AI DEVCON 2018

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

San Francisco, May 23-24, 2018

LEGAL NOTICES AND 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, Movidius, nGraph, neon, VTune, Xeon, Nervana, Atom, Core, Arria, and Myriad 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 Intel® Optimized Caffe
- Training on TensorFlow with Intel® optimizations
- Validation on CPU/GPU using the Intel® OpenVINO™ SDK – Hands On
- Validation on the Intel® Movidius™ Neural Compute Stick (NCS) – Hands On
- Deploy to an edge device (Raspberry Pi*) – Demo

INSTRUCTIONS TO PARTICIPANTS

- Intel® DevCloud Access for today

  Create an account on the Intel AI DevCloud

  https://colfaxresearch.com/aidevcon18 - Passcode: AG7WNN92
- Download and Install the Intel® OpenVINO™ SDK
- https://software.intel.com/en-us/openvino-toolkit
- Download and Install the Intel® Movidius™ Neural Compute Stick SDK
- https://developer.movidius.com/start

QUESTIONS? ASK US!

BEN ODOM
Developer Evangelist

benjamin.j.odom@intel.com

MICHAEL HERNANDEZ
Developer Evangelist

michael.j.hernandez@intel.com

MEGHANA RAO
Developer Evangelist

meghana.s.rao@intel.com

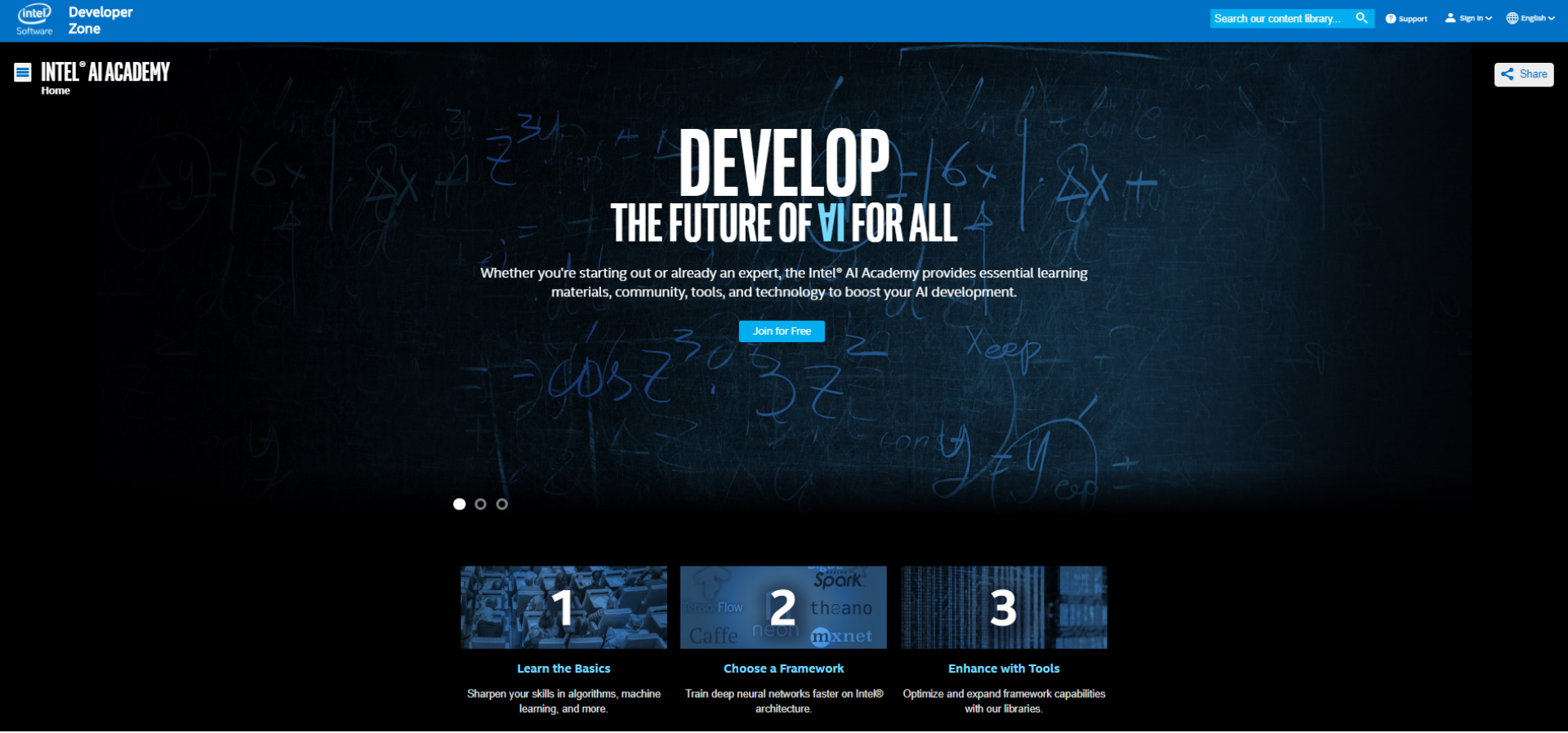
RUDY CAZABON
Developer Evangelist

rudy.cazabon@intel.com

INTEL® AI ACADEMY

AI ADOPTION IS JUST BEGINNING

Source: Forrester Research – Artificial Intelligence: Fact, Fiction. How Enterprises Can Crush It; What's Possible for Enterprises in 2017



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

intel AI

INTEL® AI ACADEMY

For developers, students, instructors, and startups

LEARN

- Online tutorials
- Webinars
- Student kits
- Support forums

DEVELOP

- Intel optimized frameworks
- Exclusive access to Intel® AI DevCloud

TEACH

- Comprehensive courseware
- Hands-on labs
- Cloud compute
- Technical support

SHARE

- Project showcase opportunities at Intel Developer Mesh
- Industry and academic events

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

intel AI

INTEL® AI PORTFOLIO

intel 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 | Mobileye

END-TO-END COMPUTE

SYSTEMS & COMPONENTS

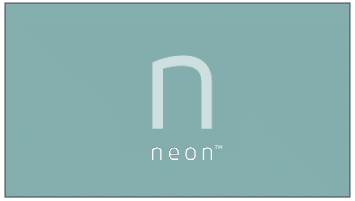
†Beta available
‡ Future
*Other names and brands may be claimed as the property of others.

intel AI

AI FRAMEWORKS OPTIMIZED BY INTEL

Popular DL frameworks are now optimized for CPU!

CHOOSE YOUR FAVORITE FRAMEWORK



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*

intel AI

AI DEVCON 2018

DEEP LEARNING FRAMEWORK OPTIMIZED FOR IA: CAFFE*

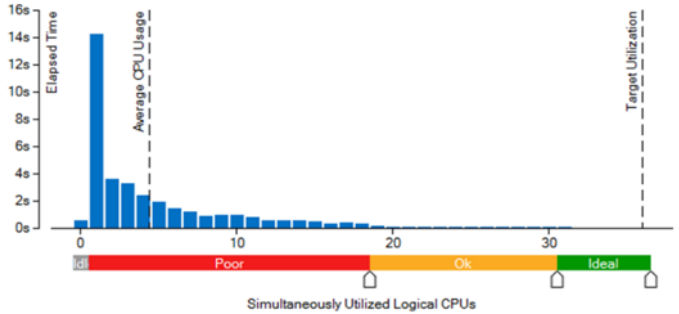
INITIAL CIFAR-10 RUN IN CAFFE*— INTEL® VTUNE™ AMPLIFIER ANALYSIS

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 Intel® Math Kernel Library (Intel® MKL)

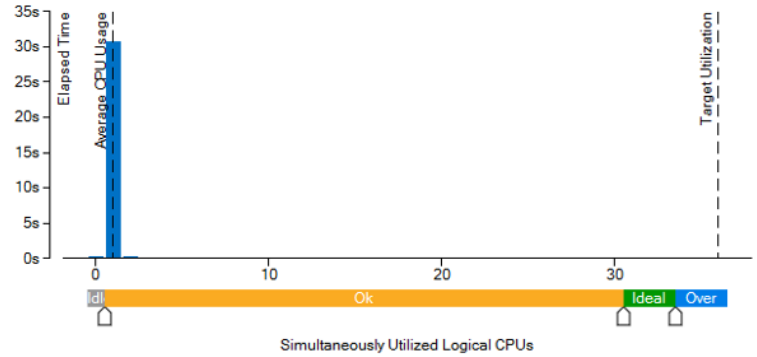
INITIAL CIFAR-10 RUN IN CAFFE—INTEL® VTUNE™ AMPLIFIER ANALYSIS

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 Intel® Math Kernel Library (Intel® MKL)

CURRENT OPTIMIZATIONS

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

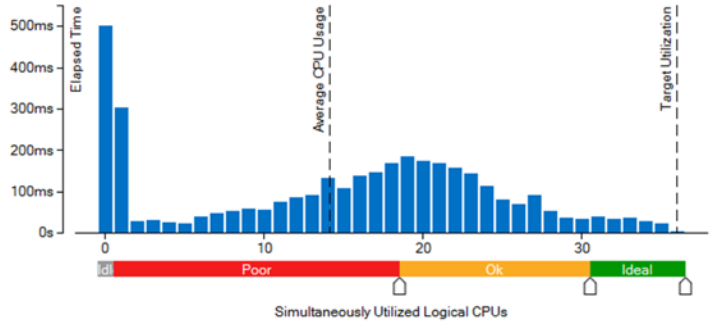
https://software.intel.com/en-us/articles/caffe-optimized-for-intel-architecture-applying-modern-code-techniques

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

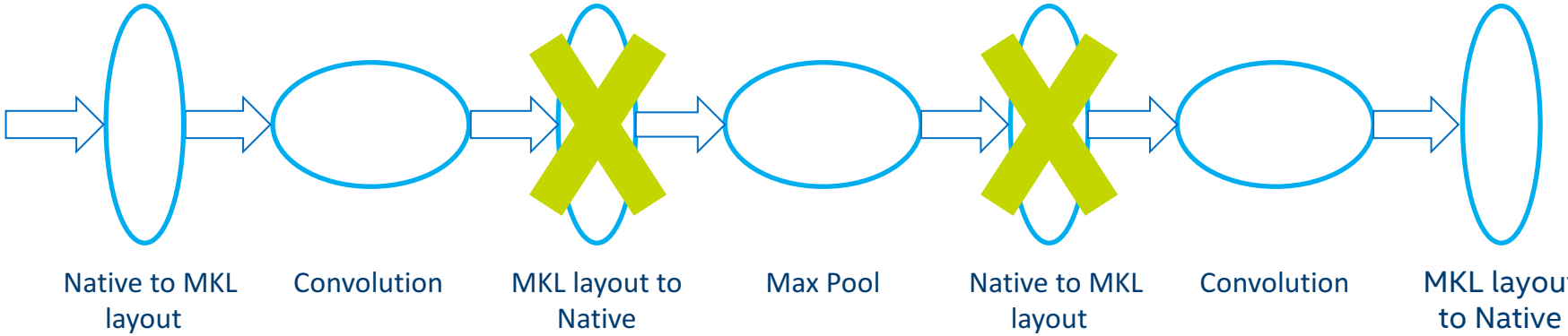
This histogram displays a percentage of the wall time the specific number of CPUs were running simultaneously. Spin and Overhead time adds to the Idle CPU usage value.



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* AND OTHER DL FRAMEWORKS FOR INTEL® ARCHITECTURE

- Leverage High-Performance Compute Libraries and Tools
  - For example, Intel® Math Kernel Library, Intel® Distribution for 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
  - For example, batch size for more parallelism
  - Learning rate and optimizer to ensure accuracy/convergence

INITIAL PERFORMANCE GAINS ON INTEL® XEON® PROCESSORS

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

- 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"

Benchmark	Metric	Batch Size	Baseline Performance Training	Baseline Performance Inference	Optimized Performance Training	Optimized Performance 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

ADDITIONAL PERFORMANCE GAINS FROM PARAMETERS TUNING

(BEST SETTING FOR INTEL® XEON® PROCESSORS (INTEL® MICROARCHITECTURE CODE NAME BROADWELL —2 SOCKET—44 CORES)

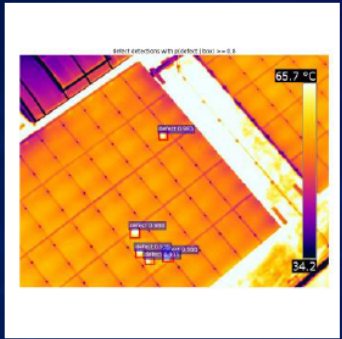
- Data format: CPU prefers NCHW data format
- Intra_op, inter_op and OMP_NUM_THREADS: set for best core utilization
- Batch size: higher batch size provides for better parallelism
  - A batch size that is too high can increase working set and impact cache/memory perf

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

AI DEVCON 2018

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

intel AI

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

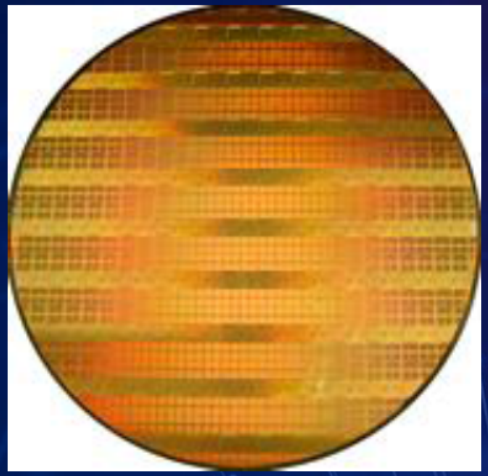
FRAMEWORK

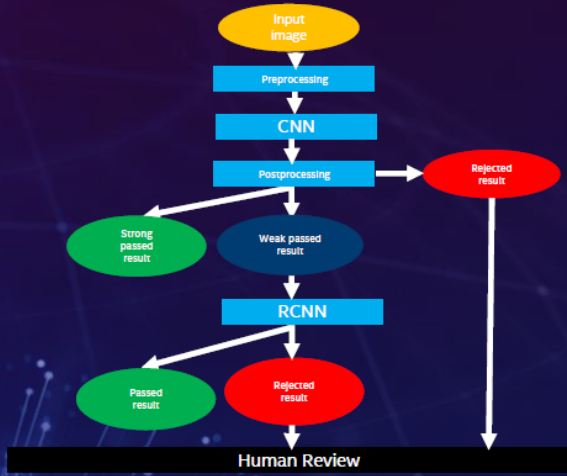
intel XEON inside

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

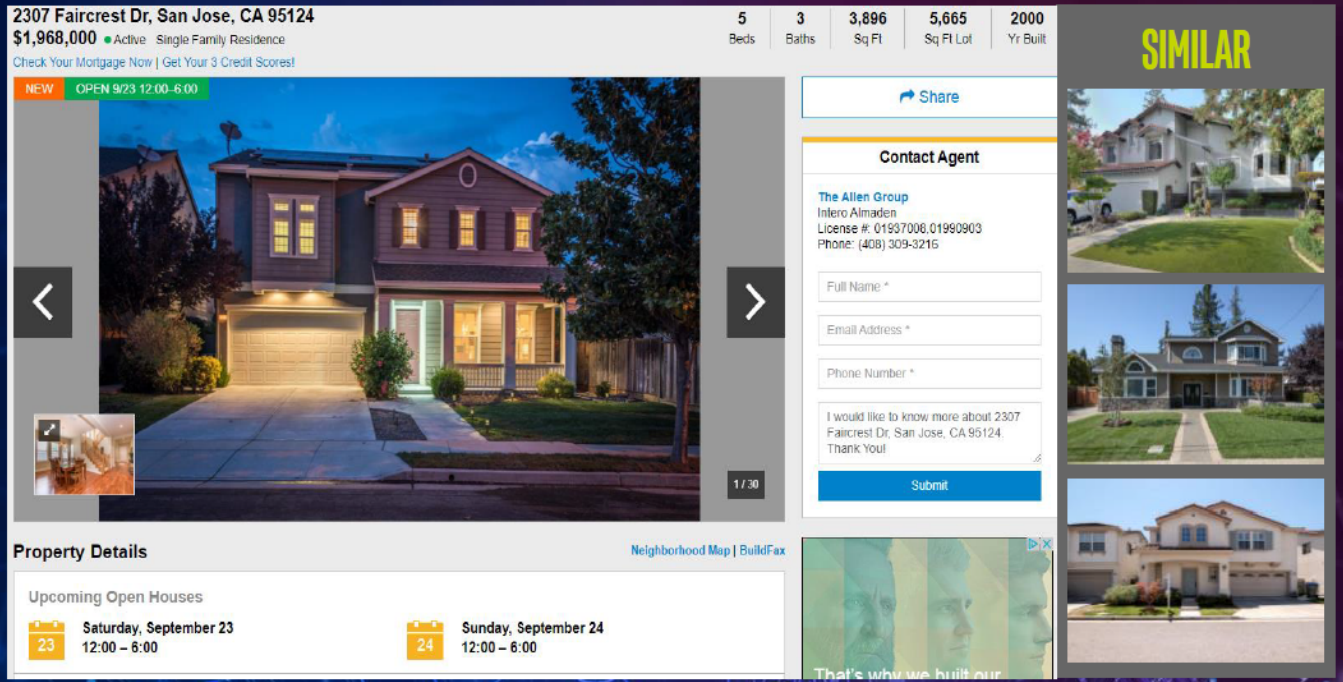
Caffe

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.

AI DEVCON 2018

HANDS-ON CODING: TRAINING A CONVOLUTIONAL NEURAL NETWORK USING THE INTEL® AI DEVCLOUD

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 Processor: Intel® Xeon® Gold 6128 processor @ 3.40 GHz, 24 cores with 2-way hyper-threading, 96 GB of on-platform RAM (DDR4), 200 GB of file storage.
- **Four weeks of initial access, with extension based on 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® Optimization for Caffe*
- Intel® Optimization for TensorFlow*
- Intel® Optimization for Theano*
- Intel® Nervana™ platform, neon™ framework
- 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

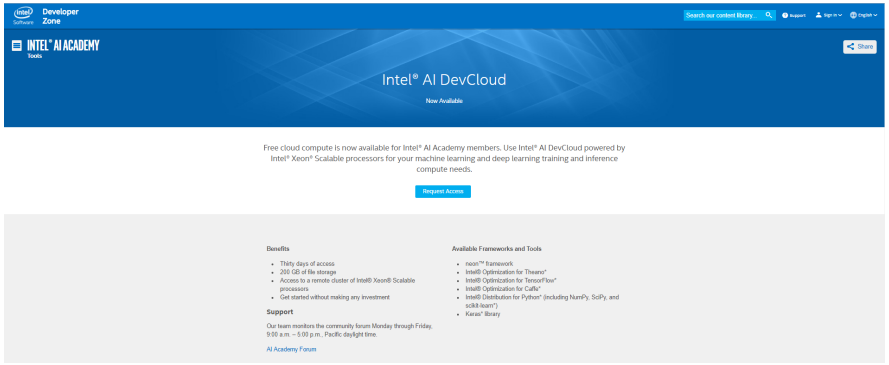
intel AI DEVCON 2018

REQUEST ACCESS

Intel® AI DevCloud

GET DEVCLOUD 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, and 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

CONNECTING TO THE DEVCLOUD

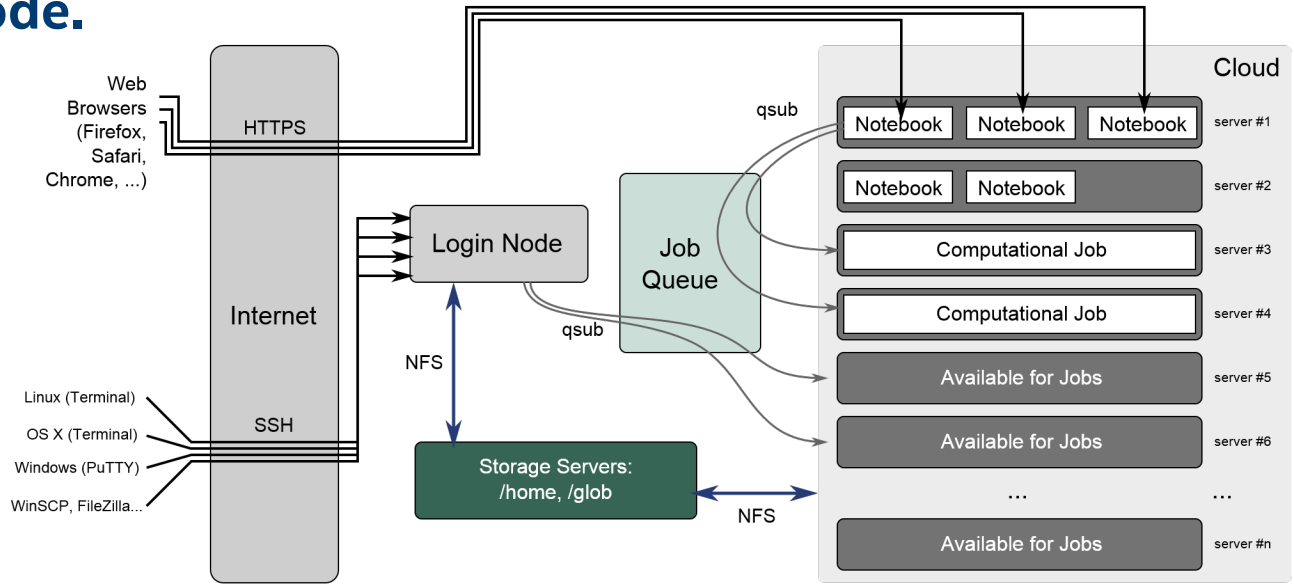
- Linux*/Mac*/Linux on Windows if you have Windows® 10
  - Download and save the Linux access key.

```
Host colfax
     User u<your usderID>
     IdentityFile ~/.ssh/colfax-access-key-<your user ID>
     ProxyCommand ssh -T -i ~/.ssh/colfax-access-key-<your user ID> guest@cluster.colfaxresearch.com
```

- If you are using PuTTY from Windows:
  - Download the ssh client PuTTY – make sure to use the 64-bit MSI installer.
  - Download and save Windows access key.
  - Right click on the downloaded key and choose “Load into Pageant.”
  - Configure PuTTY.

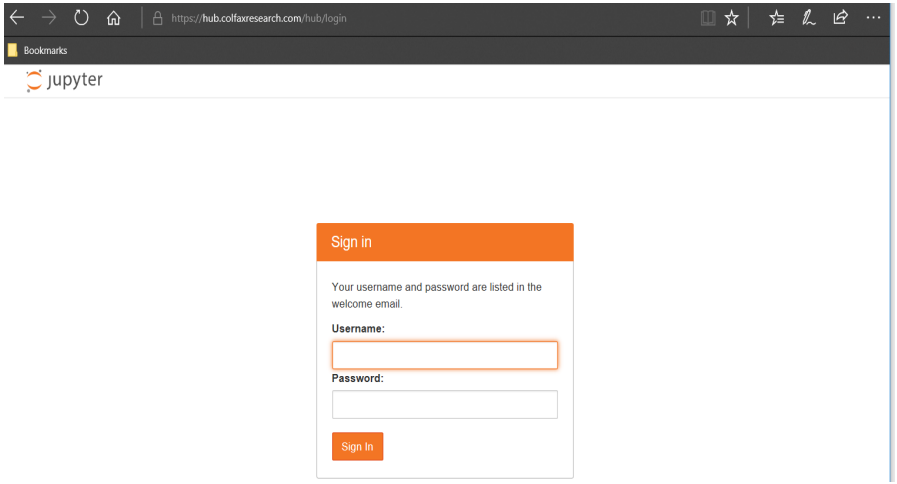
ONCE CONNECTED…

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



JUPYTER*HUB 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

AI DEVCON 2018

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. The 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!

CAFFE* WORKFLOW

TRAINING BREEDS

REPEAT STEPS FOR THE OXFORD PETS DATASET

Problem Statement	• You are here to solve an issue
Get Your Data	• Introduction to the data
Clean Your Data	• Organize it, augment it, split it, etc....
Train	• 37 breeds—learn to tell them apart
Test	• Test local sample, try from Internet

PART 1: FETCH THE DATA

The Oxford Pets Database

- 37 categories
- ~200 images of each class
- 25 Dogs
- 12 cats
- Paper

PART 1: VIEW THE BASELINE DATA

Fetch Your Data → View Original Data

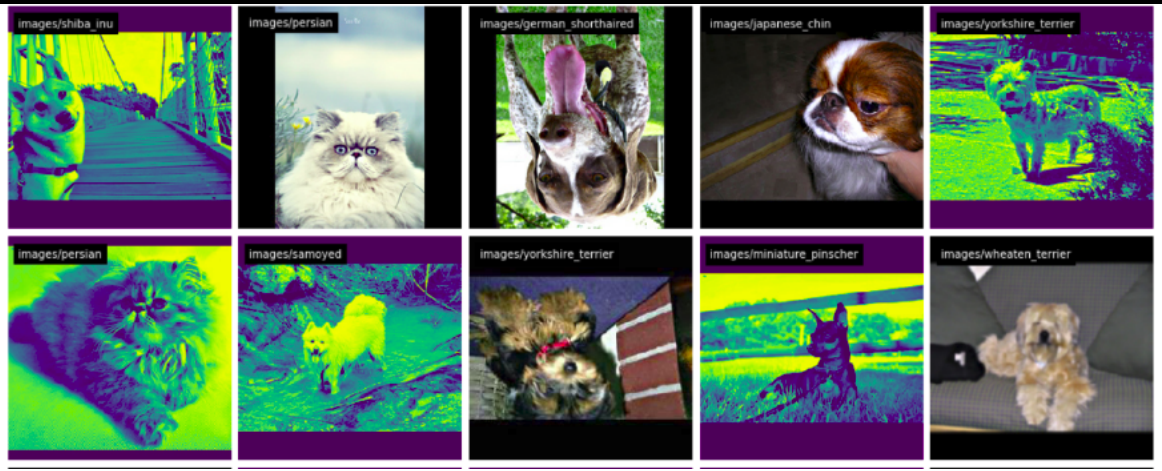
PART 1: CLEAN AND NORMALIZE THE DATA

- **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: AUGMENT THE DATA

- **Add noise to existing data**
  - Improves training and inference accuracy
- **Some ways to accomplish augmentation:**
  - Flip
  - Flop
  - Blur
  - Rotate
  - Extract color channels

PART 1: VIEW RESULTS POST AUGMENTATION



PART 1: ORGANIZE DATA FOR CONSUMPTION BY CAFFE*

PART 1: CONFIRM FOLDER STRUCTURE

PART 1: ORGANIZE DATA FOR CONSUMPTION BY CAFFE*

- Data organization is framework-specific
- Caffe expects data to be split into “train” and “val” folders
  - Non-overlapping data
  - Prevents overfitting
- **Folder structure**

- **train**
  - **Cat**
    - **Cat_t1.png**
    - **Cat_t2.png**
    - **…**
  - **Dog**
    - **Dog_t1.png**
    - **Dog_t2.png**
    - **…**

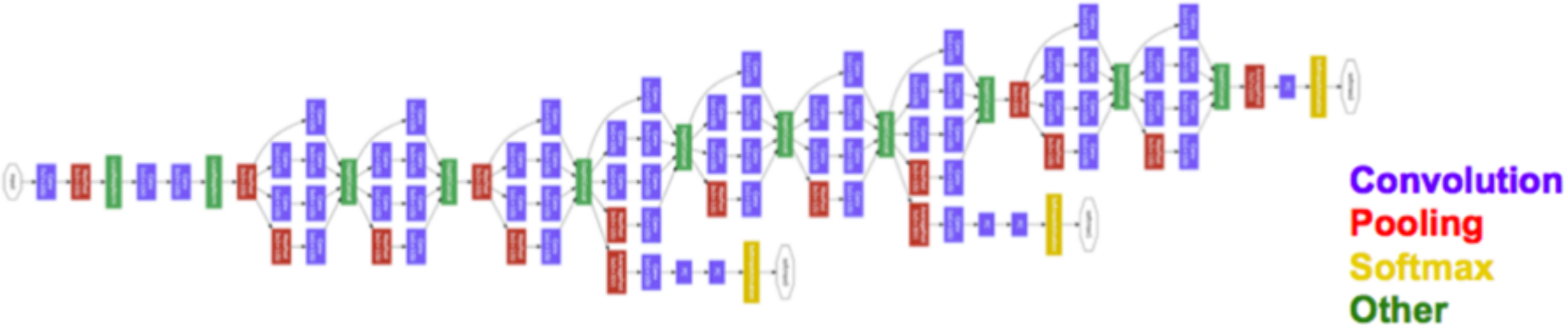
- **val**
  - **Cat**
    - **Cat_v1.png**
    - **Cat_v2.png**
    - **…**
  - **Dog**
    - **Dog_v1.png**
    - **Dog_v2.png**
    - **…**

PART 1: OPTIMIZE DATA FOR INGESTION

- **Create LMDB Dataset**
  - Database creates pointers to image files and identifies them by number.
  - Improves efficiency of image processing and training.
  - Point Caffe* to the right “train” and “val” folders created in prior step.
  - Calculate mean value of all images.
    - Defined in mean.binaryproto

PART 2: SELECT THE RIGHT TOPOLOGY

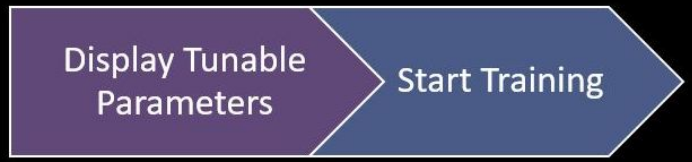
- **Criteria:**
  - Time to train: Depends on number of layers and computation required.
  - Size: Keep in mind the edge device you want to deploy to, networks it supports, and resources like memory.
  - Inference speed: Tradeoff between accuracy and latency.
  - **GoogLeNet (Inception V1) was our topology of choice.**



PART 2: DISPLAY TUNABLE PARAMETERS (HYPER-PARAMETERS)

- Train.prototxt – Network definition file
- Solver.prototxt – Tunable hyper-parameters

PART 2: START TRAINING



PART 2: ACCURACY AND LOSS FOR FULLY TRAINED NETWORK

PART 2: LOOK AT A SAMPLE IMAGE

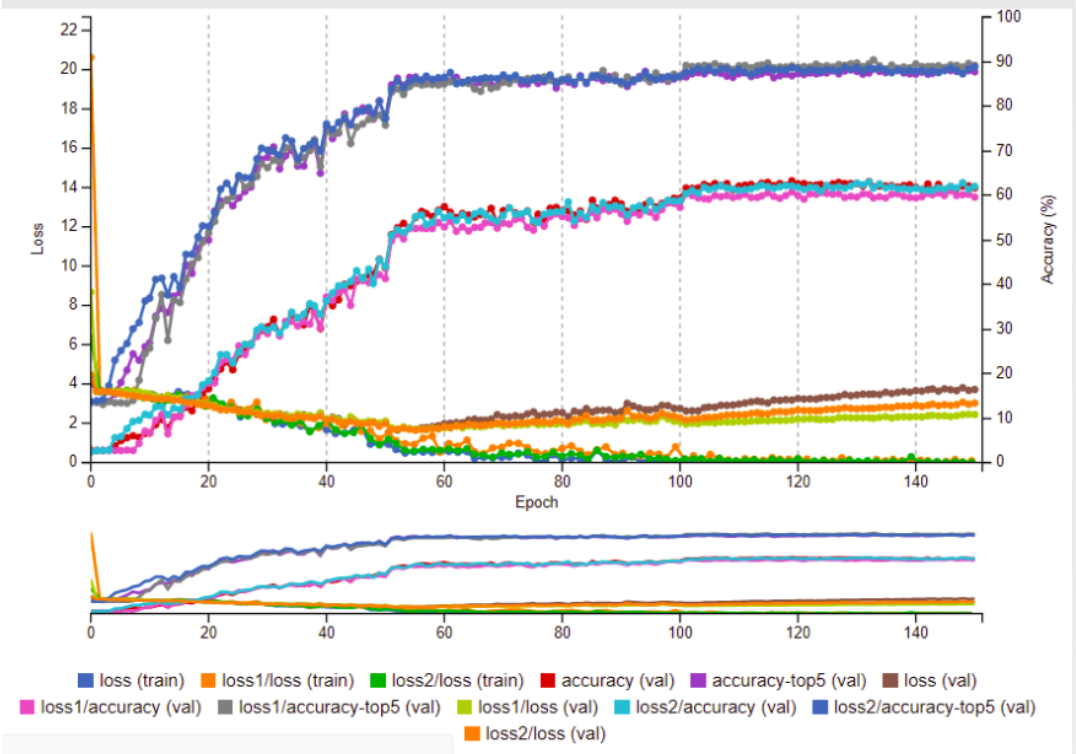
PART 2: INFERENCE ON SAMPLE IMAGE

PART 2: SUMMARY

Summary

- Getting your dataset
- Sorting your dataset
- Generating LMDB Record
- Training your dataset
- Using your Caffe* model to test image classification

RESULTS ON GOOGLENET INCEPTION V1



SAVE FILES FOR INFERENCE

Once the Caffe* model is trained, we will need the below files saved:

- deploy.prototxt – Network file that contains the layer information for the topology
- snapshotXXX.caffemodel – Weights file

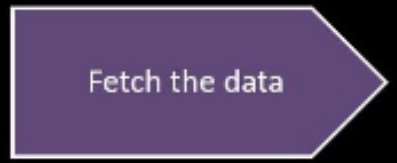
TENSORFLOW* WORKFLOW

TRAINING BREEDS

REPEAT STEPS FOR THE OXFORD PETS DATASET

Problem Statement	• You are here to solve an issue
Get Your Data	• Introduction to the data
Clean Your Data	• Organize it, augment it, split it, etc....
Train	• 37 breeds—learn to tell them apart
Test	• Test local sample, try from Internet

PART 1: FETCH THE DATA

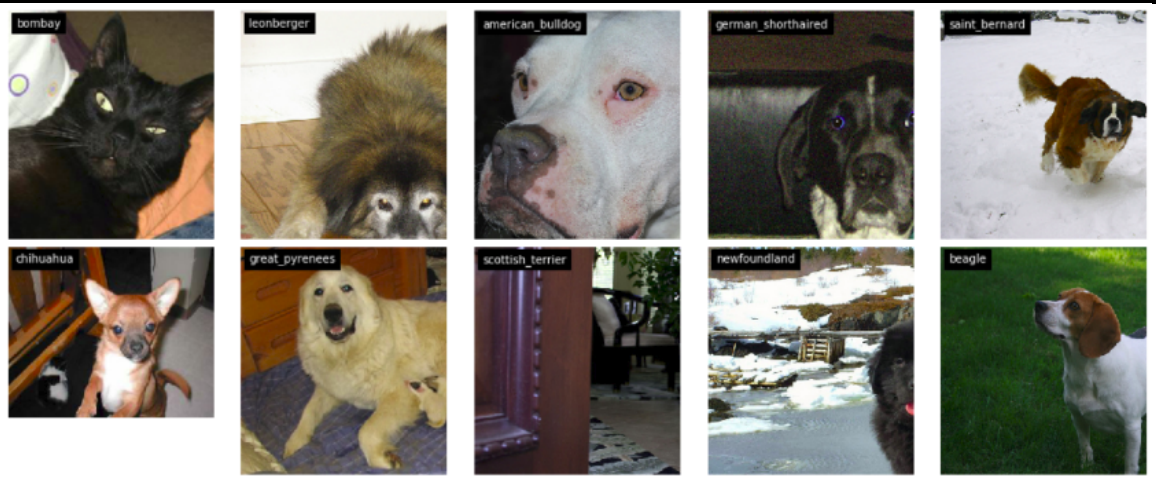


The Oxford Pets Database

- 37 categories
- ~200 images of each class
- 25 dogs
- 12 cats
- Paper talks about data and their techniques

PART 1: VIEW THE BASELINE DATA

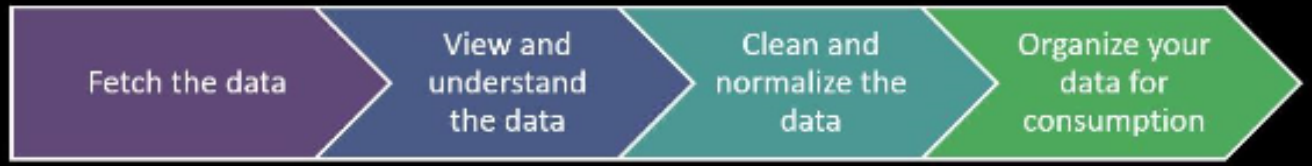
Fetch the data → View and understand the data



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*



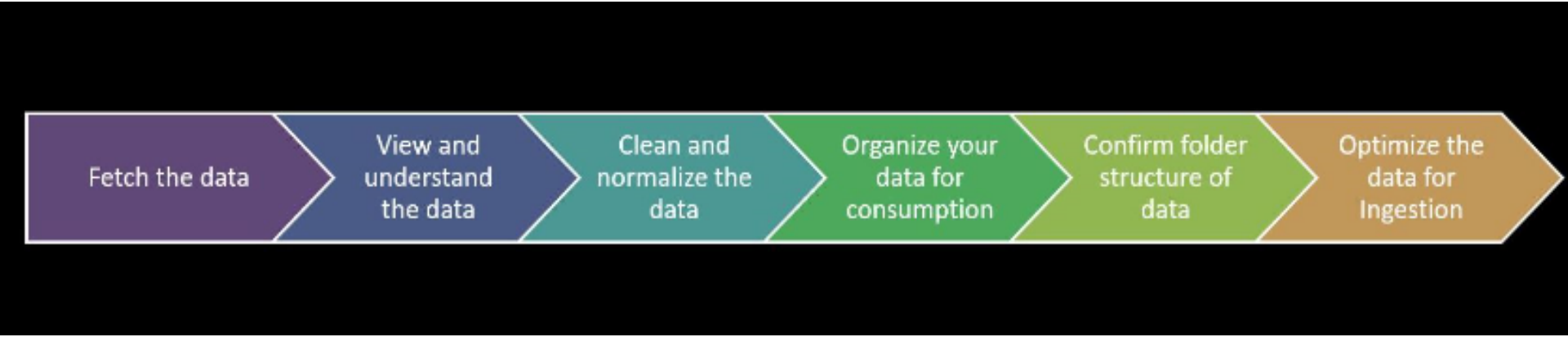
PART 1: ORGANIZE DATA FOR CONSUMPTION - CATEGORIZE

- TensorFlow* expects images to be organized into categories.
- Once complete, each category would look something like this (there are 37 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 two TFRecords, one for training and another for validation.

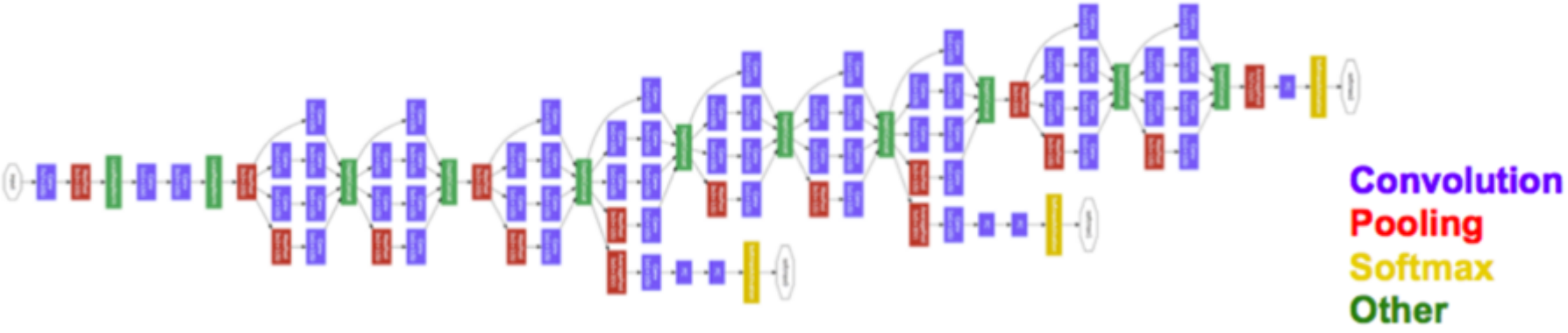
https://en.wikipedia.org/wiki/Lightning_Memory-Mapped_Database

PART 2: TRAINING

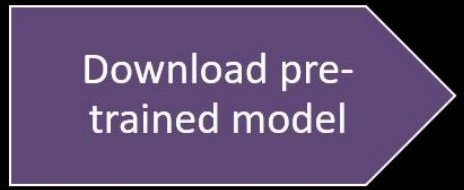
- Step 1: Choose the right topology.
- Step 2: Set up a pre-trained model to use breeds dataset.
- Step 3: Evaluate, freeze, and test results.

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.



PART 2: DOWNLOAD PRE-TRAINED MODEL



PART 2: CLONE TENSORFLOW*/MODELS GITHUB REPO

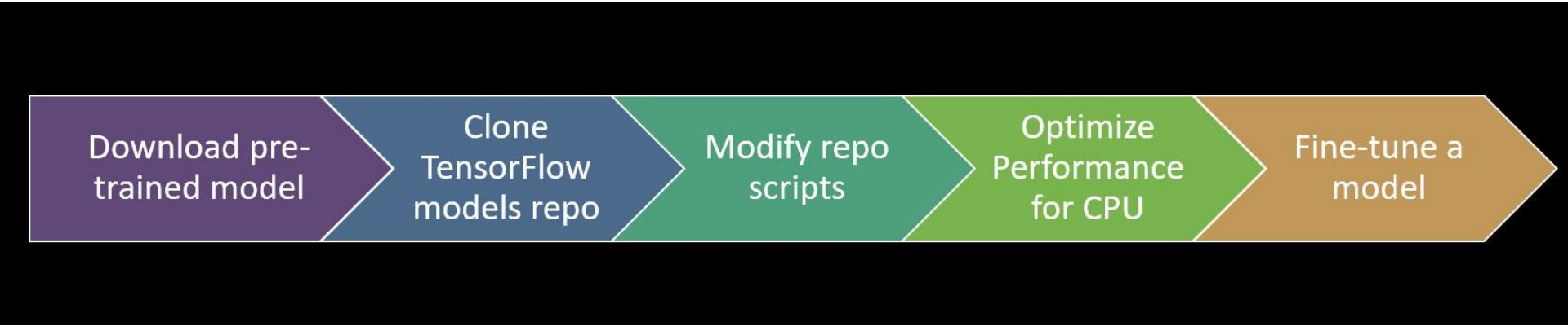
- Clone TensorFlow/models GitHub* repo

  We use transfer learning using a Convolutional Neural Network pre-trained on ISLVRC-2012-CLS image classification dataset (https://github.com/tensorflow/models)

PART 2: MODIFY/ADD FILES TO SLIM REPO TO WORK WITH BREEDS DATASET

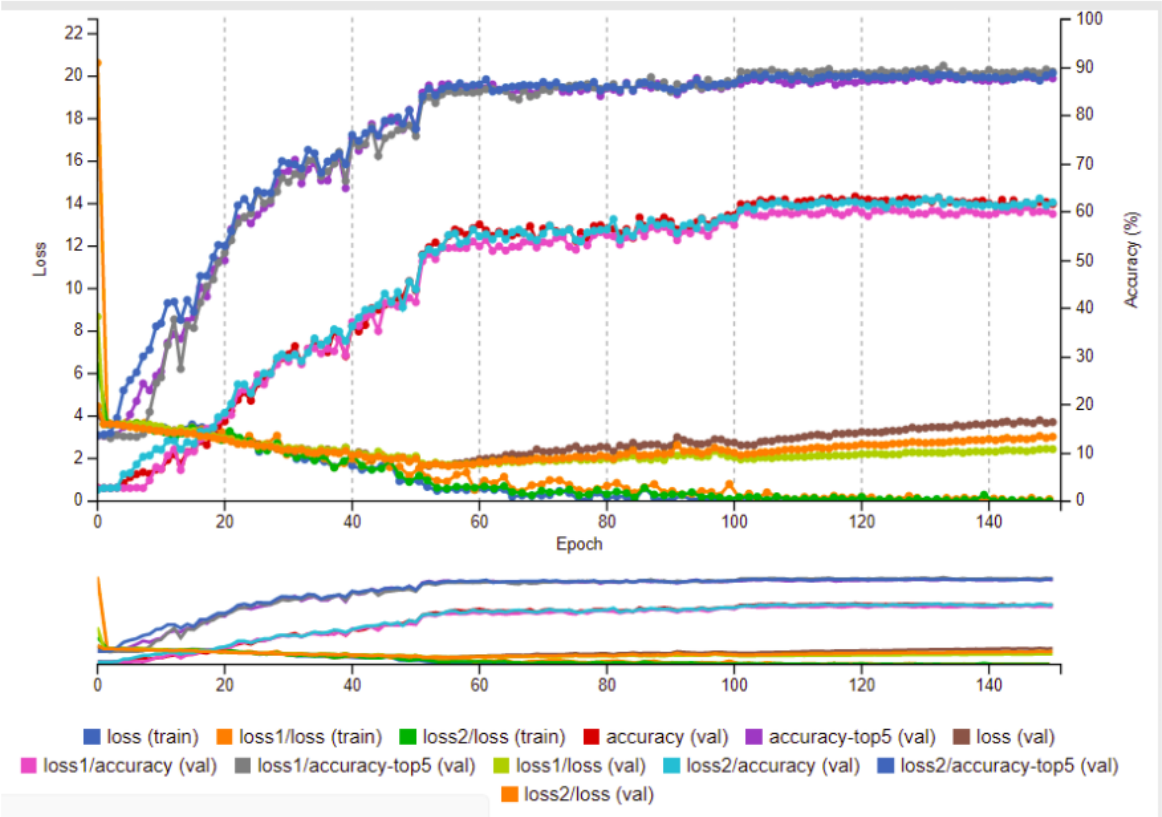
PART 2: OPTIMIZE PERFORMANCE FOR CPU

PART 2: INITIATE TRAINING

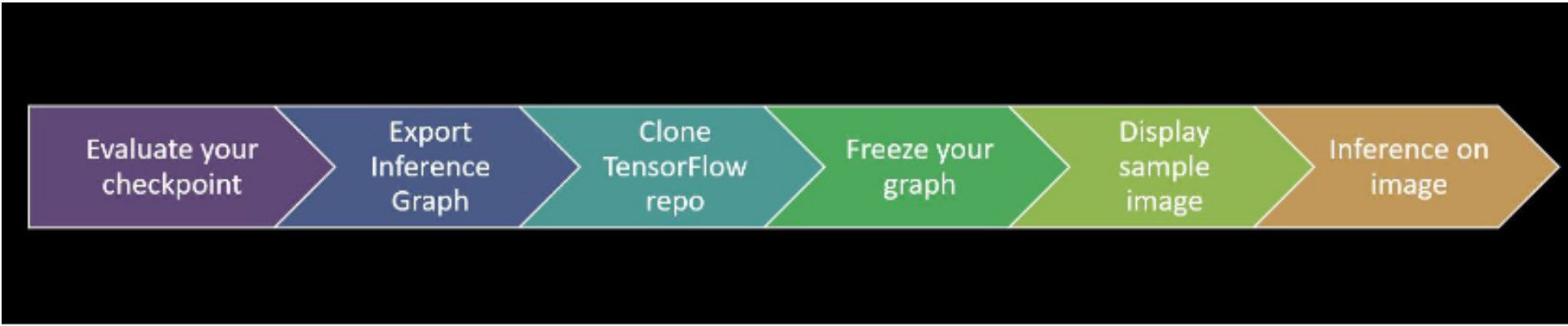


- Initiate training and review live training logs:
  - When using a pre-trained model on a different dataset, note that the final layer will change to indicate the new set of categories.
  - Indicate which subset of layers to retrain while keeping others frozen.
  - View results.

PART 2: RESULTS ON GOOGLENET INCEPTION V1 USING BREEDS



PART 3: EVALUATE, FREEZE GRAPH, AND TEST



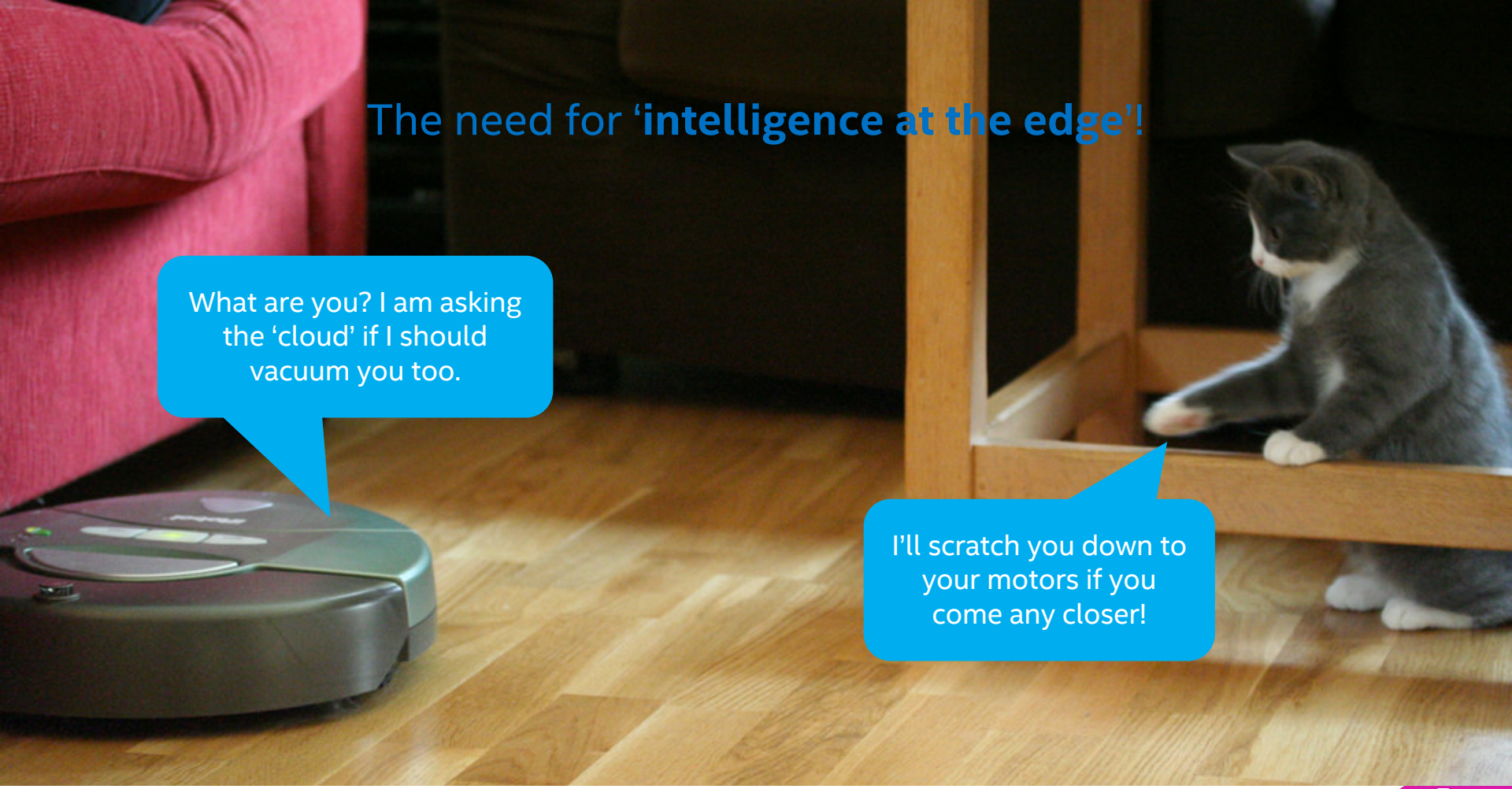
SAVE FILES FOR INFERENCE

- Save the graph def and frozen graph (.pb file).

AI DEVCON 2018

INFERENCE USING THE INTEL® MOVIDIUS™ NEURAL COMPUTE STICK

The need for '**intelligence at the edge**'!



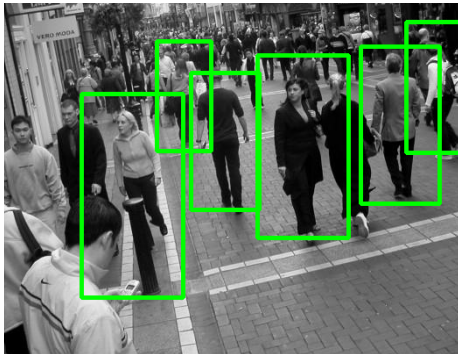
Let's look at a larger scale...

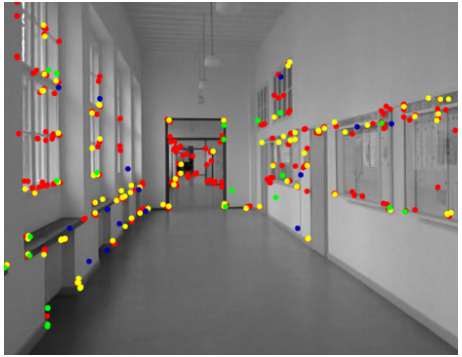
20 billion connected devices by 2020[1]

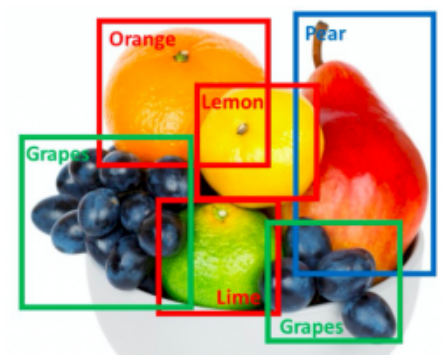
... generating **billions of petabytes of data** traffic between devices and the cloud

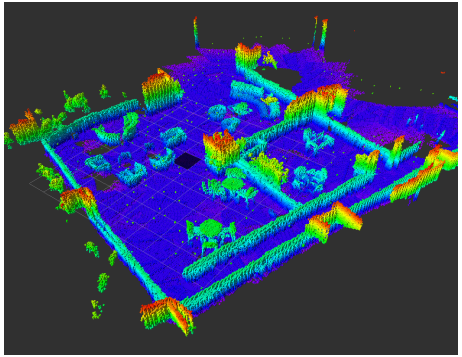
1 Source: http://www.gartner.com/newsroom/id/3598917

Movidius™
an Intel company



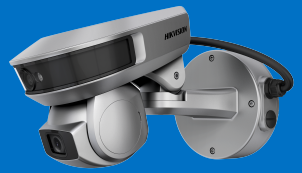






GAME-CHANGING INTELLIGENT DEVICES

Powered by Intel® Movidius™ vision processing unit (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

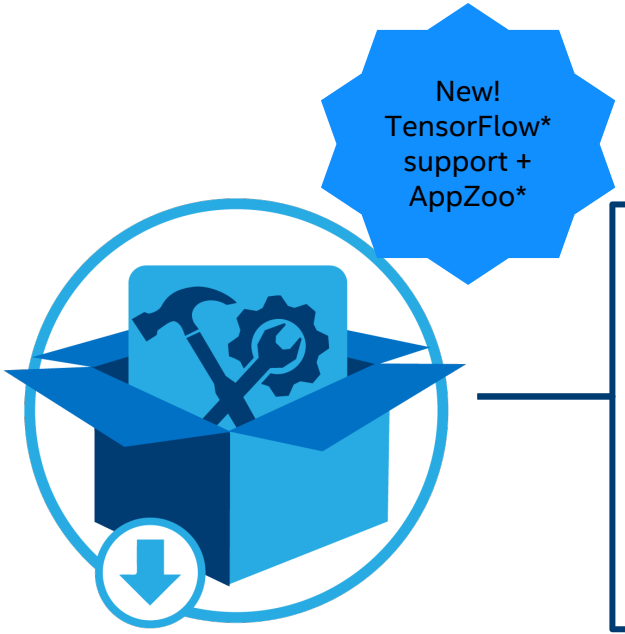
- Profiler
- Checker
- Compiler

NC API

- API

INTEL® MOVIDIUS™ NEURAL COMPUTE STICK

Redefining the AI developer kit



NC Toolkit

Profiler

Checker

Compiler

NC API

API

NC SDK

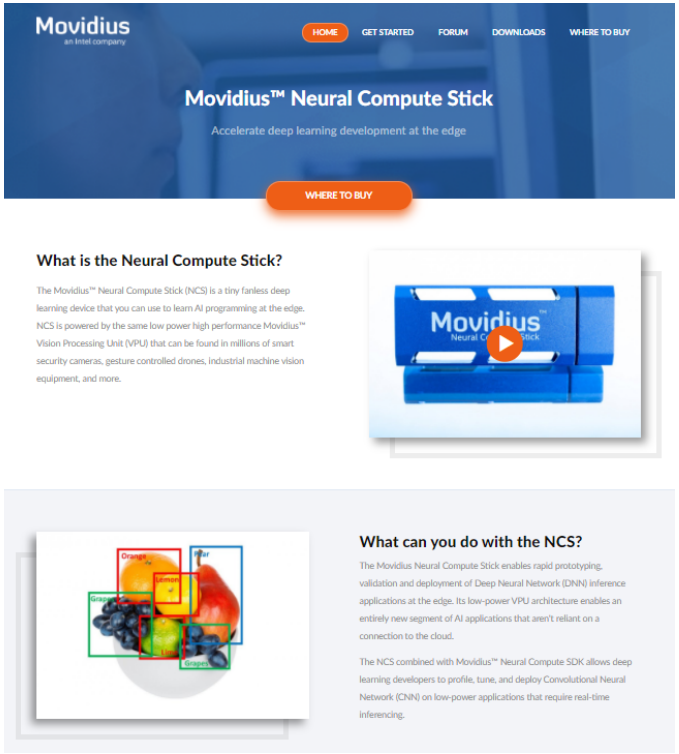
Free download @ developer.movidius.com

EXPLORE DEVELOPER.MOVIDIUS.COM

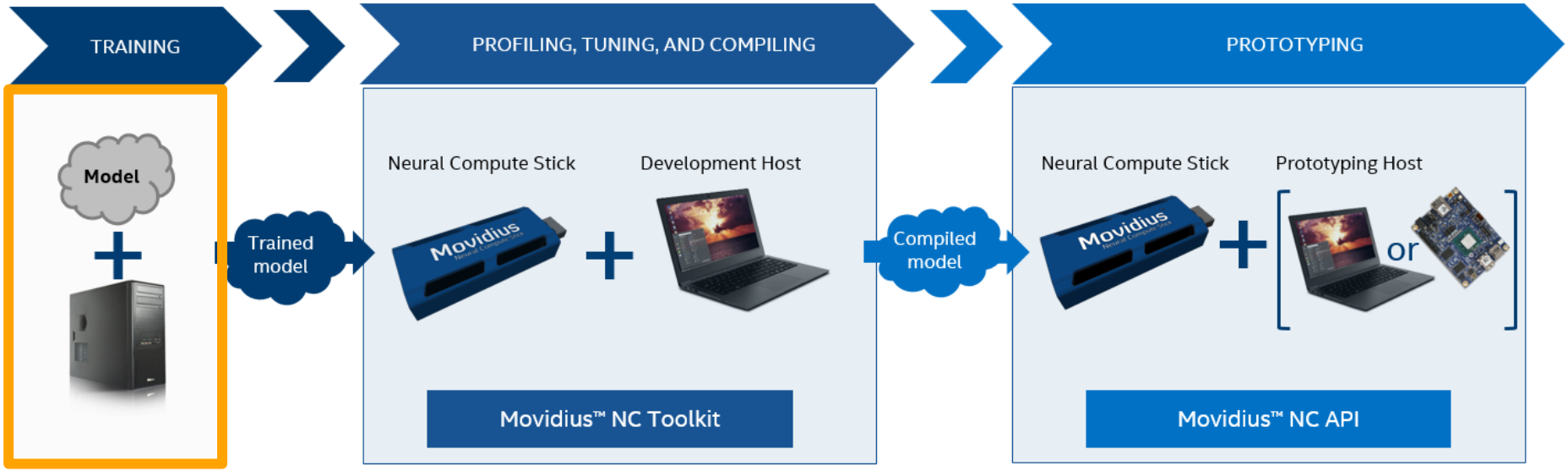
A developer-friendly website

Try out the following pages:

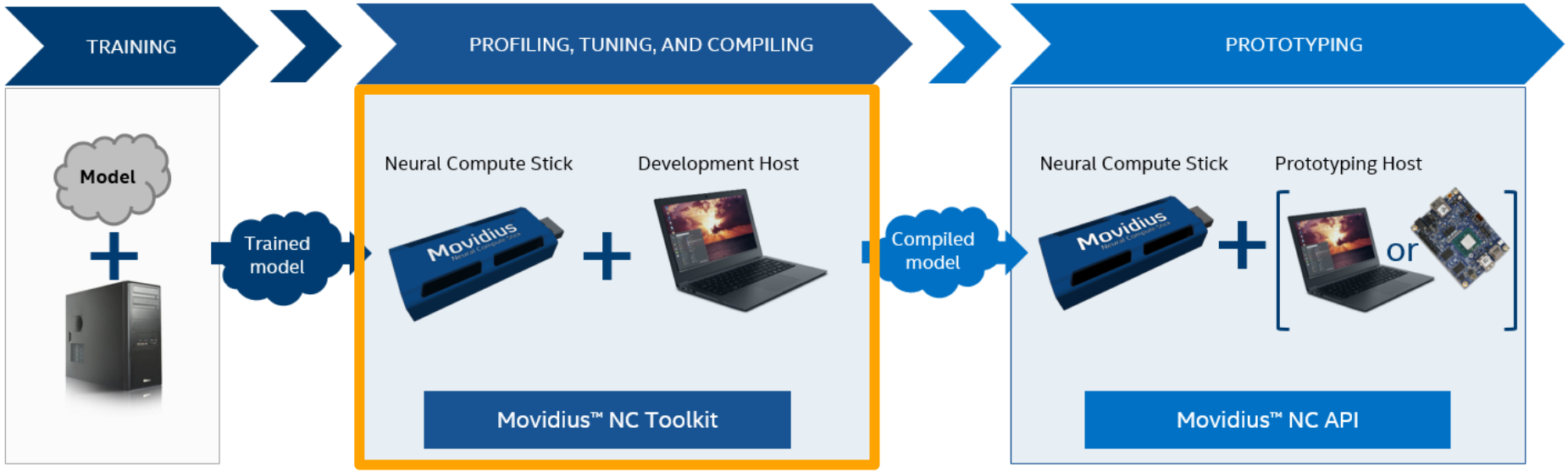
- Main page
- Getting started
- Downloads
- Docs
- Forums
- Where to buy



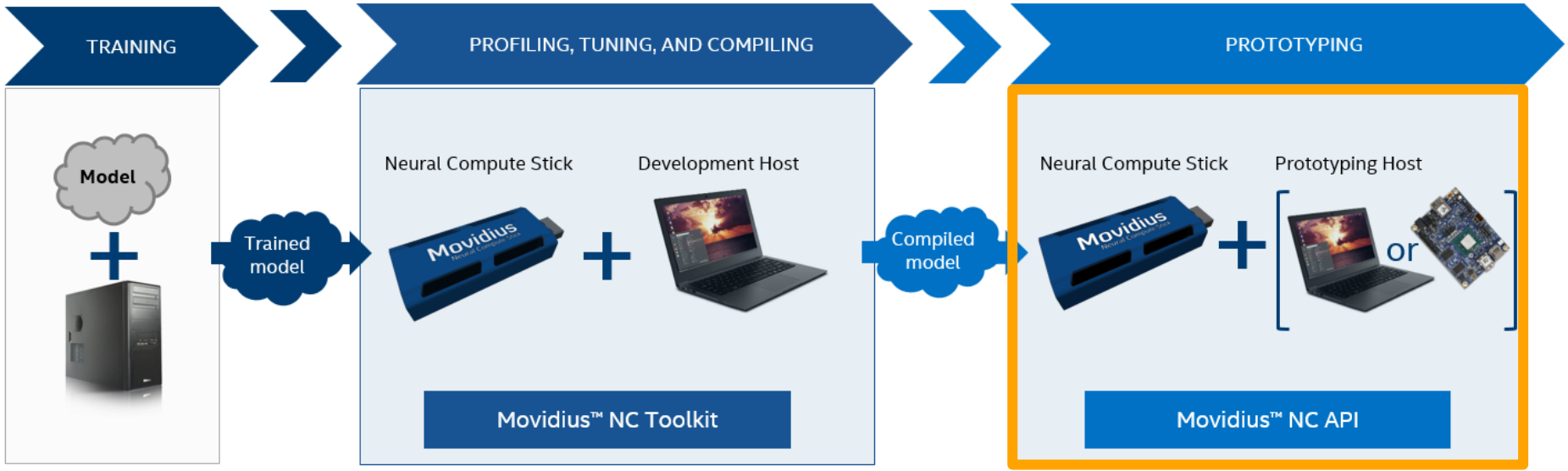
INTEL® MOVIDIUS™ SOFTWARE DEVELOPMENT KIT (SDK)



INTEL® MOVIDIUS™ SOFTWARE DEVELOPMENT KIT (SDK)



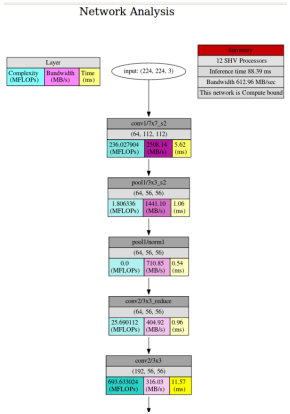
INTEL® MOVIDIUS™ SOFTWARE DEVELOPMENT KIT (SDK)



WHAT CAN I DO WITH THE NCS?

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

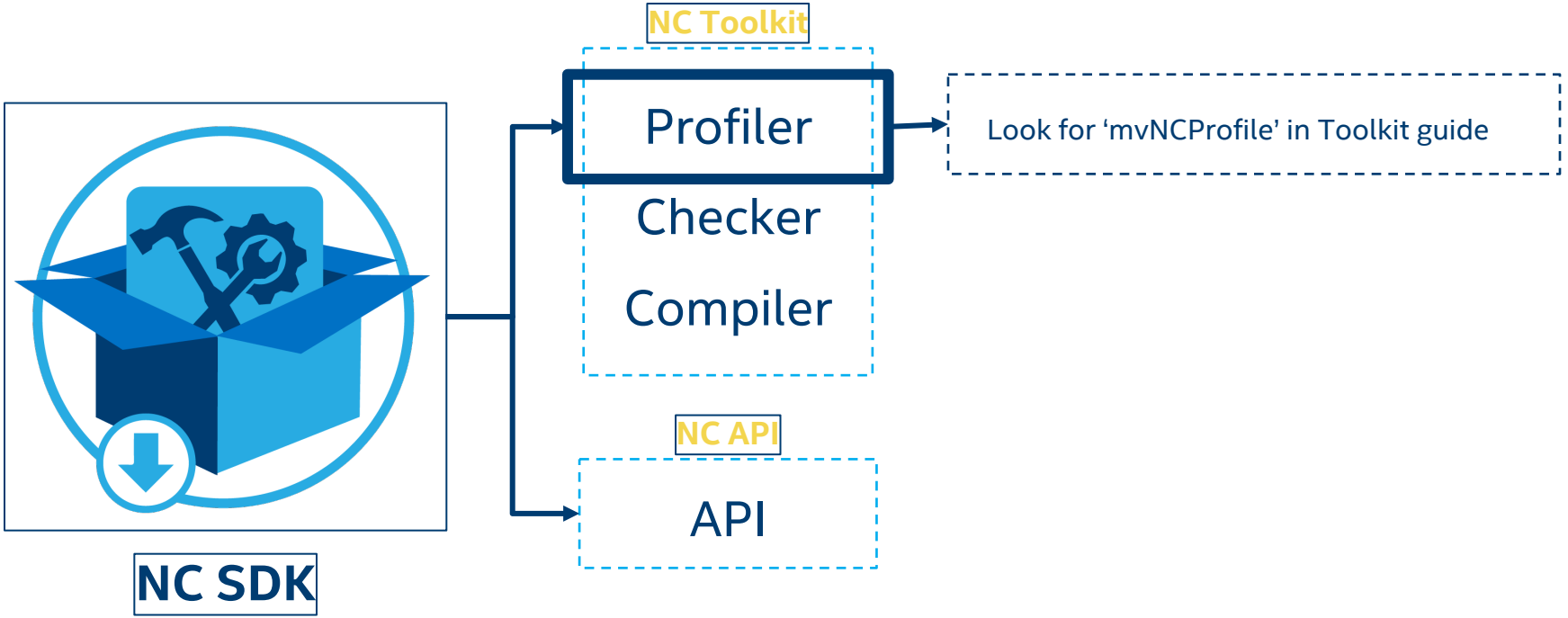
Status
GlobalOption
DeviceOption
GraphOption
EnumerateDevices
SetGlobalOption
LoadTensor
...

DNN architect/data scientist

Applications developer

PROFILER

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



CHECKER

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

NC SDK

NC Toolkit

Profiler

Checker

Compiler

NC API

API

Look for ‘mvNCCheck’ in Toolkit guide

COMPILER

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

NC SDK

NC Toolkit

Profiler

Checker

Compiler

Look for ‘mvNCCompile’ in Toolkit guide

NC API

API

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!

EXERCISE 1: INSTALL AND SET UP THE INTEL® MOVIDIUS™ NCSDK

THE FOUR-STEP PROCESS

- Step 1: Requirements
  - Ubuntu* 16.04, Rpi 3 Model B or Ubuntu VM
  - Intel® Movidius™ NCS
  - Intel® Movidius™ SDK
- Step 2: Install the SDK

```
mkdir -p ~/workspace

cd ~/workspace

git clone https://github.com/movidius/ncsdk.git

cd ~/workspace/ncsdk

make install
```

- Step 3: Test installation by running examples
  - Plug in NCS and run commands in terminal

```
cd ~/workspace/ncsdk

make examples
```

- Step 4: Well…there isn't one. You are ready to start!

EXERCISE 2: IMAGE CLASSIFICATION USING ALEXNET/GOOGLENET AND INCEPTION_V1

IMAGE CLASSIFICATION ON CAFFE* AND TENSORFLOW*

- Remember, we used :
  - AlexNet and GoogLeNet on Caffe
  - Inception_v1 on TensorFlow

- Let's create a graph file for these default models
  - Caffe/Alexnet: https://github.com/movidius/ncappzoo/tree/master/caffe/AlexNet
  - Caffe/GoogLeNet: https://github.com/movidius/ncappzoo/tree/master/caffe/GoogLeNet
  - TensorFlow/ Inception_v1
    https://github.com/movidius/ncappzoo/tree/master/tensorflow/inception_v

INSTRUCTIONS FOR CAFFE*/ALEXNET

1. mkdir -p ~/workspace
2. cd ~/workspace
3. git clone https://github.com/movidius/ncappzoo
4. cd ~/workspace/ncappzoo/caffe/AlexNet
5. make run_py

Reference: https://github.com/movidius/ncappzoo/tree/master/caffe/AlexNet

INSTRUCTIONS FOR CAFFE*/GOOGLENET

1. mkdir -p ~/workspace
2. cd ~/workspace
3. git clone https://github.com/movidius/ncappzoo
4. cd ~/workspace/ncappzoo/caffe/GoogLeNet
5. make run_py

Reference: https://github.com/movidius/ncappzoo/tree/master/caffe/GoogLeNet

INSTRUCTIONS FOR TENSORFLOW*/INCEPTION_V1

1. mkdir -p ~/workspace
2. cd ~/workspace
3. git clone https://github.com/movidius/ncappzoo
4. cd ~/workspace/ncappzoo/tensorflow/Inception_v1
5. make run

Reference: https://github.com/movidius/ncappzoo/tree/master/tensorflow/inception_v1

EXERCISE 3: CAN WE RUN ANY MODEL FROM WITHIN A SAMPLE IMAGE CLASSIFICATION APP?

YES! LET'S TRY IMAGE-CLASSIFIER APP IN NCAPPZOO

- Download the sample apps onto your computer

```
cd ~/workspace
git clone https://github.com/movidius/ncappzoo
cd ncappzoo/apps/image-classifier
```

- Edit the image-classifier app to set the following flags:
  - --graph: Set the path to graph created in Exercise 2
  - --image: Path to the static image
  - --dim: Topology specific: Use 224 X 224 for GoogLeNet/Inception_v1; 227 X 227 for AlexNet
  - --mean: Dataset specific. ILSVRC uses B=102, G=117, R=123
  - --colormode: Caffe* uses BGR, TensorFlow* uses RGB
  - --labels: Absolute path to the labels file that defines the categories

EXERCISE 4: DEPLOY THE MODEL FOR THE PETS CLASSIFICATION PROBLEM ON NCS

CREATING THE GRAPH

- We will now go back to the deploy.prototxt / snapshotxxx.caffemodel and frozen_graph.pb files saved during training.
- On Caffe*, run the below command at the terminal to generate the graph:
- **mvNCCompile deploy.prototxt –w snapshotxxx.caffemodel –s Num_of_shavecores**
- On TensorFlow*, run the below command at the terminal to generate the graph:
- **mvNCCompile frozen_graph.pb –s Num_of_shavecores –in=input –on=InceptionV1/Logits/Predictions/Reshape_1 –is 224 224 –o graph_name**
- Exercise: Try the image-classifier app from Exercise 3 to perform static inference.

  python3 image_classifier.py –graph graph_path –labels label_path –scale 0.00789 –dim 224 224 –image image_path

USE IMAGE-CLASSIFIER APP TO INFER USING THE BREEDS MODEL

- Edit the image-classifier app to set the following flags:
  - --graph: Set the path to graph created in Exercise 3
  - --image: Path to the static image
  - --dim: Topology specific: Use 224 X 224 for GoogLeNet/Inception_v1; 227 X 227 for AlexNet
  - --mean: Dataset specific. Breeds: R= 74.21, G=83.82, B= 89.90
  - --colormode: Caffe* uses BGR, TensorFlow* uses RGB
  - --labels: Absolute path to the labels file that defines the categories
  - --scale: Breeds dataset: 0.00789

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

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

Free Download

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

Accelerate Computer Vision Solutions

- **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

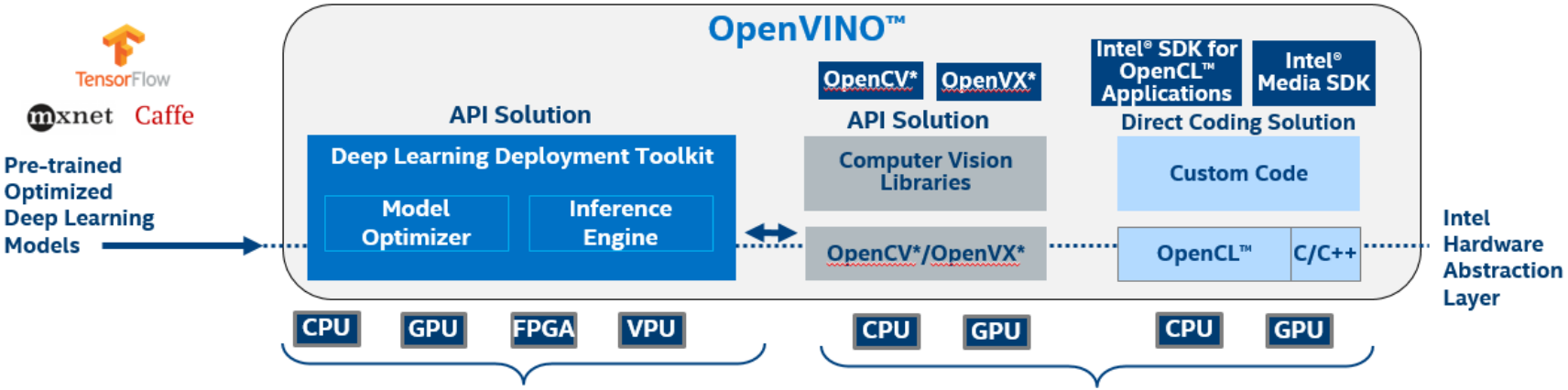
OS Support	CentOS* 7.4 (64 bit) Microsoft Windows* 10 (64 bit)	Ubuntu* 16.04.3 LTS (64 bit) Yocto Project* version Poky Jethro v2.0.3 (64 bit)
Intel® Architecture-Based Platforms Support	CELERON, ATOM, CORE, XEON, ARRIA 10, Movidius, IRIS Pro Graphics	

IR = Intermediate Representation file

intel AI

DEEP LEARNING VS. TRADITIONAL COMPUTER VISION

OPENVINO™ HAS TOOLS FOR AN END TO END VISION PIPELINE



DEEP LEARNING COMPUTER VISION

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

TRADITIONAL COMPUTER VISION

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

INTEL® DEEP LEARNING DEPLOYMENT TOOLKIT

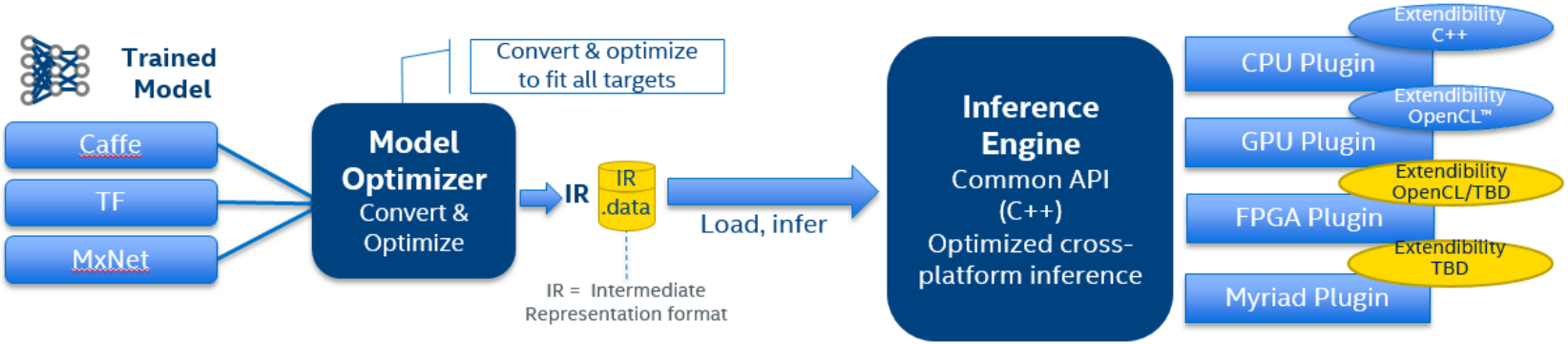
TAKE FULL ADVANTAGE OF THE POWER OF INTEL® ARCHITECTURE

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.



VALIDATE THE BREEDS CLASSIFICATION MODEL ON CPU AND GPU

STEPS TO INFER ON CPU/GPU USING THE SDK

- Install the Intel® OpenVINO™ SDK.
- Use the Model Optimizer to create the model.bin and model.xml files.
  - Caffe* command:
    - **python3 mo.py -- input_model model-file.caffemodel**
  - TensorFlow* command:
    - **python3 mo.py -- input_model model-file.pb**
- Load the Jupyter* Notebook for inference.
  - Command for CPU Inference:
    - **demo –d CPU –m model.xml –l labels.txt**
  - Command for GPU Inference:
    - **demo –d GPU –m model.xml –l labels.txt**

AI DEVCON 2018

DEPLOY THE BREEDS CLASSIFICATION MODEL TO AN EDGE DEVICE (RASPBERRY* PI)

CALL TO ACTION

LEVERAGE THE ADVANTAGES OF INTEL'S END-TO-END AI OFFERINGS

- Training
  - Take advantage of Intel® Xeon® Scalable Processors
  - Download and Install Intel® Optimized Caffe*
  - Download and install TensorFlow*
    - Pre-built wheels
- Inference
  - Download and install the Intel® Movidius™ Neural Compute Stick
  - Download and install the Intel® Computer Vision SDK
- Take advantage of AI courses and training available on Intel® Developer Zone

Q&A

THINGS TO KEEP IN MIND

1. You'll get access to the information covered in this session after the conference.

2. Visit the Intel® AI Academy for additional resources, training materials and videos related to today's presentation.
software.intel.com/AI

3. Download XYZ (NCSDK, CVSDK, Intel AI DevCloud) here to get hands on with today's tools, anytime.
<CLEAN URL>

4. Check out more examples of Intel AI/Movidius NCS/Intel AI DevCloud in action on DevMesh – Intel's Developer Network.
https://devmesh.intel.com/

SO… WHAT'S NEXT?

1. Visit the Intel® AI Academy for additional resources, training materials. and videos related to today's presentation.
   software.intel.com/AI

2. Download XYZ (NCSDK, CVSDK, Intel AI DevCloud) here to get hands on with today's tools, anytime
   <CLEAN URL>

3. Check out more examples of Intel AI/Movidius NCS/Intel AI DevCloud in action on DevMesh – Intel's Developer Network
   https://devmesh.intel.com/