|---|---|---|---|---|
| face-detection-adas-0001 | Interactive face detection | ✓ | ✓ | ✓ | ✓ |
| age-gender-recognition-retail-0013 | Interactive face detection | ✓ | ✓ | ✓ | ✓ |
| head-pose-estimation-adas-0001 | Interactive face detection | ✓ | ✓ | ✓ | |
| emotions-recognition-retail-0003 | Interactive face detection | ✓ | ✓ | ✓ | ✓ |
| facial-landmarks-35-adas-0001 | Interactive face detection | ✓ | ✓ | | |
| vehicle-license-plate-detection-barrier-0106 | Security barrier camera | ✓ | ✓ | ✓ | ✓ |
| vehicle-attributes-recognition-barrier-0039 | Security barrier camera | ✓ | ✓ | ✓ | ✓ |
| license-plate-recognition-barrier-0001 | Security barrier camera | ✓ | ✓ | ✓ | ✓ |
| person-detection-retail-0001 | Object detection | ✓ | ✓ | | |
| person-vehicle-bike-detection-crossroad-0078 | Crossroad camera | ✓ | ✓ | ✓ | ✓ |
| person-attributes-recognition-crossroad-0200 | Crossroad camera | ✓ | ✓ | | |
| person-reidentification-retail-0076 | Crossroad camera | ✓ | ✓ | ✓ | ✓ |
| person-reidentification-retail-0031 | Crossroad camera pedestrian tracker | ✓ | ✓ | ✓ | ✓ |
| person-reidentification-retail-0079 | Crossroad camera | ✓ | ✓ | ✓ | ✓ |
| road-segmentation-adas-0001 | Image segmentation | ✓ | ✓ | | |
| semantic-segmentation-adas-0001 | Image segmentation | ✓ | ✓ | | |
| person-detection-retail-0013 | Any SSD-based sample | ✓ | ✓ | ✓ | ✓ |
| face-detection-retail-0004 | Any SSD-based sample | ✓ | ✓ | ✓ | ✓ |
| face-person-detection-retail-0002 | Any SSD-based sample | ✓ | ✓ | ✓ | ✓ |
| pedestrian-detection-adas-0002 | Any SSD-based sample | ✓ | ✓ | ✓ | |
| vehicle-detection-adas-0002 | Any SSD-based sample | ✓ | ✓ | ✓ | ✓ |
| pedestrian-and-vehicle-detector-adas-0001 | Any SSD-based sample | ✓ | ✓ | ✓ | |
| face-detection-retail-0004 | Smart classroom | ✓ | ✓ | | |
| landmarks-regression-retail-0009 | Smart classroom | ✓ | ✓ | ✓ | ✓ |
| face-reidentification-retail-0095 | Smart classroom | ✓ | ✓ | | |
| person-detection-action-recognition-0003 | Smart classroom | ✓ | ✓ | | |
| human-pose-estimation-001 | Human pose estimation | ✓ | ✓ | | |
| single-image-super-resolution-0063 | Super resolution | ✓ | | | |
| single-image-super-resolution-1011 | Super resolution | ✓ | | | |
| single-image-super-resolution-1021 | Super resolution | ✓ | | | |
| text-detection-0001 | Text Detection | ✓ | ✓ | | |

## Computational details for Object Detection models

| Model name | Complexity (GFLOPs) | Size (Mp) | Face | Person | Vehicle | Bike | License plate |
|---|---|---|---|---|---|---|---|
| face-detection-adas-0001 | 1.4 | 1.1 | X | | | | |
| face-detection-retail-0004 | 1.1 | 0.6 | X | | | | |
| face-person-detection-retail-0002 | 2.8 | 0.8 | X | X | | | |
| person-detection-retail-0001 | 12.6 | 3.2 | | X | | | |
| person-detection-retail-0013 | 3.9 | 1.9 | | X | | | |
| pedestrian-detection-adas-0002 | 1.5 | 1.2 | | X | | | |
| pedestrian-and-vehicle-detector-adas-0001 | 4.0 | 1.6 | | X | X | | |
| vehicle-detection-adas-0002 | 1.4 | 1.1 | | | X | | |
| person-vehicle-bike-detection-crossroad-0078 | 3.9 | 1.2 | | X | X | X | |
| vehicle-license-plate-detection-barrier-0007 | 3.0 | 1.1 | | | X | | X |

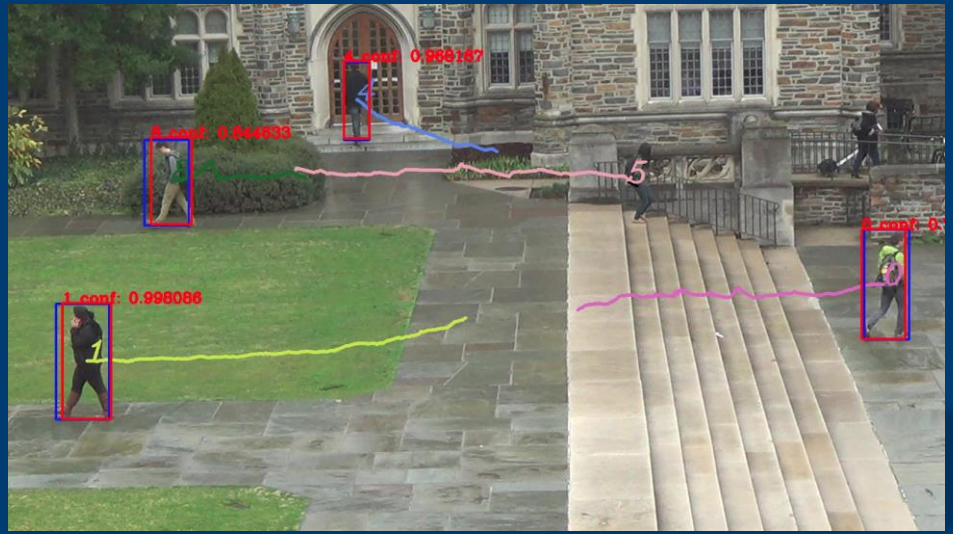
# FACIAL LANDMARKS



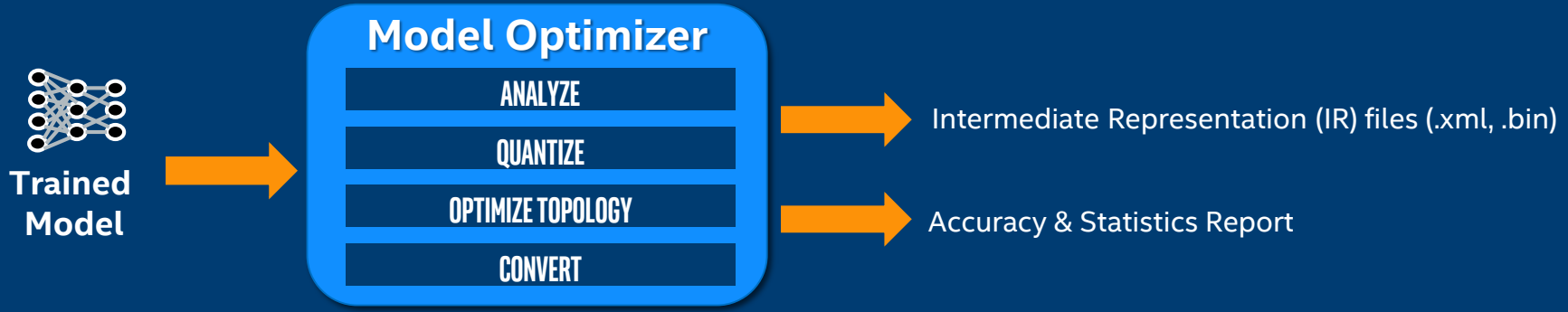

- Read to use (IR files comes with installation)
- Sample code included in the toolkit
- Custom architecture
- CNN facial landmarks estimation
- **35 landmarks**
- Can’t be re-trained at the moment

# PEDESTRIAN TRACKER

- Example of complex tracking pipeline
- Uses DL detector and re-identification networks for accurate tracking
- Model can be modified to fit use case needs (different types of object, different typical speed of objects, etc.)



# MODEL OPTIMIZER




- Python*-based workflow does not require rebuilding frameworks
- Import Models from various supported frameworks - Caffe*, TensorFlow*, MXNet*, ONNX*, Kaldi*.
- More than 100 models for Caffe, MXNet and TensorFlow validated.
  - All public models on ONNX* model zoo supported.
- With support for Kaldi, the model optimizer extends inferencing for non-vision networks.
- IR files for models using standard layers or user-provided custom layers do not require Caffe.
- Fallback to original framework is possible in cases of unsupported layers, but requires original framework

# MODEL OPTIMIZER PURPOSES

**Convert**

- Map Framework specific model format to unified IR format
- IR format is DLDT serialization format that consist of two files:
  - XML file for topology description (human-readable)
  - BIN file for weights/biases.
- There is *NO* one-to-one correspondence between every framework layer and some IR layer
- Need for framework-specific translation techniques (straightforward for Caffe than TensorFlow)

**Optimize**

- Hardware independent optimization
- No need to implement similar optimization techniques in each HW plug-in inside IE
- Frequently, model conversion means optimization: TensorFlow patterns

# MODEL OPTIMIZER STAGES

**Load**

- Parse a framework-specific model.
- Build NetworkX graph for further transformations

**Front**

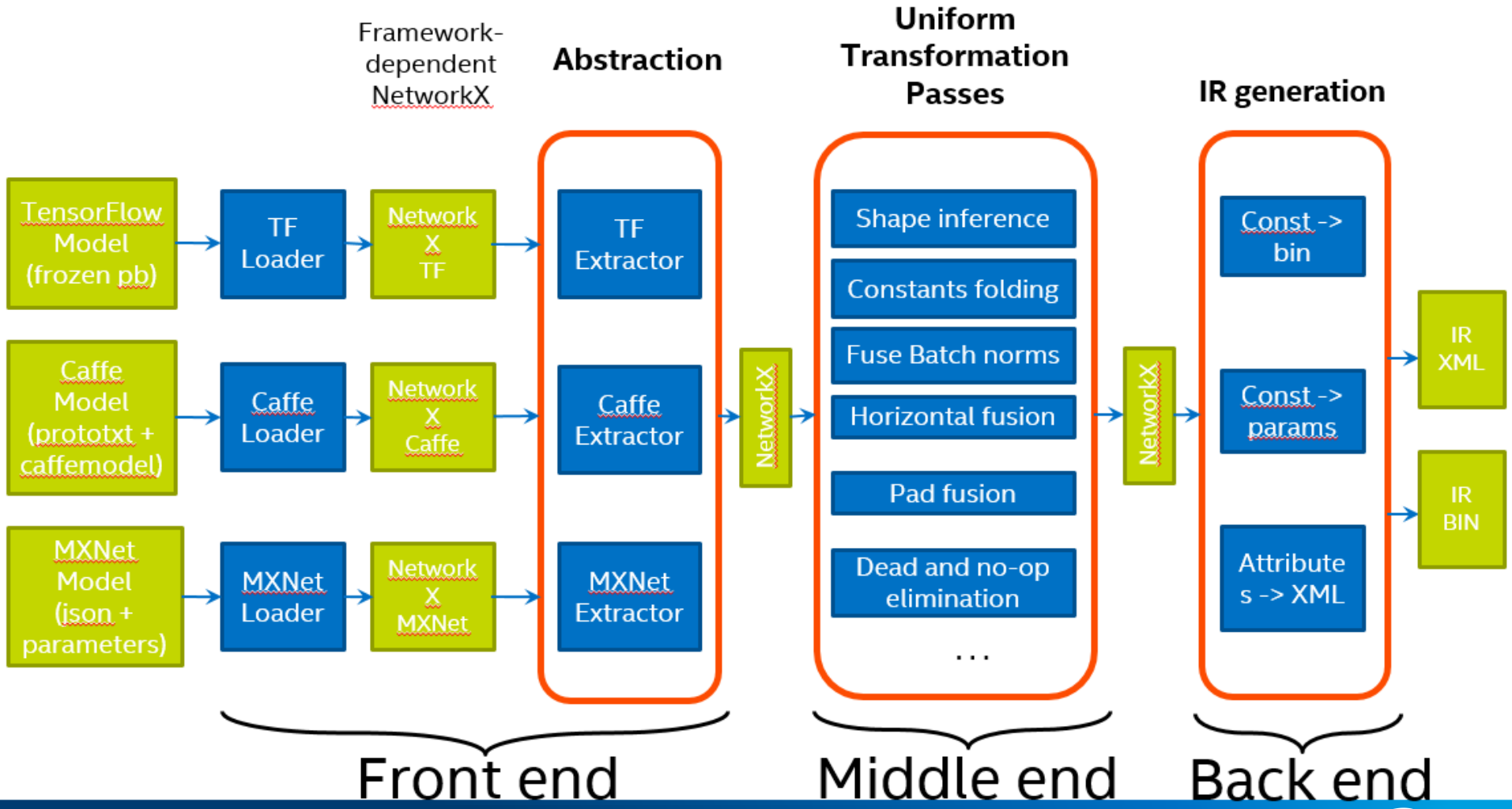
- Decode framework-specific attributes to represent them in a unified way
- Replace framework-specific specific patterns to represent them with unified set of operations

**Middle**

- Calculate and propagate shapes
- Transform graph to leave only operations (ops) that are supported by the target IR format
- **Optimize**: propagate constants, fuse operations, eliminate dead parts and ops that don't have effect (dropout)

**Back**

- Finalize graph transformation to completely fit to IR requirements
- Emit final XML and BIN files



# XML FILE

```
<net batch="1" name="AlexNet" version="2">
  <layers>
    <layer id="1" name="data" precision="FP32" type="Input">
      <output>
        <port id="1">
          <dim>1</dim>
          <dim>3</dim>
          <dim>227</dim>
          <dim>227</dim>
        </port>
      </output>
    </layer>
    <layer id="2" name="conv1" precision="FP32" type="Convolution">
      <data dilation-x="1" dilation-y="1" group="1"
            kernel-x="11" kernel-y="11" output="96" pad-x="0" pad-y="0"
            stride-x="4" stride-y="4"/>
      <input>
        <port id="2">
          <dim>1</dim>
          <dim>3</dim>
          <dim>227</dim>
          <dim>227</dim>
        </port>
      </input>
      <output>
        <port id="3">
          <dim>1</dim>
          <dim>96</dim>
```

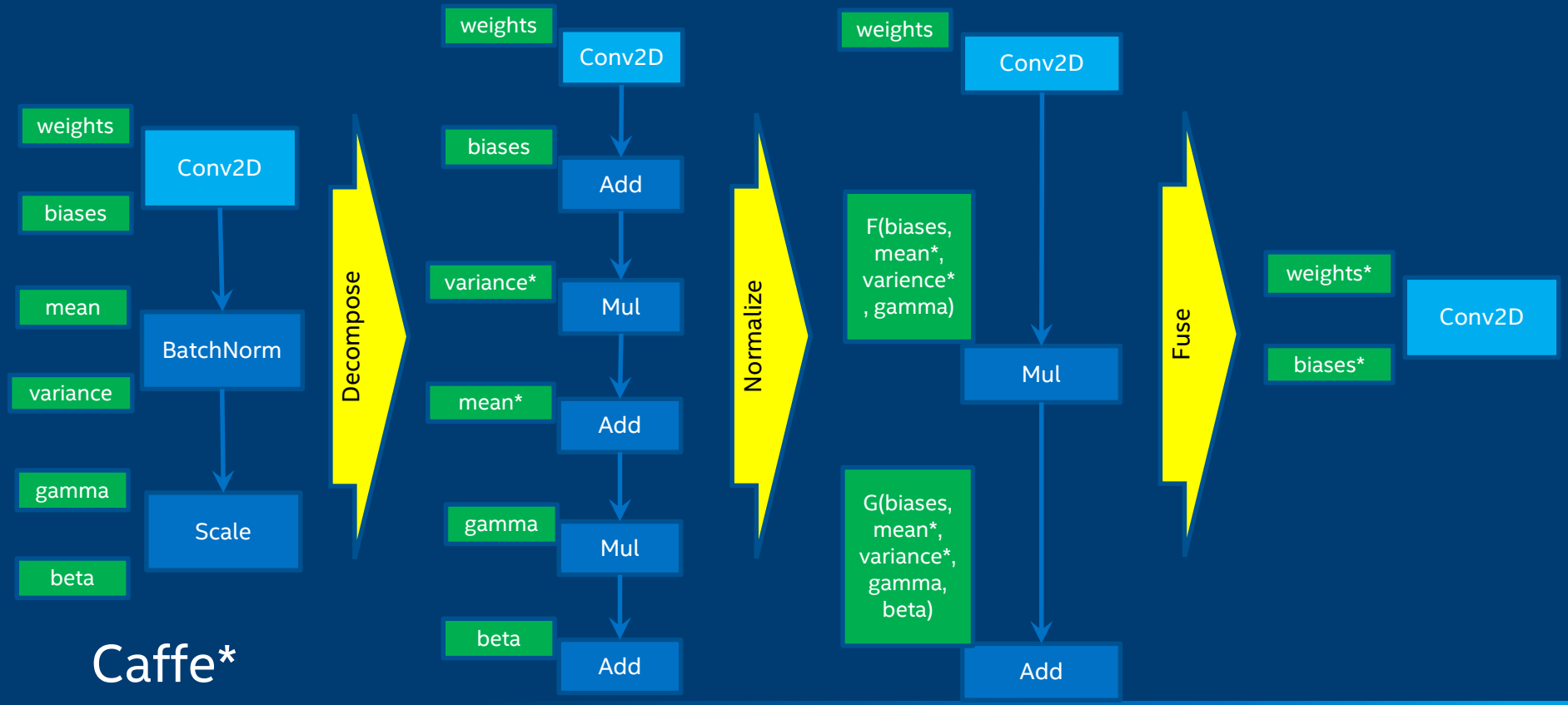
**Input Dimensions**

**Convolution Parameters**

# OPTIMIZATIONS BY MODEL OPTIMIZER

- L2 normalization pattern
- MVN (Mean Variance Normalization) pattern
- Leaky ReLU pattern
- SquaredDifference pattern
- Various batch normalization variants decomposition
- Linear operation fusion (aka 'Normalize' stage from batch norm fusion) - Works for Conv, FC, fuses from the top and the bottom
- Trivial arithmetic optimization like multiplication by 1 and add with 0
- Reshape elimination
- Constant folding
- Convolution/deconvolution grouping
- Horizontal fusion

# BATCH NORMALIZATIONS FUSION: DECOMPOSE, NORMALIZE AND FUSE



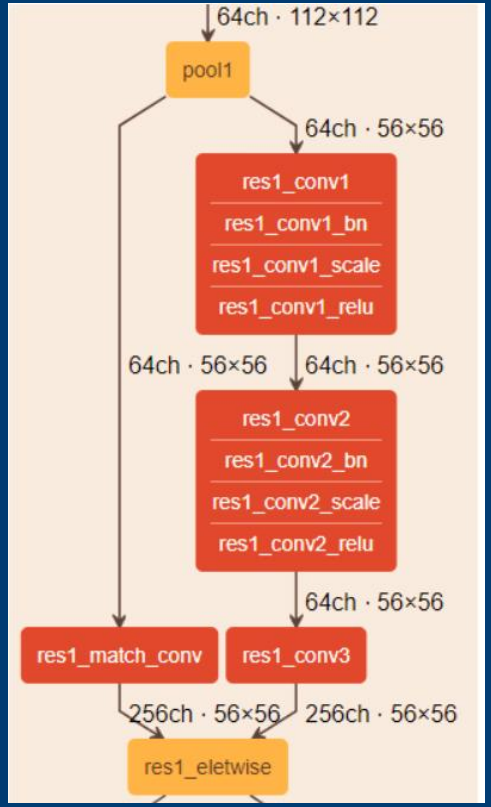

Caffe*

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

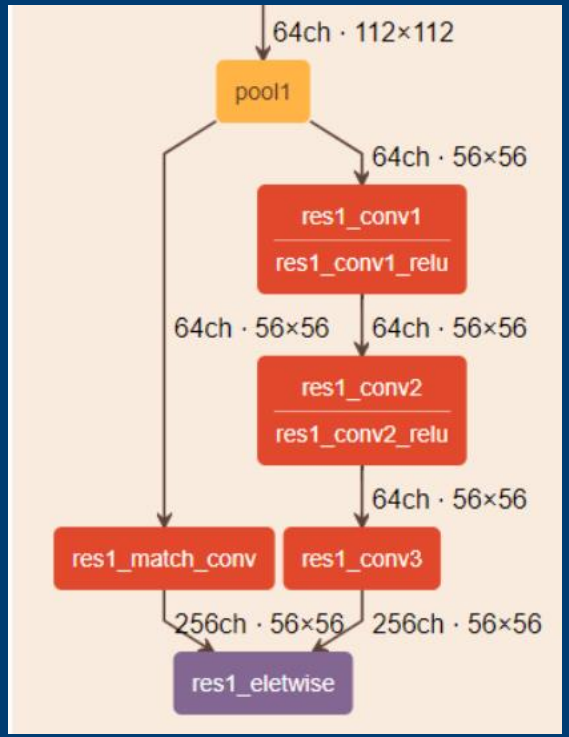
# BATCH NORMALIZATION FUSION: RESULTS

| MODEL | CPU FP32, FPS | | GPU FP16, FPS | | CPU SPEEDUP % | GPU SPEEDUP % |
|---|---|---|---|---|---|---|
| | NO FUSION | FUSION | NO FUSION | FUSION | | |
| ALEXNET | 76 | 76 | 137 | 137 | 0.0 | 0.0 |
| TF INCEPTION V1 | 86 | 99 | 59 | 105 | 15.4 | 77.0 |
| TF INCEPTION V2 | 66 | 76 | 50 | 78 | 13.9 | 57.3 |
| TF INCEPTIONRESNET V2 | 9 | 12 | 7 | 9 | 30.0 | 23.8 |
| TF RESNET V2 50 | 36 | 43 | 28 | 41 | 20.3 | 48.4 |
| TF MOBILENET V1 1_1_0_224 | 190 | 231 | 67 | 137 | 21.9 | 102.8 |
| TF MOBILENET V2 1_1_0_224 | 213 | 338 | 52 | 109 | 58.6 | 111.4 |
| TF SSD MOBILENET V1 | 94 | 97 | 47 | 59 | 3.6 | 24.8 |
| TF SSD INCEPTION V2 | 30 | 30 | 33 | 33 | 0.0 | 0.0 |
| CAFFE SQUEEZENET V1.1 | 352 | 352 | 214 | 239 | 0.0 | 11.7 |

# FUSED RESNET269 - EXAMPLE




Caffe Resnet269 block



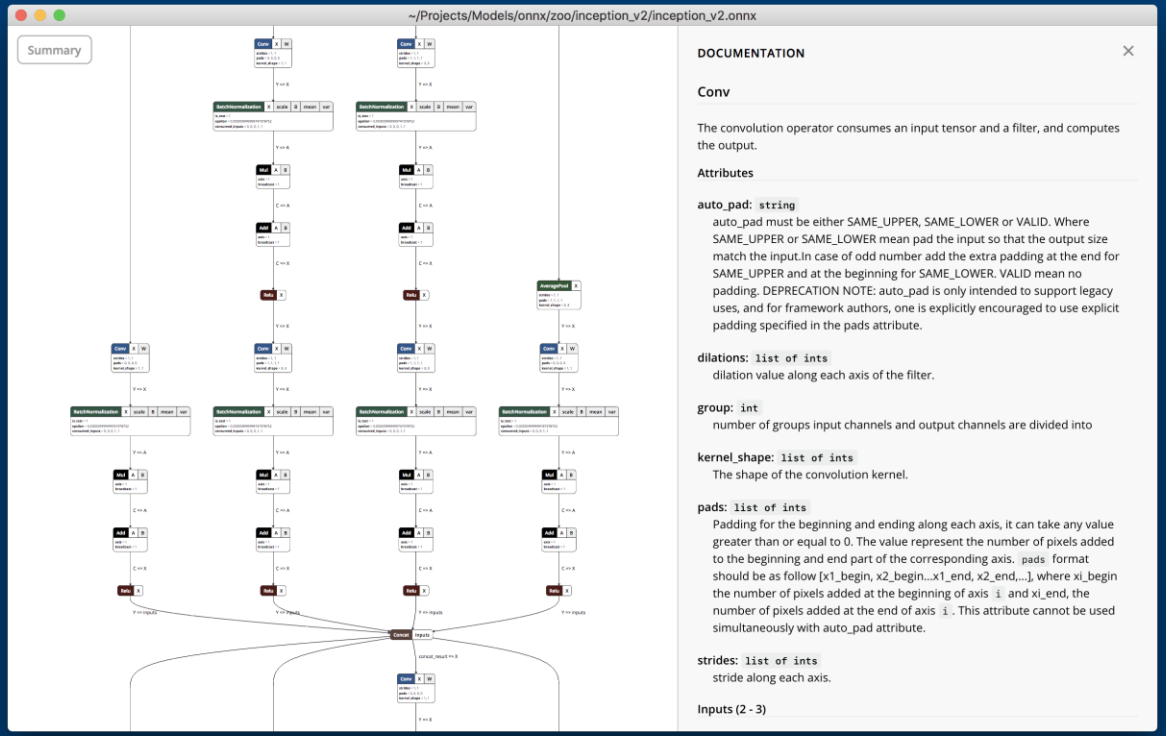

Fused Caffe Resnet269 block

- (Try - -disable_fusing to check the difference)

# DEEP LEARNING VISUALIZATION (NETRON)

- **NETRON** Deep Learning and Machine Learning Visualization tool supports IR files generated with Model Optimizer

- https://github.com/lutzroeder/netron

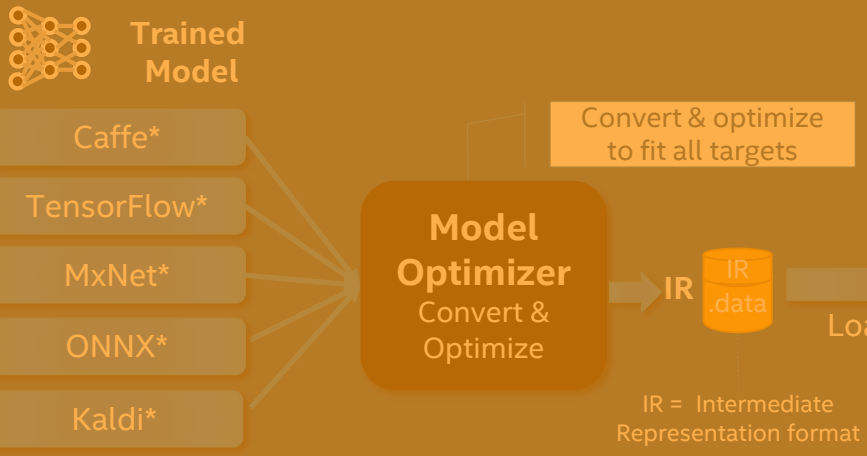


# INTEL® DEEP LEARNING DEPLOYMENT TOOLKIT

## TAKE FULL ADVANTAGE OF THE POWER OF INTEL® ARCHITECTURE

### Model Optimizer

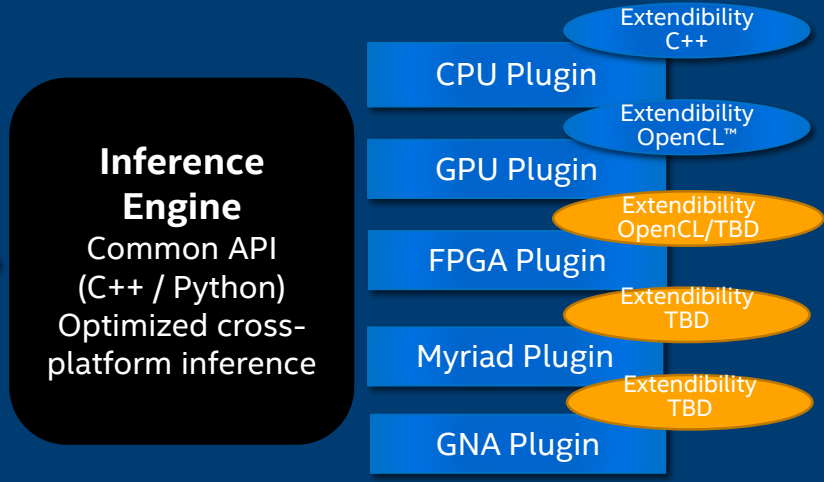
- **What it is**: Preparation step -> imports trained models
- **Why important**: Optimizes for performance/space with conservative topology transformations; biggest boost is from conversion to data types matching hardware.




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

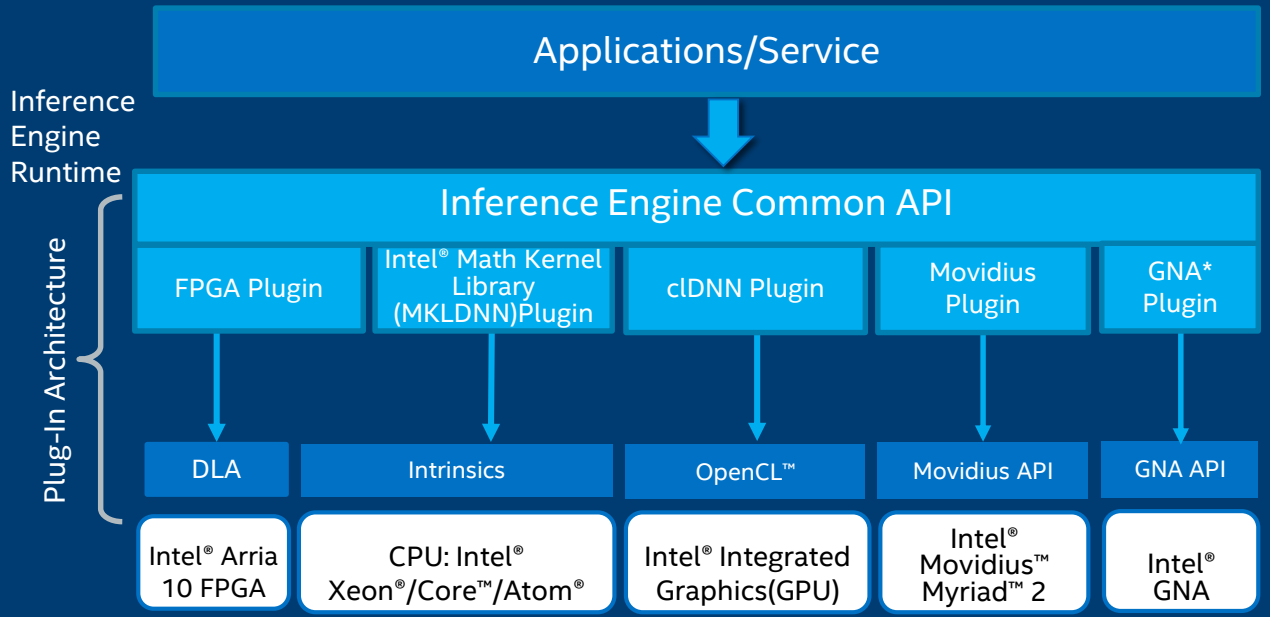
### Inference Engine

- **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.



# 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)
- Heterogeneity support allows execution of layers across hardware types
- Asynchronous execution improves performance
- Futureproof/scale your development for future Intel® processors




Transform Models & Data into Results & Intelligence

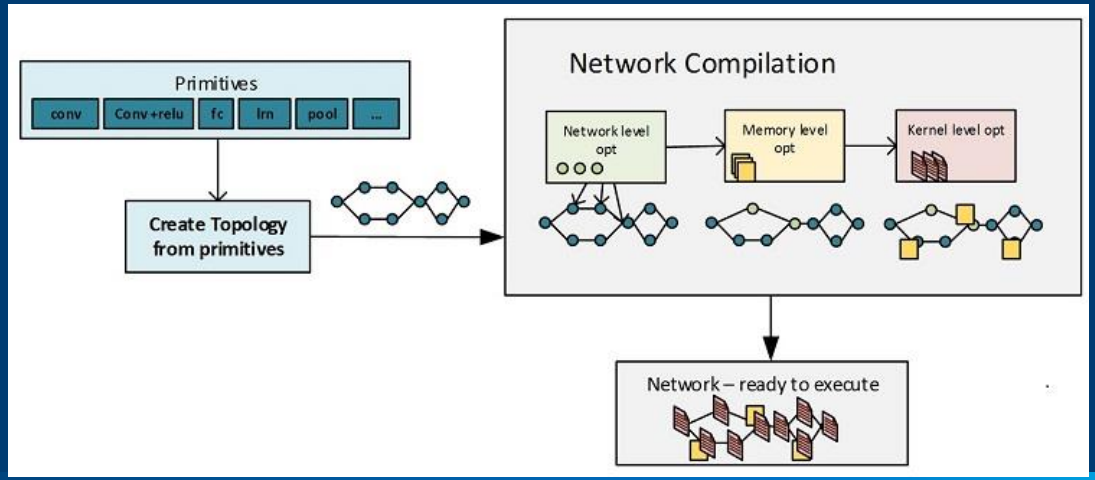
GPU = Intel CPU with integrated graphics processing unit/Intel® Processor Graphics/GEN
GNA = Gaussian mixture model and Neural Network Accelerator

# INFERENCE ENGINE

- IE is an API to deliver inference solutions on the platform of your choice: **CPU, GPU, VPU** or **FPGA**
- Loads converted/optimized IR files to execute neural network topology for inference
  - Asynchronous Execution
  - Heterogeneous Workload
  - Dynamic Batch Support
  - Supported  HW acceleration (SSE/AVX/..)
  - Optimization examples
  - Parameters/Configuration options

# INFERENCE ENGINE

- Device specific optimization is performed by the plug-in's librairies: MKL-DNN, clDNN, Movidius, DLA (Deep Learning Acceleration Suite)
- 3 main types of optimizations:
  - **Primitive/Kernel Optimization**
  - **Memory Optimization**
  - **Network Optimization**



# INFERENCE ENGINE

## Layers supported by Inference-Engine plug-ins

- MKL-DNN and cl-DNN are open source
- **CPU - MKLDNN Plugin**
  - Supports FP32
  - Supports Intel® Xeon®/Core®/Atom® platforms (https://github.com/01org/mkl-dnn )
- **GPU – clDNN Plugin**
  - Supports FP32 and FP16 (recommended for most topologies)
  - Supports Gen9 and above graphics architectures (https://github.com/01org/clDNN)
- **FPGA – DLA Plugin**
  - Supports Intel FPGA Devices Plugins
- **Movidius – MyriadPlugin**
  - Set of layers are supported on Myriad2/MyriadX, non supported layers must be inferred through other IE plugins
  - Support FP16
- The complete list of supported layers can be seen on documentation.

# INTEL® DISTRIBUTION OF OPENVINO™ CODE SAMPLES

## DL/IE Examples

- **Image Classification (AlexNet/ GoogLeNet)**
- **Image Segmentation ( FCN8.)**
- **Object Detection (Faster R-CNN )**
- **Object Detection for Single Shot Multibox Detector (SSD)**
- **Neural Style Transfer**
- **Validation Application (check model accuracy)**

## OpenCV

- People Detection (HOG)
- Colorization (DNN)
- Custom OpenCL™ Kernel
- Dense Optical Flow
- Facial Recognition

## OpenVX

- Auto Contrast
- Custom OpenCL Kernel
- Heterogeneous Basics
- Advanced OpenVX
- Domain specific OpenVX workloads

# INTEL® DISTRIBUTION OF OPENVINO™ ON DOCKER

## Dockers Containers

- Relies on Linux kernel features, architecture agnostic
- Creates and manages isolated processes
- Allows to run several containers with different run-times on the same host in the same time
- All containers connected to host over virtual network (NAT, Bridge, etc)
- Processes in Docker use the host kernel and it's capabilities
- Advanced hardware and kernel capabilities are isolated by default, but can be enabled for processes in container

# INTEL® DISTRIBUTION OF OPENVINO™ ON DOCKER

## CPU

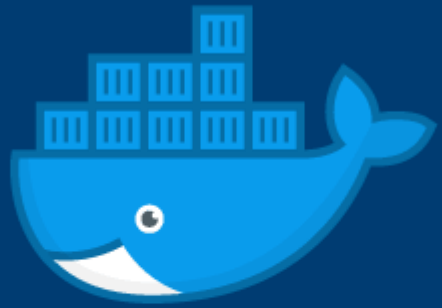
- Kernel reports the same CPU, memory. All instructions available to host also available for process in container, including AVX2, AVX512, etc.
- No virtualization penalties

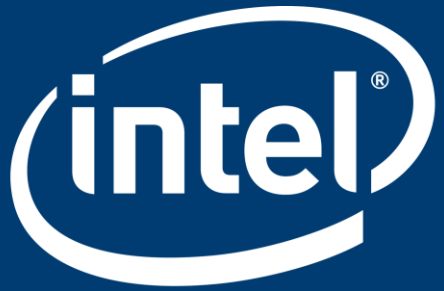
## GPU

- Not available in container by default, but can be tuned:
- Kernel driver should be installed on the host, if any OpenCL runtime should be included into container
- User in container should be in group “video”

## Myriad (Movidius)

- Device changes it’s VendorID and DeviceID during execution and looks for host system as brand new device each time. It means it cannot be mounted as usual
- UDEV events are not forwarded to container by default it does not know about device reconnection, only one device per host supported




Software




# HANDS-ON LAB 1

# HANDS-ON LAB

OpenVINO™

1. Log-in to your lab PC (intel/**P@ssw0rd**)
2. Open Firefox and **goto: localhost:8888**, run Jupyter Lab interface
3. Navigate to: **`/home/intel/Workshop`**
4. Click on "**Introduction to OpenVINO – Lab1.ipynb"**


Jupyter Notebook:

- Jupyter notebook is an interactive scripting environment with Markdown support.
- Code part is active and runs on its own environment settings.

# WHAT WE WILL COVER?

OpenVINO™

- Run Sample Applications
- Use Model Downloader & Model Optimizer
- Introduction to Inference Engine with Python API.