AI ON INTEL

AI FROM THE DATA CENTER TO THE EDGE – AN OPTIMIZED PATH USING INTEL® ARCHITECTURE

LEGAL INFORMATION

DATASET CITATION

A Large and Diverse Dataset for Improved Vehicle Make and Model Recognition

F. Tafazzoli, K. Nishiyama and H. Frigui

In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops 2017.

COURSE COMPLETION CERTIFICATE

- You have the option to receive an Intel® AI Course Completion Certificate upon completion of the end of the course quiz.
- Before taking the quiz, you may have to disable AdBlockers. (Ghostery, uBlock, AdGuard, etc.)

LEARNING OBJECTIVE

Use Intel hardware and software portfolio and demonstrate the data science process

- Hands-on understanding of building a deep learning model and deploying to the edge
  - Use an enterprise image classification problem
  - Perform Exploratory Data Analysis on the VMMR dataset
  - Choose a framework and network
  - Train the model – obtain the graph and weights of the trained network
  - Deploy the model on CPU, integrated Graphics and Intel® Movidius™ Neural Compute Stick

TRAINING OUTLINE

1. **Intel's AI Portfolio**
   - Hardware: From training to inference with emphasis on 2nd Gen Intel® Xeon™ Scalable Processors
   - Software: Frameworks, libraries and tools optimized for Intel® Architecture
   - Community resources: Intel Developer Zone Resources

2. **Exploratory Data Analysis**
   - Obtain a dataset
   - Explore data visually to understand distribution
   - Data Reduction and address imbalances

3. **Training the models**
   - Infrastructure: Intel® AI DevCloud, Amazon Web Services*, Google Compute Engine*, Microsoft Azure*
   - Process: Prepare and visualize the dataset, prepare for consumption into framework, hyper-parameter tuning, training, validate

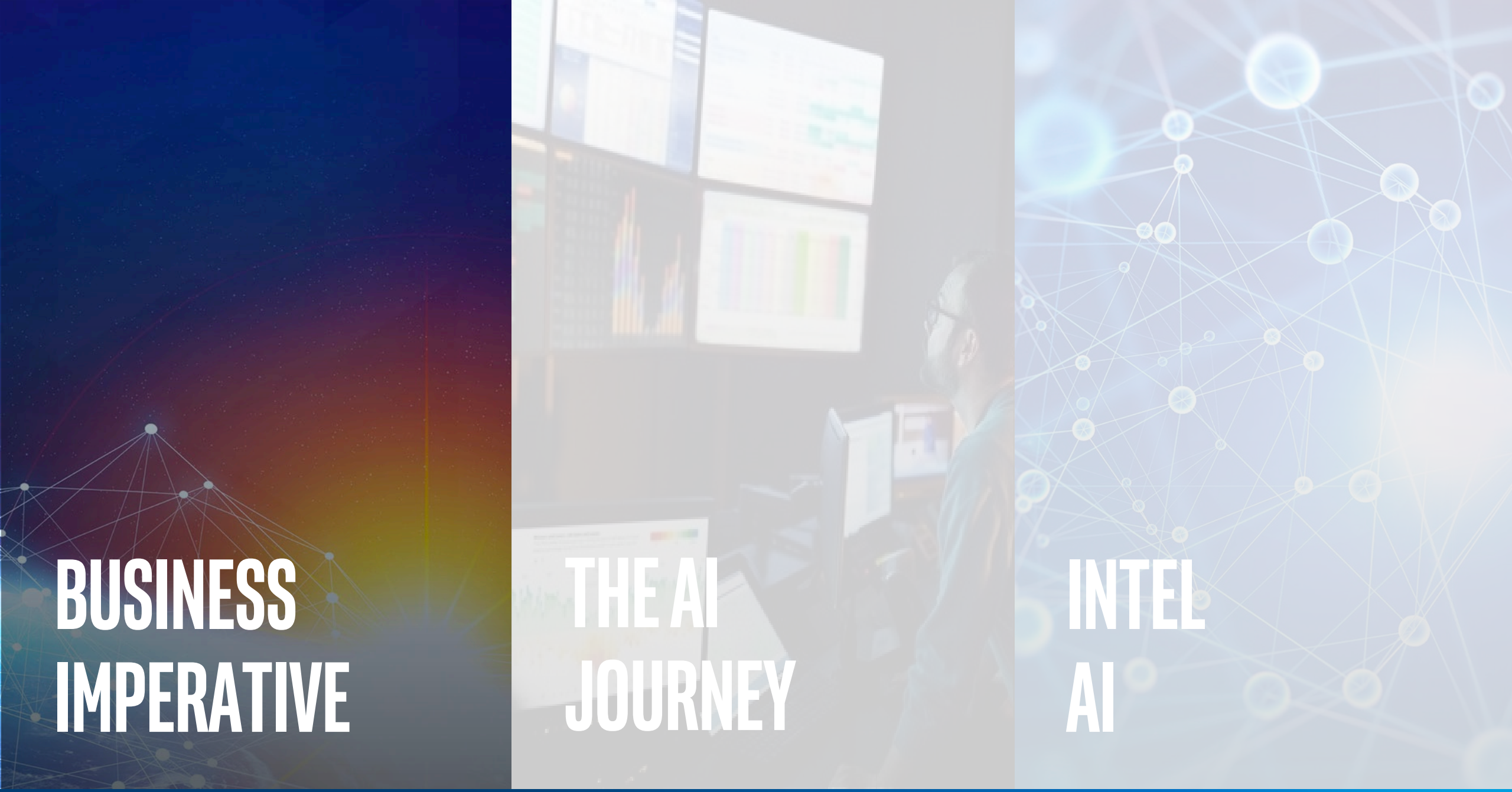
4. **Model Analysis**
   - Check your scores
   - Compare your results
   - Hyper parameter tuning
   - Pick the winner or go back to training

5. **Deploy to the edge / Inference**
   - Introduction to the Intel® OpenVINO™ Toolkit – Capabilities and benefits
   - Usage Models
   - Model Optimizer – Optimize model, generate hardware agnostic Intermediate Representation (IR) files for prebuilt and custom models
   - Inference Engine – Deploy to CPU, integrated GPU, FPGA and Intel® Movidius™ Neural Compute Stick

PREREQUISITES

- **Basic understanding of AI principles, Machine Learning and Deep Learning**
- **Coding experience with Python**
- **Some exposure to different frameworks – Tensorflow*, Caffe* etc.**
- **Here are some tutorials to get you stared**
  - Introduction to AI
  - Machine Learning
  - Deep Learning
  - Applied Deep Learning with Tensorflow*


intel AI
INTEL AI PORTFOLIO



BUSINESS IMPERATIVE

THE AI JOURNEY

INTEL AI

WHY AI NOW?

DATA DELUGE (2019)

25 GB[1] PER MONTH
Internet User

50 GB[2] PER DAY
Smart Car

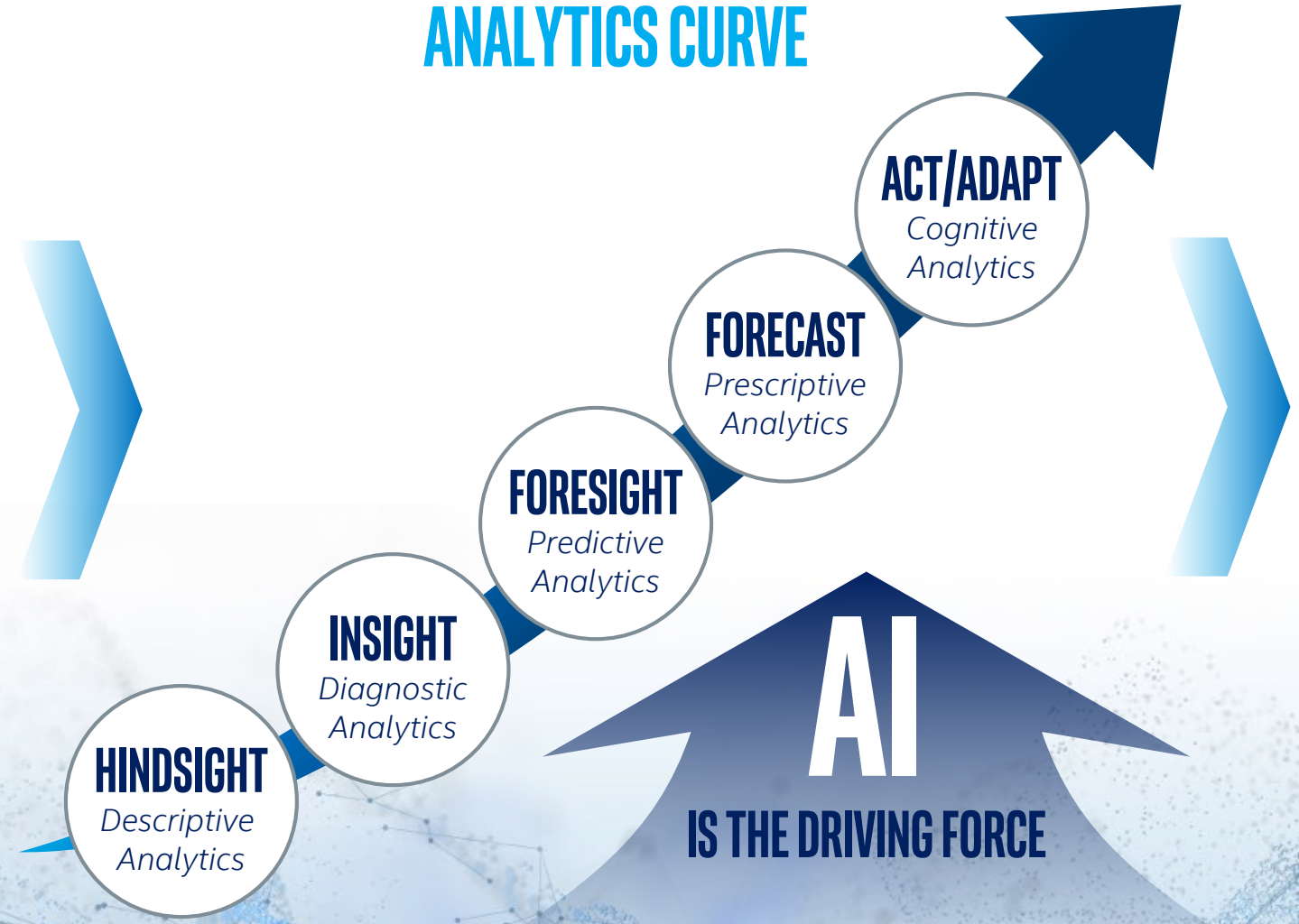
30 TB[2] PER DAY
Smart Hospital

40 TB[2] PER DAY
Airplane Data

1 PB[2] PER DAY
Smart Factory

50 PB[2] PER DAY
City Safety

ANALYTICS CURVE



INSIGHTS

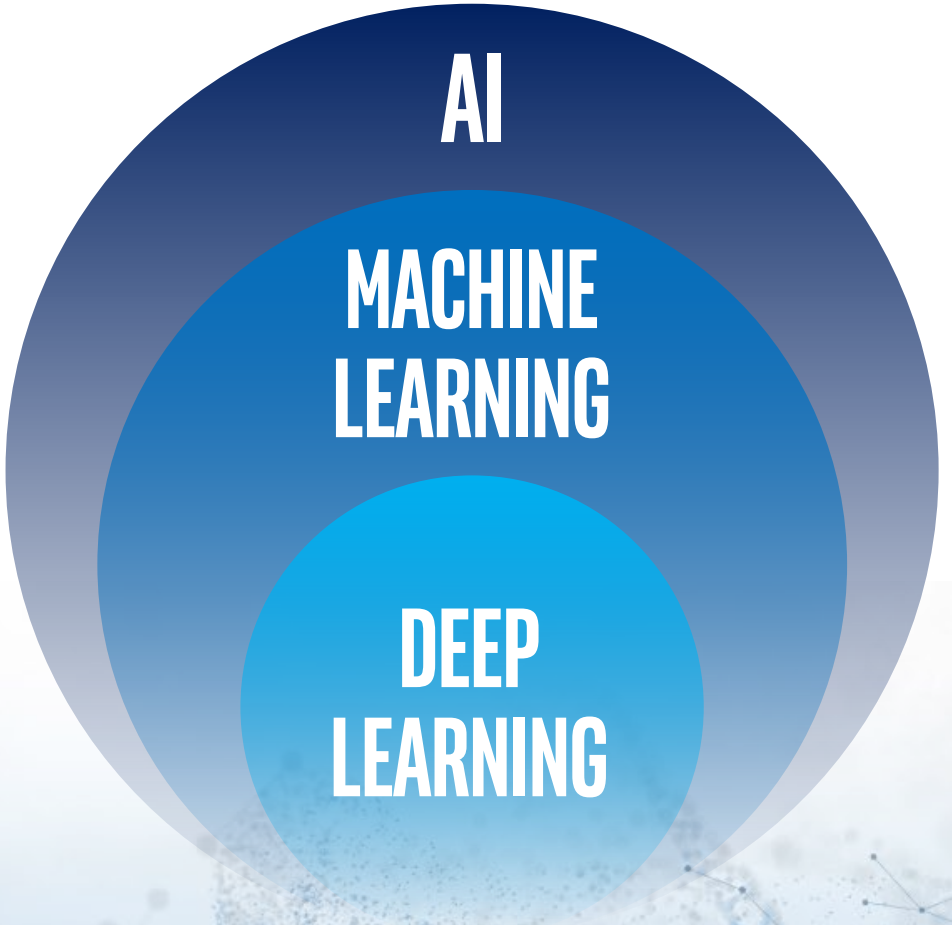
BUSINESS

OPERATIONAL

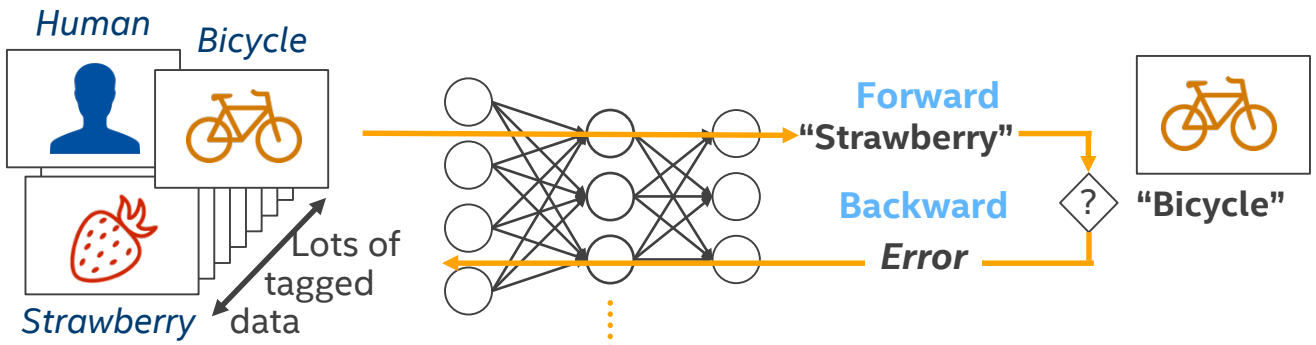
SECURITY

1. Source: http://www.cisco.com/c/en/us/solutions/service-provider/vni-network-traffic-forecast/infographic.html
2. Source: https://www.cisco.com/c/dam/m/en_us/service-provider/ciscoknowledgenetwork/files/547_11_10-15-DocumentsCisco_GCI_Deck_2014-2019_for_CKN__10NOV2015_.pdf

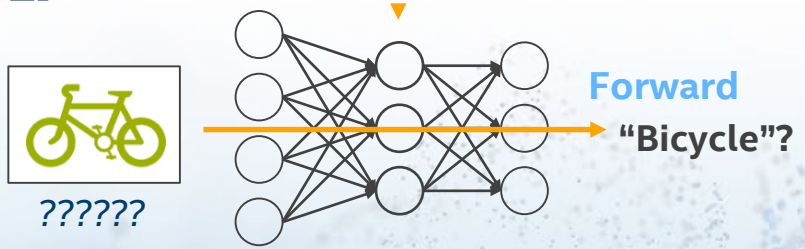
WHAT IS AI?



TRAINING:

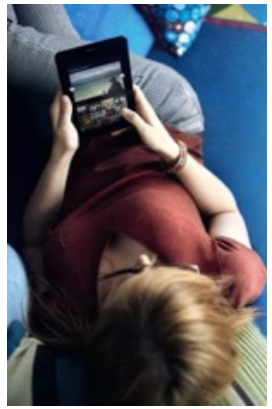


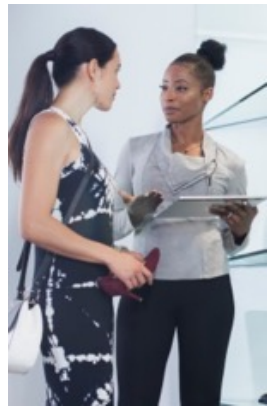
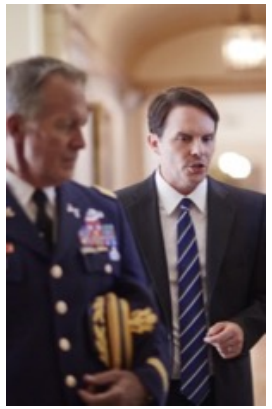
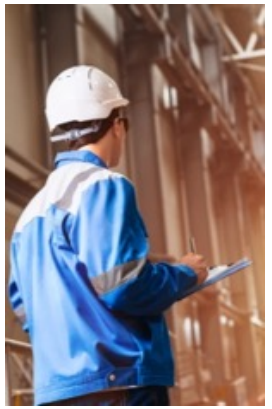
INFERENCE:



MANY DIFFERENT APPROACHES TO AI

AI WILL TRANSFORM




CONSUMER

- Smart Assistants
- Chatbots
- Search
- Personalization
- Augmented Reality
- Robots

HEALTH

- Enhanced Diagnostics
- Drug Discovery
- Patient Care
- Research
- Sensory Aids

FINANCE

- Algorithmic Trading
- Fraud Detection
- Research
- Personal Finance
- Risk Mitigation

RETAIL

- Support
- Experience
- Marketing
- Merchandising
- Loyalty
- Supply Chain
- Security

GOVERNMENT

- Defense
- Data Insights
- Safety & Security
- Resident Engagement
- Smarter Cities

ENERGY

- Oil & Gas Exploration
- Smart Grid
- Operational Improvement
- Conservation

TRANSPORT

- In-Vehicle Experience
- Automated Driving
- Aerospace
- Shipping
- Search & Rescue

INDUSTRIAL

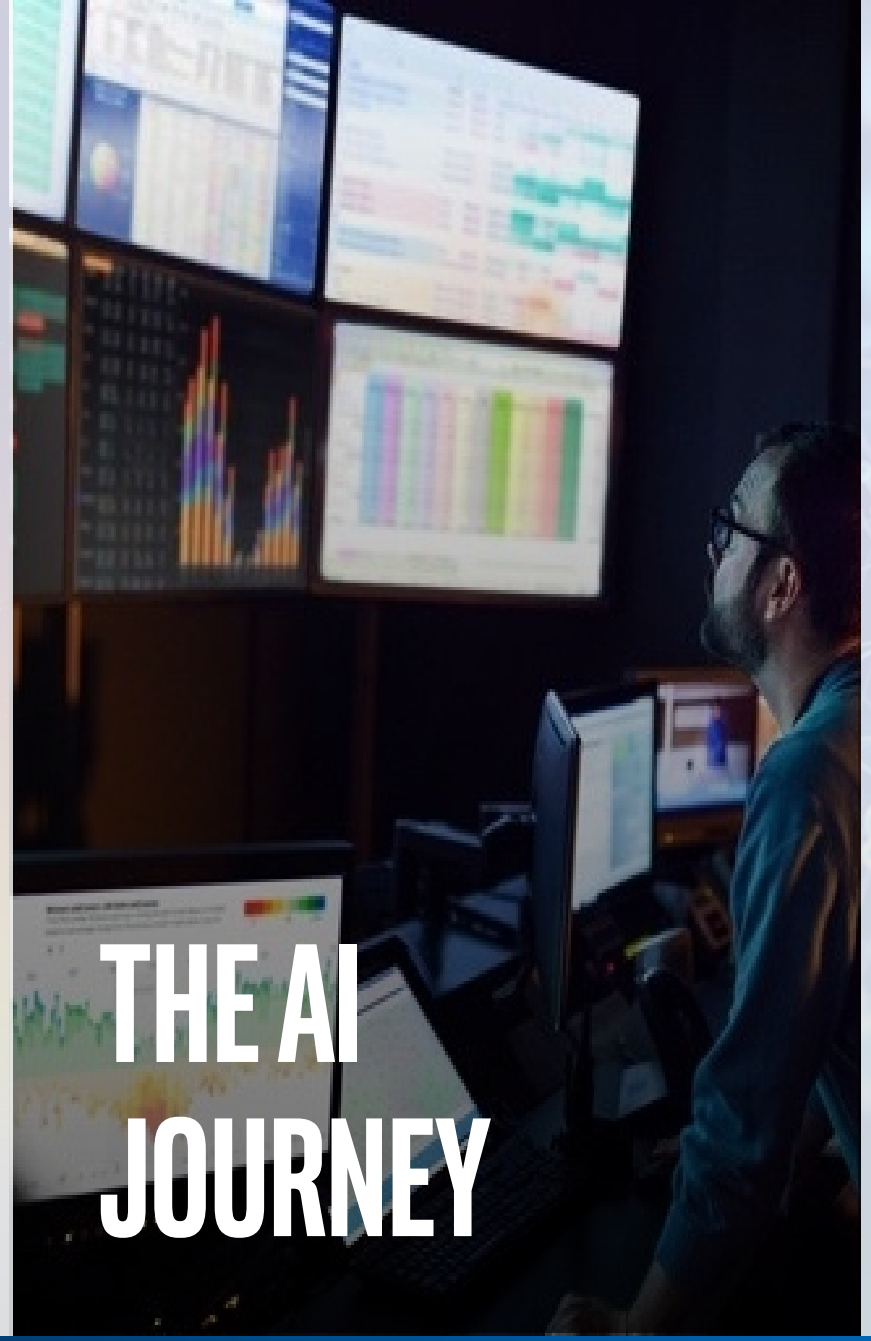
- Factory Automation
- Predictive Maintenance
- Precision Agriculture
- Field Automation

OTHER

- Advertising
- Education
- Gaming
- Professional & IT Services
- Telco/Media
- Sports

Source: Intel forecast

BUSINESS IMPERATIVE

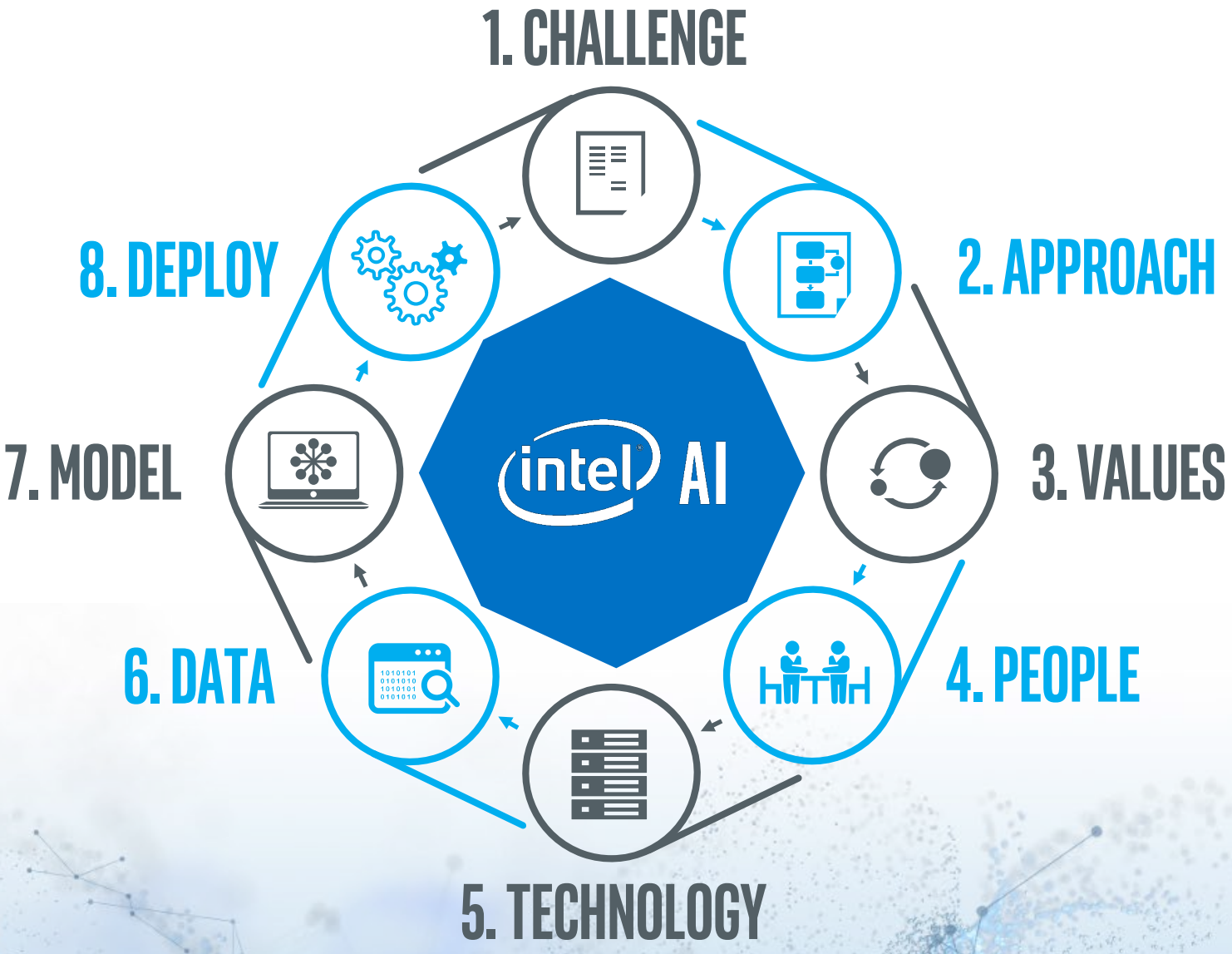


THE AI JOURNEY

INTEL AI

THE AI JOURNEY

Partner with Intel
to accelerate your AI journey



BUSINESS IMPERATIVE

THE AI JOURNEY

INTEL AI

BREAKING BARRIERS BETWEEN AI THEORY AND REALITY

PARTNER WITH INTEL TO ACCELERATE YOUR AI JOURNEY

TAME YOUR DATA DELUGE
with our data layer expertise

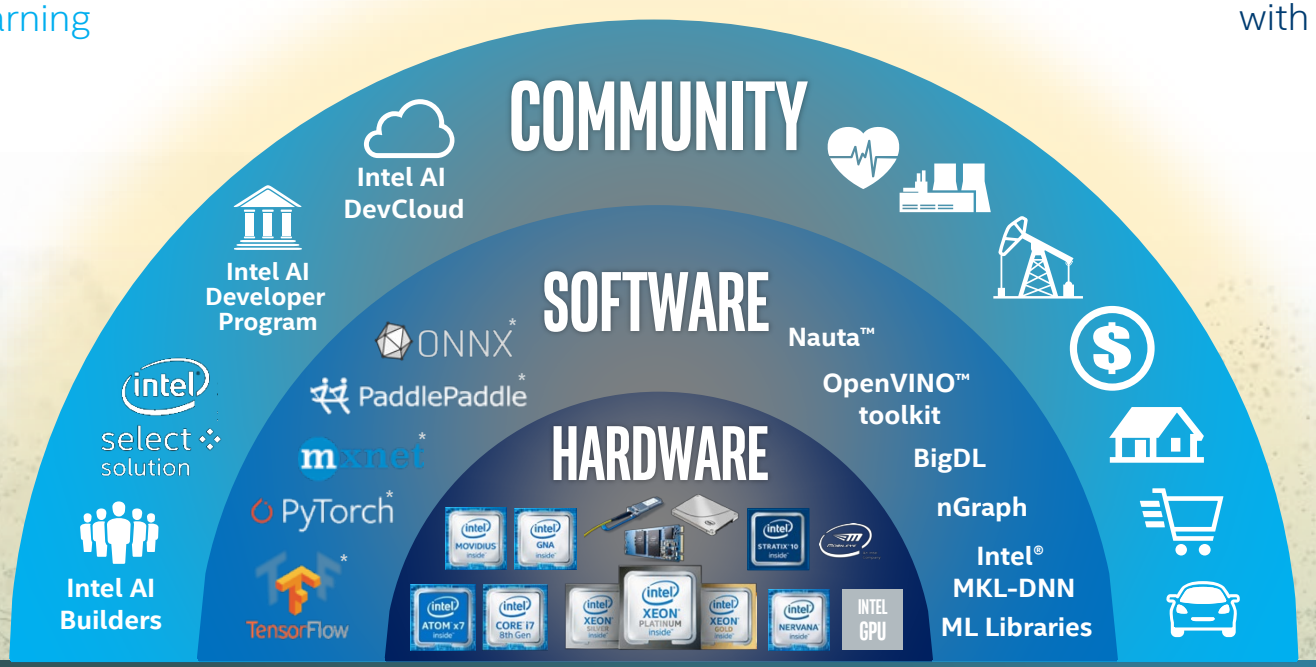
SPEED UP DEVELOPMENT
with open AI software

CHOOSE ANY APPROACH
from analytics to deep learning

DEPLOY AI ANYWHERE
with unprecedented HW choice

SIMPLIFY AI
via our robust community

SCALE WITH CONFIDENCE
on the platform for IT & cloud



HARDWARE

DEPLOY AI ANYWHERE

WITH UNPRECEDENTED HARDWARE CHOICE

DEVICE

EDGE

MULTI-CLOUD

intel ATOM inside + intel GNA inside

intel CORE i7 inside + intel Iris Graphics

intel XEON SILVER inside | intel XEON PLATINUM inside | intel XEON GOLD inside

AND/OR

ADD

ACCELERATION

DEDICATED MEDIA/VISION

intel MOVIDIUS inside

AUTOMATED DRIVING

MOBILEYE An Intel Company

DEDICATED DL TRAINING

intel 2H'19 NERVANA inside

NNP-L

FLEXIBLE ACCELERATION

intel FPGA

DEDICATED DL INFERENCE

intel 2H'19 NERVANA inside

NNP-I

GRAPHICS, MEDIA & ANALYTICS ACCELERATION

FUTURE GPU

**FPGA: (1) First to market to accelerate evolving AI workloads (2) AI+other system level workloads like AI+I/O ingest, networking, security, pre/post-processing, etc (3) Low latency memory constrained workloads like RNN/LSTM*

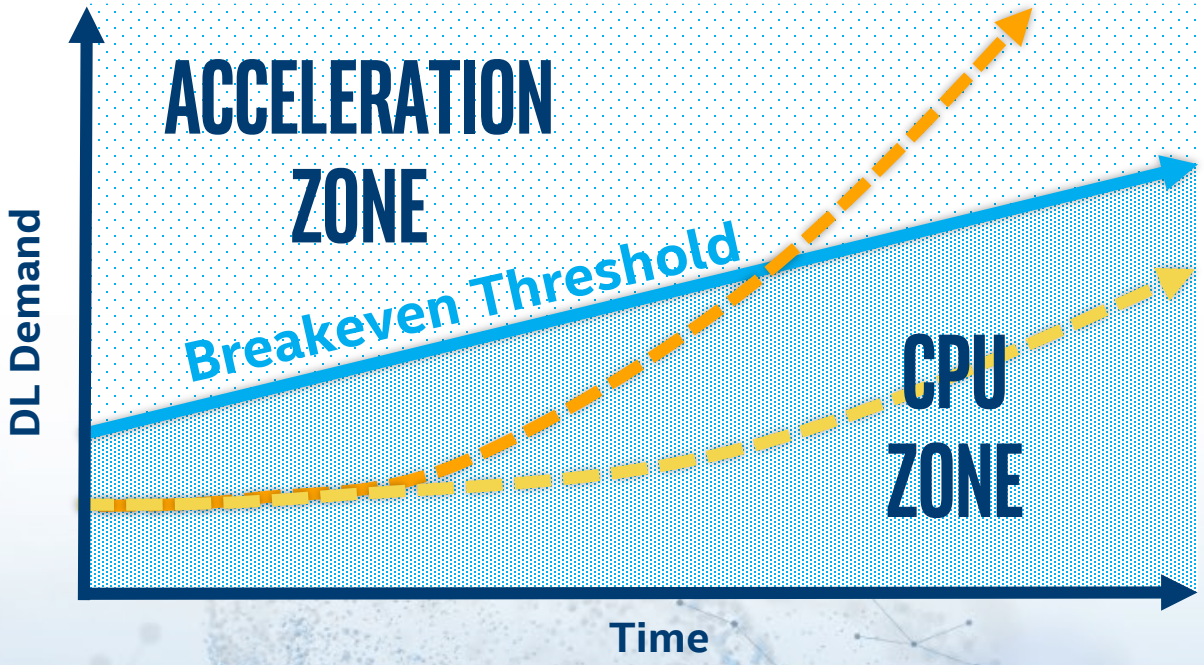
[1]GNA=Gaussian Neural Accelerator

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

Images are examples of intended applications but not an exhaustive list.

THE DEEP LEARNING MYTH

"A GPU IS REQUIRED FOR DEEP LEARNING..."



FALSE

- **Most businesses (---)**
  will use the CPU for their AI & deep learning needs
- **Some early adopters (---)**
  may reach a tipping point when acceleration is needed[1]

[1]"Most businesses" claim is based on survey of Intel direct engagements and Intel analyst market segment analysis.

SOFTWARE

SPEED UP DEVELOPMENT

WITH OPEN AI SOFTWARE

TOOLKITS

App developers

DEEP LEARNING DEPLOYMENT

Intel® Distribution of OpenVINO™ Toolkit[1]

Deep learning inference deployment on CPU/GPU/FPGA/VPU for Caffe*, TensorFlow*, MXNet*, ONNX*, Kaldi*

Nauta (Beta)

Open source, scalable, and extensible distributed deep learning platform built on Kubernetes

LIBRARIES

Data scientists

MACHINE LEARNING (ML)

Python

- Scikit-learn
- Pandas
- NumPy

R

- Cart
- Random Forest
- e1071

Distributed

- MlLib (on Spark)
- Mahout

DEEP LEARNING FRAMEWORKS

COMING SOON!

Optimized for CPU & more

TensorFlow | mxnet | Caffe2 | PyTorch | BigDL for Apache Spark

More framework optimizations underway (e.g. PaddlePaddle*, CNTK* & more)

Status & installation guides

KERNELS

Library developers

ANALYTICS & ML

Intel® Distribution for Python*

Intel distribution optimized for machine learning

Intel® Data Analytics Library

Intel® Data Analytics Acceleration Library (incl machine learning)

DEEP LEARNING

Intel® Math Kernel Library for Deep Neural Networks

(Intel® MKL-DNN)

Open source DNN functions for CPU / integrated graphics

DEEP LEARNING GRAPH COMPILER

Intel® nGraph™ Compiler (Beta)

Open source compiler for deep learning model computations optimized for multiple devices (CPU, GPU, NNP) from multiple frameworks (TF, MXNet, ONNX)

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

COMMUNITY

INTEL® AI ACADEMY

FOR DEVELOPERS, STUDENTS, INSTRUCTORS AND STARTUPS

Get smarter using online tutorials, webinars, student kits and support forums

Educate others using available course materials, hands-on labs, and more

Get 4-weeks FREE access to the Intel® AI DevCloud, use your existing Intel® Xeon® Processor-based cluster, or use a public cloud service

Showcase your innovation at industry & academic events and online via the Intel AI community forum

intel Software

SHARE YOUR STUDENT PROJECT WITH THE WORLD

LEARN MORE ON DEVMESH

OPPORTUNITIES TO SHARE YOUR PROJECTS AS AN INTEL® STUDENT AMBASSADOR

- Industry events via sponsored speakerships
- Student Workshops
- Ambassador Labs
- Intel® Developer Mesh

AI BUILDERS: ECOSYSTEM 100+ AI Partners

CROSS VERTICAL

OEM

COLFAX Customized Solutions · DELL EMC · hp · Lenovo

SYSTEM INTEGRATORS

accenture High performance. Delivered. · TATA CONSULTANCY SERVICES · Lambda · NCR · NTT DATA · Mobiliya · wipro

VERTICAL

HEALTHCARE

OrboGraph Intelligent Payment Automation · Carestream · SARADA · SIGTUPLE · HEARTVISTA · lu

FINANCIAL SERVICES

WorkFusion · NEXT labs · vPhrase · Alpaca · nuueo · Allgoo · Arya.ai

RETAIL

AiHello · NEXUS EDGE · locaweb · bigdatacorp. · GIGASPACES innovate with confidence

TRANSPORTATION

GLOBAL EDGE · Playment · ERA · WITRONIX

NEWS, MEDIA & ENTERTAINMENT

sensifai · Gramener A Data Science Company · ZIVA · keemotion · Taboola

AGRICULTURE

LeBov · tbit

LEGAL & HR

reachr · finch soluções · LEGAL LABS · seedlink PEOPLE SCIENCE

ROBOTIC PROCESS AUTOMATION

UiPath

HORIZONTAL

BUSINESS INTELLIGENCE & ANALYTCS

DataRobot · Σ SIGOPT

VISION

VISIONGENII Being Different · Matroid · restb.ai · Linker Networks · imagga

CONVERSATIONAL BOTS

[24]7.ai · SET SAIL · nama · avaamo · gamalon

AI TOOLS & CONSULTING

Mighty AI · axondata · iMerit human-empowered computing · Julia computing · cloudera · SET SAIL · LEAPMIND · QuikFynd · CALLIGO TECHNOLOGIES High Performance & Beyond! · UiPath · minds.ai · nologin

AI PAAS

Arya.ai · DOMINO · Paperspace · H2O · IoT · RocketML · KuberLab

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

PARTNER WITH INTEL TO ACCELERATE YOUR AI JOURNEY

WHY INTEL AI?

Simplify AI
via our robust community

Tame your data deluge
with our data layer experts

Choose any approach
from analytics to deep learning

Speed up development
with open AI software

Deploy AI anywhere
with unprecedented HW choice

Scale with confidence
on the engine for IT & cloud

RESOURCES

- **Intel® AI Academy**

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

- **Intel® AI Student Kit**

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

- **Intel® AI DevCloud**

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

- **Intel® AI Academy Support Community**

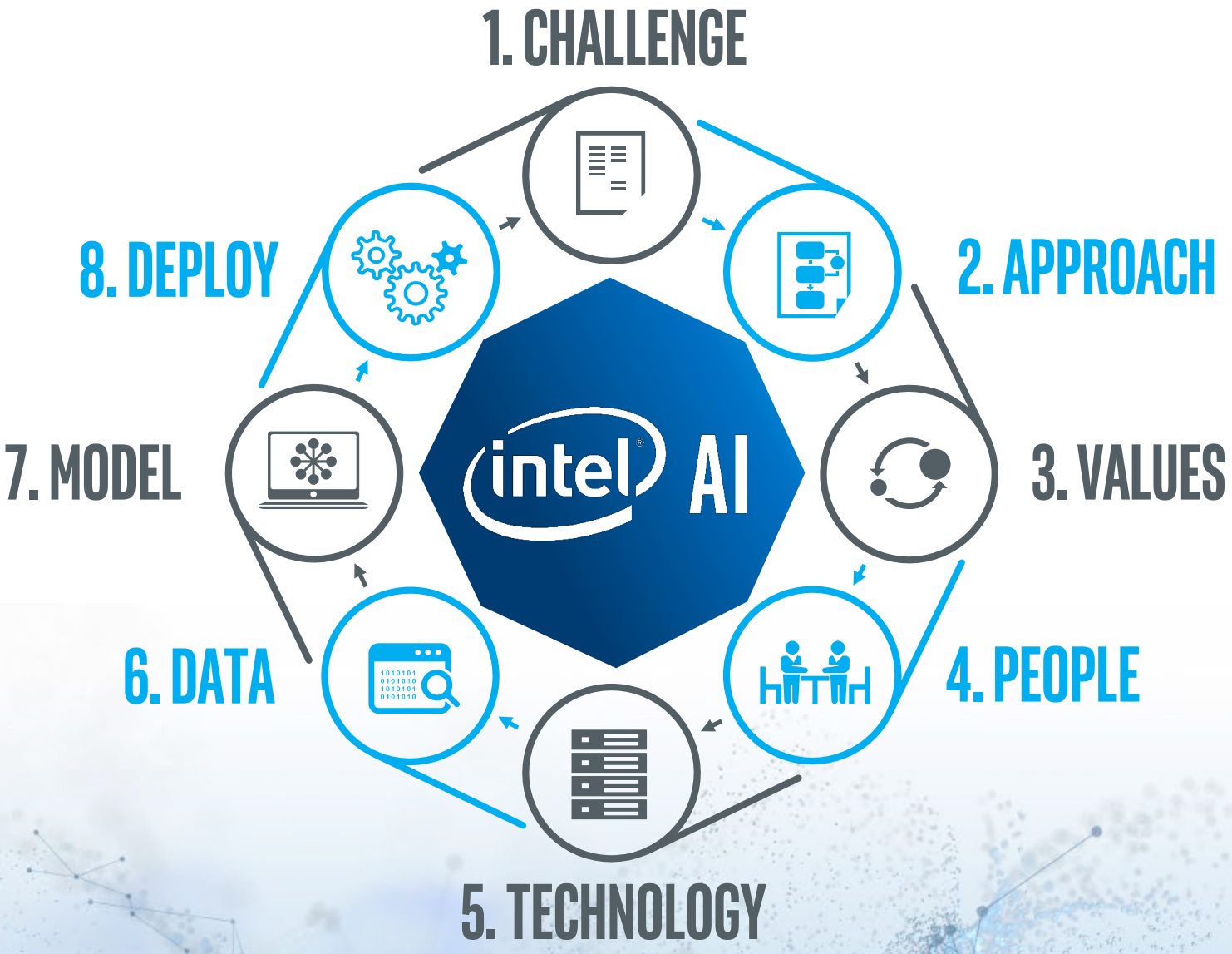
  https://communities.intel.com/community/tech/intel-ai-academy

- **DevMesh**

  https://devmesh.intel.com


intel AI
AI JOURNEY WITH INTEL

THE AI JOURNEY WITH AN INTEL CASE STUDY



INTEL AI CASE STUDY

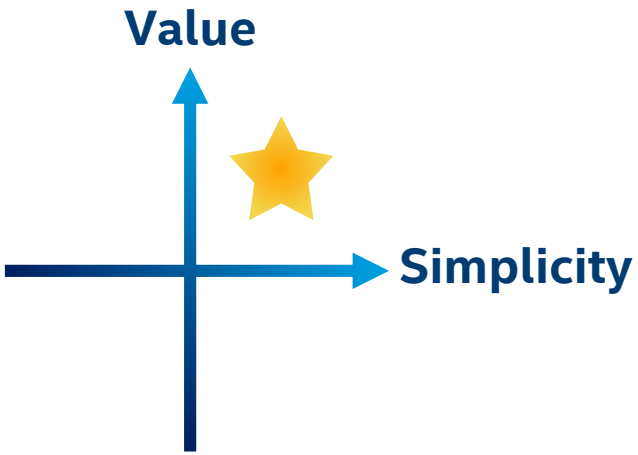
- CHALLENGE
- APPROACH
- VALUES
- PEOPLE
- TECHNOLOGY
- DATA
- MODEL
- DEPLOY

Brainstorm opportunities using the 70+ AI solutions in Intel's portfolio and rank the business value of each

Identify approach & complexity of each solution with Intel's guidance; choose high-ROI industrial defect detection using DL[1]

Discuss ethical, social, legal, security & other risks and mitigation plans with Intel experts prior to kickoff

Secure internal buy-in for AI pilot and new SW development philosophy, grow talent via Intel AI developer program



intel AI DEVELOPER PROGRAM

[1]DL = Deep Learning

INTEL AI CASE STUDY (CONT'D)

CHALLENGE
APPROACH
VALUES
PEOPLE
TECHNOLOGY
DATA
MODEL
DEPLOY

Prepare **data for model development working with Intel and/or partner to get the time-consuming data layer right (~12 weeks)**

Source Data	Transmit Data	Ingest Data	Cleanup Data	Integrate Data	Stage Data
10%	*5%*	*5%*	*60%*	*10%*	*10%*

Develop **model by training, testing inference and documenting results working with Intel and/or partner for the pilot (~12 weeks)**

Train *(Topology Experiments)*	Train *(Tune Hyper-parameters)*	Test Inference	Document Results
30%	*30%*	*20%*	*20%*

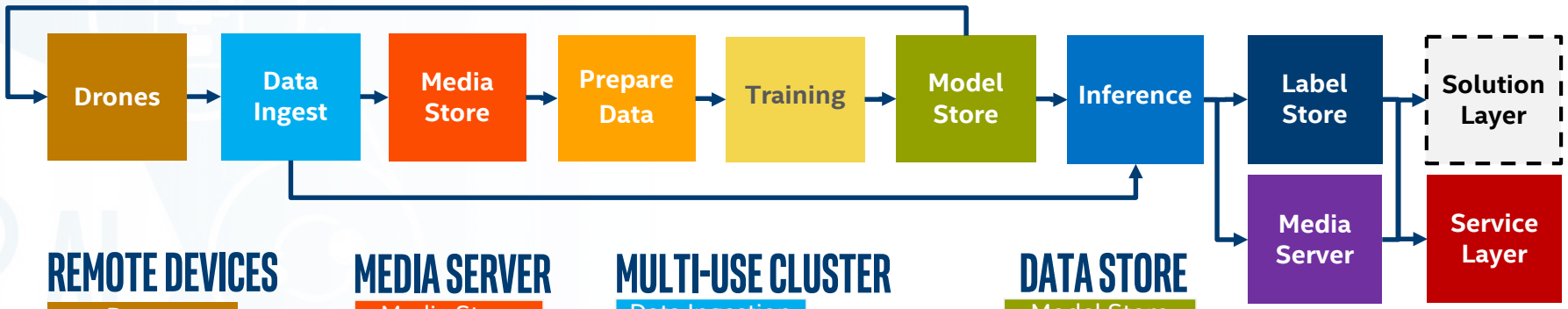
Project breakdown is approximated based on engineering estimates for time spent on each step in this real customer POC/pilot; time distribution is expected to be similar but vary somewhat for other deep learning use cases

INTEL AI CASE STUDY (CONT'D)

CHALLENGE
APPROACH
VALUES
PEOPLE
TECHNOLOGY
DATA
MODEL
DEPLOY

Engage Intel AI Builders partner to deploy & scale

intel AI BUILDERS



REMOTE DEVICES

Drone
Drone
Drone

10 Drones

Real-time object detection and data collection

Drone
Drone
Drone

Per Drone

1x Intel® Core™ processor
1x Intel® Movidius™ VPU

MEDIA SERVER

Media Store
Media Store
Media Store

110 Nodes

8 TB/day per camera

10 cameras

3x replication

1-year video retention

4 mgmt nodes

Media Store
Media Store
Media Store

Per Node

1x 2S 61xx 20x 4TB SSD

MULTI-USE CLUSTER

Data Ingestion
Data Ingestion
Data Ingestion
Data Ingestion
Inference
Inference
Inference
Inference
Prepare Data
Prepare Data
Service Layer
Service Layer
Service Layer
Media Server
Media Server
Media Server
Media Server

4 Nodes
One ingestion per day, one-day retention

4 Nodes
20M frames per day

2 Nodes
Infrequent op

3 Nodes
Simultaneous users

3 Nodes
10k clips stored

DATA STORE

Model Store
Model Store
Model Store
Model Store
Label Store
Label Store
Label Store
Label Store

Per Node

1x 2S 81xx
5x 4TB SSD

4 Nodes
1-year of history

4 Nodes
Labels for 20M frames /day

ADV. ANALYTICS

Training

16 Nodes

Intermittent use
1 training/month
for <10 hours

Training

Per Node

1x 2S 81xx
1x 4TB SSD

SOFTWARE

- *OpenVino™ Toolkit*
- *Intel® Movidius™ SDK*
- *TensorFlow**
- *Intel® MKL-DNN*

KEY LEARNING

AI in the real world is much more involved than in the lab

In most cases, acquiring the data for the challenge at hand, preparing it for training is as time consuming as training and model analysis phases

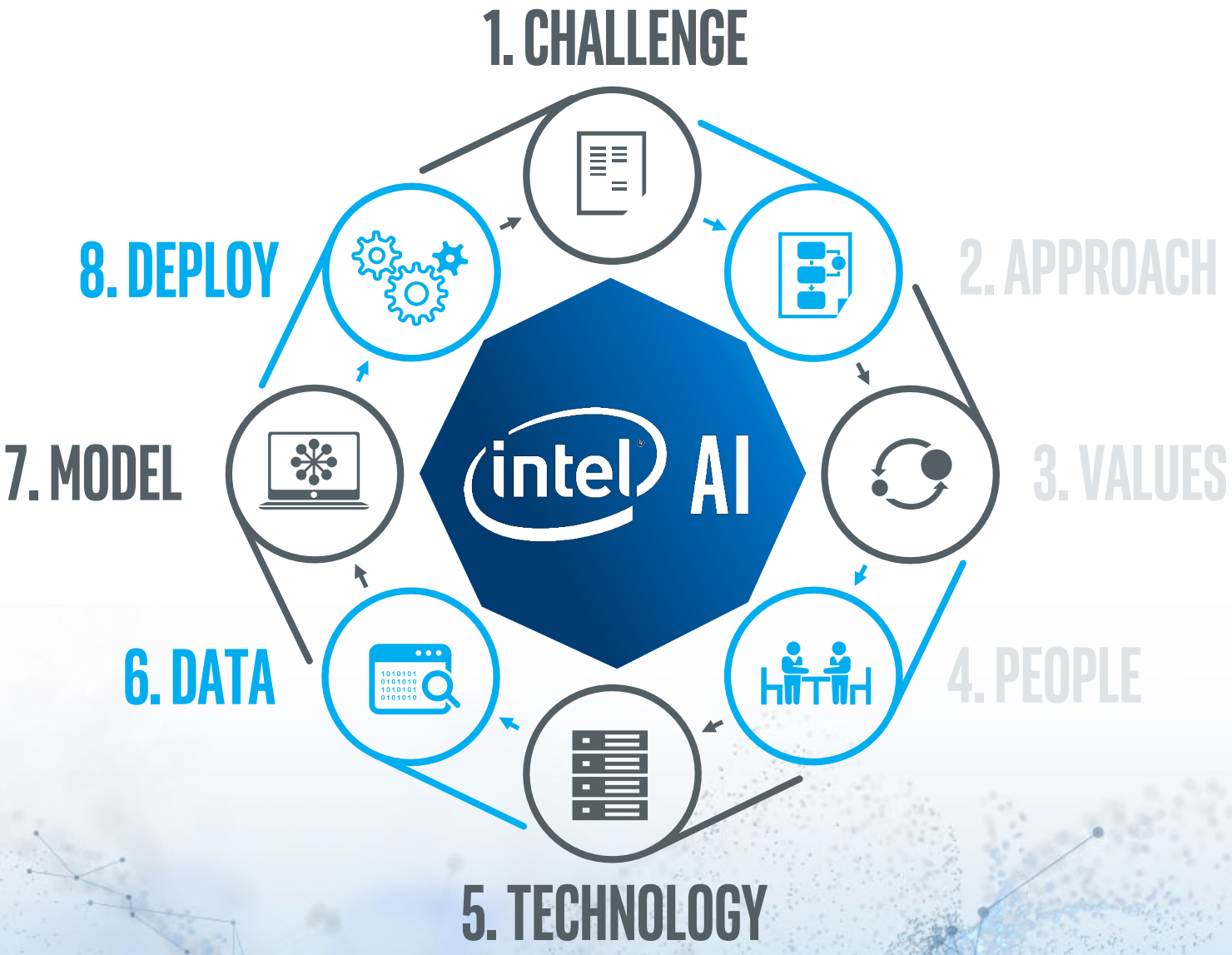
Most often, the entire process takes weeks to months to complete

ADDRESSING THE AI JOURNEY IN THE CLASSROOM

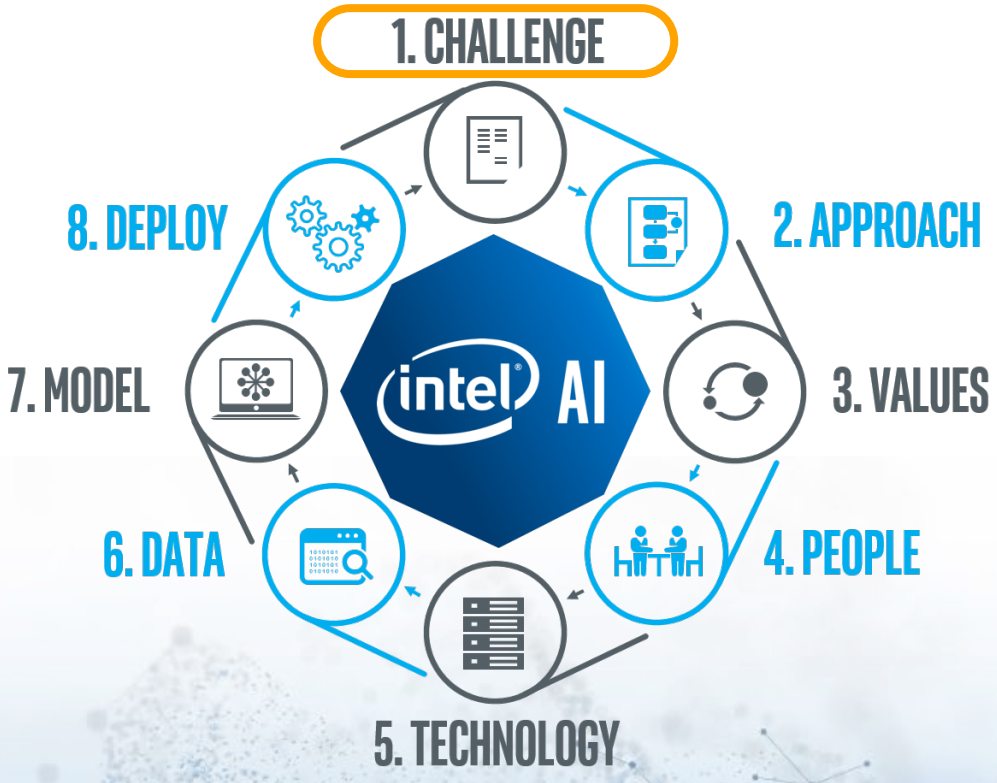
- **An enterprise problem is too large and complex to address in a classroom**
- **Pick a smaller challenge and understand the steps to later apply to your enterprise problems**
- **The AI journey in the class today will focus on:**
  - Defining a challenge
  - Technology choices
  - Obtaining a dataset and exploratory data analysis
  - Training a model and deploying it on CPU, integrated Graphics, Intel® Movidius™ Neural Compute Stick


intel AI
HANDS-ON PROBLEM SOLVING

THE AI JOURNEY – STEPS WE WILL COVER IN THIS COURSE



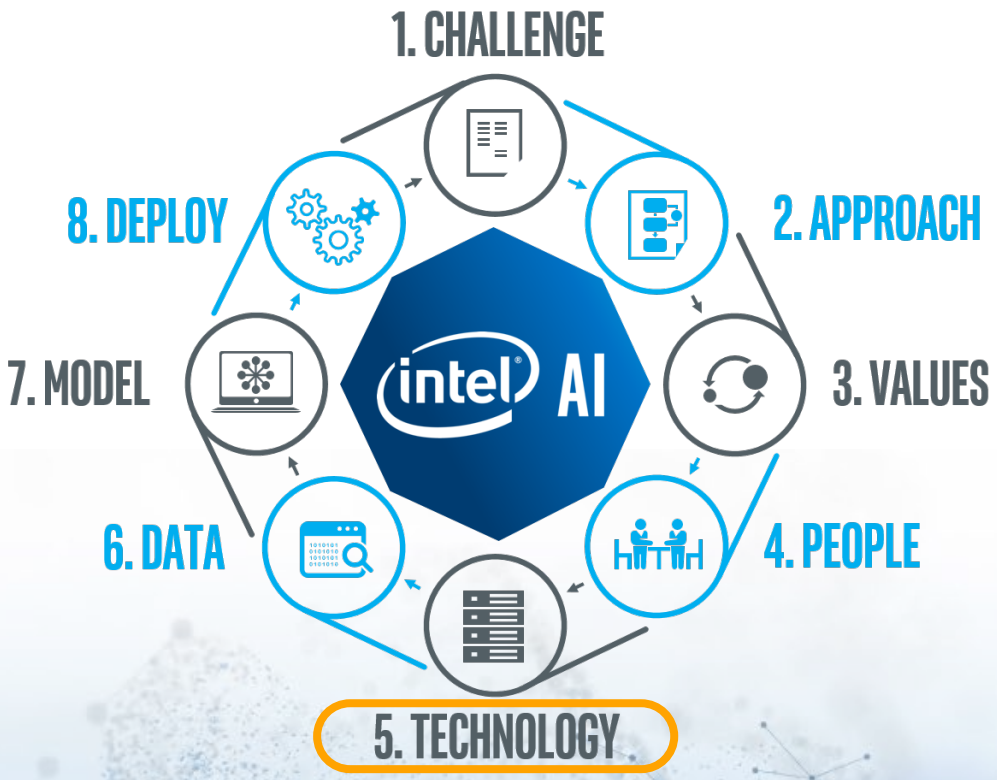
STEP 1 – THE CHALLENGE



- **Identify the challenge – Identification of most stolen cars in the US**
  - Image recognition problem
- **Application – Traffic surveillance**
  - Extensible to License Plate Detection (not included in the class)

TECHNOLOGY CHOICES

STEP 5 - COMPUTE CHOICES FOR TRAINING AND INFERENCE



- Intel® AI DevCloud
- Amazon Web Services* (AWS)
- Microsoft Azure*
- Google Compute Engine* (GCE)

INTEL® AI DEVCLOUD

INTEL® AI DEVCLOUD

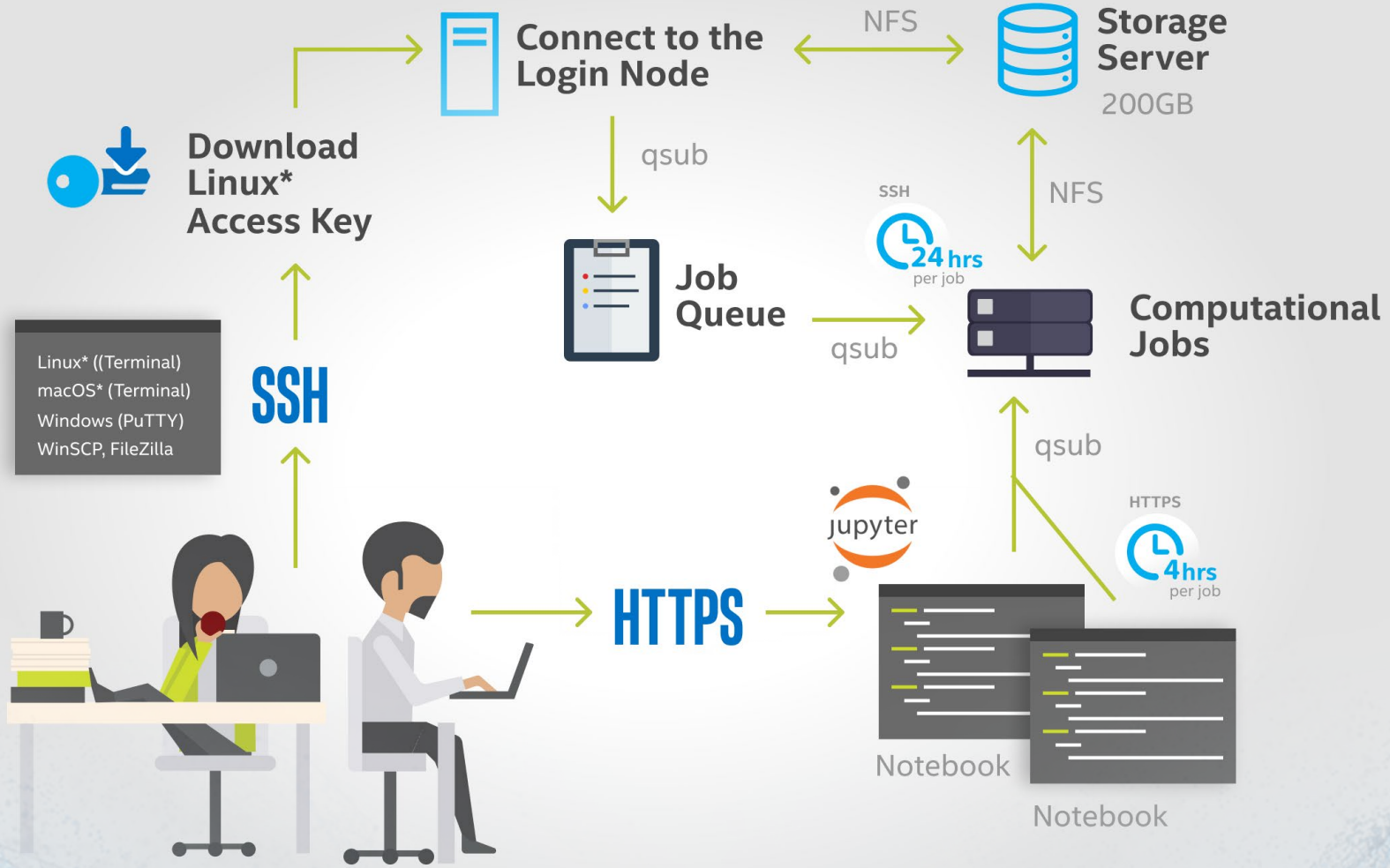
- A cloud hosted hardware and software platform available to Intel® AI Academy members to learn, sandbox and get started on Artificial Intelligence projects
- Intel® Xeon® Scalable Processors(Intel(R) Xeon(R) Gold 6128 CPU @ 3.40GHz 24 cores with 2-way hyper-threading, 96 GB of on-platform RAM (DDR4), 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**
- **Intel® Optimized Caffe***
- **Intel® Optimized TensorFlow***
- **Intel Optimized Theano***
- **Keras library**
- **More Frameworks coming as they are optimized**

- **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

DEV CLOUD OVERVIEW



OTHER CHOICES WITH INTEL PROCESSOR SUPPORT

CHOOSING YOUR CLOUD COMPUTE

Amazon Web Services* (AWS)

- Name: C5 or C5n
- vCPUs: 2 - 72
- Memory: 4gb - 144gb

Microsoft Azure* (Azure):

- Name: Fsv2
- vCPUs: 2 - 72
- Memory: 4gb - 144gb

Google Compute Engine* (GCE):

- Name: n1-highcpu
- vCPUs: 2 - 96
- Memory: 1.8gb - 86.4gb

What to look for in your compute choices:

- Better: Intel® Xeon™ Scalable Processor (code named Skylake) / **Best**: 2nd Gen Intel® Xeon™ Scalable Processor (code named Cascade Lake)
- AVX512 and VNNI Support
- Compute Intensive Instance Type per Cloud Service Provider
- Memory and vCPU are specific to your dataset

SETTING UP YOUR CLASS ENVIRONMENT ON YOUR WORKSTATION

SYSTEM CONFIGURATION

Supported hardware:

- 6th to 8th generation Intel® Core™ processors and Intel® Xeon® processors
- Intel Pentium® processor N4200/5, N3350/5, or N3450/5 with Intel® HD Graphics

Supported operating systems:

- Windows® 10 (64 bit)
- Ubuntu* 16.04.3 LTS (64 bit)
- CentOS* 7.4 (64 bit)
- Yocto Project* version Poky Jethro 2.0.3 (64 bit)
- macOS* (64 bit)

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

CREATE ANACONDA ENVIRONMENT

1. **Navigate to the root directory of the class**
2. **Run** `conda env create -f environment.yml` **to create the environment.**
3. **Now, to add this environment to the list of available environments you'll see in your Jupyter notebook by running:**

```
python -m ipykernel install --user --name intel_dc2edge --display-name "Python (intel_dc2edge)"
```

4. **Run** `source activate intel_dc2edge or conda activate intel_dc2edge` **to activate the environment.**
5. **Now run** `jupyter notebook` to start your notebook.
6. **In your notebook, select "Kernel -> Change kernel" and select "Python (intel_dc2edge)" as your kernel.**

Now you'll be able to use all the libraries you'll need to complete the exercises!

Note: if you run into any problems while creating the environment, deactivate then delete the environment and start back at step 1.

`conda deactivate` followed by `conda env remove –n inteldc2edge`

SETTING UP YOUR CLASS ENVIRONMENT ON INTEL® AI DEVCLOUD

CONNECT TO YOUR DEVCLOUD ACCOUNT

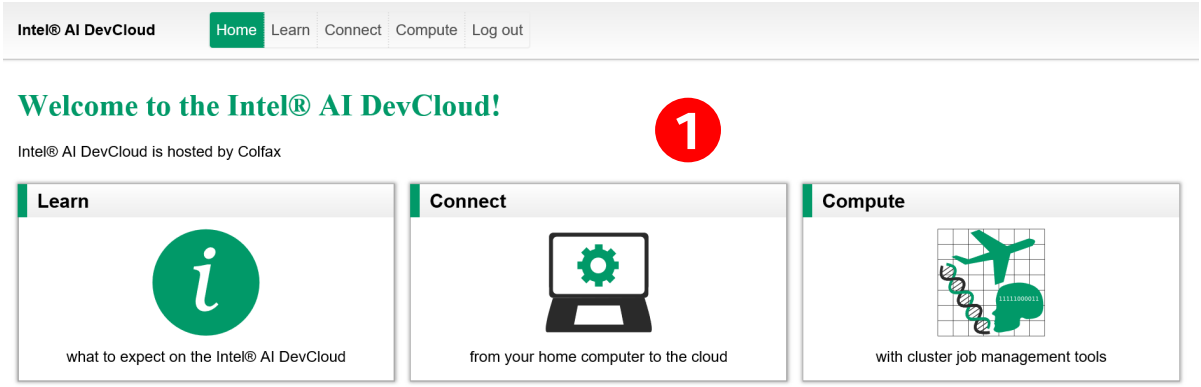
Obtain an account on Intel® AI DevCloud

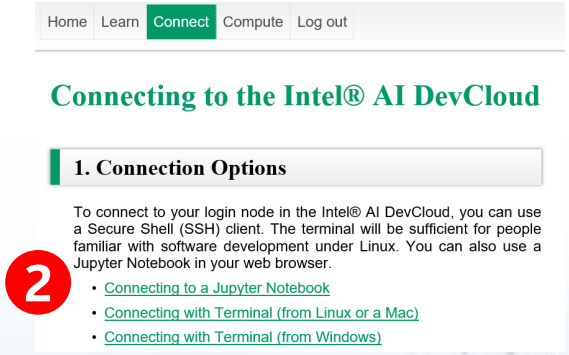
Start by connecting to the URL from the DevCloud welcome email to access your account. This will open the DevCloud home page.

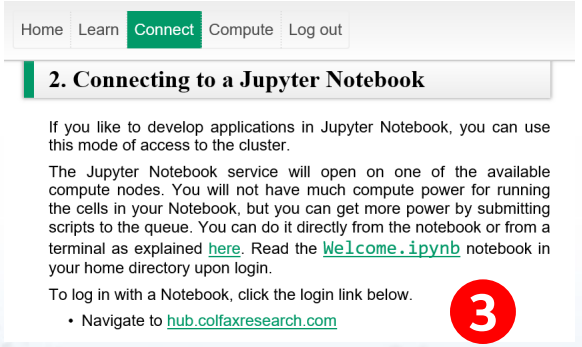
1. Click on the Connect icon to connect to your account.

2. Choose one of three connection options.
   1. Connecting to a Jupyter Notebook
   2. Connecting with Terminal (from Linux or a Mac)
   3. Connecting with Terminal (from Windows)

3. We are choosing option 1 since most of the class exercise will be done on a jupyter notebook. This will open the connect page where we get the username and password.
   - Copy the username and password before leaving this page.
   - Navigate to your jupyter notebook account by clicking on the hub.colfaxresearch.com link.





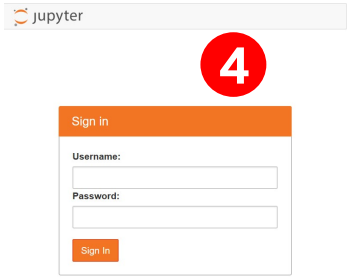


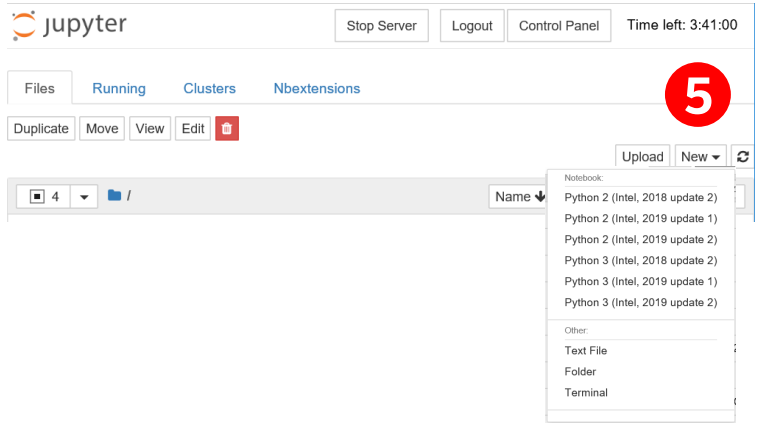
ACCESS YOUR DEVCLOUD JUPYTER NOTEBOOK ACCOUNT

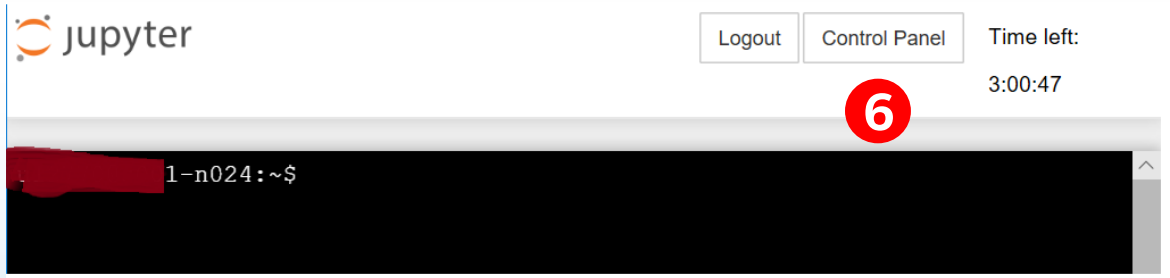
4. Enter the previously copied username and password to access your jupyter notebook account.

5. Click on the **'New'** menu on the right side of the page and select the **'Terminal'** to access the terminal

6. Now you are connected to your DevCloud account terminal via jupyter notebook. You can always return to the jupyter homepage by clicking on the **'Control Panel'** button.

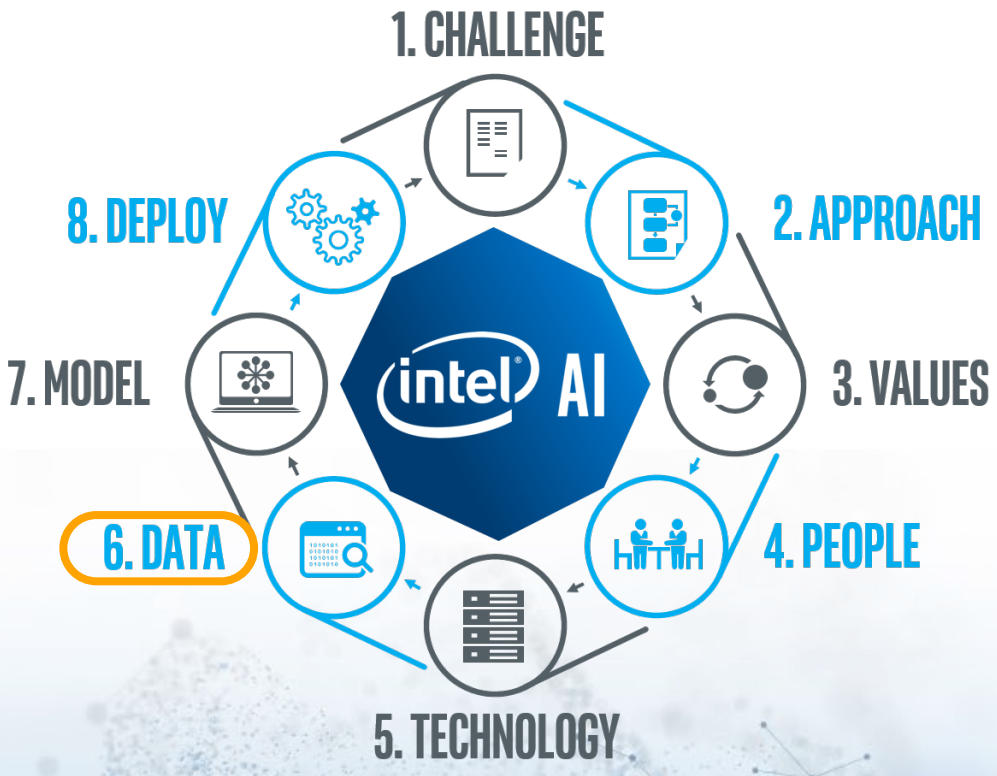






EXPLORATORY DATA ANALYSIS

STEP 6 – EXPLORATORY DATA ANALYSIS



- **Obtain a starter dataset**
- **Initial assessment of data**
- **Prepare the dataset for the problem at hand**
  - Identify relevant classes and images
  - Preprocess
  - Data augmentation

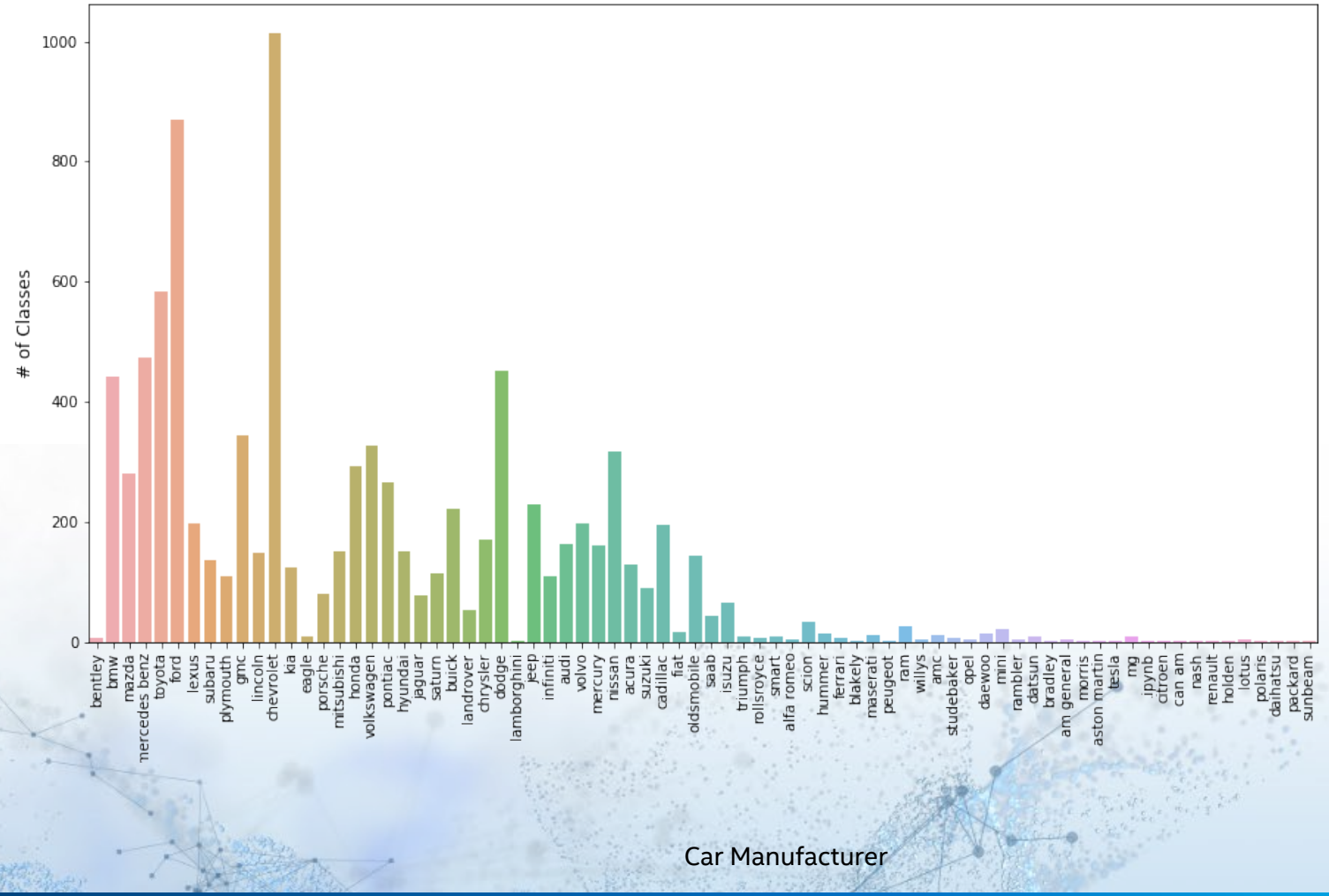
OBTAIN A STARTER DATASET

- **Look for existing datasets that are similar to or match the given problem**
  - Saves time and money
  - Leverage the work of others
  - Build upon the body of knowledge for future projects
  - We begin with the VMMRdb dataset

INITIAL ASSESSMENT OF THE DATASET

A Large and Diverse Dataset for Improved Vehicle Make and Model Recognition

- Large in scale and diversity
- Images are collected from Craigslist
- Contains 9170 classes
- Identified 76 Car Manufacturers
- 291,752 images in total
- Manufactured between1950-2016



DATASET FOR THE STOLEN CARS CHALLENGE

Hottest Wheels: The Most Stolen New And Used Cars In The U.S.

Choose the 10 classes in this problem – shortens training time

- Honda Civic (1998): 45,062
- Honda Accord (1997): 43,764
- Ford F-150 (2006): 35,105
- Chevrolet Silverado (2004): 30.056
- Toyota Camry (2017): 17,276
- Nissan Altima (2016): 13,358
- Toyota Corolla (2016): 12,337
- Dodge/Ram Pickup (2001): 12,004
- GMC Sierra (2017): 10,865
- Chevrolet Impala (2008): 9,487

# indicates number of stolen cars in each model in 2017

PREPARE DATASET FOR THE CHALLENGE

- **Map multiple year vehicles to the stolen car category (based on exterior similarity)**
- **Provides more samples to work with**

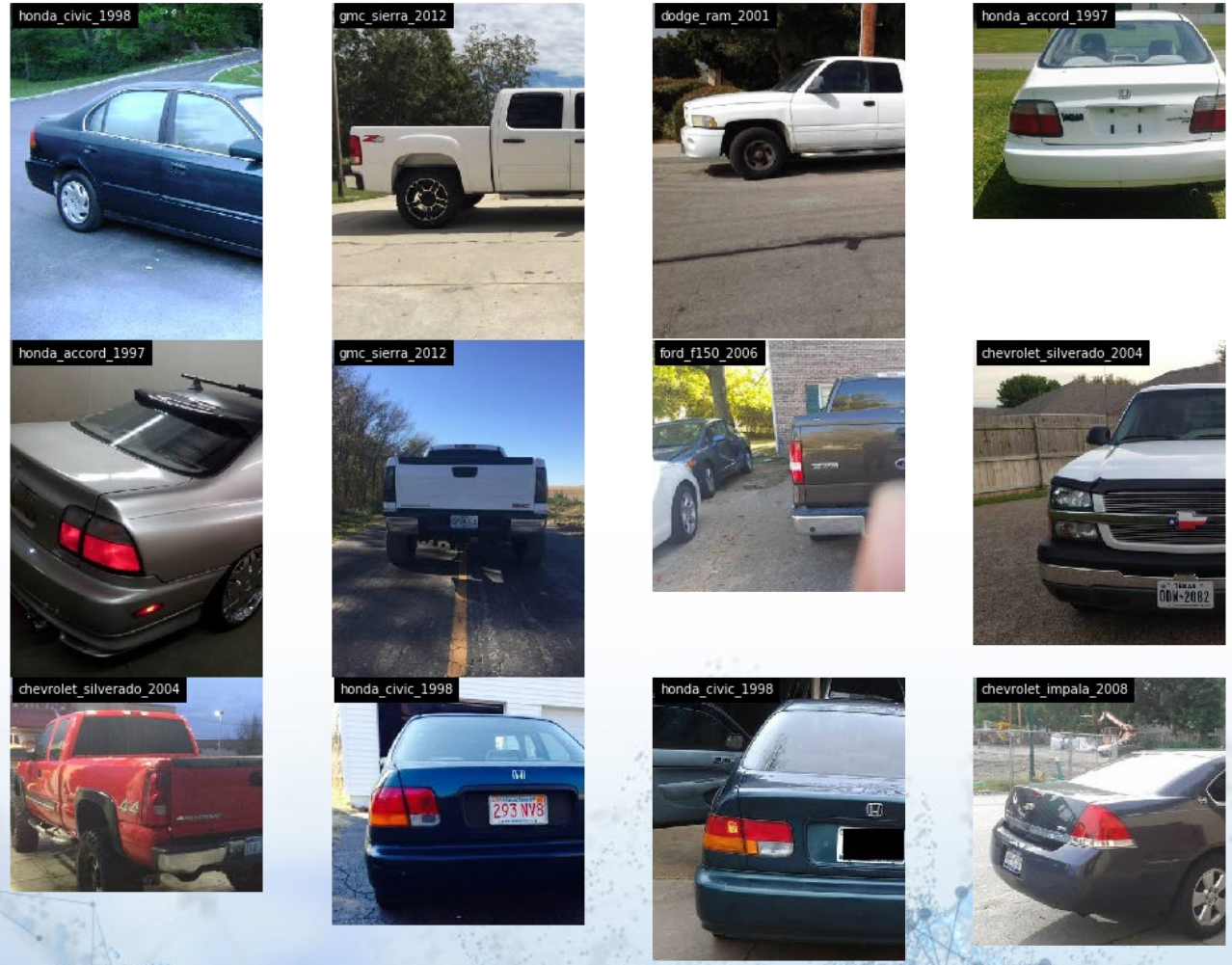
- Honda Civic (1998): 45,062	→ Honda Civic (1997 - 1998)
- Honda Accord (1997): 43,764	→ Honda Accord (1996 - 1997)
- Ford F-150 (2006): 35,105	→ Ford F150 (2005 - 2007)
- Chevrolet Silverado (2004): 30.056	→ Chevrolet Silverado (2003 - 2004)
- Toyota Camry (2017): 17,276	→Toyota Camry (2012 - 2014)
- Nissan Altima (2016): 13,358	→Nissan Altima (2013 - 2015)
- Toyota Corolla (2016): 12,337	→Toyota Corolla (2011 - 2013)
- Dodge/Ram Pickup (2001): 12,004	→Dodge Ram 1500 (1995 - 2001)
- GMC Sierra (2017): 10,865	→GMC Sierra 1500 (2007 - 2013)
- Chevrolet Impala (2008): 9,487	→Chevrolet Impala (2007 - 2009)

PREPROCESS THE DATASET

- **Fetch and visually inspect a dataset**
- **Image Preprocessing**
  - Address Imbalanced Dataset Problem
  - Organize a dataset into training, validation and testing groups
  - Augment training data
    - Limit overlap between training and testing data
    - Sufficient testing and validation datasets
- **Complete Notebook: Part1-Exploratory_Data_Analysis.ipynb**

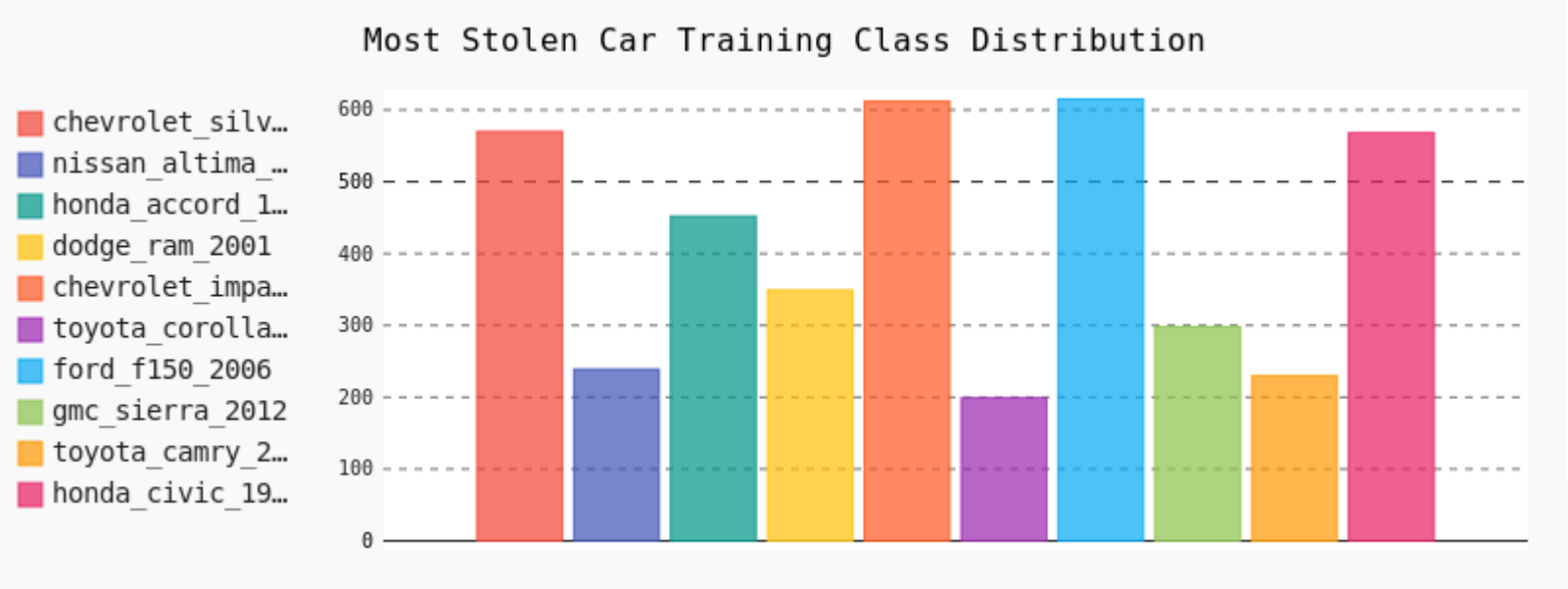
INSPECT THE DATASET

- **Visually Inspecting the Dataset**
  - Taking note of variances
    - ¾ view
    - Front view
    - Back view
    - Side View, etc.
    - Image aspect ratio differs
- **Sample Class name:**
  - Manufacturer
  - Model
  - Year



DATA CREATION

- Honda Civic (1998)
- Honda Accord (1997)
- Ford F-150 (2006)
- Chevrolet Silverado (2004)
- Toyota Camry (2014)
- Nissan Altima (2014)
- Toyota Corolla (2013)
- Dodge/Ram Pickup (2001)
- GMC Sierra (2012)
- Chevrolet Impala (2008)



PREPROCESSING & AUGMENTATION

PREPROCESSING

- **Removes inconsistencies and incompleteness in the raw data and cleans it up for model consumption**
- **Techniques:**
  - Black background
  - Rescaling, gray scaling
  - Sample wise centering, standard normalization
  - Feature wise centering, standard normalization
  - RGB → BGR

DATA AUGMENTATION

- **Improves the quantity and quality of the dataset**
- **Helpful when dataset is small or some classes have less data than others**
- **Techniques:**
  - Rotation
  - Horizontal & Vertical Shift, Flip
  - Zooming & Shearing

Learn more about the preprocessing and augmentation methods in Optional-VMMR_ImageProcessing_DataAugmentation.ipynb

PREPROCESSING

GRAY SCALING

SAMPLE WISE CENTERING

SAMPLE STD NORMALIZATION

ROTATED

RGB CHANNELS

- **Images are made of pixels**
- **Pixels are made of combinations of Red, Green, Blue, channels.**

RGB – BGR

- **Depending on the network choice RGB-BGR conversion is required.**
- **One way to achieve this task is to use Keras*** `preprocess_input`

```
>> keras.preprocessing.image.ImageDataGenerator(preprocessing_function=preprocess_input)
```

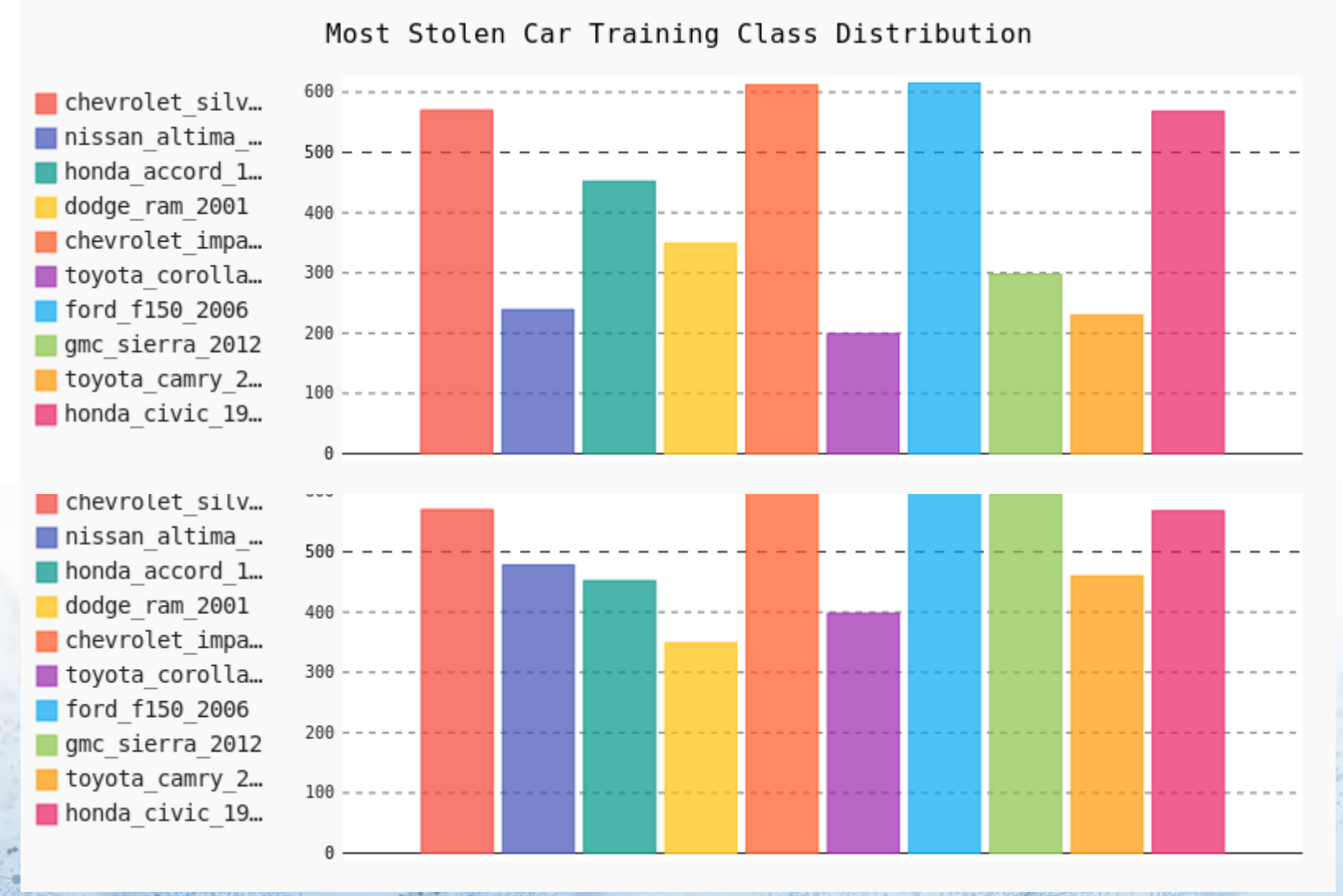
DATA AUGMENTATION

- **Oversample Minority Classes in Training**

SUMMARY

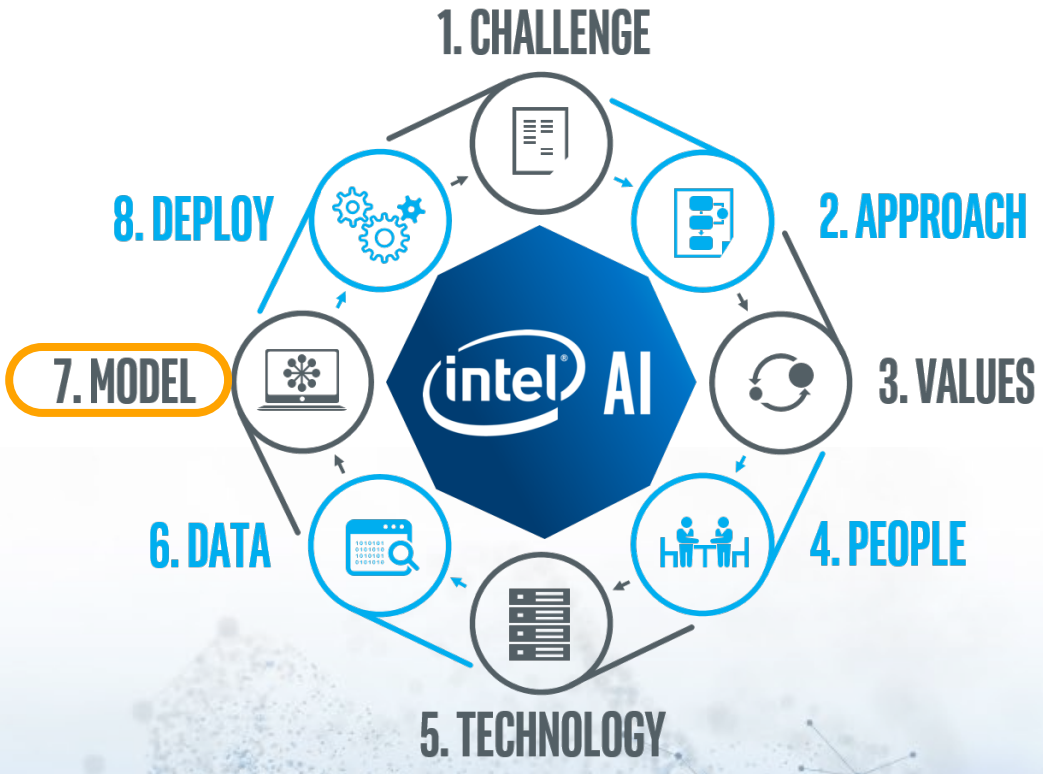
Before Preprocessing

After Preprocessing



THE TRAINING PHASE

STEP 7 – THE TRAINING/MODEL PHASE



- **Generating a trained model involves multiple steps**
  - Choose a framework (Tensorflow*, Caffe*, PyTorch)
  - Choose a network (InceptionV3, VGG16, MobileNet, ResNet etc. or custom)
  - Train the model and tune it for better performance
  - Hyper parameter tuning
  - Generate a trained model (frozen graph/ caffemodel etc.)

SELECTING A FRAMEWORK

DECISION METRICS FOR CHOOSING A FRAMEWORK

WHICH FRAMEWORKS IS INTEL OPTIMIZING?

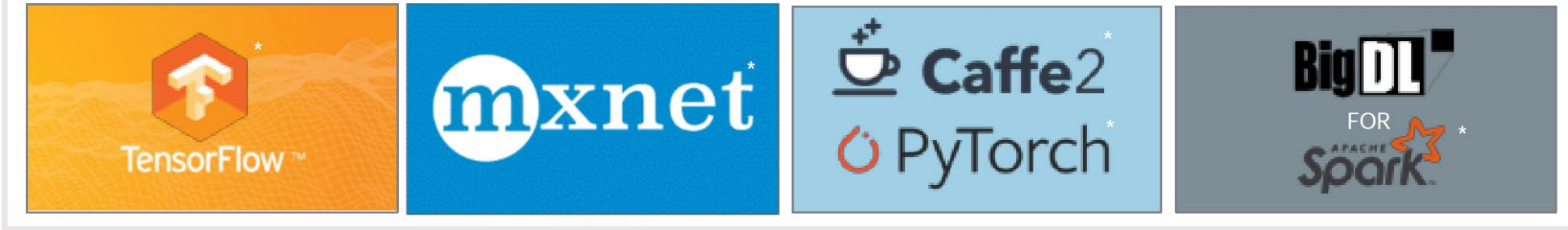
WHAT ARE THE DECISION FACTORS FOR CHOOSING A SPECIFIC FRAMEWORK?

WHY DID WE CHOOSE TENSORFLOW?

OPTIMIZED DEEP LEARNING FRAMEWORKS

INSTALL AN INTEL-OPTIMIZED FRAMEWORK AND FEATURED TOPOLOGY

FRAMEWORKS OPTIMIZED BY INTEL



More under optimization: PaddlePaddle and more.

GET STARTED TODAY AT AI.INTEL.COM/FRAMEWORK-OPTIMIZATIONS/

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.

CAFFE / TENSORFLOW / PYTORCH FRAMEWORKS

Developing Deep Neural Network models can be done faster with Machine learning frameworks/libraries. There are a plethora of choices of frameworks and the decision on which to choose is very important. Some of the criteria to consider for the choice are:

1. Opensource and Level of Adoption
2. Optimizations on CPU
3. Graph Visualization
4. Debugging
5. Library Management
6. Inference target (CPU/ Integrated Graphics/ Intel® Movidius™ Neural Compute Stick /FPGA)

Considering all these factors, we have decided to use the Google Deep Learning framework **TensorFlow**

WHY DID WE CHOOSE TENSORFLOW ?

The choice of framework was based on:

Opensource and high level of Adoption

- Supports more features, also has the 'contrib' package for the creation of more models which allows for support of more higher-level functions.

Optimizations on CPU

- TensorFlow with CPU optimizations can give up to 14x Speedup in Training and 3.2x Speedup in Inference! TensorFlow is flexible enough to support experimentation with new deep learning models/topologies and system level optimizations. Intel optimizations have been up-streamed and are part of public TensorFlow* GitHub repo.

Inference target (CPU/GPU/Movidius/FPGA)

- TensorFlow can be scaled or deployed on different types of devices ranging from CPUs, GPUs and Inferenced on devices as small as mobile phones. TensorFlow has seamless integration with CPU, GPU, TPU with no need for any explicit configuration. Support for small-scale, mobile, TF serving for server-sided deployment. TensorFlow graphs are exportable graph – pb/onnx

WHY DID WE CHOOSE TENSORFLOW ?

The choice of framework was base on ..

- **Graph Visualization**: compared to its closest rivals like Torch and Theano, TensorFlow has better computational graph visualization with Tensor Board.
- **Debugging**: TensorFlow uses its debugger called the 'tfdbg' TensorFlow Debugging, which lets you execute subparts of a graph to observe the state of the running graphs.
- **Library Management**: TensorFlow has the advantage of the consistent performance, quick updates and regular new releases with new features. This course uses Keras which will enable an easier transition to TensorFlow 2.0 for training and testing models.

SELECTING A NETWORK

HOW TO SELECT A NETWORK?

We started this project with the plan for inference on an edge device in mind as our ultimate deployment platform. To that end we always considered three things when selecting our topology or network: time to train, size, and inference speed.

- **Time to Train**: Depending on the number of layers and computation required, a network can take a significantly shorter or longer time to train. Computation time and programmer time are costly resources, so we wanted a reduced training times.

- **Size**: Since we're targeting edge devices and an Intel® Movidius™ Neural Compute Stick, we must consider the size of the network that is allowed in memory as well as supported networks.

- **Inference Speed**: Typically the deeper and larger the network, the slower the inference speed. In our use case we are working with a live video stream; we want at least 10 frames per second on inference.

- **Accuracy**: It is equally important to have an accurate model. Even though, most pretrained models have their accuracy data published, but we still need to discover how they perform on our dataset.

INCEPTION V3 - VGG16 – MOBILENET NETWORKS

We decided to train our dataset on three networks that are currently supported on our edge devices (CPU, Integrated GPU, Intel® Movidius™ Neural Compute Stick).

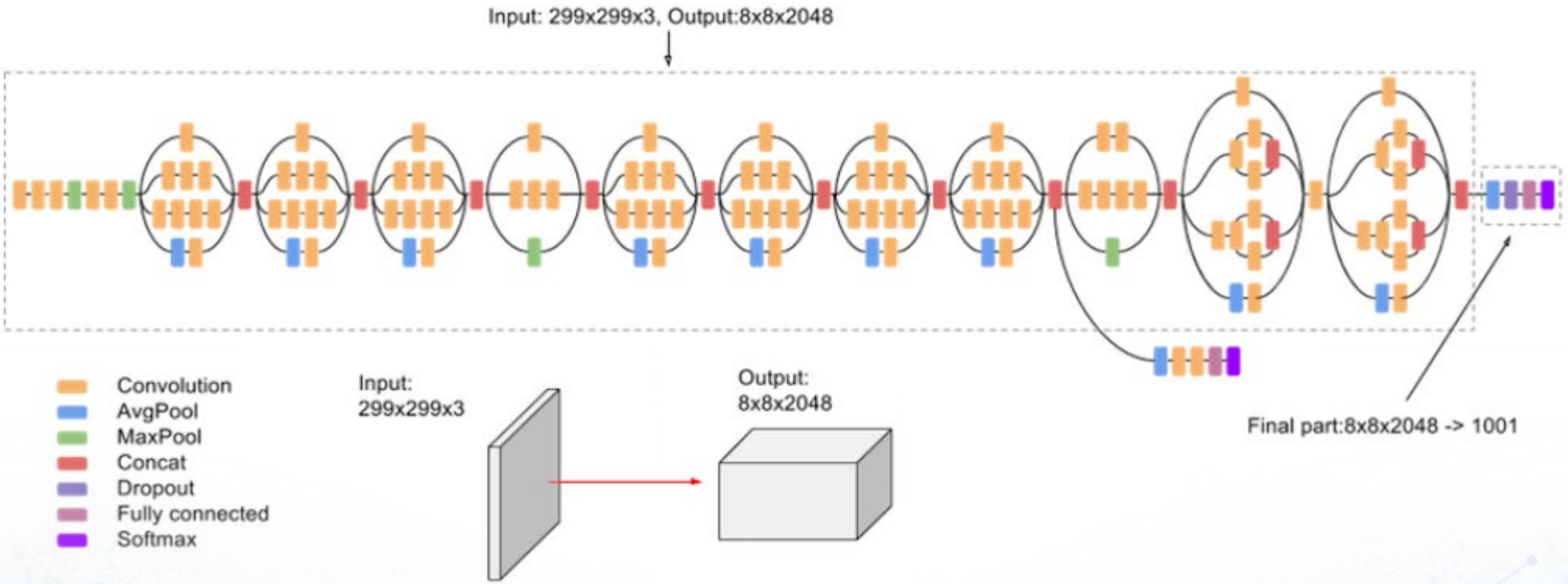
The original paper* was trained on ResNet-50. However, it is not supported currently on Intel® Movidius™ Neural Compute Stick.

The supported networks that we trained the model on:

- Inception v3
- VGG16
- MobileNet

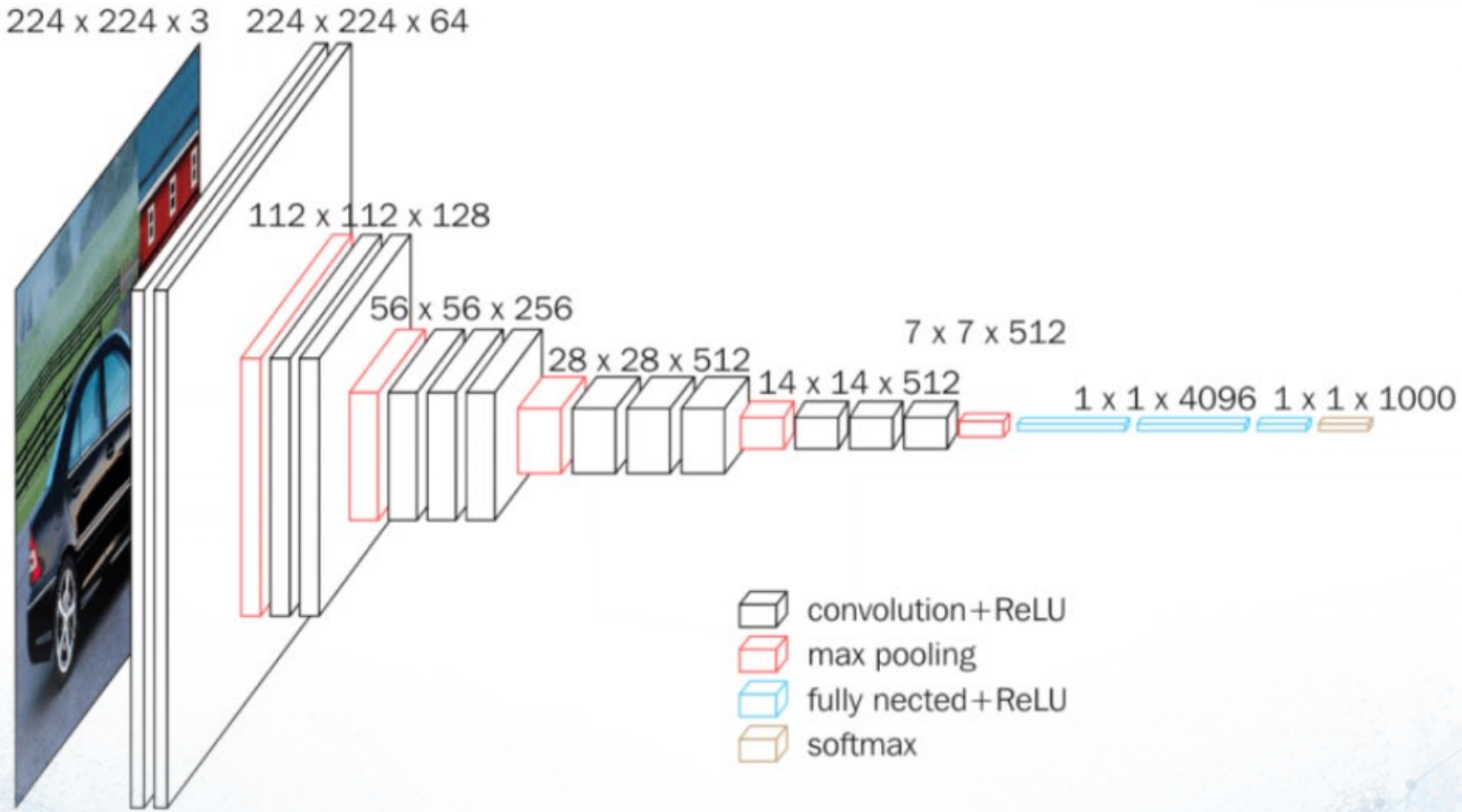
*http://vmmrdb.cecsresearch.org/papers/VMMR_TSWC.pdf

INCEPTION V3



https://arxiv.org/abs/1512.00567

VGG16



Very Deep Convolutional Networks for Large-Scale Image Recognition
Karen Simonyan and Andrew Zisserman, 2014

MOBILENET

Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

https://arxiv.org/pdf/1704.04861.pdf

INCEPTION V3 - VGG16 - MOBILENET

After training and comparing the performance and results based on the previously discussed criteria, our final choice of Network was **Inception V3**.

This choice was because, out of the three networks:

- MobileNet was the least accurate model (74%) but had the smallest size (16mb)
- VGG16 was the most accurate (89%) but the largest in size (528mb)
- InceptionV3 had median accuracy (83%) and size (92mb)

SUMMARY

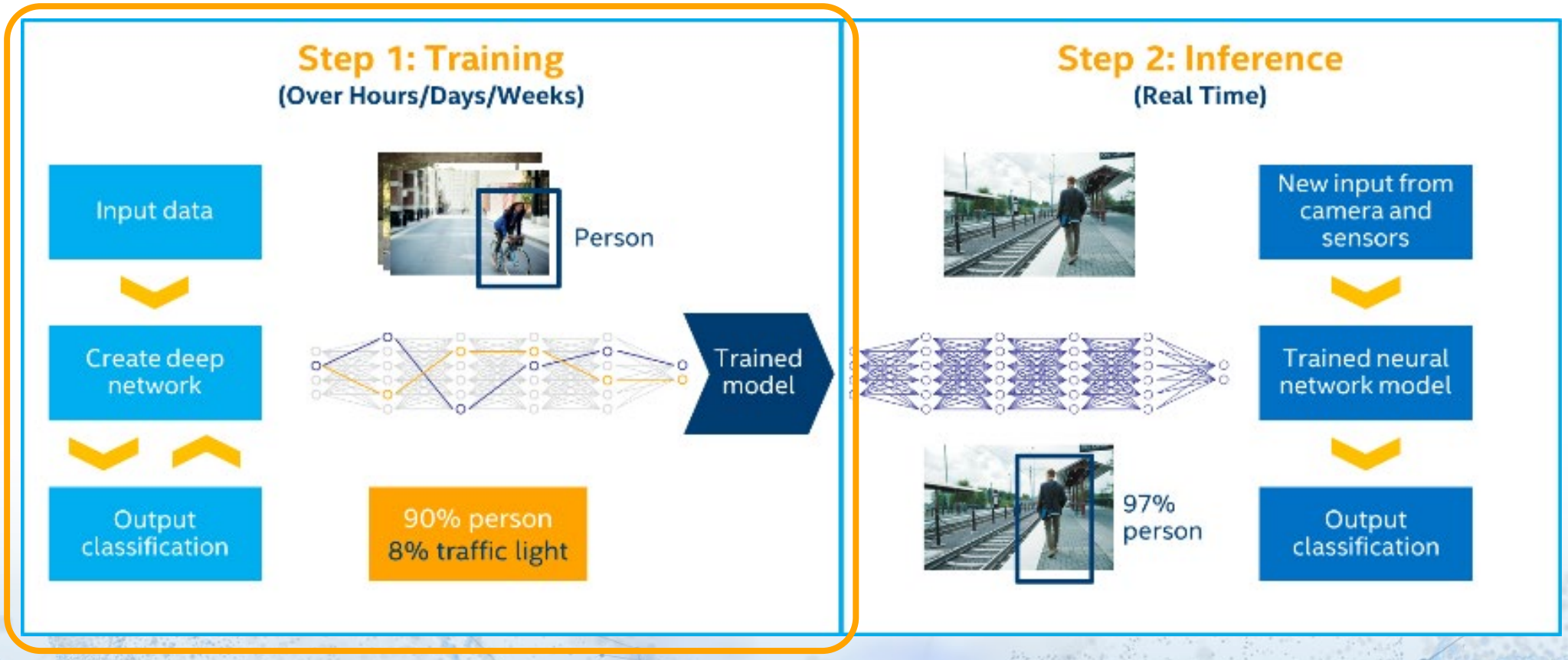
Based on your projects requirements the choice of framework and topology will differ.

- Time to train
- Size of the model
- Inference speed
- Acceptable accuracy

There is no one size fits all approach to these choices and there is trial and error to finding your optimal solution.

TRAINING
JUPYTER NOTEBOOK EXERCISE

TRAINING AND INFERENCE WORKFLOW



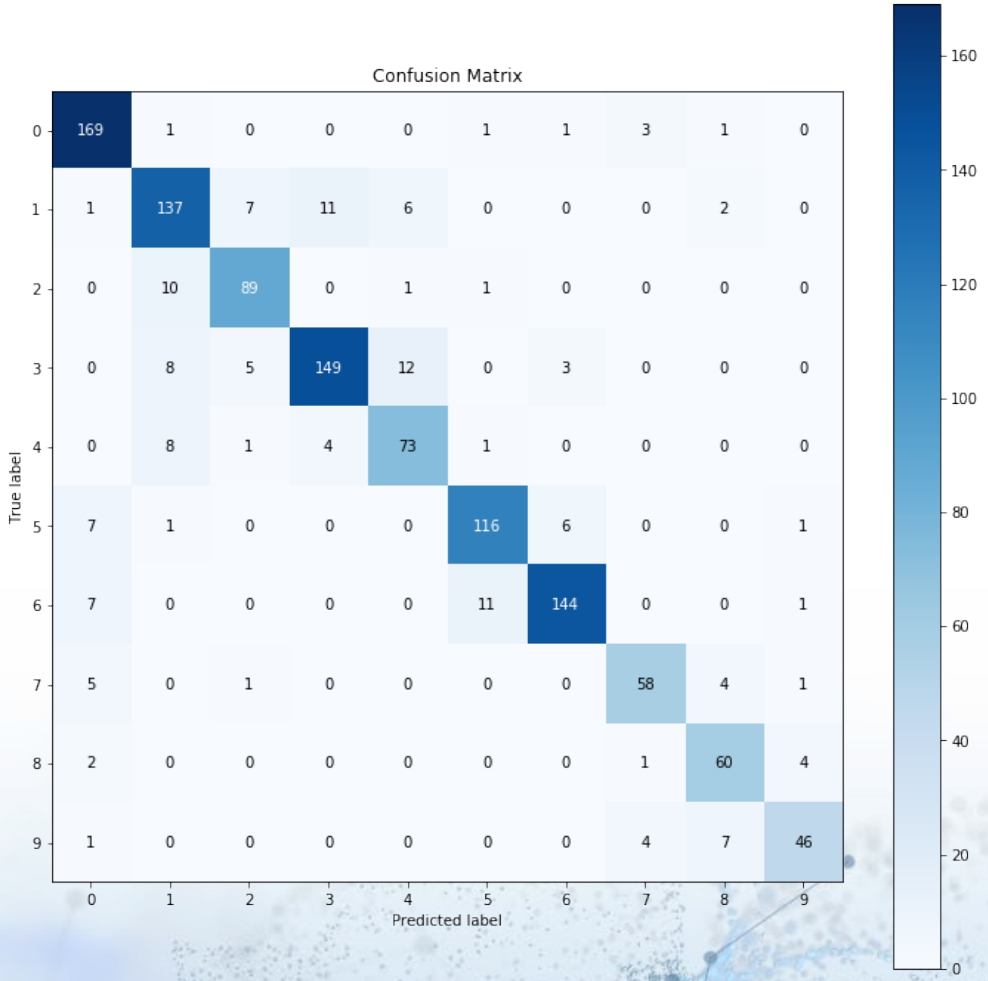
Complete Notebook : Part2-Training_InceptionV3.ipynb

(OPTIONAL) TRAINING USING VGG16 AND MOBILENET

- Try out **Optional-Training_VGG16.ipynb**
- Try out **Optional-Training_Mobilenet.ipynb**
- See how your training results differ from inceptionV3

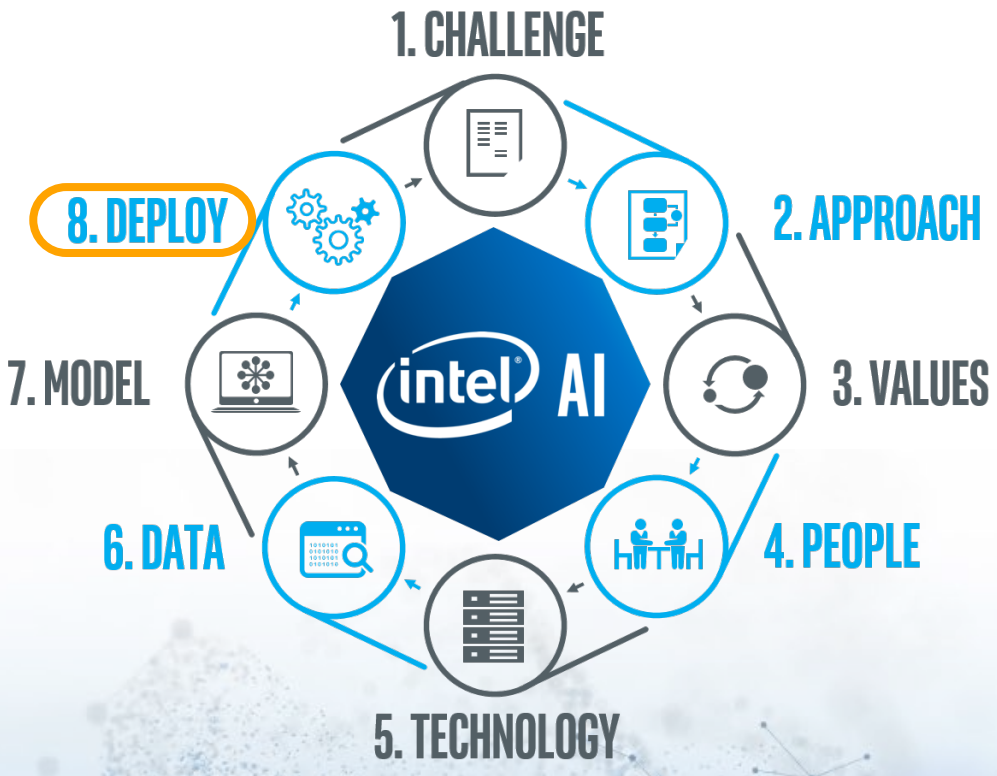
MODEL ANALYSIS

- Understand how to interpret the results of the training by analyzing our model with different metrics and graphs
  - Confusion Matrix
  - Classification Report
  - Precision-Recall Plot
  - ROC Plot
- **Complete Notebook – Part3-Model_Analysis.ipynb**



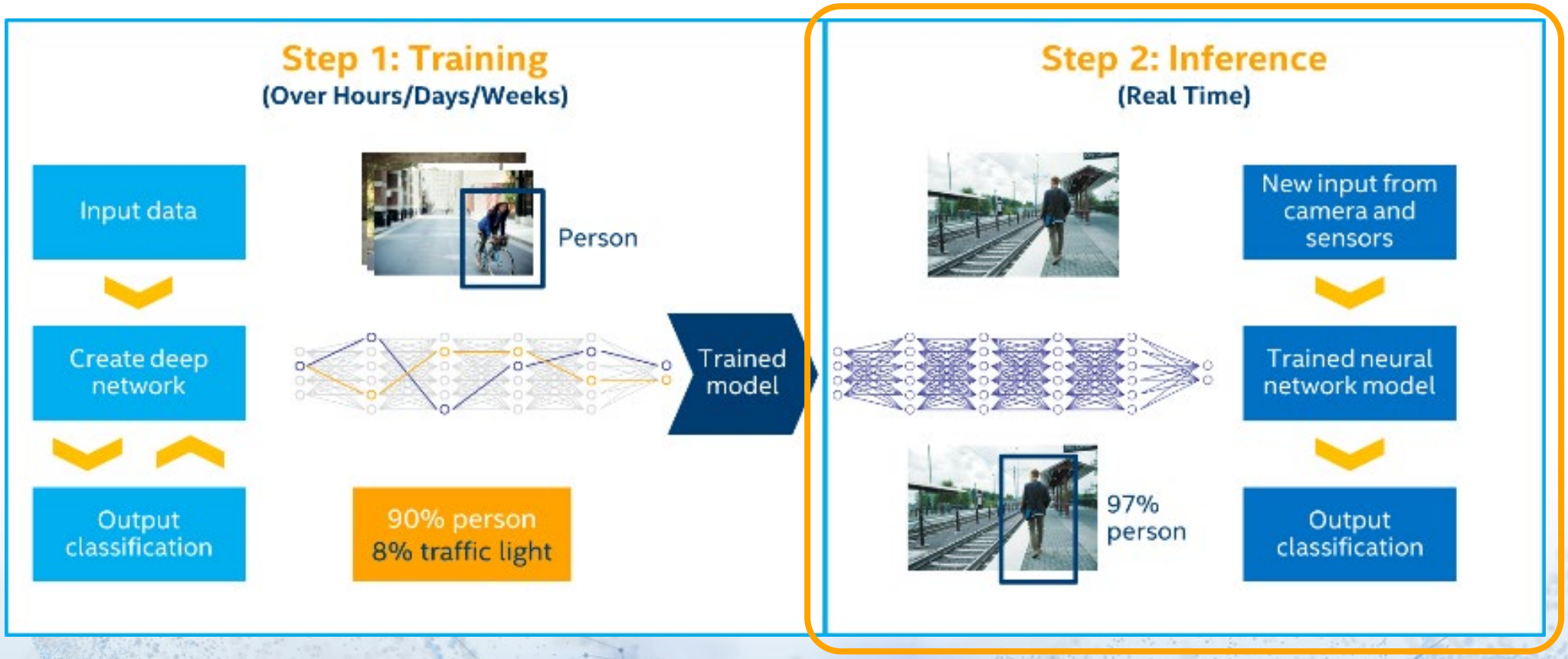
THE DEPLOYMENT PHASE

STEP 8 – THE DEPLOYMENT PHASE



- **What does deployment or inference mean?**
- **What does deploying to the edge mean?**
- **Understand the Intel® Distribution of OpenVINO™ Toolkit**
  - Learn how to deploy to CPU, Integrated Graphics, Intel® Movidius™ Neural Compute Stick

WHAT DOES DEPLOYMENT/INFERENCE MEAN?



WHAT IS INFERENCE ON THE EDGE?

Real-time evaluation of a model subject to the constraints of power, latency and memory

Requires AI models that are specially tuned to the above-mentioned constraints

Models such SqueezeNet, for example, are tuned for image inferencing on PCs and embedded devices


intel AI
INFERENCE AT THE EDGE WITH INTEL®
DISTRIBUTION OF OPENVINO™ TOOLKIT

USE CASES

PEOPLE COUNTER SOLUTION

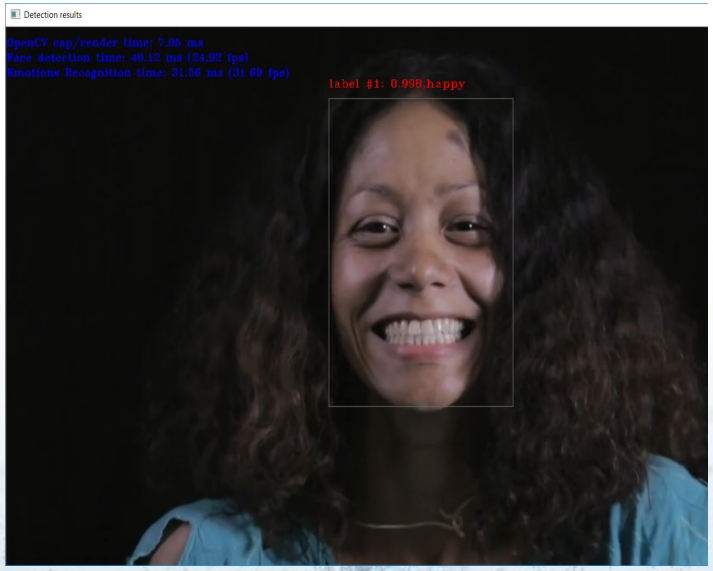
(COMES WITH THE INTEL® DISTRIBUTION OF OPENVINO™ TOOLKIT INSTALLATION)

DESCRIPTION	An application capable of counting the number of people in a given input video frame, a cumulative count of people detected so far and the duration for which a person was present on the screen. This solution can be leveraged to a people traffic monitor in retail stores. The data can be utilized by the store owners to optimize staffing, analyzing the store sections and identifying the hours that bring in maximum traffic etc. The application uses a “ResMobNet_v4 (LReLU) with single SSD head” model as its backbone
USE CASES	Store Monitoring, Video Surveillance, Traffic Monitor etc.
SOFTWARE REQUIREMENTS	OpenVINO
HARDWARE REQUIREMENTS	Intel Core System, Intel Integrated GPU, Movidius VPU
INPUT SOURCE	Video stored locally
APPLICATION CODE BASE	C++ API
USER INTERFACE	Offline video stream

MICRO EMOTION RECOGNITION SOLUTION

(COMES WITH THE INTEL® DISTRIBUTION OF OPENVINO™ TOOLKIT INSTALLATION)

DESCRIPTION	This application demonstrates how to create a micro emotion recognition solution using Intel® hardware and software tools. This solution is capable of mapping emptions to five categories - 'neutral', 'happy', 'sad', 'surprise', 'anger'. It can be leveraged to behavioral analysis solutions for the market research industry where video feeds of customer product interaction is captured and analyzed in the interest of optimizing marketing strategies. The application uses a pipeline of two models, one with a default MobileNet backbone that uses depth-wise convolutions and another that is a full convolutional network
USE CASES	Emotion recognition for interviews, Market research, Video surveillance
SOFTWARE REQUIREMENTS	OpenVINO
HARDWARE REQUIREMENTS	Intel Core System, Intel Integrated GPU, Movidius VPU
INPUT SOURCE	Video stored locally
APPLICATION CODE BASE	C++ API
USER INTERFACE	Offline video stream



PRE-TRAINED MODELS AND SAMPLES

PRE-TRAINED MODELS OPTIMIZED FOR INTEL ARCHITECTURE

OpenVINO™ toolkit includes optimized pre-trained models that can expedite development and improve deep learning inference on Intel® processors. Use these models for development and production deployment without the need to search for or to train your own models.

PRE-TRAINED MODELS

- Age & Gender
- Face Detection – standard & enhanced
- Head Position
- Human Detection – eye-level & high-angle detection
- Detect People, Vehicles & Bikes
- License Plate Detection: small & front facing
- Vehicle Metadata
- Vehicle Detection
- Retail Environment
- Pedestrian Detection
- Pedestrian & Vehicle Detection
- Person Attributes Recognition Crossroad
- Emotion Recognition
- Identify Someone from Different Videos – standard & enhanced
- Identify Roadside objects
- Advanced Roadside Identification
- Person Detection & Action Recognition
- Person Re-identification – ultra small/ultra fast
- Face Re-identification
- Landmarks Regression

SAVE TIME WITH DEEP LEARNING SAMPLES & COMPUTER VISION ALGORITHMS

SAMPLES

Use Model Optimizer & Inference Engine for both public models as well as Intel pre-trained models with these samples.

- Object Detection
- Standard & Pipelined Image Classification
- Security Barrier
- Object Detection for Single Shot Multibox Detector (SSD) using Asynch API
- Object Detection SSD
- Neural Style Transfer
- Hello Infer Classification
- Interactive Face Detection
- Image Segmentation
- Validation Application
- Multi-channel Face Detection

COMPUTER VISION ALGORITHMS

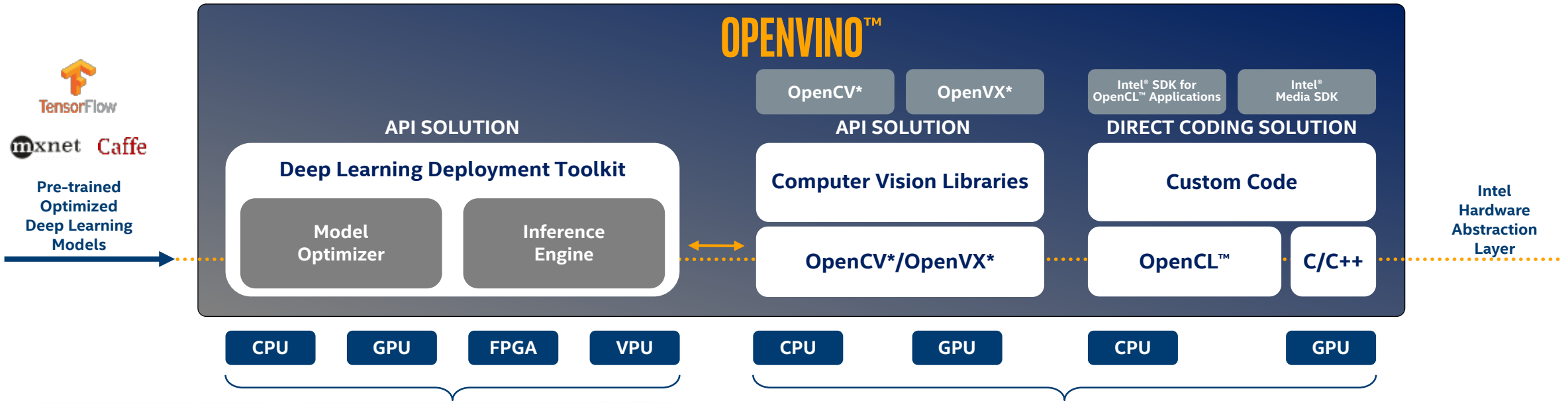
Get started quickly on your vision applications with highly-optimized, ready-to-deploy, custom built algorithms using the pre-trained models.

- Face Detector
- Age & Gender Recognizer
- Camera Tampering Detector
- Emotions Recognizer
- Person Re-identification
- Crossroad Object Detector

INTEL® OPEN VISUAL INTERFACE & NEURAL NETWORK OPTIMIZATION (OPENVINO™) TOOLKIT

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

USAGE MODEL

INTEL® DEEP LEARNING DEPLOYMENT TOOLKIT

TRAIN

Train a DL model.
Currently supports:
- Caffe*
- Mxnet*
- TensorFlow*

TensorFlow
Caffe
mxnet

.prototxt
.caffemodel

PREPARE OPTIMIZE

Model optimizer:
- Converting
- Optimizing
- Preparing to inference

(device agnostic, generic optimization)

Run Model Optimizer

IR
.xml
.bin

INFERENCE

Inference engine lightweight API to use in applications for inference.

User Application

Inference Engine

OPTIMIZE/ HETEROGENEOUS

Inference engine supports multiple devices for heterogeneous flows.

(device-level optimization)

MKL-DNN	CPU: Intel® Xeon®/Intel® Core™/Intel Atom®
cl-DNN	GPU
DLA	FPGA
Intel® Movidius™ API	Movidius Myriad™ 2/X

EXTEND

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

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

STEP 1 – TRAIN A MODEL

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

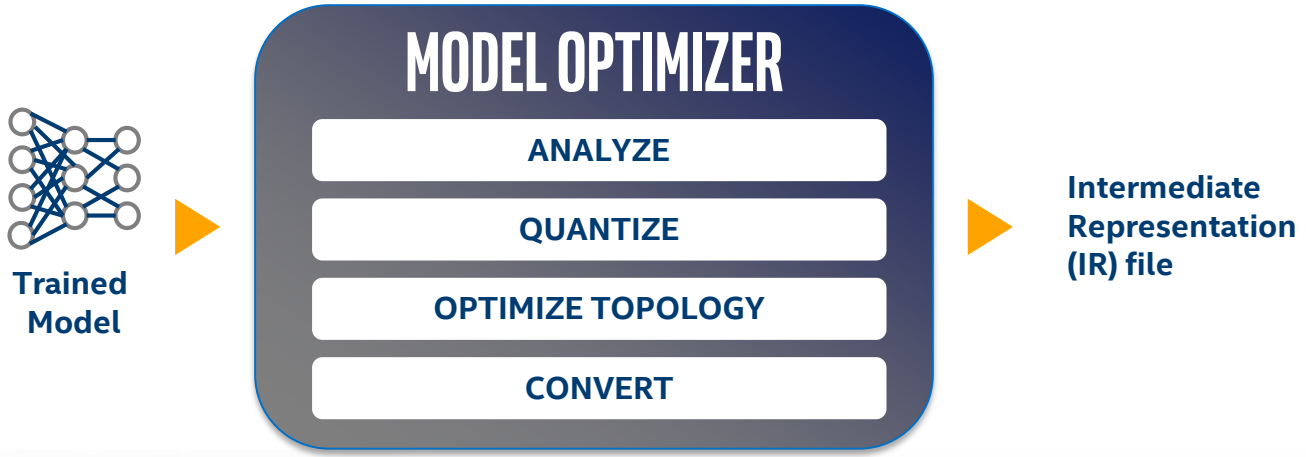
2. Use the frozen graph (.pb file) from the Stolen Cars model training as input

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

STEP 2 – MODEL OPTIMIZER (MO)

STEP 2 – MODEL OPTIMIZER (MO)

IMPROVE PERFORMANCE WITH MODEL OPTIMIZER



- Easy to use, Python*-based workflow does not require rebuilding frameworks.
- Import Models from various frameworks (Caffe*, TensorFlow*, MXNet*, more are planned…)
- More than 100 models for Caffe*, MXNet* and TensorFlow* validated.
- 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

IMPROVE PERFORMANCE WITH MODEL OPTIMIZER (CONT'D)

Model optimizer performs generic optimization:

- Node merging
- Horizontal fusion
- Batch normalization to scale shift
- Fold scale shift with convolution
- Drop unused layers (dropout)
- FP16/FP32 quantization

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

Model optimizer can cut out a portion of the network:

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

IMPROVE PERFORMANCE WITH MODEL OPTIMIZER

EXAMPLE

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

PROCESSING STANDARD LAYERS

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

PROCESSING CUSTOM LAYERS (OPTIONAL)

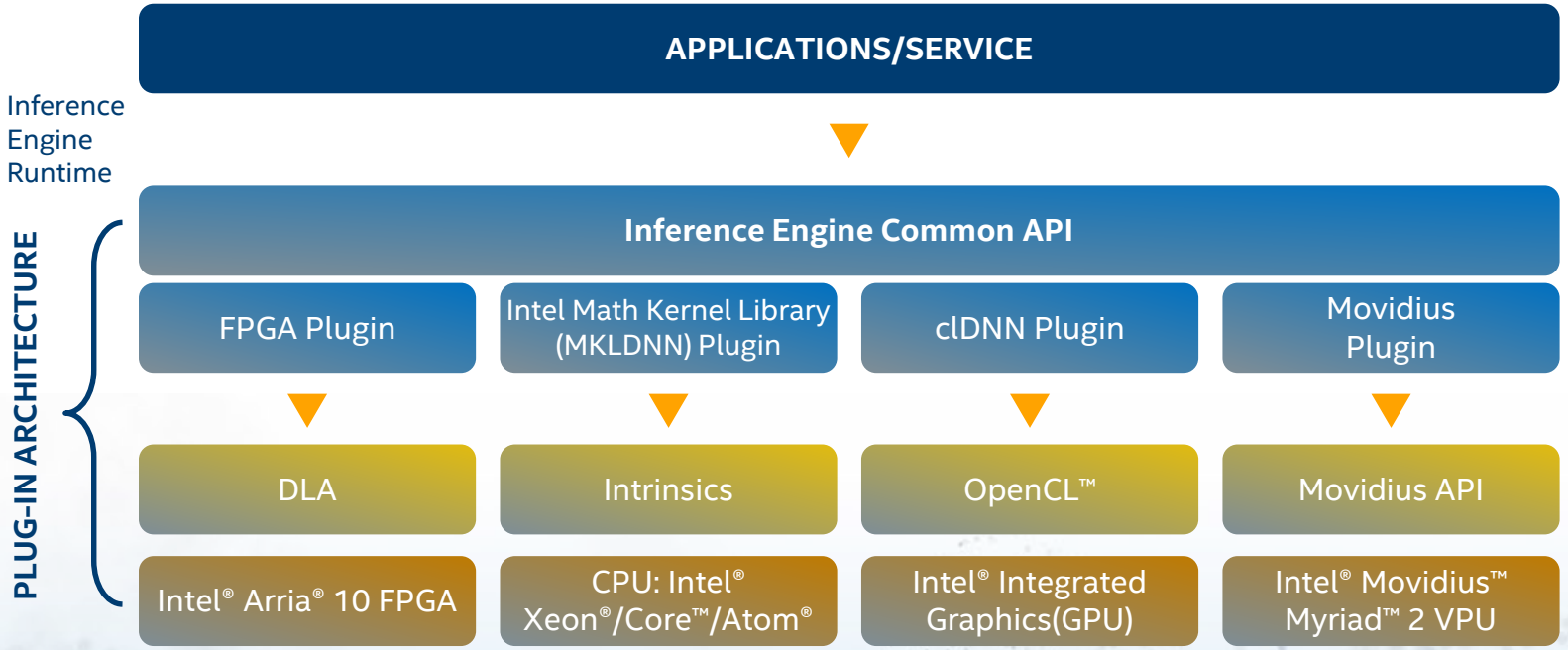
- **Custom layers are layers not included in the list of layers known to MO**
- **Register the custom layers as extensions to the Model Optimizer**
  - Is independent of availability of Caffe* on the computer
- **Register the custom layers as Custom and use the system Caffe to calculate the output shape of each Custom Layer**
  - Requires Caffe Python interface on the system
  - Requires the custom layer to be defined in the CustomLayersMapping.xml file
- **Process is similar in Tensorflow* as well**

STEP 3 AND 4 INFERENCE ENGINE AND SUPPORT FOR CPU/INTEGRATED GRAPHICS/MOVIDIUS/FPGA

OPTIMAL MODEL PERFORMANCE USING THE INFERENCE ENGINE

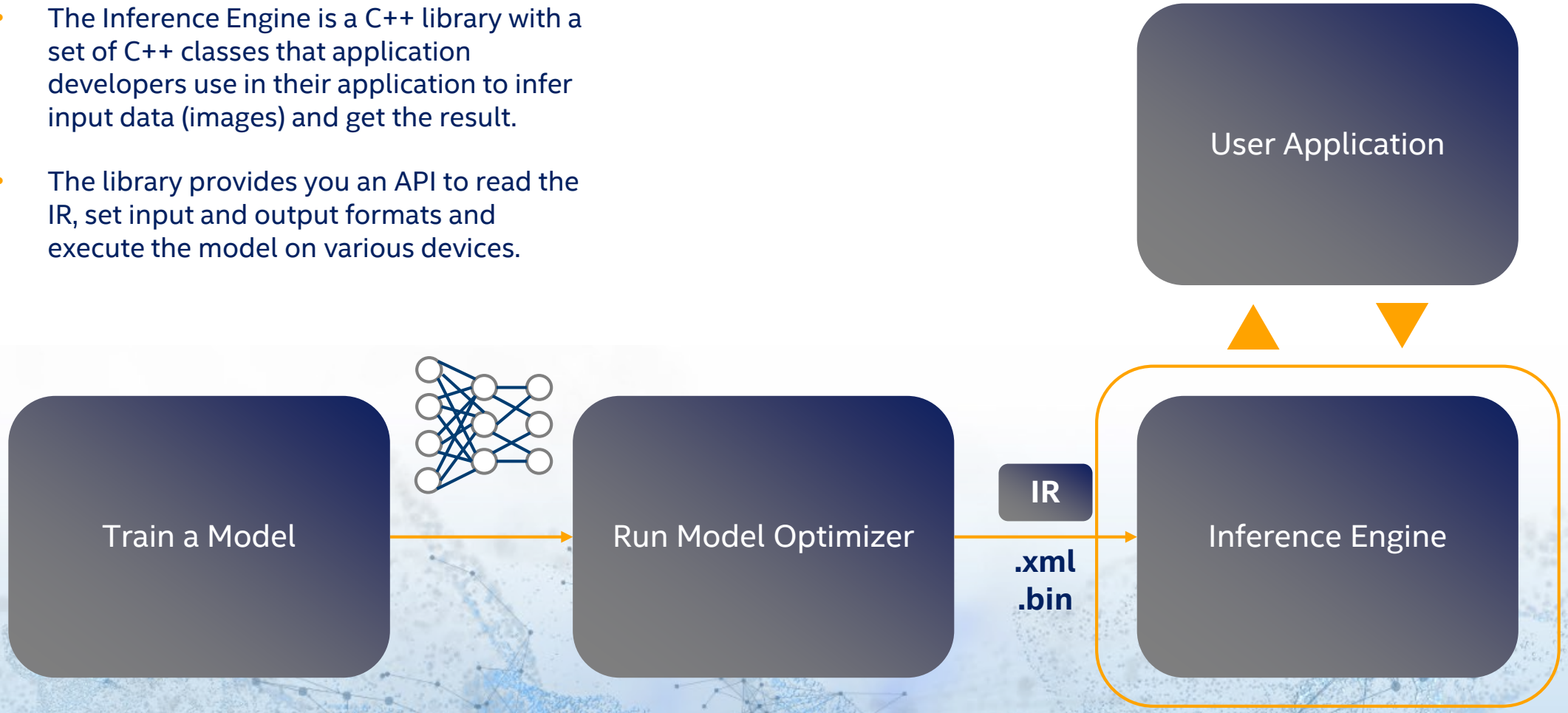
TRANSFORM MODELS & DATA INTO RESULTS & INTELLIGENCE

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



INFERENCE ENGINE

- The Inference Engine is a C++ library with a set of C++ classes that application developers use in their application to infer input data (images) and get the result.
- The library provides you an API to read the IR, set input and output formats and execute the model on various devices.



LAYERS SUPPORTED BY INFERENCE ENGINE PLUGINS

- **CPU – Intel® MKL-DNN Plugin**
  - Supports FP32, INT8 (planned)
  - Supports Intel® Xeon®/Intel® Core™/Intel 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® Arria® 10
  - FP16 data types, FP11 is coming
- **Intel® Movidius™ Neural Compute Stick– Intel® Movidius™ Myriad™ VPU Plugin**
  - Set of layers are supported on Intel® Movidius™ Myriad™ X (28 layers), non-supported layers must be inferred through other inference engine (IE) plugins . Supports FP16

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

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

INTEL® DISTRIBUTION OF OPENVINO™ TOOLKIT INSTALLATION

INSTALL THE INTEL® OPENVINO™ TOOLKIT

- Installation instructions can be found on this link: https://software.intel.com/en-us/openvino-toolkit/choose-download
- Follow the instructions for TensorFlow*
- Test out some of the samples before we begin
- Before running inference, you will need to convert the frozen graph obtained from training to Intermediate Representation using the Model Optimizer (MO)

CREATING INTERMEDIATE REPRESENTATION(IR) FILES USING MO

GENERATE OPTIMIZED INTERMEDIATE REPRESENTATION (IR) USING MO

Configure the Model Optimizer for TensorFlow*:

- Configure the Model Optimizer for the TensorFlow* framework running the configuration bash script (Linux* OS) or batch file (Windows* OS) from:

```
<INSTALL_DIR>/deployment_tools/model_optimizer/install_prerequisites folder:

    install_prerequisites_tf.sh

    install_prerequisites_tf.bat
```

GENERATE OPTIMIZED INTERMEDIATE REPRESENTATION (IR) USING MO

To convert a TensorFlow* model:

```
Go to the <INSTALL_DIR>/deployment_tools/model_optimizer directory
```

- Use the mo_tf.py script to simply convert a model with the path to the input model .pb file with the output Intermediate Representation called result.xml and result.bin that are placed in the specified ../../models/:

```
python mo_tf.py --input_model <TRAIN_DIR>/frozen_inception_v3.pb
--model_name result \
--output_dir ../../models/
```

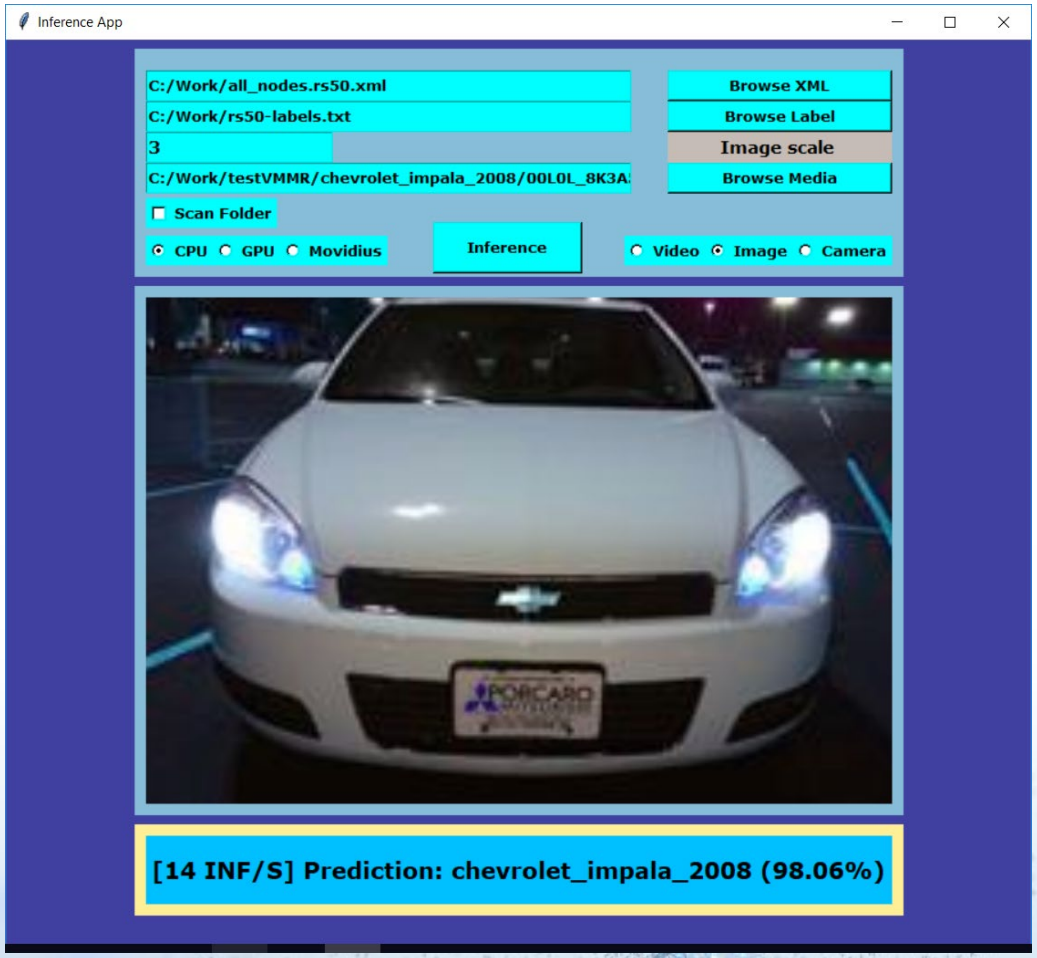
- Launching the Model Optimizer for model .pb file, with reversing channels order between RGB and BGR, specifying mean values for the input and the precision of the Intermediate Representation to be FP16:

```
python mo_tf.py --input_model <TRAIN_DIR>/frozen_inception_v3.pb \
--reverse_input_channels \
--mean_values [255,255,255] \
--data_type FP16
. . . . . .
```

RUNTIME INFERENCE

HANDS ON INFERENCE ON THE EDGE - TUTORIAL:

- **Introduction**
  - Identification of stolen cars
- **What it does**
  - The implementation instructs users on how to develop a working solution to the problem of creating a car theft classification application using Intel® hardware and software tools.



HOW IT WORKS

The app uses the pre-trained models from the earlier exercises.

The model is based on the modified Inception_V3 network that was derived from a check-point trained for ImageNet with1000 categories. For purposes of this exercises the model was modified in the last layer to only account for the 10 categories of most stolen cars.

Upon getting a frame from the OpenCV's VideoCapture, the application performs inference with the model. The results are displayed in a frame with the classification text and performance numbers.

- To execute the inference demo application, run:

```
$ python Inference_GUI.py
```

OPENVINO™ APP EXECUTION FLOW

STEPS TO INFERENCE

1. **Load plugin**

   plugin = IEPlugin(device=device_option)

2. **Read IR / Load Network**

   net = IENetwork(model=model_xml,weights=model_bin)

3. **Configure Input and Output**

   input_blob, out_blob = next(iter(net.inputs)),
   next(iter(net.outputs))

4. **Load Model**

   n, c, h, w = net.inputs[input_blob].shape

   exec_net = plugin.load(network=net)

5. **Prepare Input**

   inputs={input_blob: [cv2.resize(frame_, (w, h)).transpose((2, 0, 1))]}

6. **Infer**

   res = exec_net.infer(inputs=inputs)

   res = res[out_blob]

7. **Process Output**

   top = res[0].argsort()[-1:][::-1]

   pred_label = labels[top[0]]

RUNNING INFERENCE ON JUPYTER NOTEBOOK

- **You can also create IR files (bin/xml) by running the MO through a jupyter notebook and infer using the Inference Engine**

- **Refer to Part4-OpenVINO_Video_Inference.ipynb**
  - Set the “arg_device” parameter to “CPU”, “GPU” or “MYRIAD” to run on the CPU, integrated graphics or the Intel® Movidius™ Neural Compute Stick

ACCELERATED AI/DL INFERENCE (INT8) ON 2ND GEN INTEL® XEON™ SCALABLE PROCESSORS

2ND GENERATION INTEL® XEON® SCALABLE PROCESSOR

Drop-in compatible CPU on Intel® Xeon® Scalable platform

TCO/FLEXIBILITY

Begin your AI journey efficiently, now with even more agility…

- ✓ IMT – Intel® Infrastructure Management Technologies
- ✓ ADQ – Application Device Queues
- ✓ SST – Intel® Speed Select Technology

PERFORMANCE

Built-in Acceleration with Intel® Deep Learning Boost…

deep learning throughput!1

Throughput (img/s)

SECURITY

Hardware-Enhanced Security…

- ✓ Intel® Security Essentials
- ✓ Intel® SecL: Intel® Security Libraries for Data Center
- ✓ TDT – Intel® Threat Detection Technology

[1] Based on Intel internal testing: 1X,5.7x,14x and 30x performance improvement based on Intel® Optimization for Café ResNet-50 inference throughput performance on Intel® Xeon® Scalable Processor. See Configuration Details 3
Performance results are based on testing as of 7/11/2017(1x) ,11/8/2018 (5.7x), 2/20/2019 (14x) and 2/26/2019 (30x) and may not reflect all publically available security updates. No product can be absolutely secure. See configuration disclosure for details. ,
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 of 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. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit: http://www.intel.com/performance

INTEL® DEEP LEARNING BOOST (DL BOOST)

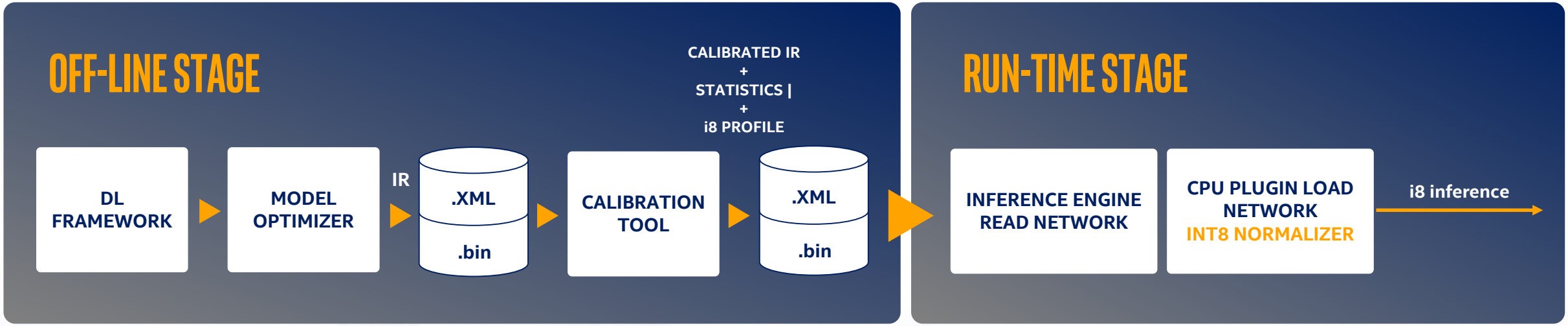
FEATURING VECTOR NEURAL NETWORK INSTRUCTIONS (VNNI)

Current AVX-512 instructions to perform INT8 convolutions: **vpmaddubsw, vpmaddwd, vpaddd**

Future AVX-512 (VNNI) instruction to accelerate INT8 convolutions: **vpdpbusd****

INT8 INFERENCE USING THE INTEL® DISTRIBUTION OF OPENVINO™ TOOLKIT

WORKFLOW



The workflow is similar to FP32, EXCEPT for the use of “Calibration Tool” for INT8.

STEPS TO CONVERT A TRAINED MODEL AND INFER

OpenVINO toolkit support for int8 model inference on Intel processors:

- Convert the model from original framework format using the Model Optimizer tool. This will output the model in Intermediate Representation (IR) format.
- **Perform model calibration using the calibration tool within the Intel Distribution of OpenVINO toolkit. It accepts the model in IR format and is framework-agnostic.**
- Use the updated model in IR format to perform inference.


intel AI
COURSE COMPLETION CERTIFICATE

COURSE COMPLETION CERTIFICATE

- You have the option to receive an Intel® AI Course Completion Certificate upon completion of the end of the course quiz.
- Before taking the quiz, you may have to disable AdBlockers. (Ghostery, uBlock, AdGuard, etc.)
- **Take the quiz**


intel AI
LEARN MORE ABOUT INTEL'S AI OFFERINGS

RESOURCES

- Intel® Distribution of OpenVINO™ Toolkit
- Reinforcement Learning Coach
- NLP Architect
- Nauta
- BigDL
- Intel Optimizations to Caffe*
- Intel Optimizations to TensorFlow*

Learn more through the AI webinar series

AI Courses:

- Introduction to AI
- Machine Learning
- Deep Learning
- Applied Deep Learning with Tensorflow*


intel®
AI