FROM TRAINING TO INFERENCE: CREATE AN END-TO-END DEEP LEARNING PROJECT USING OPTIMIZED HARDWARE AND SOFTWARE FROM INTEL

San Francisco, 23/24th May 2018

LEGAL NOTICES & DISCLAIMERS

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 http://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 and others are trademarks of Intel Corporation in the U.S. and/or other countries.
*Other names and brands may be claimed as the property of others.

AGENDA

- Intel® AI Academy
- Intel® AI Portfolio
- Overview of Intel® Optimized Caffe* and Tensorflow*
- Intel AI Use Cases
- Training on Tensorflow* with Intel optimizations
- Validation on the Intel® Movidius Neural Compute Stick (NCS) - Demo
- Deploy to an edge device (Raspberry Pi) – Demo

QUESTIONS? ASK US!

SULAIMON IBRAHIM
Developer Evangelist

Sulaimon.ibrahim@intel.com

RUDY CAZABON
Developer Evangelist

rudy.cazabon@intel.com

MEGHANA RAO
Developer Evangelist

meghana.s.rao@intel.com

MICHAEL HERNANDEZ
Developer Evangelist

michael.j.hernandez@intel.com

INTEL® AI PORTFOLIO

AI PORTFOLIO

SOLUTIONS

Data Scientists | Technical Services | Reference Solutions

PLATFORMS

Intel® AI DevCloud

Intel® Deep Learning System‡

Intel Saffron™ REASONING

TOOLS

Intel® Deep Learning Studio‡

Intel® Deep Learning Deployment Toolkit†

Intel® Computer Vision SDK†

Intel® Movidius™ Software Development Kit (SDK)

FRAMEWORKS

TensorFlow* Caffe* mxnet* BigDL on Spark* neon Caffe2*‡ PYTORCH*‡ Microsoft CNTK*‡ PaddlePaddle*‡

LIBRARIES

Intel® MKL/MKL-DNN, clDNN, DAAL, Intel Python Distribution, etc.

DIRECT OPTIMIZATION

Intel® nGraph™ Library†

CPU Transformer† | NNP Transformer‡ | Other

TECHNOLOGY

Intel Xeon | Intel Nervana | Intel Arria 10 | Intel Atom | Intel Core i7 | Intel Iris Graphics | Intel Movidius

END-TO-END COMPUTE

SYSTEMS & COMPONENTS

†Beta available
‡ Future
*Other names and brands may be claimed as the property of others.

intel AI

INTEL AI FRAMEWORKS

Popular DL Frameworks are now optimized for CPU!

See installation guides at ai.intel.com/framework-optimizations/

More under optimization: Caffe2* PYTORCH* Microsoft CNTK* PaddlePaddle* *and others to be enabled via Intel® nGraph™ Library*

SEE ALSO: Machine Learning Libraries for Python (Scikit-learn, Pandas, NumPy), R (Cart, randomForest, e1071), Distributed (MlLib on Spark, Mahout)
**Limited availability today*
Other names and brands may be claimed as the property of others.

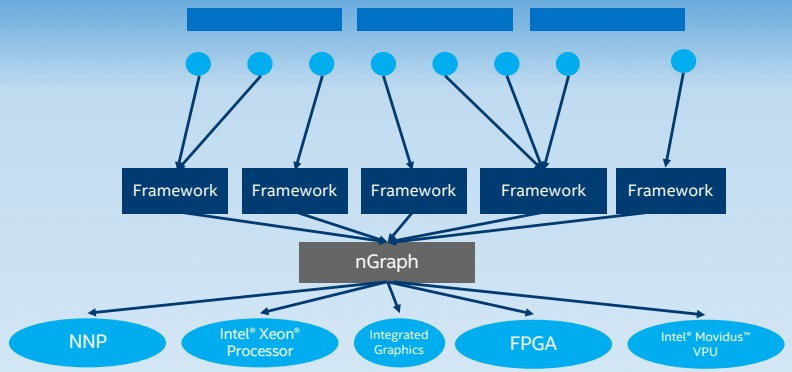
INTEL AI LIBRARIES

DIRECT OPTIMIZATION

MKL-DNN	Open-source optimized deep neural network functions for new frameworks
clDNN	Open-source optimized deep neural network functions for Intel GPUs
DAAL	Data Analytics Acceleration Library for analytics and machine learning
Intel Python Distribution	Optimized distribution of most popular & fastest growing language for machine learning

INTEL® NGRAPH™ LIBRARY

BETA



Translates participating deep learning framework compute graphs into hardware-optimized executables for many different targets (CPU, GPU, NNP, FPGA, VPU, etc.)

DEEP LEARNING FRAMEWORK OPTIMIZED FOR IA: CAFFE*

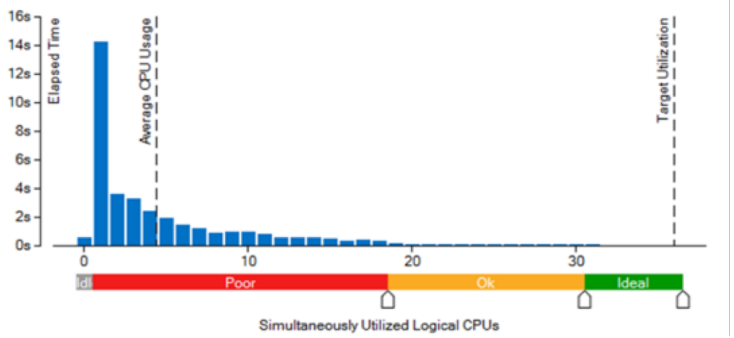
INITIAL CIFAR-10 RUN IN CAFFE—VTUNE ANALYSIS

Elapsed Time: 37.026s

CPU Time:	1306.422s
Effective Time:	162.646s
Spin Time:	1134.014s
Imbalance or Serial Spinning (OpenMP):	1100.758s
Lock Contention (OpenMP):	0.019s
Other:	33.238s
Overhead Time:	9.762s
Total Thread Count:	38
Paused Time:	0s

CPU Usage Histogram

This histogram displays a percentage of the wall time the specific number of CPUs were running simultaneously. Spin and Overhead time adds to the Idle CPU usage value.



Hardware Details:

- 36 available physical cores
- Dual-socket Intel Xeon processor E5-2699 v3 at 2.30 GHz with 18 cores/socket (HT disabled)
- 64 GB of DDR4 @ 2,133 MHz

Conclusions:

- multithreading scalability
- Only used in GEMM operations of MKL

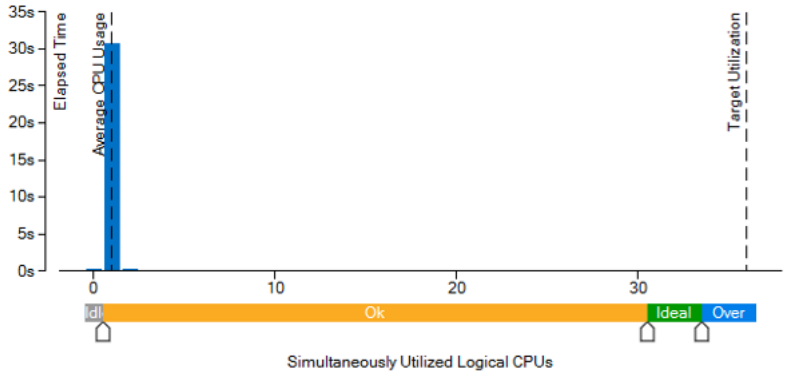
INITIAL CIFAR-10 RUN IN CAFFE—VTUNE ANALYSIS

Elapsed Time: 31.149s

- CPU Time: 31.240s
- Total Thread Count: 3
- Paused Time: 0s

CPU Usage Histogram

This histogram displays a percentage of the wall time the specific number of CPUs were running simultaneously. Spin and Overhead time adds to the Idle CPU usage value.



New Run Details:

- Export OMP_NUM_THREADS=1
- Same hardware and execution setup
- Execution time reduced (37.0s → 31.2s)

Conclusions:

- Threads re-initialization and data distribution introduce significant (15.7%) overhead
- Only used in GEMM operations of MKL

CURRENT OPTIMIZATIONS

- LEVERAGE OPTIMIZATION TOOLS & LIBRARIES
- SCALAR, SERIAL OPTIMIZATIONS
- VECTORIZATION
- THREAD PARALLELIZATION
- SCALE FROM MULTICORE TO MANY CORE

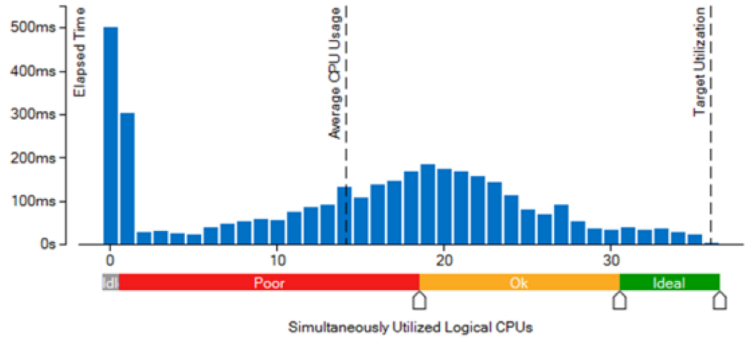
https://software.intel.com/en-us/articles/caffe-optimized-for-intel-architecture-applying-modern-code-techniques

Elapsed Time: 3.602s

CPU Time:	111.070s
Effective Time:	50.819s
Spin Time:	58.437s
Imbalance or Serial Spinning (OpenMP):	55.477s
Lock Contention (OpenMP):	0.340s
Other:	2.620s
Overhead Time:	1.814s
Total Thread Count:	37
Paused Time:	0s

CPU Usage Histogram

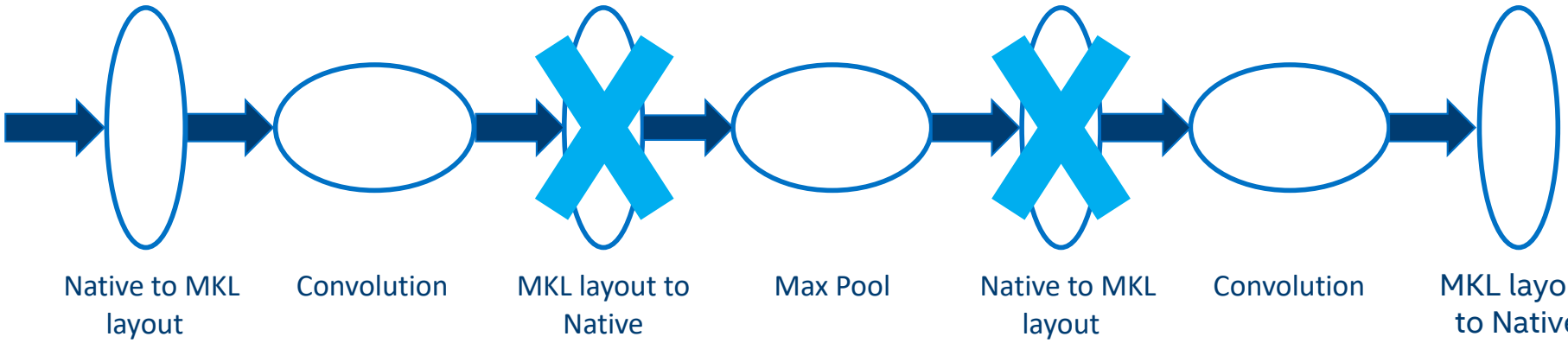
This histogram displays a percentage of the wall time the specific number of CPUs were running simultaneously. Spin and Overhead time adds to the Idle CPU usage value.



DEEP LEARNING FRAMEWORK OPTIMIZED FOR IA: TENSORFLOW*

MINIMIZE CONVERSIONS OVERHEAD

- End to end optimization can reduce conversions
- Staying in optimized layout as long as possible becomes one of the tuning goals
- Minimize the number of back and forth conversions
  - Use of graph optimization techniques



OPTIMIZING TENSORFLOW* & OTHER DL FRAMEWORKS FOR INTEL® ARCHITECTURE

Leverage High Performant Compute Libraries and Tools
e.g. Intel® Math Kernel Library, Intel® Python, Intel® Compiler etc.

Data Format/Shape:
Right format/shape for max performance: blocking, gather/scatter

Data Layout
Minimize cost of data layout conversions

Parallelism
Use all cores, eliminate serial sections, load imbalance

Memory Allocation
unique characteristics and ability to reuse buffers

Data Layer Optimizations
parallelization, vectorization, IO

Optimize Hyper Parameters
- e.g. batch size for more parallelism
- learning rate and optimizer to ensure accuracy/convergence

INITIAL PERFORMANCE GAINS ON INTEL® XEON® PROCESSORS

(2 SOCKETS INTEL® MICROARCHITECTURE CODE NAME BROADWELL—22 CORES)

Benchmark	Metric	Batch Size	Baseline Performance Training	Baseline Perf Inference	Optimized Perf Training	Optimized Perf Inference	Speedup Training	Speedup Inference
ConvNet-Alexnet	Images/ sec	128	33.52	84.2	524	1696	15.6x	20.2x
ConvNet-GoogleNet v1	Images/ sec	128	16.87	49.9	112.3	439.7	6.7x	8.8x
ConvNet-VGG	Images/ sec	64	8.2	30.7	47.1	151.1	5.7x	4.9x

- Baseline using TensorFlow* 1.0 release with standard compiler knobs
- Optimized performance using TensorFlow with Intel optimizations and built with
  - bazel build --config=mkl --copt="-DEIGEN_USE_VML"

ADDITIONAL PERFORMANCE GAINS FROM PARAMETERS TUNING

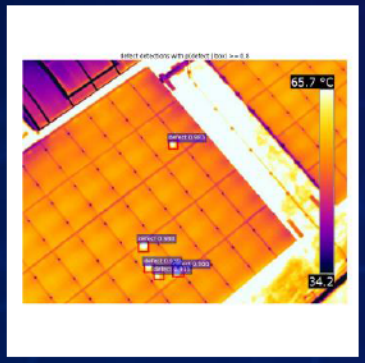
- Data Format: CPU prefers NCHW data format
- Intra_op, inter_op and OMP_NUM_THREADS: set for best core utilization
- Batch size: higher batch size provides for better parallelism
  - Too high a batch size can increase working set and impact cache/memory perf

Best Setting for Intel® Xeon® Processors (Intel® microarchitecture code name Broadwell —2 Socket—44 Cores)

Benchmark	Data Format	Inter_op	Intra_op	KMP_BLOCKTIME	Batch size
ConvNet- AlexnetNet	NCHW	1	44	30	2048
ConvNet-Googlenet V1	NCHW	2	44	1	256
ConvNet-VGG	NCHW	1	44	1	128

INTEL AI USE CASES

HIGH RISK INSPECTION BY DRONES: 1 CPU NODE



Caffe

intel XEON inside

FRAMEWORK HARDWARE

Time to train: 6 hours

Chong Y., Yiqiang Z and Jiong G., "Automatic Defect Inspection Using Deep Learning for Solar Farm" Dec. 2017. https://software.intel.com/en-us/articles/automatic-defect-inspection-using-deep-learning-for-solar-farm

DRUG DESIGN: 1 CPU NODE

- Deep learning training with huge dataset (4 Million compound-protein interactions)
- Stunning accuracy (98.2%)
- Training in 1.1 – 8.8 days

theano

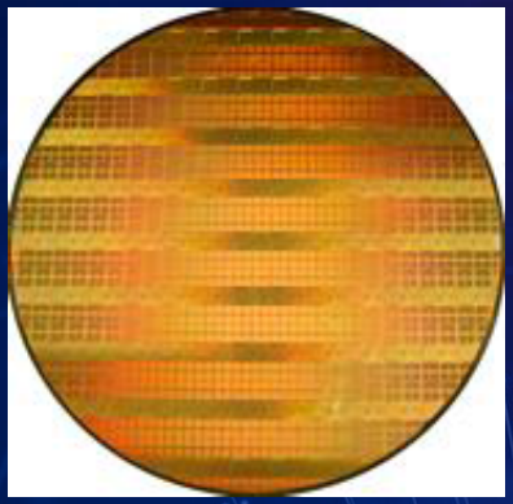
FRAMEWORK

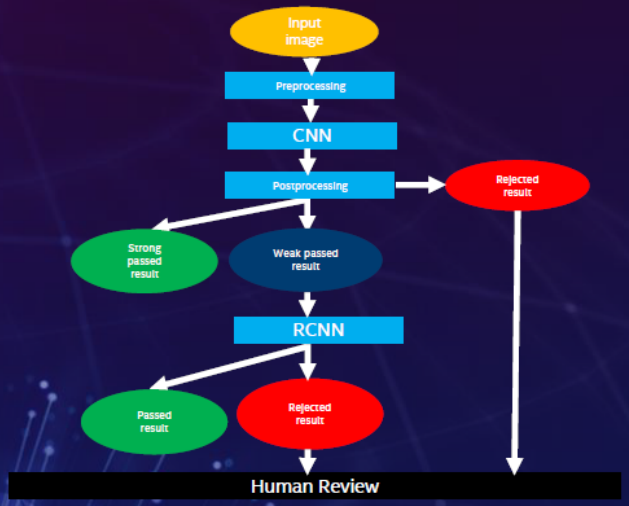
intel XEON inside

HARDWARE

M. Hamanaka et al, "CGBVS-DNN: Prediction of Compound-protein Interactions Based Deep Learning" http://onlinelibrary.wiley.com/doi/10.1002/minf.201600045/full

SILICON PACKAGE DEFECT DETECTION: 8 CPU NODES





Training within one hour on 8 CPU nodes.

Z. Yiqiang and J. Gong, "Manufacturing package fault detection using deep learning." Aug. 2017. https://software.intel.com/en-us/articles/manufacturing-package-fault-detection-using-deep-learning

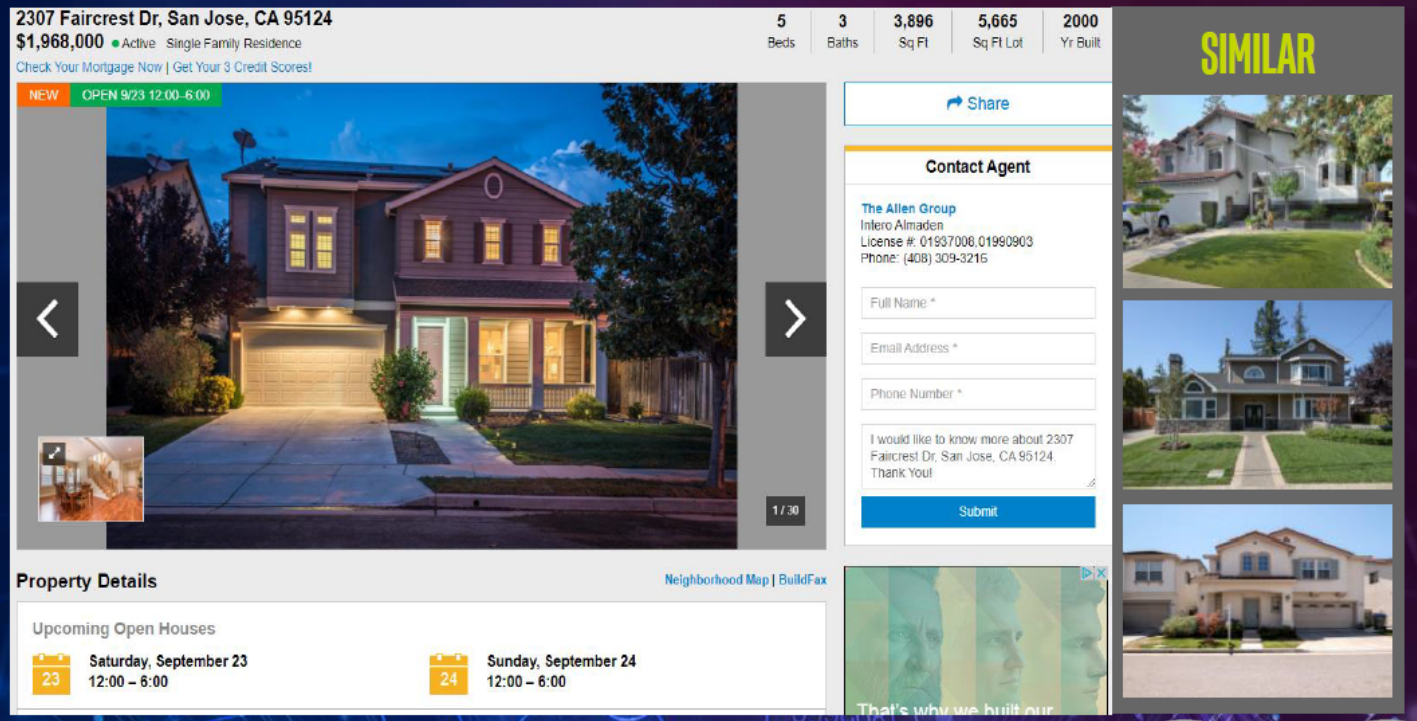
Caffe

FRAMEWORK

intel XEON inside

HARDWARE

HOME BUYING ASSISTANT: 10 CPU NODES



J. Dai, Y. Yuhao and J. Wang, "Using BigDL to build image similarity-based house recommendations." Nov. 2017.
https://software.intel.com/en-us/articles/using-bigdl-to-build-image-similarity-based-house-recommendations

FRAMEWORK HARDWARE

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

CREDIT CARD ANOMALY DETECTION: 32 CPU NODES

PAYMENT PROCESSING COMPANY

BigDL

FRAMEWORK

intel XEON inside

HARDWARE

https://www.intel.com/content/www/us/en/financial-services-it/union-pay-case-study.html

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

INTEL® AI DEVCLOUD

Intel® AI DevCloud

- A cloud hosted hardware and software platform available to 200K Intel® AI Academy members to learn, sandbox and get started on Artificial Intelligence projects
- Intel® Xeon® Scalable Processors(Intel(R) Xeon(R) Gold 6128 CPU @ 3.40GHz 24 cores with 2-way hyper-threading, 96 GB of on-platform RAM (DDR4), 200 GB of file storage
- **4 weeks of initial access, with extension based upon project needs**
- Technical support via Intel® AI Academy Support Community
- Available now to all AI Academy Members

https://software.intel.com/ai-academy/tools/devcloud

Optimized Software – No install required

- Intel® distribution of Python* 2.7 and 3.6 including NumPy, SciPy, pandas, scikit-learn, Jupyter, matplotlib, and mpi4py, Keras
- Intel® Optimized Caffe*
- Intel® Optimized TensorFlow*
- Intel Optimized Theano*
- Intel Nervana Neon*
- More Frameworks as they are optimized
  - MXNet
  - Py-Faster-RCnn

Intel® Parallel Studio XE Cluster Edition and the tools and libraries included with it:

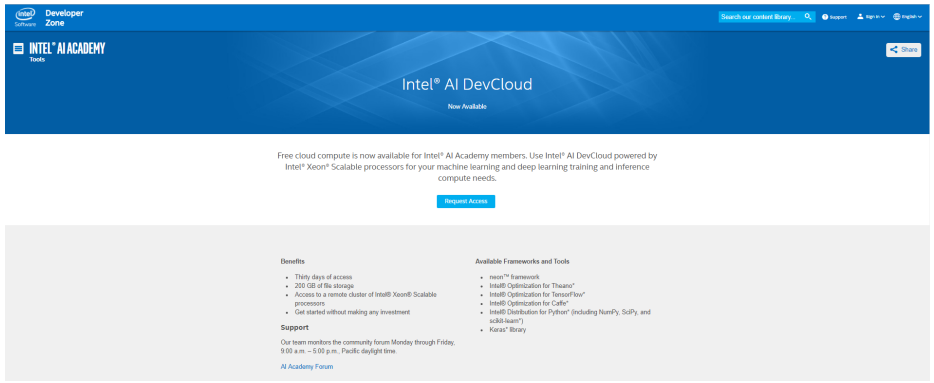
- Intel C, C++ and Fortran compilers
- Intel® MPI library
- Intel® OpenMP* library
- Intel® Threading Building Blocks library
- Intel® Math Kernel Library-DNN
- Intel® Data Analytics Acceleration Library

REQUEST ACCESS

Intel® AI DevCloud

Get Dev Cloud Access

- Click the request access button to open the application page
- Fill in the required information and submit the application
- After submitting your application, you will normally receive an email within 2 business days, including account number, node & user's guide
- Try not to loose this email it has your user and UUID = PW



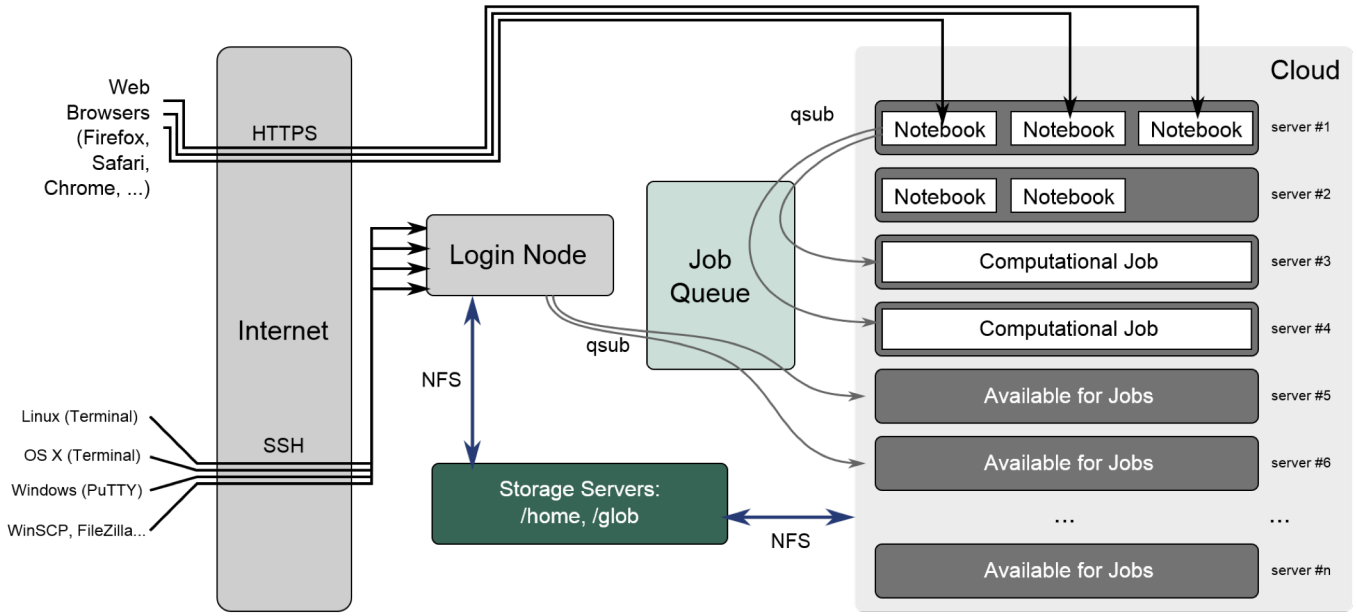
https://software.intel.com/en-us/ai-academy/tools/devcloud

CONNECT VIA TERMINAL AND JUPYTER NOTEBOOKS

Intel® AI DevCloud

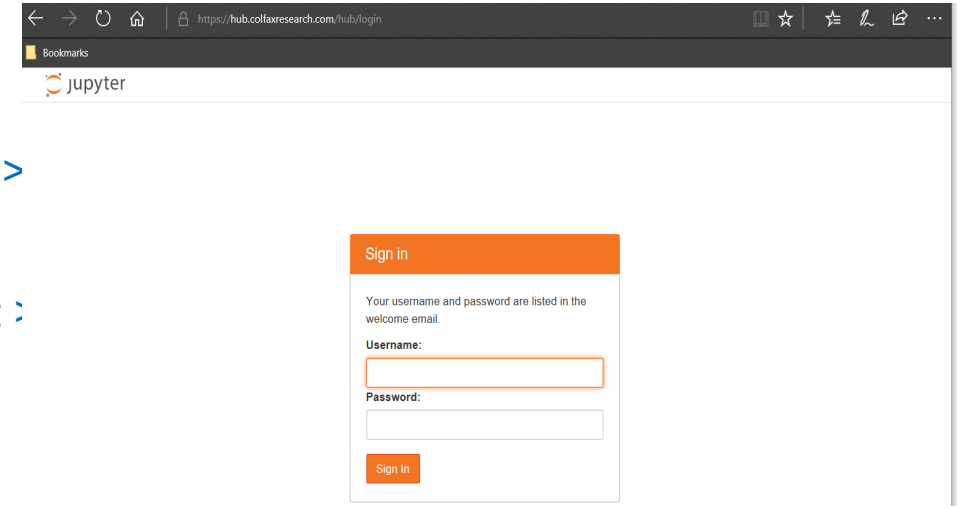
Once Connected:

- You are officially connected to the Login Node
- **This is not your compute node --- c009 is always your login node**



JupyterHUB Notebook

- Navigate to hub.colfaxresearch.com
- Username: <available on your DevCloud account>
- Password: < available on your DevCloud account >
- Refer Welcome.ipynb notebook in your home directory upon login



WE WILL USE THE JUPYTER NOTEBOOK INTERFACE FOR TODAY'S SESSION

HANDS-ON CODING:
TRAINING A CNN USING THE INTEL® AI DEVCLOUD

PROBLEM STATEMENT

Animal ID Startup

Natural and man-made disasters create havoc and grief. Lost and abandoned pets/livestock only add to the emotional toll.

How do you find your beloved dog after a flood? What happens to your daughter's horse?

Our charter is to unite pets with their families.

YOUR JOB: DATA SCIENTIST

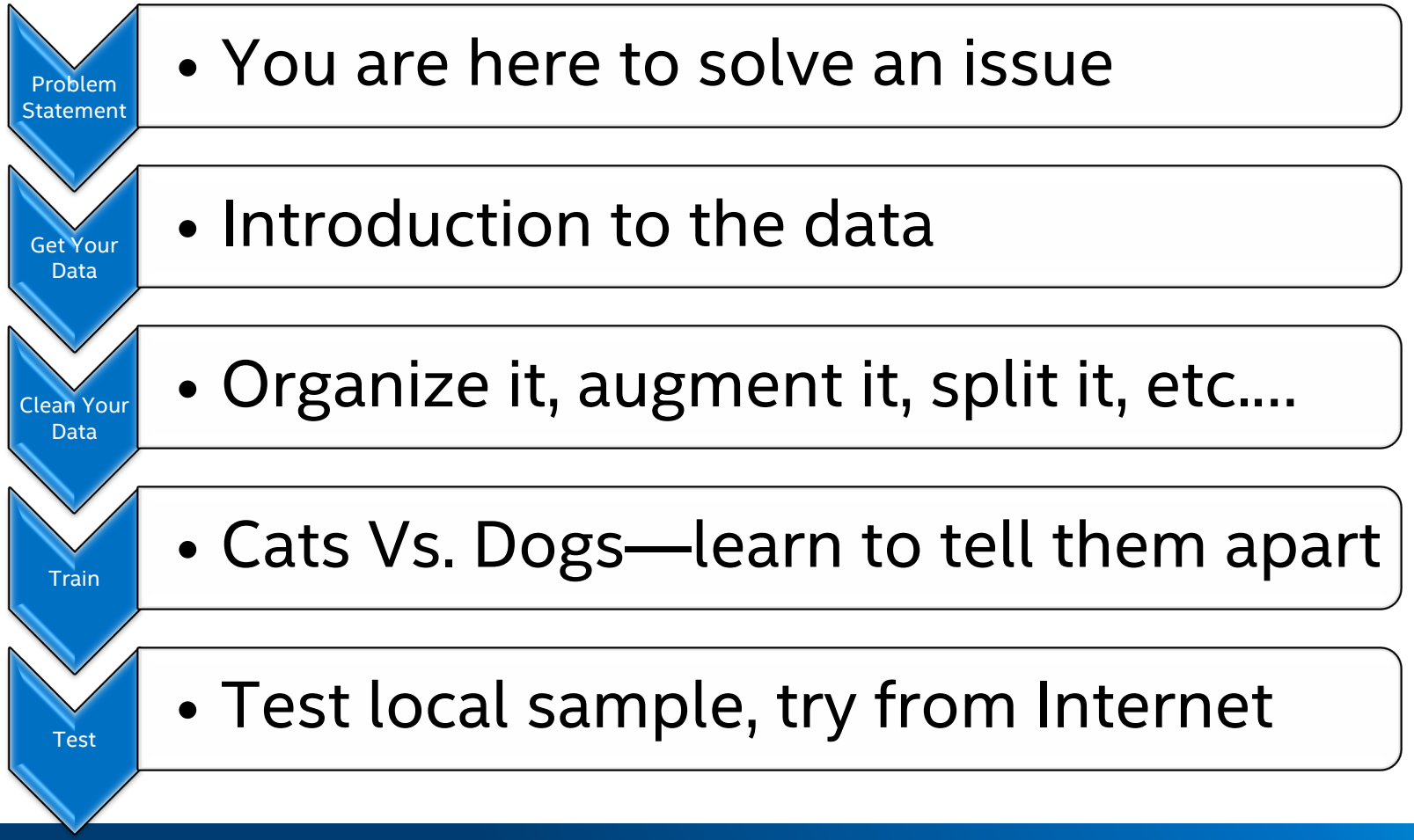
We need your help creating a way to identify animals. Initial product is focused on cat/dog breed identification. Your app will be used by rescuers and the public to document found animals and to search for lost pets.

Welcome aboard!

TENSORFLOW* WORKFLOW

TRAINING BREEDS

BASIC STEPS TO HANDS ON WORKSHOP



PART 1 - FETCH THE DATA

The Oxford Pets Database

- 37 categories
- ~200 images of each class.
- 25 Dogs
- 12 cats
- Paper talks about Data and their techniques

ISSUE COPY COMMAND

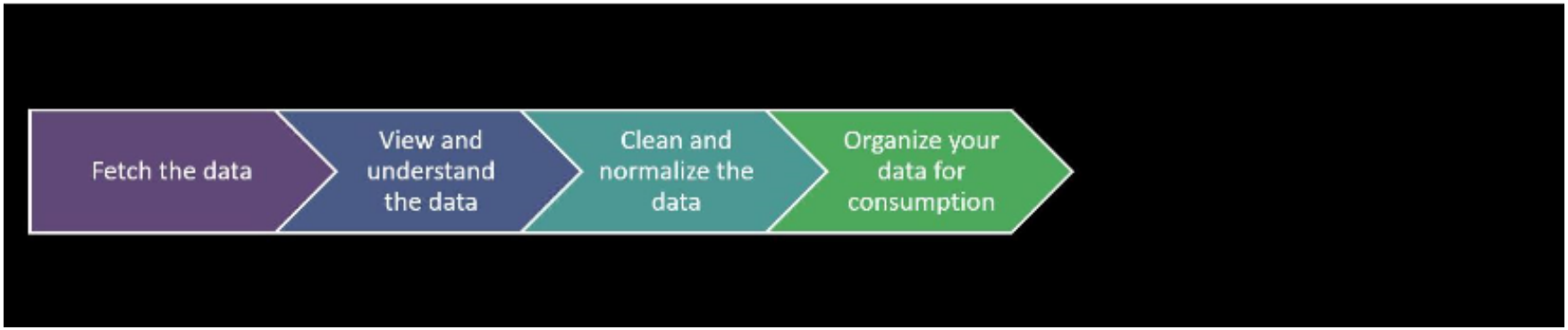
```
cp -r /data/aiworkshop/TF_Slim_Breeds/ .
```

PART 1 – VIEW THE BASELINE DATA

PART 1 – CLEAN AND NORMALIZE THE DATA

- **Extract, Transform and Load (ETL)**
  - **Data cleaning** – Eliminates noise and resolves inconsistencies in the data.
  - **Data integration** – Migrates data from various different sources into one coherent source, such as a data warehouse.
  - **Data transformation** – Standardizes or normalizes any form of data.
  - **Data reduction** – Reduces the size of the data by aggregating it.
- **Prepare data as expected by topology**
- **Ensure you have enough processing and storage capacity**

PART 1 – ORGANIZE DATA FOR CONSUMPTION BY TENSORFLOW

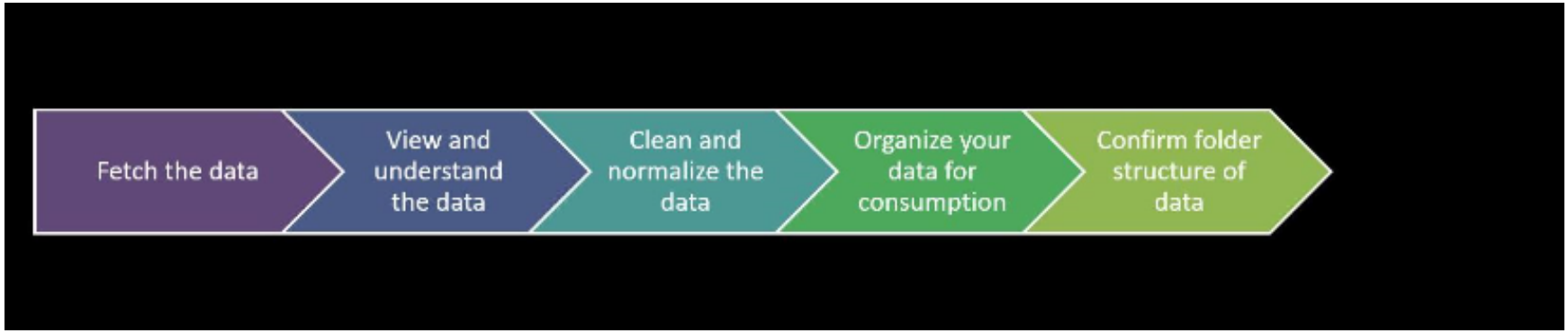


PART 1: ORGANIZE DATA FOR CONSUMPTION - CATEGORIZE

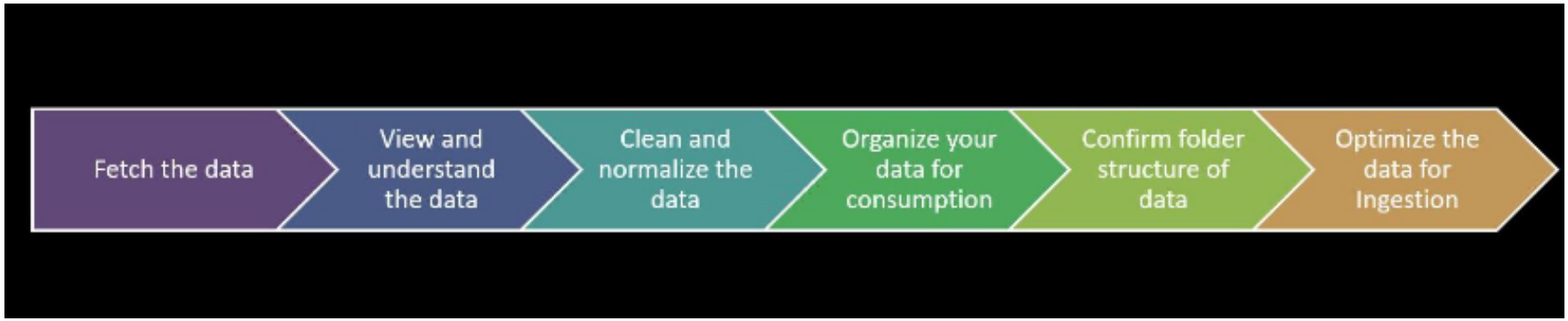
- Data organization is framework specific
- Tensorflow expects images to be organized into categories
- Once complete, each category would look something like this (and there are 39 categories)

```
breeds/
    sorted/
        british_shorthair/
            British_Shorthair_184.jpg
            British_Shorthair_269.jpg
            British_Shorthair_37.jpg
            British_Shorthair_71.jpg
            British_Shorthair_167.jpg
        japanese_chin/
            japanese_chin_167.jpg
            japanese_chin_182.jpg
            japanese_chin_191.jpg
            japanese_chin_38.jpg
            japanese_chin_17.jpg
        wheaten_terrier/
            wheaten_terrier_74.jpg
            wheaten_terrier_128.jpg
            wheaten_terrier_137.jpg
            wheaten_terrier_4.jpg
            wheaten_terrier_9.jpg
```

PART 1 – CONFIRM FOLDER STRUCTURE



PART 1: OPTIMIZE DATA FOR INGESTION



PART 1: OPTIMIZE DATA FOR INGESTION - CREATE TFRECORDS

- TFRecord is the Tensorflow recommended format for ingestion
- It is a sequence of binary strings
- If the dataset is too large, we could create multiple shards of the TFRecords to make it more manageable
- We create 2 TFRecords – One for training and another for validation

https://en.wikipedia.org/wiki/Lightning_Memory-Mapped_Database

PART 2 – TRAINING

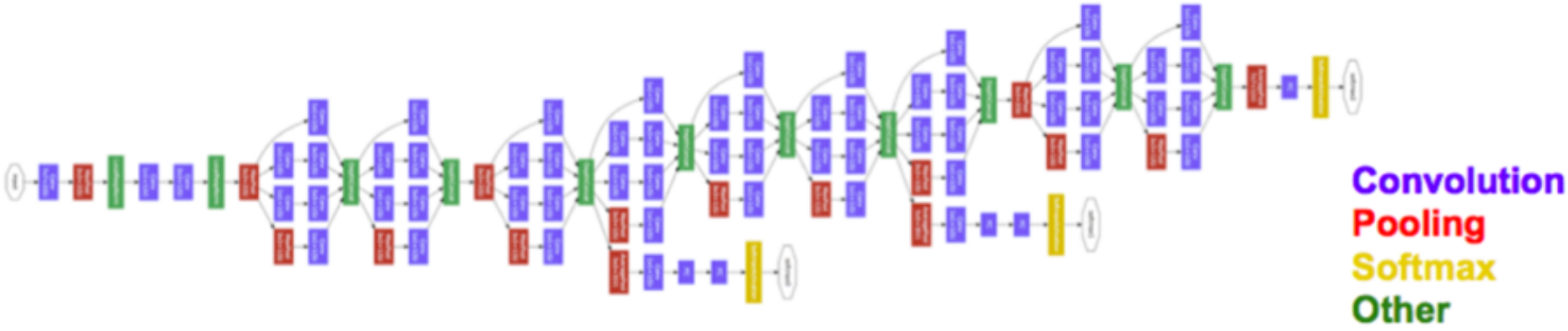
Step 1: Choose the right topology

Step 2: Setup a pre-trained model to use breeds dataset

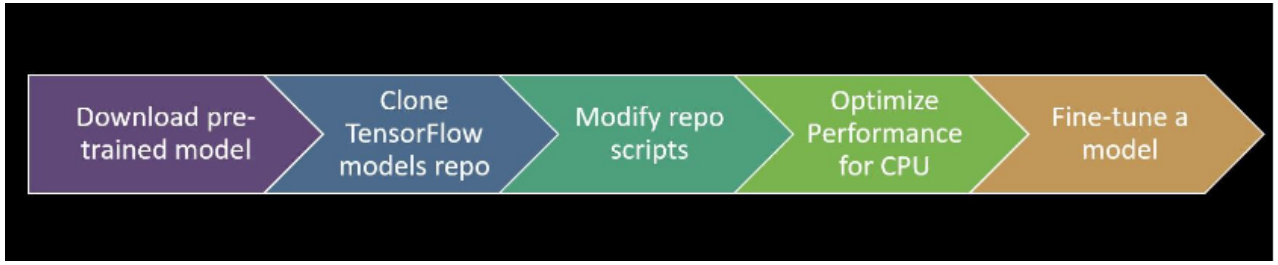
Step 3: Evaluate, freeze and test results

PART 2: STEP 1 - SELECT THE RIGHT TOPOLOGY

- **Criteria:**
  - Time to train: Depends on number of layers and computation required
  - Size: Keep in mind the edge device you want to deploy to, networks it supports and resources like memory
  - Inference speed: Tradeoff between accuracy and latency
  - **GoogLeNet Inception_V1 was our topology of choice**

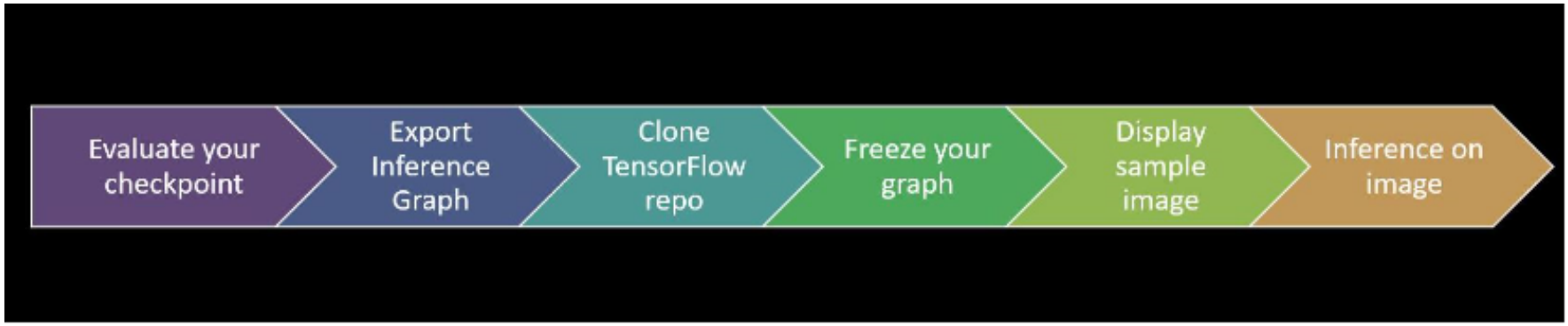


PART 2: STEP 2 – TRAINING FROM PRE-TRAINED MODEL

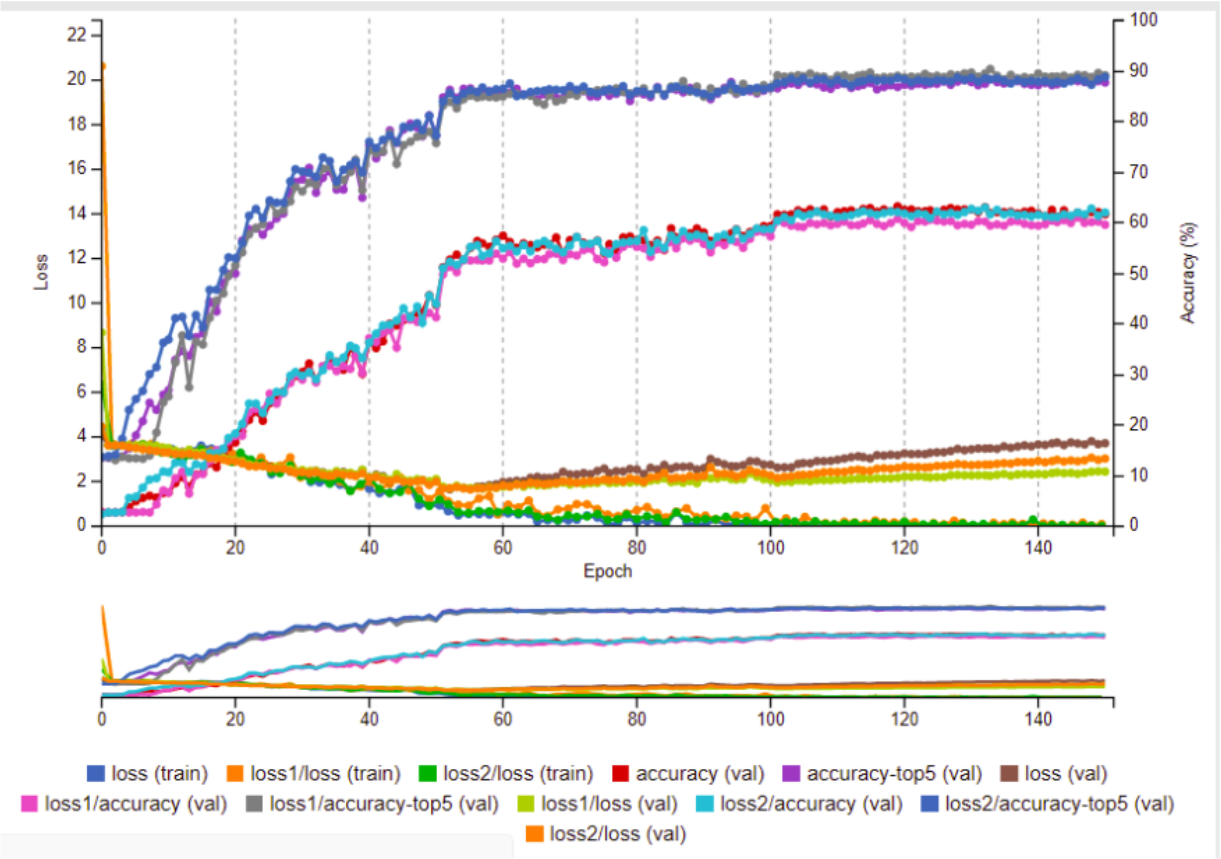


- **Clone tensorflow/models github repo**
  - We use transfer learning using a CNN pre-trained on ISLVRC-2012-CLS image classification dataset (https://github.com/tensorflow/models)
- **Modify/Add files to slim repo to work with breeds dataset**
- **Initiate training and review live training logs**
  - When using a pre-trained model on a different dataset, note that the final layer will change to indicate the new set of categories
  - Indicate which subset of layers to retrain while keeping others frozen
  - View results

PART 2: STEP 3 – EVALUATE, FREEZE GRAPH AND TEST



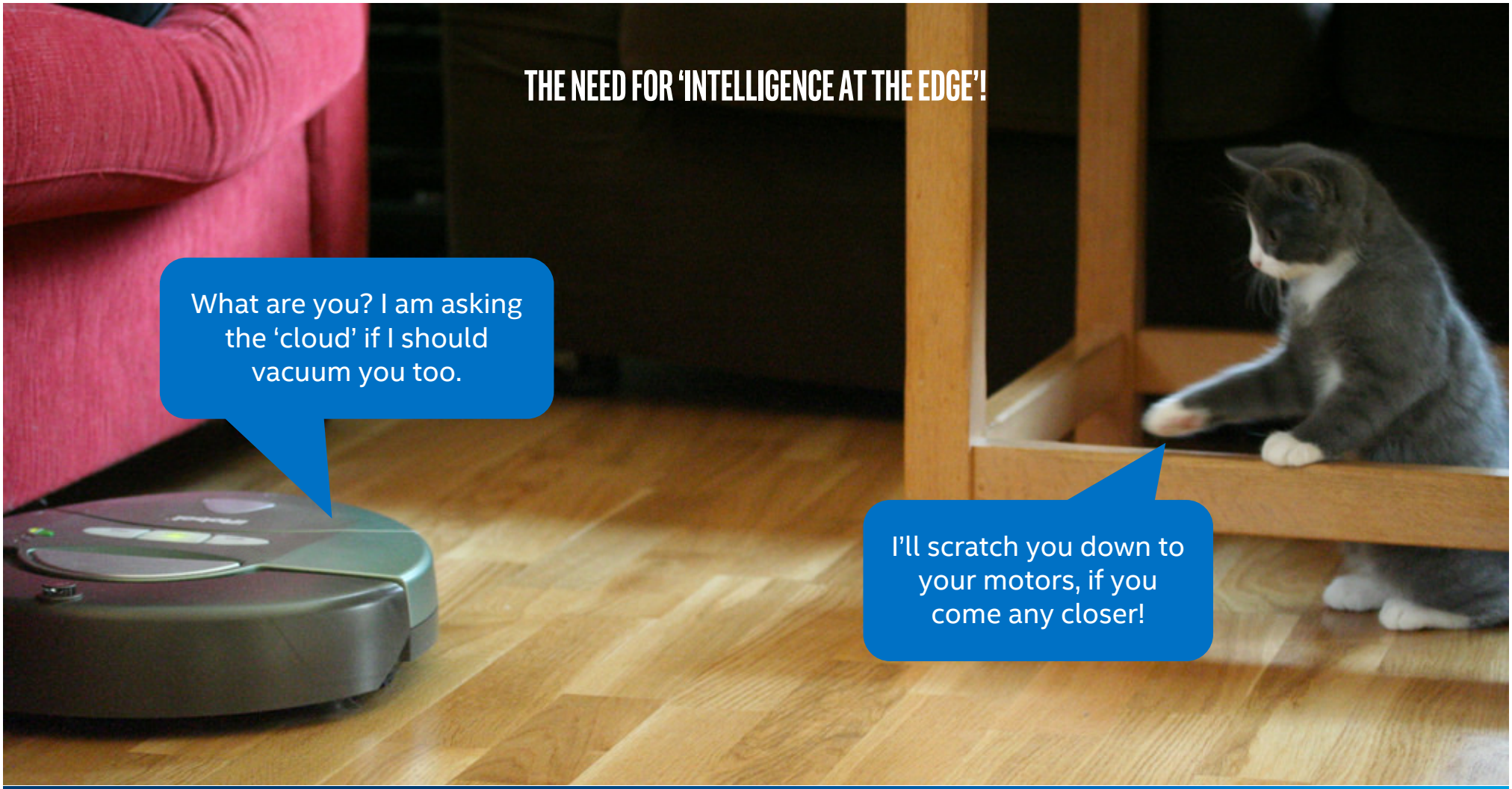
PART 2: RESULTS ON GOOGLENET INCEPTION V1 USING BREEDS



SAVE FILES FOR INFERENCE

- **Save the frozen graph (.pb file)**

INFERENCE USING THE INTEL® MOVIDIUS™ NEURAL COMPUTE STICK


THE NEED FOR 'INTELLIGENCE AT THE EDGE'!
What are you? I am asking the 'cloud' if I should vacuum you too.
I'll scratch you down to your motors, if you come any closer!

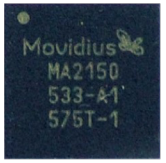
Let's look at a larger scale...

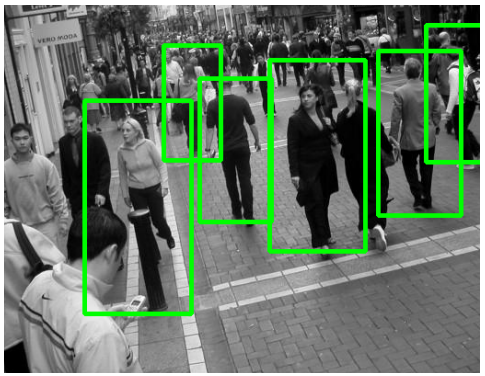
20 billion connected devices by 2020[1]

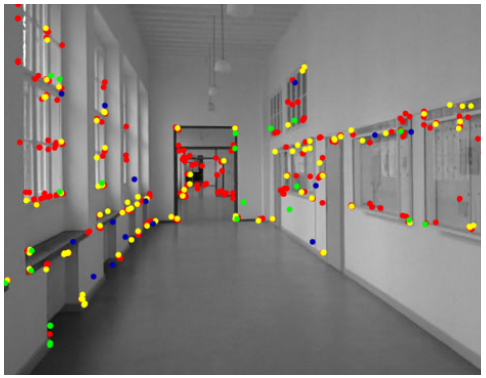
generating **billions of petabytes of data** traffic between devices & the cloud

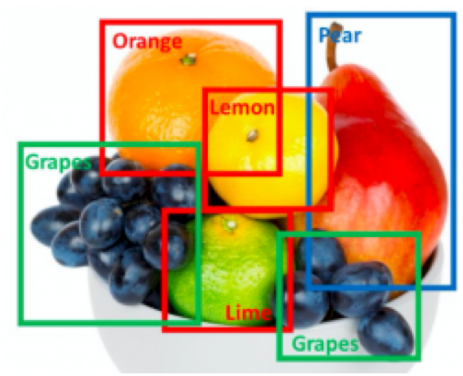
1 Source: http://www.gartner.com/newsroom/id/3598917

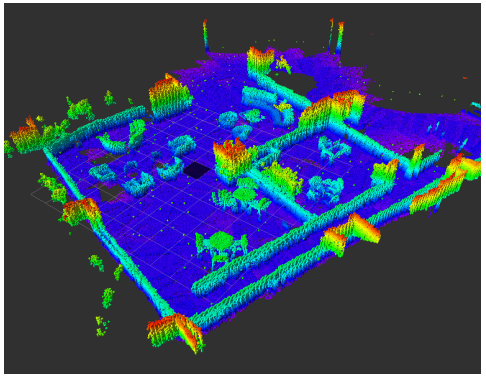
Movidius™
an Intel company











Game-changing intelligent devices

Powered by 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

Intel® Movidius™ Neural Compute Stick

Redefining the AI developer kit

- Neural Network Accelerator in USB Stick Form Factor
- No additional heat-sink, no fan, no cables, no additional power supply
- Prototype, tune, validate and deploy deep neural networks at the edge
- Features the same Intel® Movidius™ Myriad™ Vision Processing Unit (VPU) used in drones, surveillance cameras, VR headsets, and other low-power intelligent and autonomous products

Intel® Movidius™ Neural Compute Stick

Redefining the AI developer kit

NC SDK

Free download @ developer.movidius.com

NC Toolkit

- Profiler
- Checker
- Compiler

NC API

- API

Intel® Movidius™ Neural Compute Stick

Redefining the AI developer kit

New! TensorFlow* support + AppZoo

NC Toolkit

Profiler

Checker

Compiler

NC API

API

NC SDK

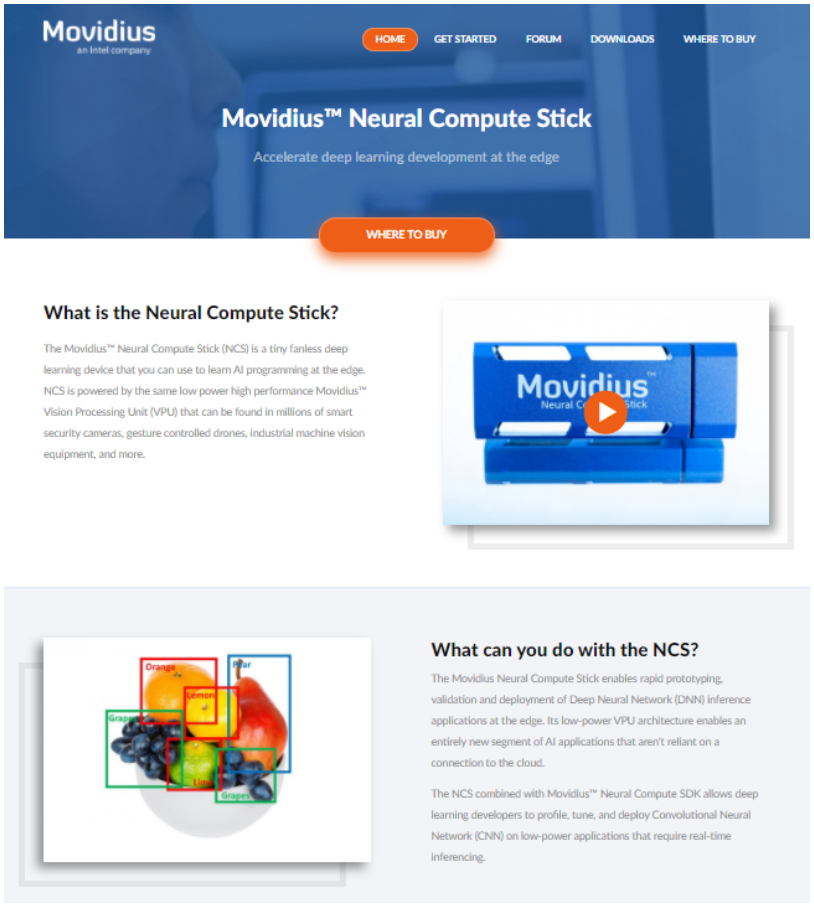
Free download @ developer.movidius.com

Explore developer.movidius.com

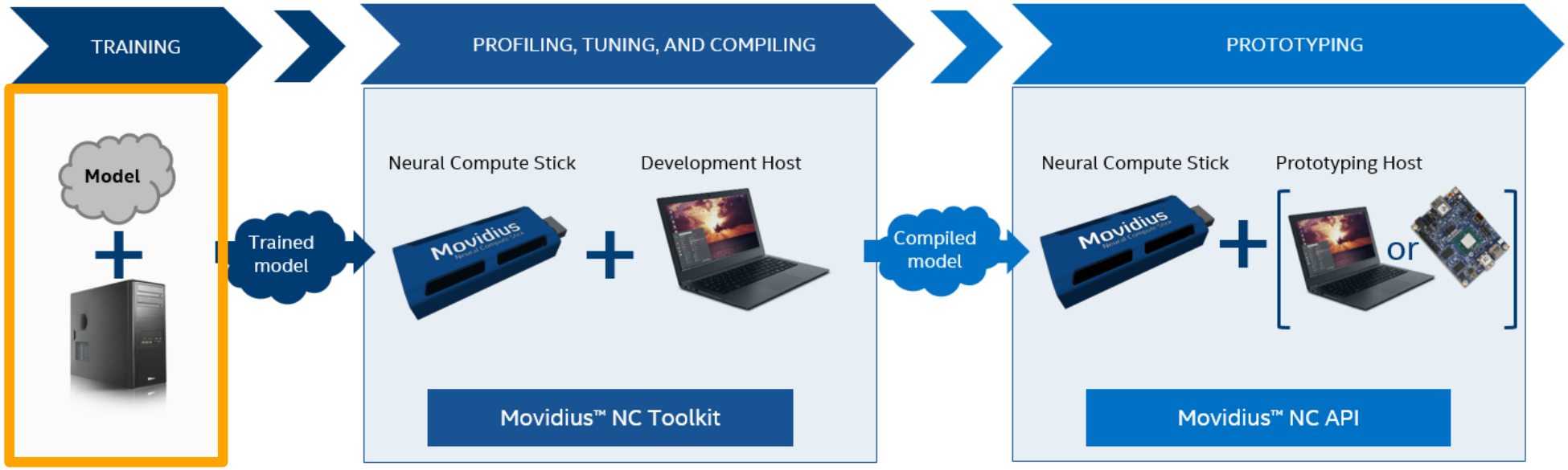
A developer-friendly website

Try out the following pages:

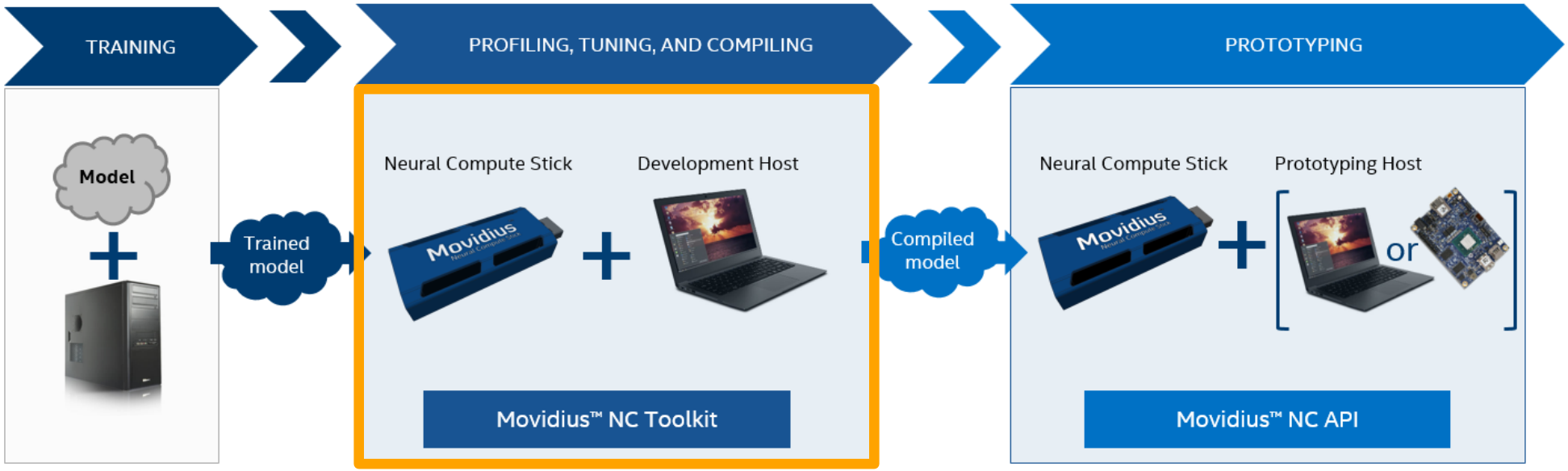
- Main page
- Getting started
- Downloads
- Docs
- Forums
- Where to buy



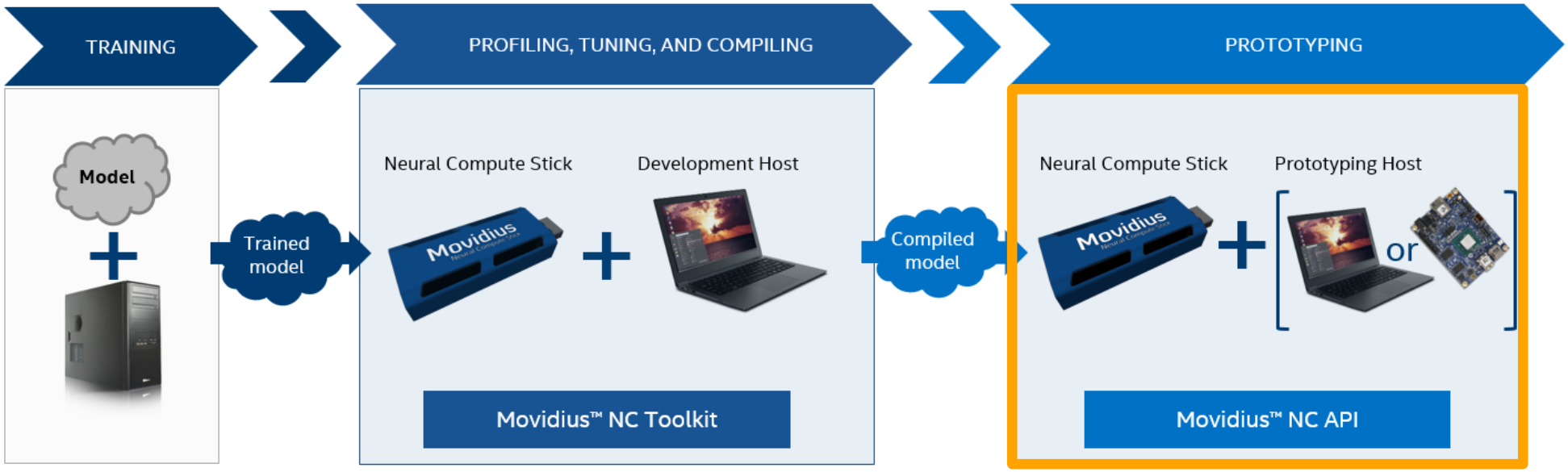
Intel® Movidius™ Software Development Kit (SDK) workflow



Intel® Movidius™ Software Development Kit (SDK) workflow



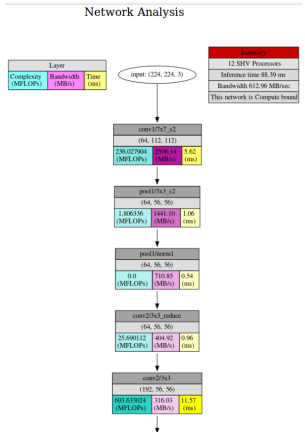
Intel® Movidius™ Software Development Kit (SDK) workflow



What can I do with the Intel® Movidius™ Neural Compute Stick ?

Profiler

A tool that provides a detailed stage-by-stage breakdown of where the bottlenecks are in your system.



Checker

Runs a single inference on the NCS using the provided model, allowing for the calculation of classification correctness.

Compiler

-

The compiler is used to create a graph which is an optimized binary file that can be processed by the NCS.

C API

GetDeviceName
OpenDevice
AllocateGraph
DeallocateGraph
LoadTensor
SetGraphOption
CloseDevice
...

Python bindings

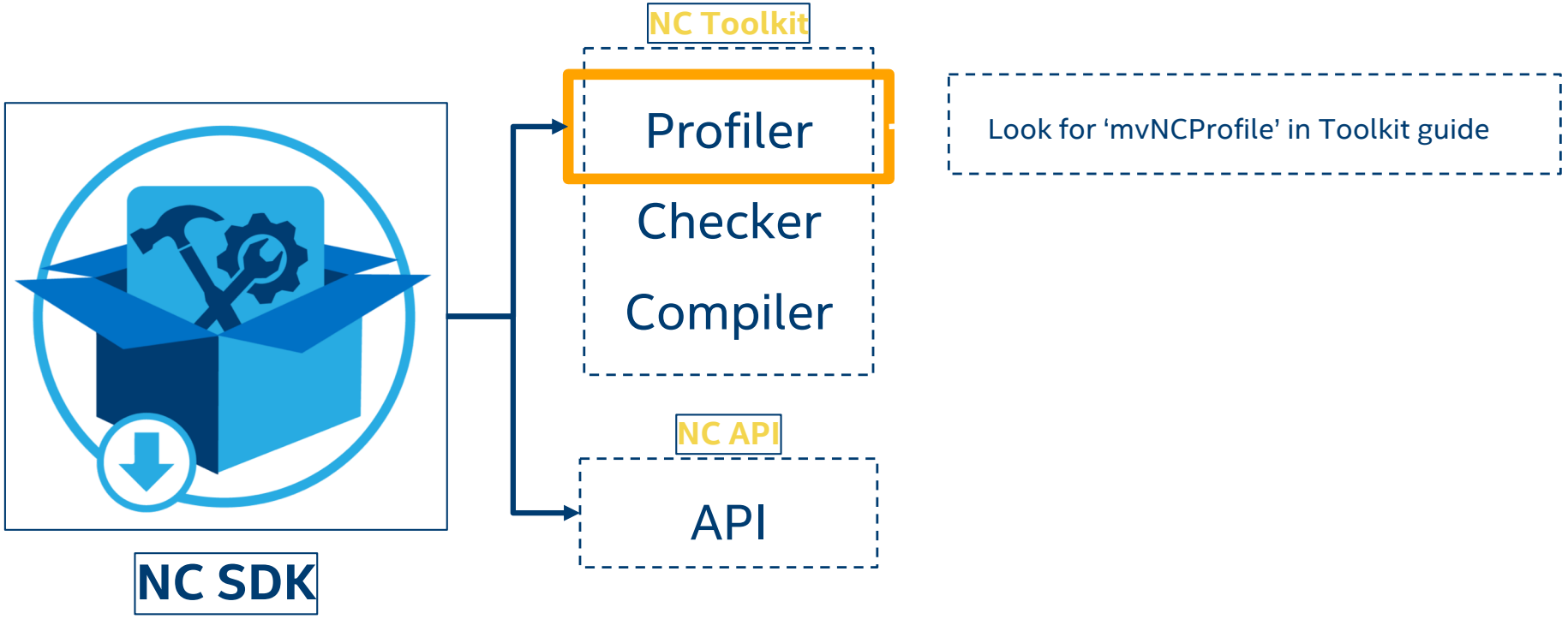
Status
GlobalOption
DeviceOption
GraphOption
EnumerateDevices
SetGlobalOption
LoadTensor
...

DNN architect / data scientist

Applications developer

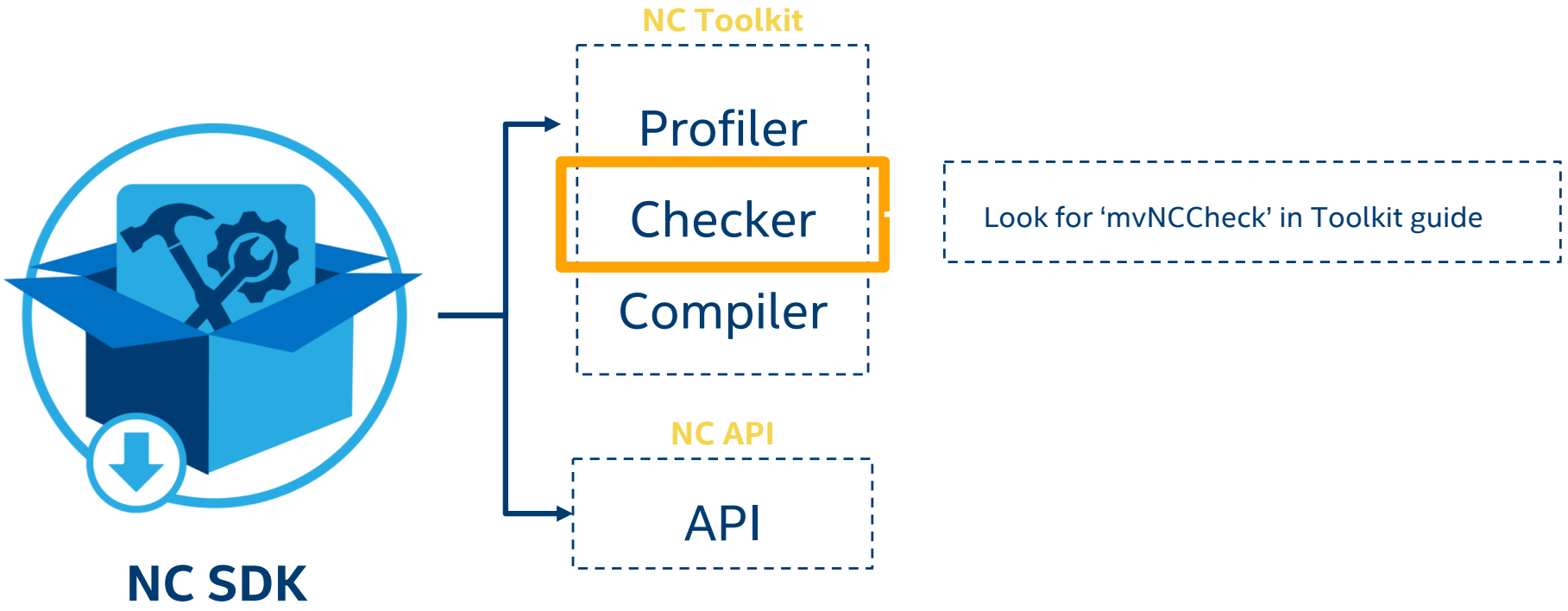
NC SDK Profiler

Get a better insight into your network's complexity, bandwidth & execution time



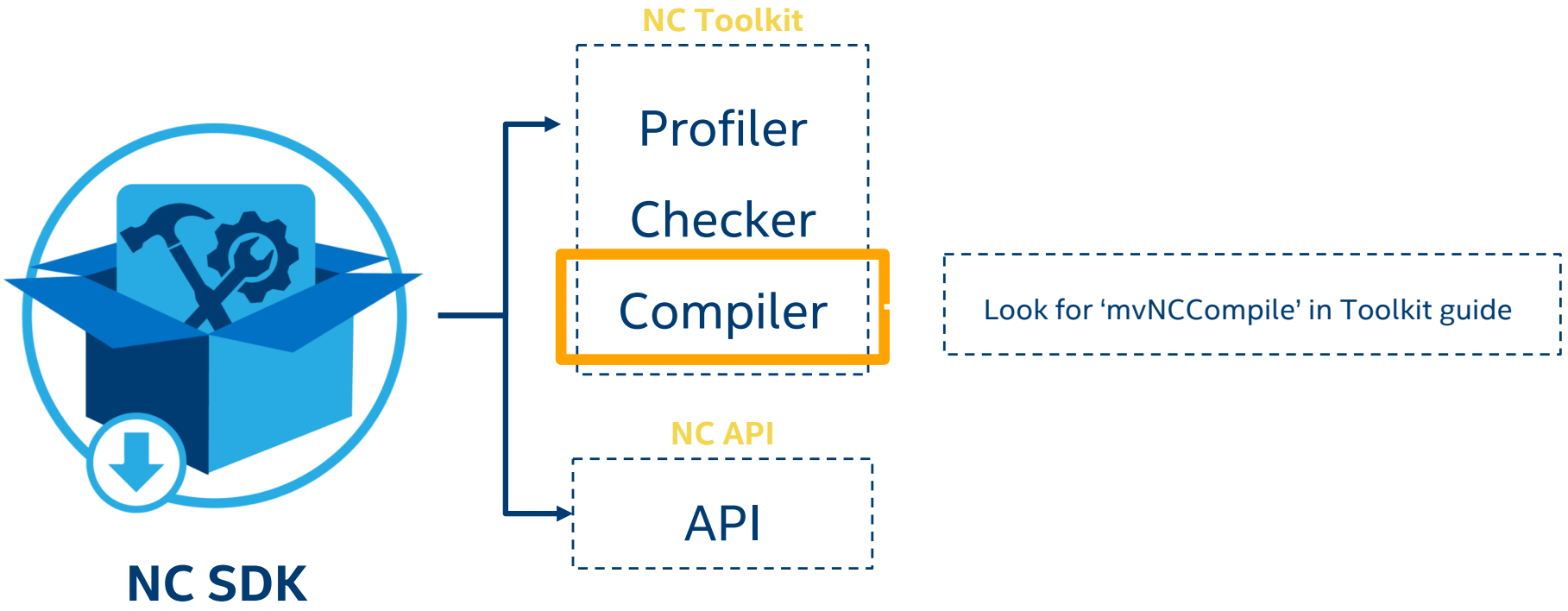
NC SDK Checker

Run a single inference on the NCS and compare results with that of Caffe



NC SDK Compile

Convert your network into a binary graph file that can be loaded onto the NCS



NC SDK API framework

Develop you own embedded application with deep-learning accelerated image processing

NC SDK

NC Toolkit

Profiler

Checker

Compiler

NC API

API

Let's develop a simple embedded app!

INFERENCE ON CPU AND GPU USING THE INTEL® OPENVINO™ SDK

Open Visual Inference & Neural network Optimization (OpenVINO™) toolkit

Accelerate Computer Vision Solutions

Free Download

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

What it is

A toolkit to fast-track development of **high performance computer vision** and **deep learning into vision applications**. It enables deep learning on hardware accelerators and easy **heterogeneous** execution across Intel® platforms.
Components include:

- Intel® Deep Learning Deployment Toolkit (model optimizer, inference engine)
- Optimized functions for OpenCV* and OpenVX*

Why important

Demand is growing for intelligent vision solutions. **Deep learning revenue** is estimated to grow from $655M in 2016 to **$35B by 2025**[1]. This requires **developer tools** to integrate computer vision, deep learning, and analytics processing capabilities into applications, so they can help **turn data into insights that fuel artificial intelligence**.



Users: Software developers, data scientists working on vision solutions for surveillance, robotics, healthcare, office automation, autonomous vehicles, & more.

OpenVINO™ version is 2018 R1

[1]Tractica 2Q 2017

What's Inside the OpenVINO™ toolkit

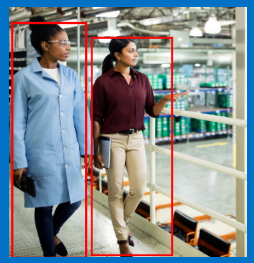
INTEL® DEEP LEARNING DEPLOYMENT TOOLKIT

Model Optimizer
Convert & Optimize

IR

Inference Engine
Optimized Inference

Code Samples & 10 Pre-trained Models



TRADITIONAL COMPUTER VISION TOOLS & LIBRARIES

Optimized Computer Vision 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) Microsoft Windows* 10 (64 bit)	Ubuntu* 16.04.3 LTS (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

IR = Intermediate Representation file

Intel® Deep Learning Deployment Toolkit

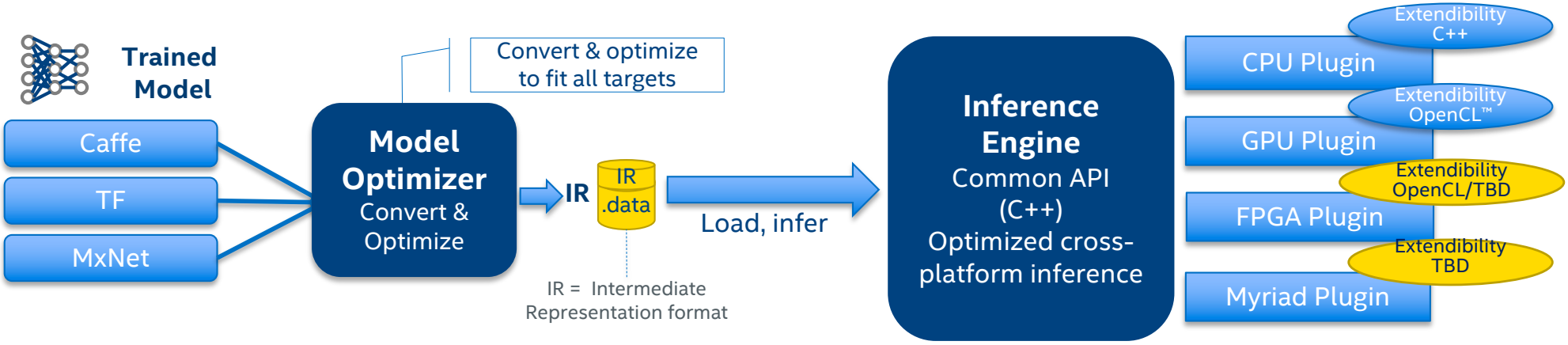
Take Full Advantage of the Power of Intel® Architecture

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.

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.



Q&A