



AI ON INTEL

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

LEGAL INFORMATION

These materials are provided for educational purposes only and is being provided subject to the CC_BY_NC_ND 4.0 license which can be found at the following location:
<https://creativecommons.org/licenses/by-nc-nd/4.0/>

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

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

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.

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.

Intel, the Intel logo, Arria, Myriad, Atom, Xeon, Core, Movidius, neon, Stratix, OpenCL, Celeron, Phi, VTune, Iris, OpenVINO, Nervana, Nauta, and nGraph 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.

© 2019 Intel Corporation. All rights reserved



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
(http://vmmrdb.cecsresearch.org/papers/VMMR_TSWC.pdf).



3

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 started
 - Introduction to AI (<https://software.intel.com/en-us/ai/courses/artificial-intelligence>)
 - Machine Learning (<https://software.intel.com/en-us/ai/courses/machine-learning>)
 - Deep Learning (<https://software.intel.com/en-us/ai/courses/deep-learning>)
 - Applied Deep Learning with Tensorflow* (<https://software.intel.com/en-us/ai/courses/tensorflow>)



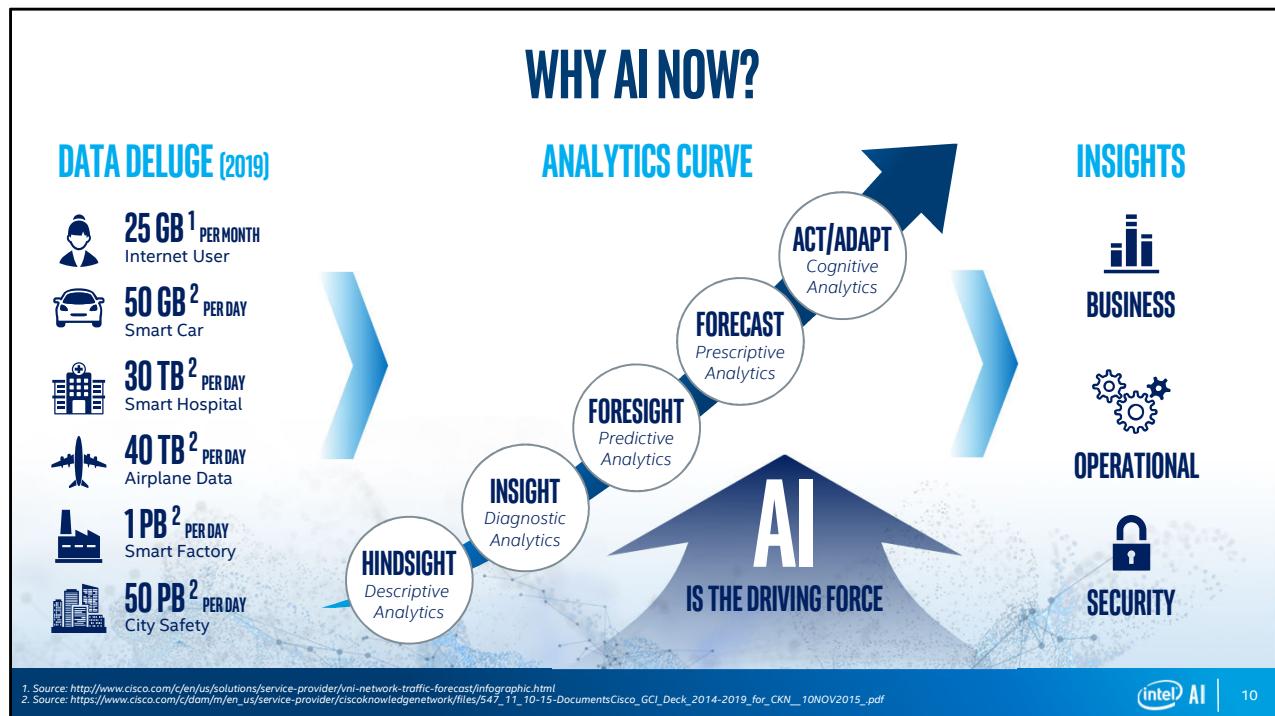
INTEL AI PORTFOLIO



© 2019 Intel Corporation

intel AI |

9



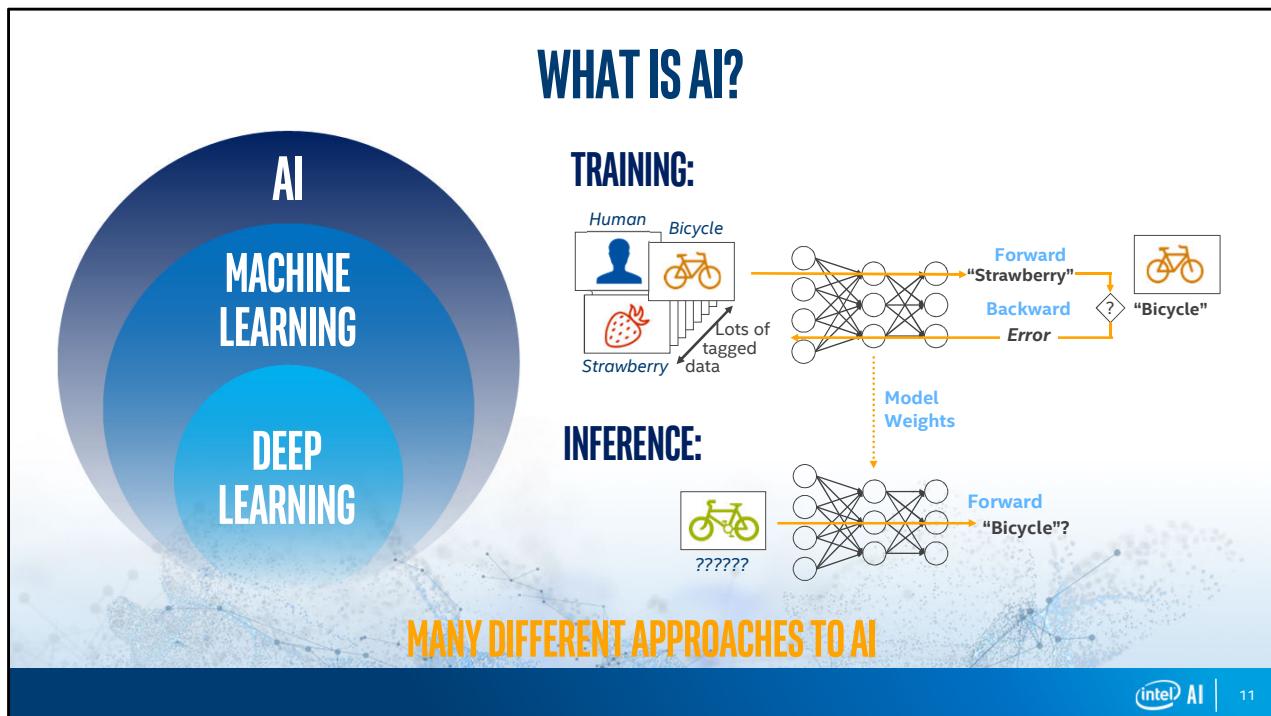
So what's driving this AI surge?

Data, for one. In 2019, the average internet user will generate ~25GB of IP traffic **per month**. By comparison, in a **single day**, a smart car will generate 2X that amount of data (50 GB), a smart hospital will generate 120X (3TB or 3,000 GB), a plane will generate 1,600X (40TB or 40,000 GB), a smart factory will generate 40,000X (1PB or 1,000,000 GB), and a city safety system will generate 800,000X (50PB or 50,000,000 GB). And we're only talking about 2019! When you consider that there will be 3X more smart connected devices than the global population by 2022, a growth from 17.7 billion networked devices in 2019 to 28.5 billion in 2022, the quantity of data generated is difficult to fathom. This data contains a treasure trove of valuable insights, in business, operations and security that we really want to extract, and in order to do that efficiently we need tools like analytics and AI by our side.

And when you think about analyzing troves of data, the first thing that may come to mind is “data analytics”, the longstanding but constantly evolving science that companies leverage for insight, innovation, and competitive advantage. Analytics has changed a lot over the years, but continues to advance through ‘more or less’ five stages of increasing scale & maturity:

- Descriptive & Diagnostic analytics, sometimes called “operational analytics”, help us understand what happened and why.
- Predictive, Prescriptive & Cognitive analytics, sometimes called “advanced analytics”, help us predict and plan for the future.

AI is its own category, applied to all phases of the analytics pipeline (especially more advanced analytics), and a vital tool for reaching higher maturity & scale data analytics. And with the recent breakthroughs in computation performance and an in-pouring of innovation into the A+AI realm, we now have the tools to extract valuable insights from troves of data.



So, what is AI, exactly? Well, we're a long way from how Artificial Intelligence (AI for short) is portrayed in science fiction movies. The definition continues to evolve, but fundamentally, **AI** is the ability of machines to learn from experience, without explicit programming, in order to perform functions typically associated with the human mind. There's no one-size fits all approach to AI, so it's helpful to explore some of the more prominent approaches to AI.

One such leading approach is **machine learning**, a category includes algorithms that improve with exposure to more data over time, and there are countless such algorithms that perform functions like regression, classification, clustering, decision trees, extrapolation, and more.

A fast-growing subset within the machine learning category is **deep learning**. This approach uses layered neural networks that learn from vast amounts of data to solve problems that are difficult to reverse engineer, such as computer vision, speech recognition, and many more. With deep learning, we avoid feature some of the 'reverse engineering' required with traditional machine learning algorithms, instead letting the neural network automatically adjust and adapt to every new piece of training data.

Now, let's explore the difference between the two key stages of deep learning: training and

inference.

In the example shown here, the job of the deep neural network is to classify a picture into one of three different categories – a person, a bicycle, or a strawberry.

First, labeled image (e.g. picture of a bicycle labeled as “bicycle”) is input into the network in a “forward pass” and the untrained network predicts that the bicycle is a strawberry, which is an error. Next, in the “backward pass”, the error propagates back through the network and the weights (i.e., the interconnections between the artificial neurons) are updated to account for the error. Once these updates have been made, the next time the same image is passed into the network, it will be more likely to predict that it’s a bicycle. Over billions upon billions of iterations like this example, you end up with a “trained” neural network that can accurately identify a given input image. When you’re satisfied with the trained accuracy of your neural network, the model weights are frozen, and the trained model can be used for inference.

Inference, in this example, is the process of feeding an unknown image into the trained neural network and allowing it to “infer” what’s in that image. If you did a good job training the model weights, it should predict “bicycle” for an image of a bicycle. Inference is really just the “forward pass” portion of the training phase, but while training is dense compute intensive and typically done in the data center, inference can take place there or even in a smart car or on a smartphone. The compute demands for inferencing really depend on the use case, and vary significantly in throughput, latency, power and size. So while you could use the same processor for inference as you do for training, it often makes sense to use a different more efficient approach.

AI WILL TRANSFORM



Which industries are the earliest adopters of AI? Generally, those segments with clear use cases, high purchasing power, and high rewards for making decisions quickly and/or more accurately will adopt AI fastest. Here are the segments that we believe will lead AI through 2020, ordered roughly by market opportunity (earliest at left).

Consumer

- Smart Assistants – personal assistant that anticipates, optimizes, automates daily life (e.g. Amazon Alexa, Apple Siri, Google Assistant, Microsoft Cortana, Facebook Jarvis home automation, X.ai virtual assistant Amy)
- Chatbots – 24/7/365 no waiting access to an informative or helpful agent (e.g. WeChat, Bank of America, Uber, Pizza Hut, Alaska Airlines, Amtrak, etc.)
- Search – ability to more intelligently search more data types including image, video, context, etc (e.g. Improved Google search, Google Photos, ReSnap)
- Personalization – ability to automatically adjust content/recommendations to suit individuals (e.g. Entefy, Netflix recommendation engine, Amazon personalized shopping recommendations)
- Augmented Reality – overlay information on our field of view in real-time to identify interesting or undesirable things (e.g. Google Translate using smartphone camera)
- Robots – personal robots that are able to perform household, yard, or other chores (e.g. Jibo robot for day-to-day functions, Roomba follow-ons)

Health

- Enhanced Diagnosis – a tool for doctors to augment their own diagnosis with more data, experience, precision and accuracy (e.g. radiology image analysis, Journal of American Medicine Association paper on retina scan for diabetic retinopathy, skin lesion classification to recognize melanoma with 98% accuracy, medical history scraping, treatment outcome prediction)
- Drug Discovery – computational drug discovery that intelligently hones in on the most promising treatments (e.g. speeding pharma drug development)
- Patient Care – machines that aid with monitoring, treatment, and/or recovery of patients (e.g. visual patient monitoring, autonomous robotic surgery, friendly medication and/or physical therapy robots)
- Research – instantly sifting through hundreds of new research papers and clinical trials that are published each day to make new connections (e.g. AI at University of North Carolina's Lineberger Comprehensive Cancer Center)
- Sensory Aids – filling in for various senses that are absent or challenged (e.g. visual aid, audio aid)

Finance

- Algorithmic Trading – augment rule-based algorithmic trading models and data sources using AI (e.g. Kensho analysis of myriad data to predict stock movement)
- Fraud Detection – ability to identify fraudulent transactions and/or claims (e.g. USAA identifies insurance fraud)
- Research – ability to intelligently assemble, parse, and extract meaning from troves of data that influence asset prices (e.g. Quid, FSI firm reducing time to insight for portfolio managers through smart knowledge management system)
- Personal Finance – smarter recommendations, lower risk lending, greater efficiency (e.g. active portfolio recommendations, quickly parsing more data before issuing loan, automatic reading of check scans, etc.)
- Risk Mitigation – detect risk factors and/or reduce the burden of regulation and minimize errors through automated compliance (e.g. IBM+Promontory Financial Group using natural language processing to detect excursions)

Retail

- Support – bots providing shopping, ordering and support in lifelike interaction (e.g. My Starbucks Barista, KLM Dutch Airline customer support via social media, Nieman Marcus visual search, Pizza Hut order pizza via bot, Adobe Digital's digital mirror that recommends clothes, intelligent phone menu routing based on NLP, ViSenze recommending similar items based on image, Adobe Digital's digital mirror that recommends clothes)
- Experience – deliver winning consumer experiences in-store (e.g. Amazon Go checkout-free grocery store, Macy's mobile shopping assistant, Lowes Lowebots that roam stores answering simple questions and tracking inventory)
- Marketing – precision marketing to consumers, promoting products and services how and where they want to hear (e.g. North Face "Expert Personal Shopper" on website)

- Merchandising – better planning through accelerated and expanded insight into consumer buying patterns (e.g. Stitch Fix virtual styling, Skechers.com analyzing clicks in real-time to bring similar catalog items forward, Wal-mart pairing products that sell together, Cosabella evolutionary website tweaks)
- Loyalty – transform the consumer experience through segmentation (e.g. Under Armour health app that constantly collects user data to deliver personalized fitness recommendations)
- Supply Chain – optimize the supply chain and inventory management for efficiency and innovate new business models (e.g. OnProcess technology's use of predictive analytics for inventory management)
- Security – improve security of all consumer and business digital assets, such as real-time shoplifting/lifter detection, multi-factor identity verification, data breach detection (e.g. Mastercard pay with your face, Walmart facial recognition to catch shoplifters)

Government

- Defense – drones, connected soldiers, defense strategy (e.g. military/surveillance drones, autonomous rescue vehicles, augmented connected soldier, real-time threat assessment and strategy recommendation)
- Data Insights – analyze massive amounts of data to identify opportunities/inefficiencies in bureaucracy, cybersecurity threats and more, to ultimately implement better systems and policies (e.g. MIT AI that detects cyber security threats)
- Crime Prevention using AI to predict and help recover from disasters thanks to ability to quickly process large amounts of unstructured data and optimize limited resources (e.g. 1Concern, BlueLineGrid)
- Safety & Security – crowd analytics, behavioral/sentiment analytics, social media analytics, face/vehicle recognition, online identity recognition, real-time video analytics, using AI to predict and help recover from disasters thanks to ability to quickly process large amounts of unstructured data and optimize limited resources (e.g. police analyzing social media to adjust police presence, license plate readers in police cars, 1Concern, BlueLineGrid)
- Resident Engagement – new tools to facilitate citizen engagement like chatbots, at-risk citizen identification, (e.g. Amelia chatbot in North London Enfield council, North Carolina chatbot to help state employees with IT inquiries)
- Smarter Cities – traffic/pedestrian management, lighting management, weather management, energy conservation, services analytics (e.g. San Francisco and Pittsburgh using sensors and AI to optimize traffic flow)

Energy

- Oil & Gas Exploration – automated geophysical feature detection (e.g. oil & gas producers using AI to augment traditional modeling & simulation)
- Smart Grid – predictive and real-time intelligent generation, allocation, and storage of power to meet variable demand (e.g. GridSense, SoloGrid)
- Operational Improvement – safety and efficiency improvements through predictive and/or insightful AI (e.g. GE Oil and Gas using predictive analytics and AI to predict and preempt potential operational problems)

- Conservation – intelligent buildings, computing and appliances that reduce power consumption and are more efficient than producing another kWh of electricity (e.g. Google DeepMind datacenter energy reductions)

Transport

- Automated Cars – autonomous cars driving on the roadways (e.g. BMW, Google, Uber, many others)
- Automated Trucking – autonomous trucks driving on the roadways (e.g. Daimler)
- Aerospace – autonomous planes and other aerial vehicles (e.g. Boeing's evolution of autopilot and drones)
- Shipping – autonomous package delivery via drone or other vehicle (e.g. Amazon package delivery drone)
- Search & Rescue – ability to deploy autonomous robot to search and rescue victims in potentially hazardous environments (e.g. war casualty extraction, miner rescue, firefighting, avalanche rescue)

Industrial

- Factory Automation – highly-productive, efficient and safe factories with robots that can see, hear and adapt to their environment to produce goods with incredible quality and speed (e.g. assembly line)
- Predictive Maintenance – ability to detect patterns that indicate the likelihood of an upcoming fault that would require maintenance (e.g. airline being able to adjust schedule to perform preventive maintenance before a failure)
- Precision Agriculture – ability to deliver the precise amount of water, nutrients, sunlight, weed killer, etc to a particular crop or individual plant (e.g. farmer using visual weed search to zap only weeds with RoundUp, automated sorting of produce for market)
- Field Automation – ability to automate heavy equipment beyond the factory walls (e.g. mining, excavation, construction, road repair)

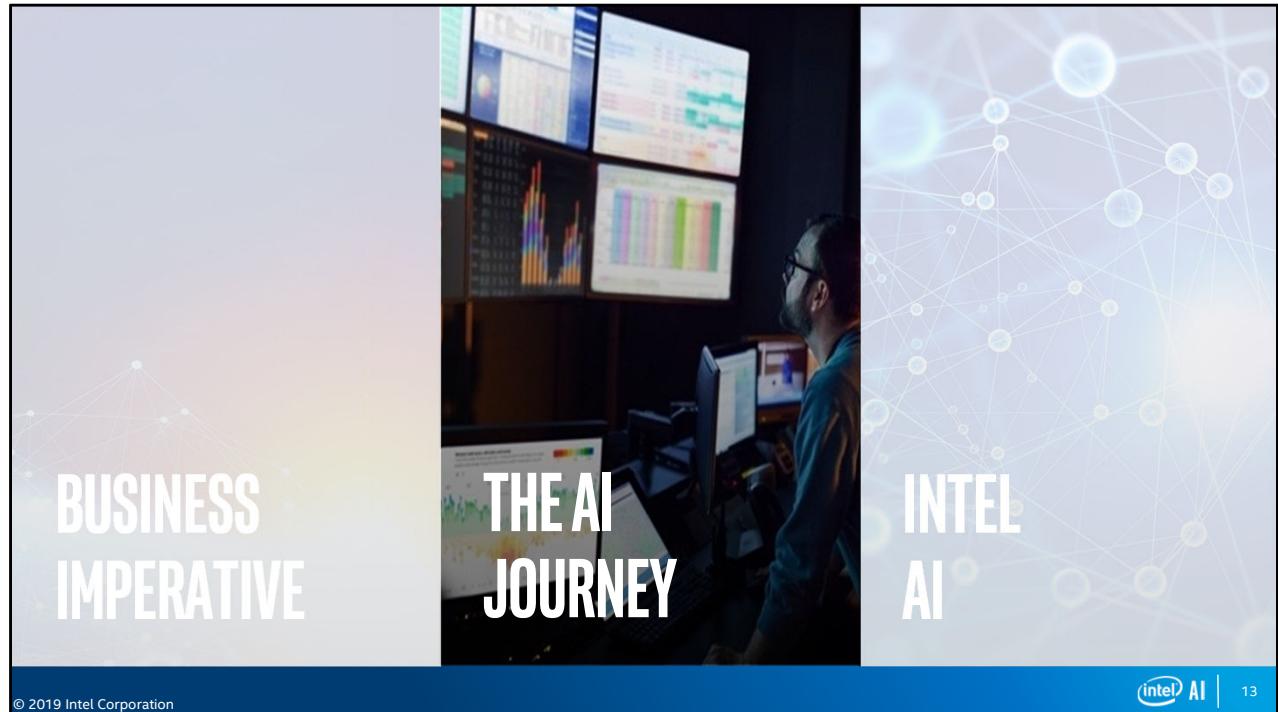
Other

- Advertising – interactive ads, adaptive ads, personalized ads, real-time ads (e.g. AdBrain, MetaMarkets, Proxicom, RocketFuel)
- Education – virtual mentors, foreign language instruction, automated study sheets, personalized assignments, cheating detection, deliberate practice, machine-to-machine instruction (e.g. Intelligent Tutor Systems, Content Technologies Inc, PR2 robot from Cornell)
- Gaming – dynamic and interactive video game experiences (e.g. Xbox Kinect, Playstation Eye, Wii)
- Professional & IT Services – sales, marketing, legal research, accounting/tax, assisted counseling, customized IT recommendations (e.g. Pinsent Masons law firm that emulates human decision-making, Salesforce use of AI)
- Telco/Media – customized content/ads, network optimization, quality of service, mobile/home security (e.g. media company customizing tv show recommendations and ads, network operator ensuring efficient and high-quality delivery/repair, wireless

- company using multi-factor security)
- Sports – intelligent analytics for injury prevention and betting (e.g. Kinduct injury prevention, Microsoft Cortana predicting football games)

Here is an even broader list of industries that will be impacted by AI: Advertising, Aerospace, Agriculture, Automotive, Building Automation, Business, Education, Fashion, Finance, Gaming, Government, Healthcare, IT, Investment, Legal, Life Sciences, Logistics, Manufacturing, Media & Entertainment, Oil/Gas/Mining, Real Estate, Retail, Sports & Fitness, Telecommunications, Transportation

Sources: Intel forecast (IDC, GII Research, Tractica, Technavio, Market Research Store, Allied Market Research, BCC Research)

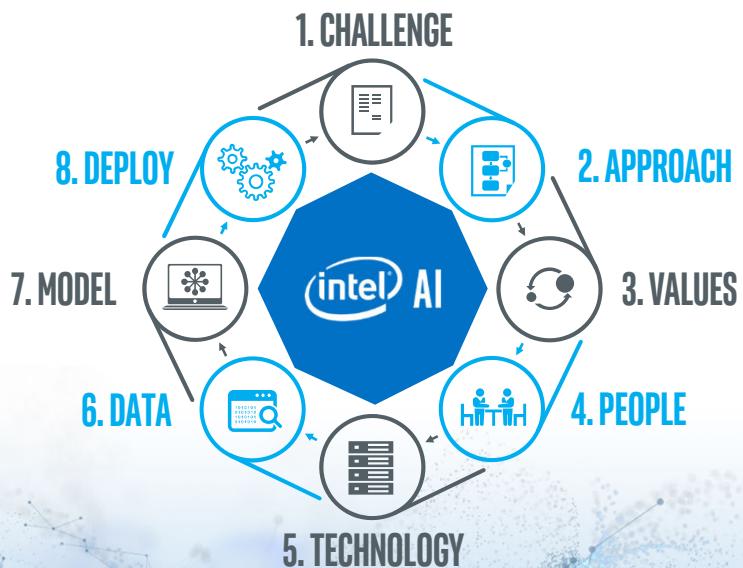


© 2019 Intel Corporation

intel AI | 13

THE AI JOURNEY

Partner with Intel to accelerate your AI journey



intel AI | 14

Before we venture any further, it's important to understand that implementing AI in your organization will be a journey, and to think about which technology partner can help you accelerate each step to take you full circle.



© 2019 Intel Corporation

intel AI | 15

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



SIMPLIFY AI

via our robust community



COMMUNITY

SOFTWARE

HARDWARE

DEPLOY AI ANYWHERE

with unprecedented HW choice



SCALE WITH CONFIDENCE

on the platform for IT & cloud



www.intel.ai



16

The Intel AI commitment to our customers is simple: we're here to help them break barriers between AI theory and reality. With Intel AI, our customers can simplify AI, choose any approach, tame their data deluge, speed up development, deploy AI anywhere, and scale with confidence. There's no other company on the planet that brings these unique capabilities together to accelerate AI from start to finish.

At the heart of these capabilities, are three things:

First is our AI **hardware** portfolio. Intel brings unprecedented AI hardware choice, from Intel® Xeon® Scalable processors optimized for faster deep learning, to Intel® Movidius™ VPUs for leading on-device computer vision inference, to Intel® FPGAs for real-time inference, to forthcoming ASICs built from the ground up to accelerate deep learning, and more. Through these compute engines and our innovative memory, storage & connectivity technologies, we're helping our customers move, store and process data for AI.

Second is our AI **software** stack, which is inextricable from hardware. From research to deployment, Intel is contributing open software & tools to speed up AI, including optimizations for the most popular open-source deep learning frameworks and topologies to get the most out of our hardware.

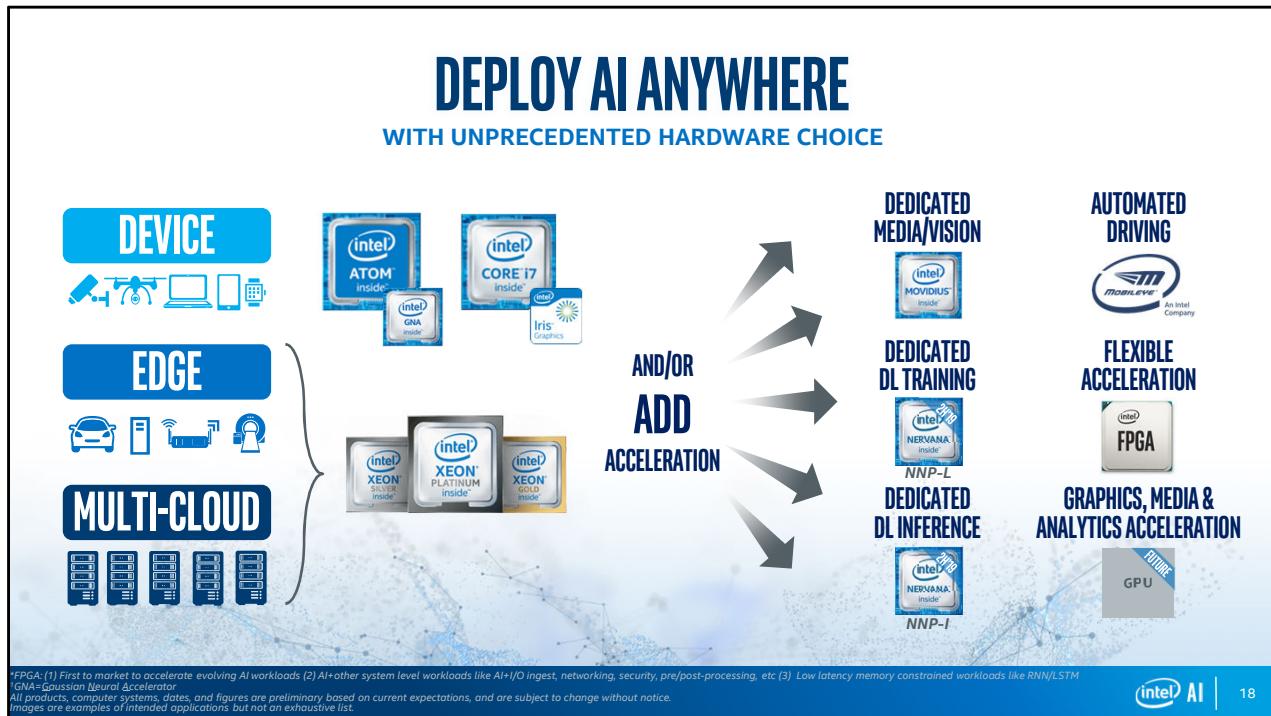
Third is our AI **community** which brings it all together. Intel's community helps

customers truly move from data strategy to enterprise-scale AI deployment, through AI direct engagements, market-ready solutions, reference designs, and many more offerings across all industries.

For more information about all these and more, visit www.intel.ai



HARDWARE



With Intel AI, you can deploy AI anywhere with unprecedented hardware choice.

As you can infer (pun intended), each AI use case has very different requirements in terms of compute, power, size/form factor, latency, cost, resilience, etc. It's helpful to break these requirements down into a few buckets:

- On the top-left is the **device** category, where end point uses with lower power interactive technology reside such as personal computers, cameras, smart speakers, drones, robots and more. For this category, the Intel® Atom™ and Intel® Core™ processors are frequently the host processor, but we're seeing a growing demand in this space for domain specific inference SOC's tailored to individual applications
 - For internet of things sensors (IOT) in security, home, retail, industrial and many more verticals, high performance with very low power is crucial. For vision & inference applications in drones and cameras for example, the Intel® Movidius™ Vision Processing Units (VPU) deliver high quality image recognition in a <1 watt power envelope. You can experience this platform through the Intel® Movidius™ Neural Compute Stick NCS (<https://developer.movidius.com>). Similarly, for speech recognition in smart speakers and robots for example, the combination of the Intel® Atom™ processor with the Intel GNA (Gaussian Neural Accelerator), you can enable always-on listening using only milliwatts of power. You can experience this platform through the Intel speech enabling developer kit

(<https://software.intel.com/en-us/iot/speech-enabling-dev-kit>).

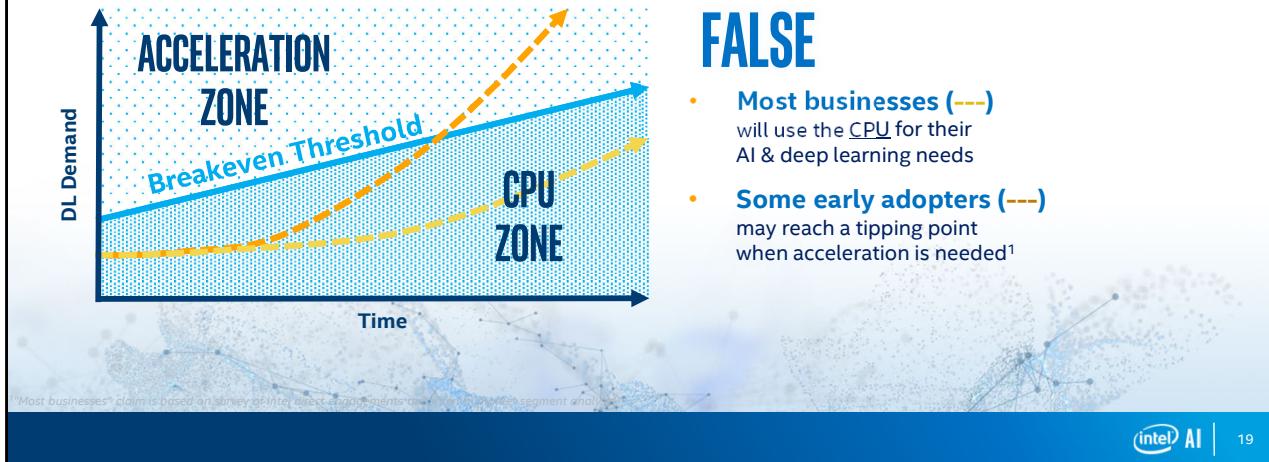
- For self-driving vehicles, Intel® Mobileye™ technology is your autonomous driving solution – it's a comprehensive self-driving vehicle platform that's been in development for years. For transportation-as-a-service oriented companies that want control over their IP and access to the bare silicon, the Intel® Nervana™ Neural Network Processor for inference is the best solution.
- For personal computing, including desktops, laptops, convertibles, tablets, smartphones and more, Intel is combining several of our dedicated accelerators (Movidius VPU, GNA) with our CPU technology (Atom, Core) and integrated processor graphics (Intel® Iris graphics) to deliver game-changing display, video/vision, AR/VR, speech and gesture capabilities.
- On the left-middle is the **edge**, which could be a small distributed cluster located at a company's factories around the world, an aggregation point like a network video recorder (NVR), a complex system like a car or MRI (magnetic resonance imaging) machine, or even a few servers or workstations acting as gateway devices. In other words, the “edge” is a broad category for localized compute. For the most part, most customers are doing all their deep learning inferencing on Xeon/CPU, unless they're consistently doing a tremendous amount of it and/or have specific use case requirements, which drives demand for general purpose acceleration. Even in that case, for customers who run into problems running inference on CPU, upgrading to the latest generation Xeon and utilizing the latest Intel-optimized deep learning software (frameworks & topologies) can help meet their demands.
 - For dedicated inference applications, the Intel® Nervana™ Neural Network Processor for inference will likely be the most efficient solution
 - For vision & inference workloads with higher performance/watt requirements, the Intel® Movidius™ vision processing unit (VPU) is a great option, available as a PCIe add-in card called the “Intel® Vision Accelerator Design with Intel® MovidiusTM MyriadTM X VPU”
 - For streaming latency-bound workloads with “real-time” inference demands, particularly in media & vision, the highly-flexible Intel® Arria® 10 FPGA is another option, , available as a PCIe add-in card called the “Intel® Vision Accelerator Design with Intel® Arria® 10 FPGA”
- Finally, on the left-bottom is **multi-cloud**, which consists of the largest ‘hyperscale’ deployments such as public clouds (AWS, GCP, etc), communication service providers, government labs, academic clusters, large enterprise IT (private and/or hybrid cloud) and more. For the most part, most customers are running their deep learning inferencing and training on Xeon/CPU, unless they're consistently doing a tremendous amount of it, which drives demand for general purpose acceleration. Even in that case, for customers who run into problems running on CPU, upgrading to the latest generation Xeon and utilizing the latest Intel-optimized deep learning software (frameworks & topologies) can help meet their demands.
 - For dedicated deep learning training environments, like those where that

workload is persistent and accounts for a large share of all compute cycles, the Intel® Nervana™ Neural Network Processor (NNP) is your purpose-built AI accelerator solution with multi-model and multi-user support – coming in 2019.

- For customers with memory-bandwidth bound (e.g. RNN/recurrent neural networks) and/or flexible acceleration needs, the Intel® Stratix® 10 FPGA is an option – especially such as with very specific custom use cases, unique IP flows, data types, and/or multi-function workload flows. If you understand how to use FPGA's overall, similar to how they're used in other accelerated applications, then this may be a good acceleration solution.

THE DEEP LEARNING MYTH

“A GPU IS REQUIRED FOR DEEP LEARNING...”



It's a myth that you need to use GPUs for AI or deep learning.

As you see in this chart, most enterprises are below the blue line, successfully using Intel® Xeon® processors for AI and deep learning inference, and are now increasingly using them for deep learning training too, thanks to optimizations that have led to breakthrough performance increases. This performance continues to improve over time, and many enterprises will never need acceleration to meet their needs.

That said, at some point down the line in your AI journey, you may reach an inflection point where acceleration does becomes necessary. This could be driven by a particular usage model at initial deployment or once your application “takes off” with huge growth in inference demand (e.g. your app explodes like Instagram). However, for the initial proof-of-concept (POC) – which can take a few weeks to several months, don’t waste your money on a limited-purpose GPU accelerator that will sit idle most of the time, and hardly save you any time (if at all) due to the added time required to manage/deploy/duplicate data/etc. If and when you are truly ready to benefit from acceleration, there will be exciting new options on the market to select from, some of which we cover in this presentation.

So, as a rule of thumb, if you are like most enterprises and are just beginning your AI journey, forget about acceleration and start with Xeon. It’s already the standard for deep

learning inference in the data center, and is now more capable than ever for deep learning training thanks in large part to all the software optimizations in the past 1-2 years. At some point during the AI journey, you may need acceleration for a particular use case or because deep learning has grown to be significant in your overall compute mix, but cross that bridge when (and if) it comes... in fact, you may be more than satisfied with the continuous extension of Xeon AI performance that comes with each new generation, especially now that new AI features are being built into the silicon architecture.

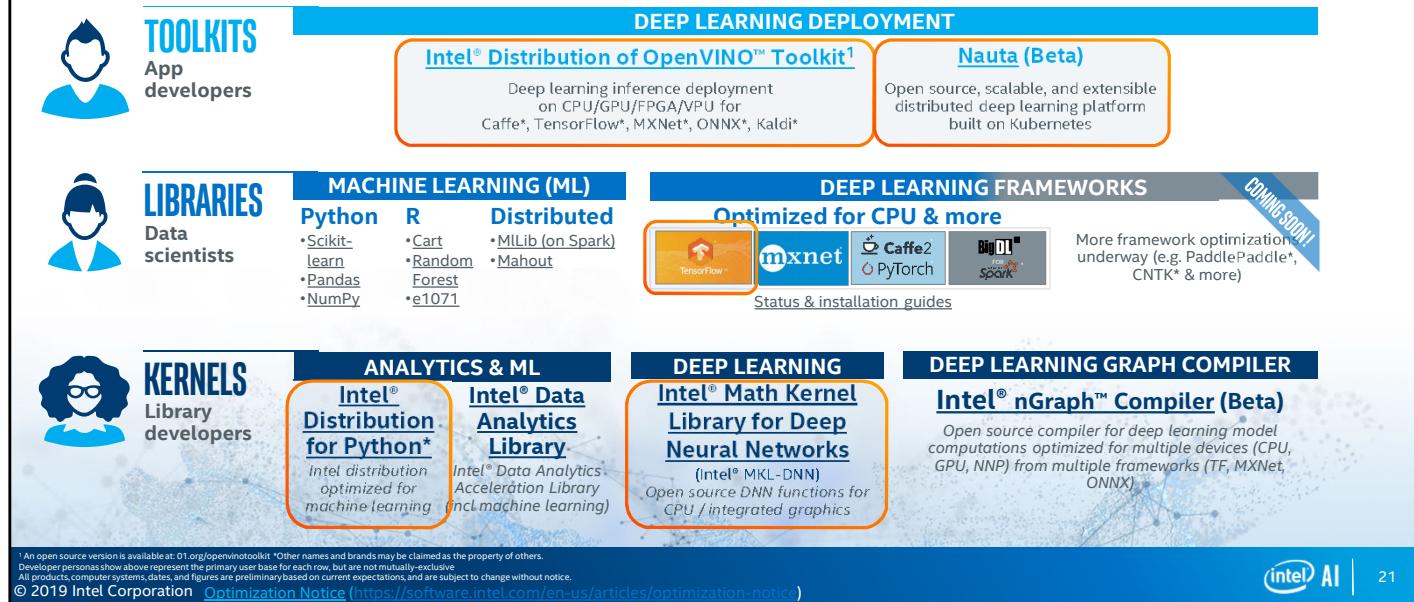


A large, abstract graphic occupies the top half of the page. It features a dark blue background with a lighter blue gradient. Overlaid on this are numerous small, white, glowing dots connected by thin lines, forming a complex network or molecular structure that spans the width of the frame.

SOFTWARE

SPEED UP DEVELOPMENT

WITH OPEN AI SOFTWARE



The infographic is organized into several sections:

- TOOLKITS** (App developers): Shows the Intel Distribution of OpenVINO Toolkit¹ and Nauta (Beta).
- LIBRARIES** (Data scientists): Shows Python, R, and Distributed libraries.
- MACHINE LEARNING (ML)**: Details Python, R, and Distributed libraries.
- DEEP LEARNING DEPLOYMENT**: Shows the Intel Distribution of OpenVINO Toolkit¹ and Nauta (Beta).
- DEEP LEARNING FRAMEWORKS**: Shows Optimized for CPU & more (TensorFlow, mxnet, Caffe2, PyTorch, BigDL, spark) with a "Status & installation guides" link and a "COMING SOON!!" note.
- KERNELS** (Library developers): Shows the Intel Distribution of OpenVINO Toolkit¹ for Python*, Intel Data Analytics Library, and Intel Math Kernel Library for Deep Neural Networks.
- ANALYTICS & ML**: Shows Intel Data Analytics Library.
- DEEP LEARNING**: Shows Intel Math Kernel Library for Deep Neural Networks.
- DEEP LEARNING GRAPH COMPILER**: Shows Intel nGraph™ Compiler (Beta).

1 An open source version is available at: 01.org/openvino/toolkit. *Other names and brands may be claimed as the property of others.
Developer personas shown 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.
© 2019 Intel Corporation. [Optimization Notice](https://software.intel.com/en-us/articles/optimization-notice) (<https://software.intel.com/en-us/articles/optimization-notice>)

With Intel AI, you can speed up development with open AI software.

Intel is investing in AI tools that get the most out of, and streamline development across, each hardware option in our portfolio. This ultimately accelerates total time-to-solution.

- **For application developers**—those who deploy solutions using AI-based algorithms—Intel has several tools to optimize performance and accelerate time-to-solution. For deep learning, the Intel Distribution of OpenVINO Toolkit facilitates model deployment for inference by converting and optimizing trained models for whichever hardware target is downstream. It offers support for models trained in TensorFlow, Caffe, and MXNet on CPU, integrated GPU, VPU (Movidius Myriad 2/Neural Compute Stick), and FPGA. Intel also launched the beta of a tool to help compress the end-to-end deep learning development cycle. This open source, scalable and extensible distributed deep learning platform, built on Kubernetes, is called Nauta (pronounced as ‘nau·ta’; means ‘sailor’ in Latin), formerly known as the Intel Deep Learning Studio.
- **For data scientists**—those who create AI-based algorithms—Intel contributes to and optimizes a set of open source libraries that are widely used for machine and deep learning. There are a number of such machine learning libraries that can be used to get the most out of Intel hardware today, spanning Python, R, and Distributed. For deep

learning, Intel aims to ensure that all the major deep learning frameworks and topologies run well on Intel hardware, and customers are of course free to choose whichever frameworks best suit their needs. We've been directly optimizing the most popular AI frameworks first, based on market demand, and producing huge improvements. Today, we have many optimized topologies available for TensorFlow, MXNet, Caffe2/PyTorch, and BigDL on Spark, and you can download and install the optimized version of these frameworks by clicking on the link in this slide. Going forward, we intend to enable even more frameworks like PaddlePaddle, CNTK & many more through the Intel nGraph compiler.

- **For library developers**—those who develop and optimize APIs, libraries, and frameworks to support new algorithms and topologies on the underlying hardware—Intel offers a host of foundational building blocks to get the most out of our hardware. Beginning on the left with the primitives category, the Data Analytics Acceleration Library and Intel Python distribution are important building blocks for machine learning. The DNN (deep neural network) open source libraries contain CPU-optimized functions that are most relevant for, you guessed it, deep learning model development. On the right side of this row is a description of the Intel nGraph library (formerly the Nervana Graph), which takes the computational graph from each deep learning framework and creates an intermediate representation, which is executed by calling the math accelerator software libraries of each Intel hardware target. This compiler reduces the need for framework and model direct optimization for each hardware target using low-level software and math accelerator libraries. Today, it supports Intel Xeon CPUs, GPU (CUDA), and the Crest family, with more hardware targets planned going forward.



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
portal on Intel® DevMesh

SOFTWARE.INTEL.COM/AI

intel AI | 23

So where can you get started? The Intel AI academy is a great place to start for developers, students, instructors and startups. There, you can learn all about AI, download tools and resources to begin development with AI, find course materials to teach others & spread the knowledge, and share things that you've learned and created with the AI community. To get started, go to software.intel.com/ai.



intel Software

SHARE YOUR AI PROJECTS WITH THE WORLD

GET NOTICED WITH INTEL® DEVMESH

CONNECT WITH DEVELOPERS, SHARE YOUR SKILL, GAIN REPUTATION

- 9K+ Member Profiles
- 1900+ Developer Projects
- Community Repos
- Community Groups
- Developer Blogs
- Developer Speakership Programs

DEVMESH.INTEL.COM

intel AI | 24

AI BUILDERS: ECOSYSTEM 100+

AI Partners

CROSS VERTICAL



SYSTEM INTEGRATORS



VERTICAL



HORIZONTAL



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

BUILDERS.INTEL.COM/AI

intel AI | 25

And last but certainly not least, you can turn to Intel or one of our many ecosystem partners to help you get started on your AI journey. Visit builders.intel.com/ai to find out more about one of our 100 and counting list of AI builder partners.



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

www.intel.ai

intel AI | 26

Intel is the only company that spans the entire AI journey. The Intel AI commitment to our customers is simple: we're here to help them break barriers between AI theory and reality. With Intel AI, our customers can **simplify AI, choose any approach, tame their data deluge, speed up development, deploy AI anywhere, and scale with confidence**. We look forward to building what was once thought to be impossible, with YOU. For more information about all these and more, visit www.intel.ai

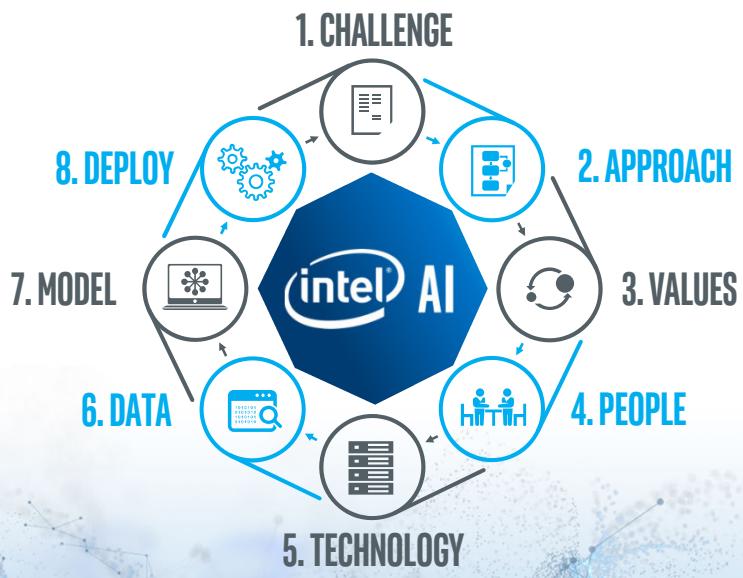
RESOURCES

- Intel® AI Developer Program
 - <https://software.intel.com/en-us/ai>
- Intel® AI Courses
 - <https://software.intel.com/en-us/ai/courses>
- Intel® DevCloud
 - <https://software.intel.com/en-us/devcloud>
- Intel® AI Support Forum
 - <https://software.intel.com/en-us/forums/intel-optimized-ai-frameworks>
- Intel® DevMesh
 - <https://devmesh.intel.com>



AI JOURNEY WITH INTEL

THE AI JOURNEY WITH AN INTEL CASE STUDY



intel AI | 29

In the next four slides, we'll walk through an Intel AI case study to illustrate this journey.

INTEL AI CASE STUDY

The diagram illustrates the Intel AI Case Study process flow. It starts with a 'Challenge' phase, followed by 'Approach', 'Values', 'People', 'Technology', 'Data', 'Model', and finally 'Deploy'. Each phase is accompanied by an icon and a brief description. A callout box highlights the 'Approach' phase, which involves identifying the complexity of each solution and choosing high-ROI industrial defect detection using Deep Learning (DL). Another callout box highlights the 'Deploy' phase, which involves securing internal buy-in and growing talent via the Intel AI developer program.

Value

Simplicity

CHALLENGE

APPROACH

VALUES

PEOPLE

TECHNOLOGY

DATA

MODEL

DEPLOY

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 DL¹

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

Corrosion

AI DEVELOPER PROGRAM

intel AI | 30

¹DL = Deep Learning

This is a case study based on Intel's AI engagement with an industrial sector company.

1. Challenge

This journey began with a survey of potential AI opportunities, starting with internal brainstorming, surveying the external landscape, and combing through the 70+ solutions in the Intel AI builders program. Once we identified some promising opportunities, the next step was to assess and rank the business value of implementing each AI solution.

2. Approach

The next step was to work with Intel's experts to identify the best approach (analytics, ML, DL, etc.) and estimate the associated complexity/cost of each solution. For example, building a new deep learning model from scratch is more costly than building off an existing deep learning model, which in turn is more costly than using a known machine learning method. We then plotted the top AI opportunities on a value/simplicity chart, it became clear which project would deliver the highest ROI: automating underwater industrial defect detection using deep learning image recognition.

3. Values

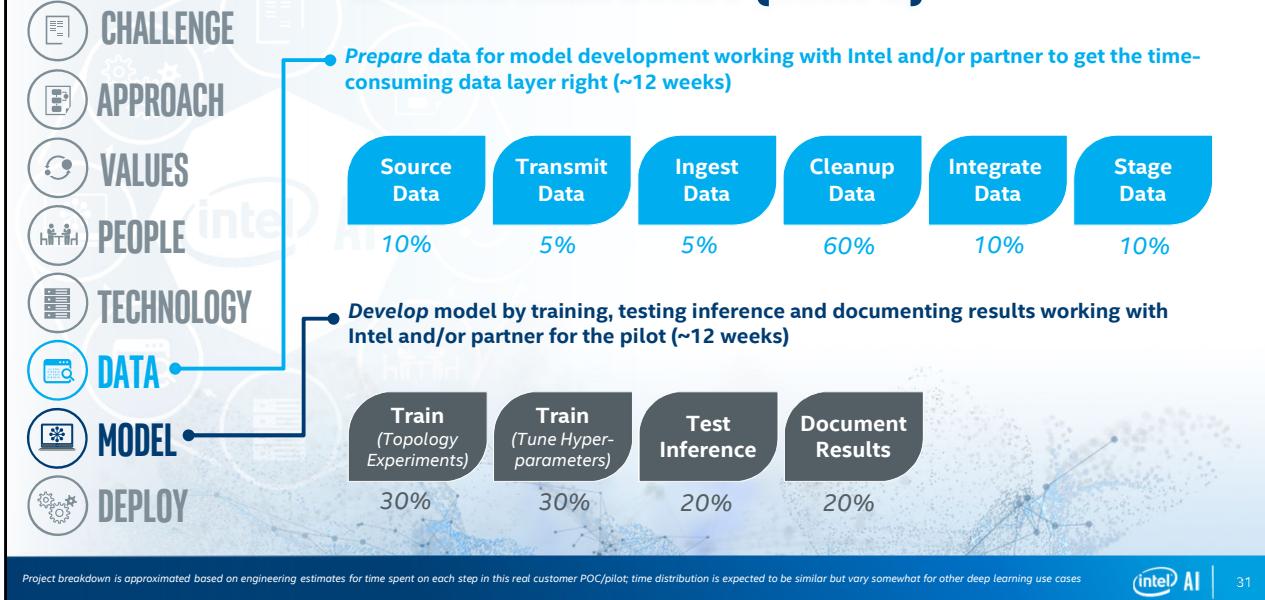
Before going any further down the chosen path, it's important to assess the "other"

ramifications of an AI implementation, beyond the dollars and cents. In this case, we discussed the legal, social, and ethical issues that may arise, what we could do to mitigate them, and whether there were any showstopper risks. We also documented the assessment and mitigation plan to revisit if/when this pilot goes into production.

4. People

The next step was to secure organizational buy-in and build up the right talent. This step is crucial, because if key stakeholders aren't ready to accept data-driven insights, then all the work ahead may be for naught. A classic example is the initial resistance to data analytics in sports, where general managers and scouts scoffed at the idea of computer algorithms outsmarting their years of experience and tribal knowledge. We used other Intel AI solution briefs and customer testimonials to get buy-in, as well as ensure that the organization was ready to embrace the fact that AI development is *different*, involving more trial & error and uncertainty than traditional software development. Next, we assessed the talent situation and determined that training up existing developers through Intel's free AI developer program was the best approach.

INTEL AI CASE STUDY (CONT'D)



While the time slice breakdowns you'll see on this slide are only based on this example, other projects will vary slightly but generally follow the same process.

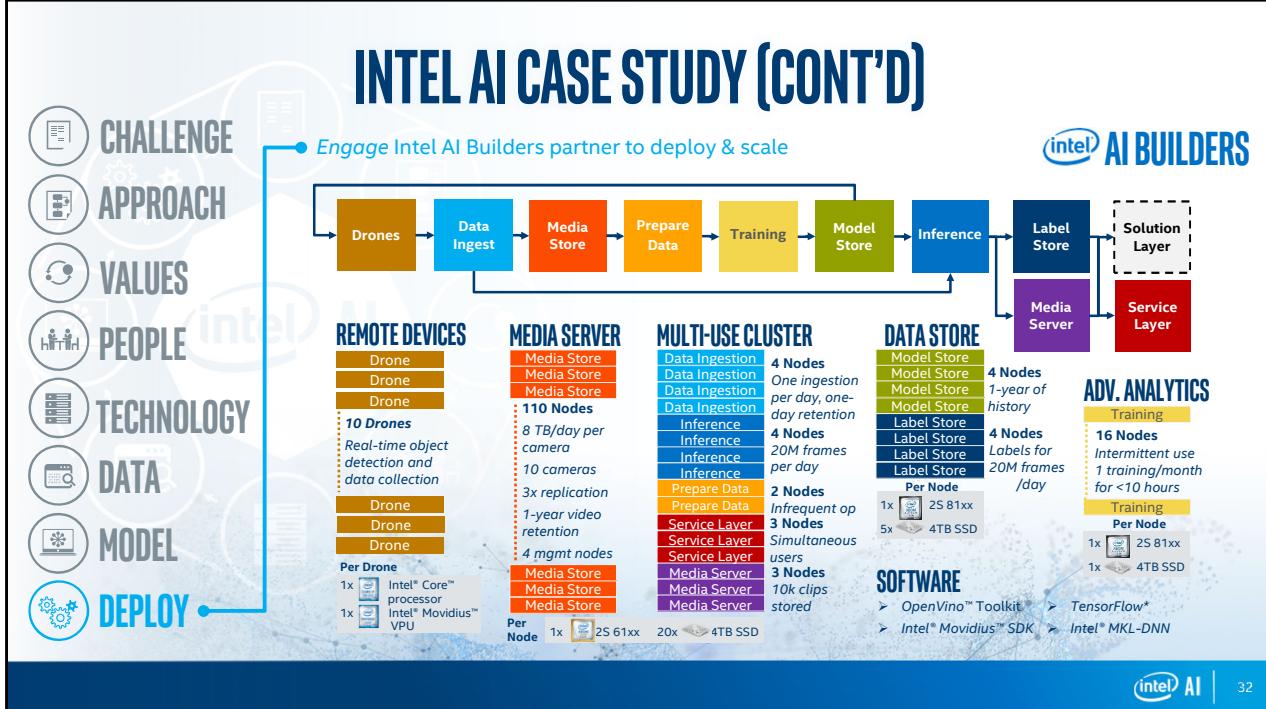
6. Data

One of the biggest barriers to AI, which is often overlooked, is getting your data ready. From sourcing to storing to preparing/cleansing data for analysis, Intel worked with this customer to get their data layer right – a stage that took about as long as the actual model development itself!

7. Model

Once the data was ready, the team began experimenting with various topologies and tuning hyperparameters through iterative training runs. Once a sufficiently high accuracy was reached, the trained model was tested against a control data set, and inference results achieved a high enough accuracy to proceed to the cleanup & documentation phase. About 60% of the time was spent training, whereas the rest was testing & documentation.

INTEL AI CASE STUDY (CONT'D)



8. Deploy

The final (and arguably most complex) stage is to take the pilot to production, deploying the model at scale.

In this case, our customer joined forces with a partner from the Intel AI Builders program to put together a “real world” AI solution.

The colorful block diagram at the top is a functional description of each step in this industrial defect detection scenario. There are 10 underwater drones that are equipped with video cameras to monitor heavy industrial equipment in order to detect potential defects. These drones capture videos of the underwater equipment, which is then ingested into the data center. Those videos are stored, for use in re-training the model and future reference, as well as passed to the inference cluster to determine if and where defects are. For re-training, human experts label images where the equipment was present or not, and where defects were present or not, in order to continue build the dataset and achieve even higher levels of accuracy. The latest trained models are stored, with one being deployed to perform object recognition inference on the drone (to aim the cameras at the equipment itself), and the other deployed to perform defect image recognition inference in the data center on the ingested video streams. As possible defects are identified, the inference output is sent

to both the service layer (for human audit) and the solution layer, where it is used as part of a larger decision process to determine whether to call a technician and/or shut down the equipment.

The stacks at the bottom of this slide illustrate the infrastructure – both hardware and software – underlying each colored step in the solution. This includes a whole lot of Xeon-based servers in the data center with SSD storage, Movidius VPU's in the drones, and Intel AI software like the OpenVINO toolkit, the Movidius SDK, and the latest Intel-optimized version of TensorFlow with MKL-DNN.

THE BOTTOM LINE here is that AI in the real world is much more involved than in the lab, and Intel & our partners are here to help you... not only with your deployment at scale, but to accelerate each and every step in your AI journey. Next, we'll see what Intel AI brings to the table.

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

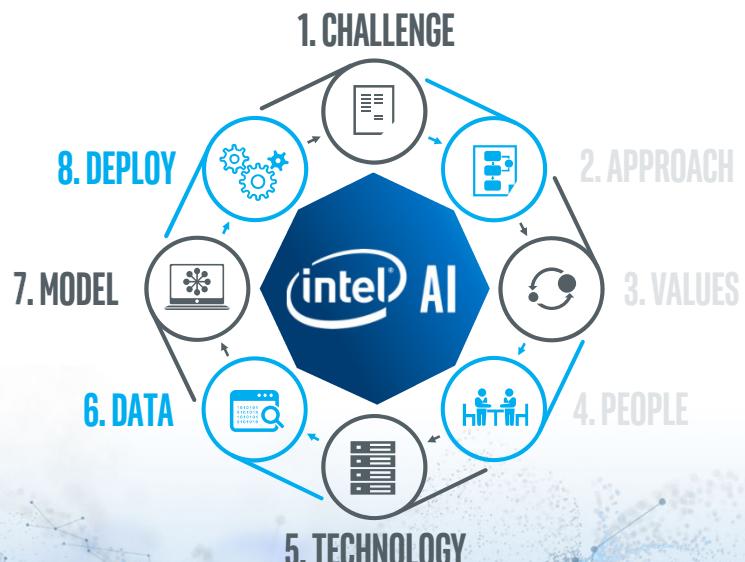
ADDRESSING THE AI JOURNEY IN THE CLASSROOM

- 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



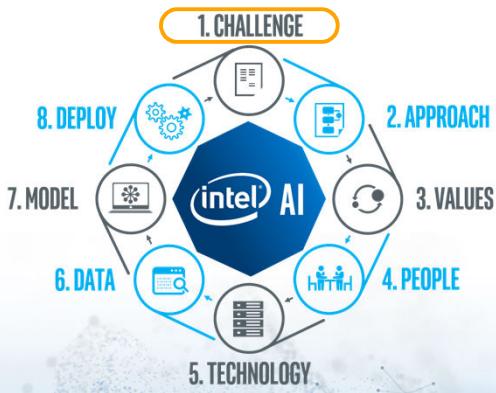
HANDS-ON PROBLEM SOLVING

THE AI JOURNEY - STEPS WE WILL COVER IN THIS COURSE



In the next couple of slides, we'll walk through the challenge

STEP 1 - THE CHALLENGE

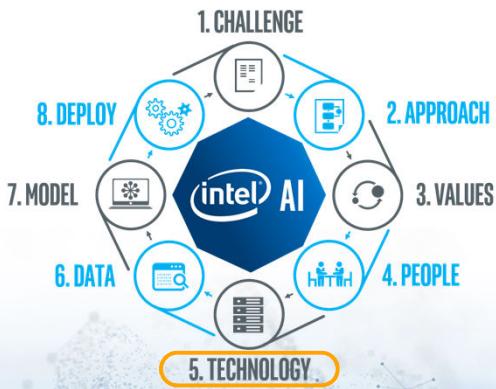


- Identify the challenge – Identification of most stolen cars in the US
 - Image recognition problem
- Application – Traffic surveillance
 - Extensible to License Plate Detection (not included in the class)



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 DEV CLOUD

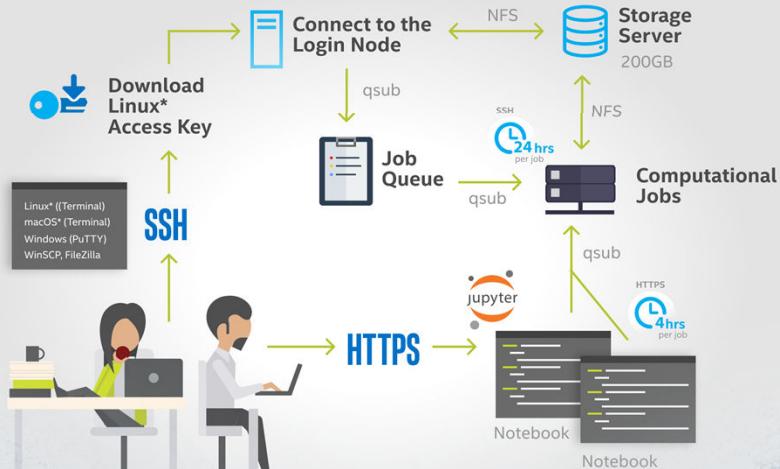
INTEL® DEV CLOUD

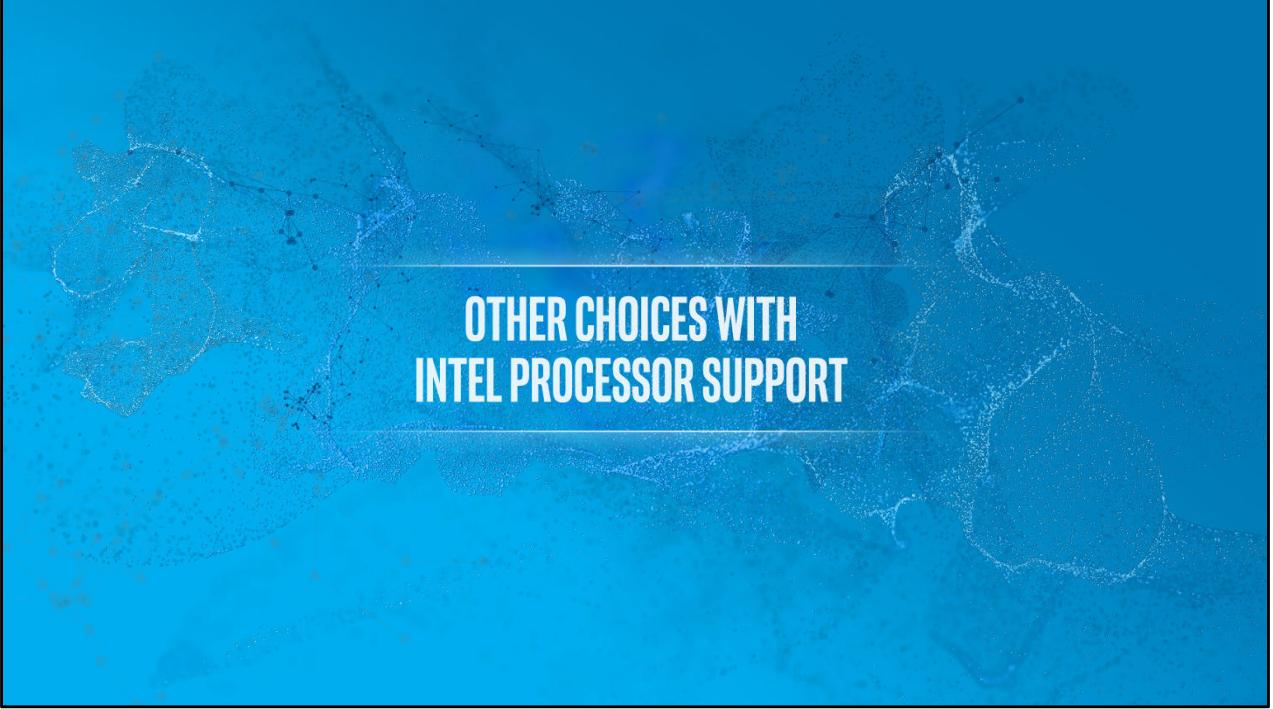
- A cloud hosted hardware and software platform available to Intel® AI Academy members to learn, sandbox and get started on Artificial Intelligence projects
- Intel® Xeon® Scalable Processors(Intel(R) Xeon(R) Gold 6128 CPU @ 3.40GHz 24 cores with 2-way hyper-threading, 96 GB of on-platform RAM (DDR4), 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/en-us/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

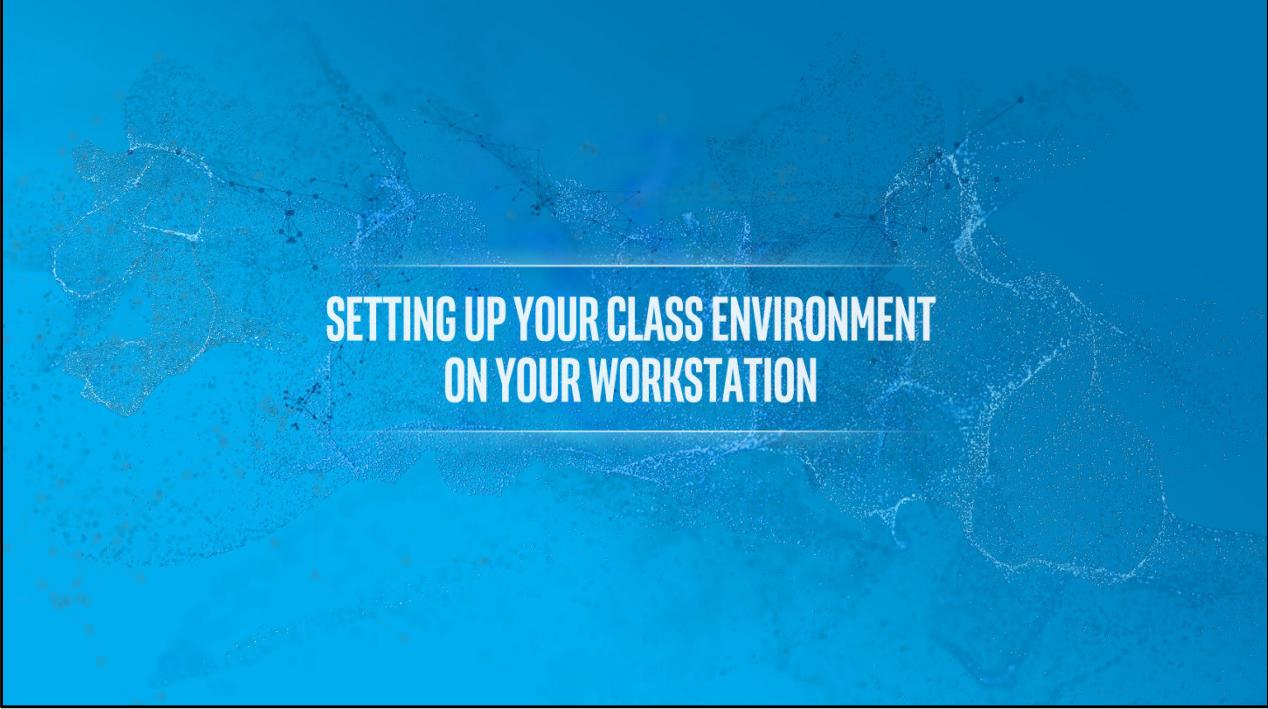
Google Compute Engine* (GCE):

- 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

Not all servers are equal, each CSP has different choices for servers. Depending on your favorite CSP we recommend looking for these types of instances to get the necessary Processors that support the optimized software features., e.g., AVX512 and or VNNI



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>



47

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. Run `conda activate tf_training` to activate the environment
4. `python -m ipykernel install --user --name tf_training --display-name "tf_training"`
5. Now run `jupyter notebook` to start your notebook.
6. In your notebook, select "Kernel -> Change kernel" and select "tf_training" 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 tf_training`



SETTING UP YOUR CLASS ENVIRONMENT ON INTEL® AI DEV CLOUD

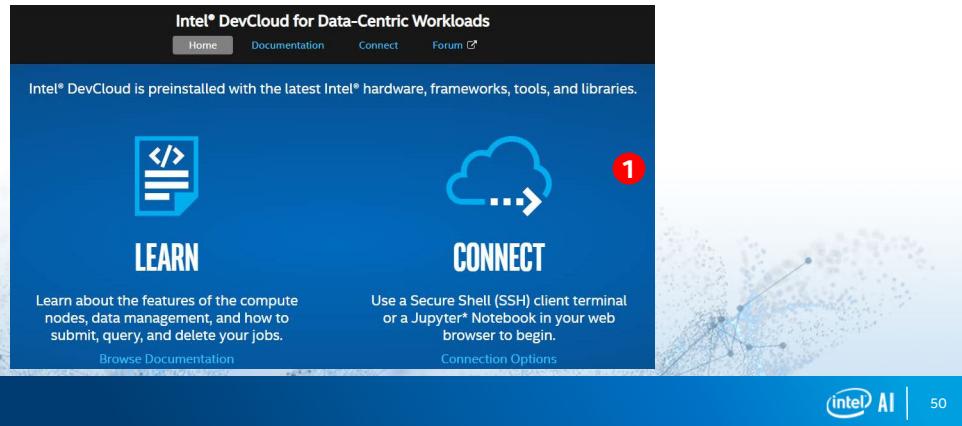
CONNECT TO YOUR DEV CLOUD ACCOUNT

Obtain an account on **Intel® DevCloud**:

<https://software.intel.com/en-us/devcloud/datacenter>

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.



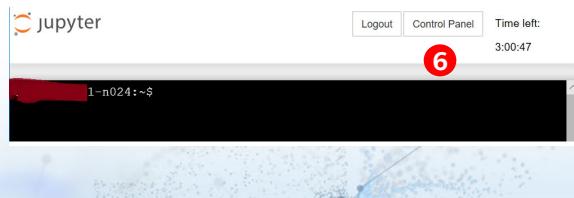
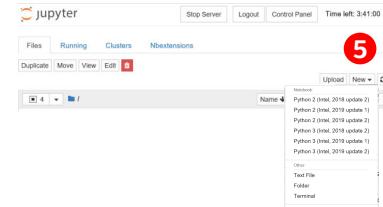
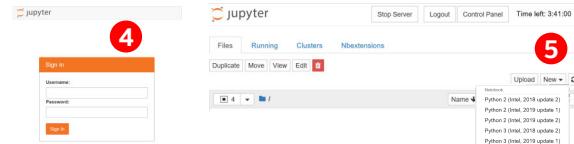
CONNECT TO YOUR DEV CLOUD ACCOUNT

2. Choose one of three connection options.
 1. Connecting with Terminal (from Linux/Mac/Windows)
 2. Connecting to a Jupyter Notebook
3. We are choosing option 2 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 <https://jupyter.devcloud.intel.com/hub/login> link.

The screenshot shows the 'CONNECT TO INTEL® DEV CLOUD' page. At the top, there's a navigation bar with links for Home, Documentation, Connect (which is highlighted), and Forum. Below the navigation bar, a heading says 'CONNECT TO INTEL® DEV CLOUD'. A sub-instruction reads 'Use a Secure Shell (SSH) client terminal or a Jupyter® Notebook in your web browser to begin.' Two numbered options are shown: '2' with a terminal icon and '3' with a Jupyter notebook icon. Option 2 has a 'Connect with a Terminal' section with a note 'Select your operating system to get started.' It lists 'Windows® (PuTTY)' and 'Linux® or macOS (SSH client)'. Option 3 has a 'One-click Log In' button and fields for 'URL', 'Username', and 'Password'. The URL field contains 'https://jupyter.devcloud.intel.com/'. The username field contains 'u27501' and the password field contains 'show'. The bottom right corner of the screenshot features the Intel AI logo and the number '51'.

ACCESS YOUR DEVCLOUD JUPYTER NOTEBOOK ACCOUNT

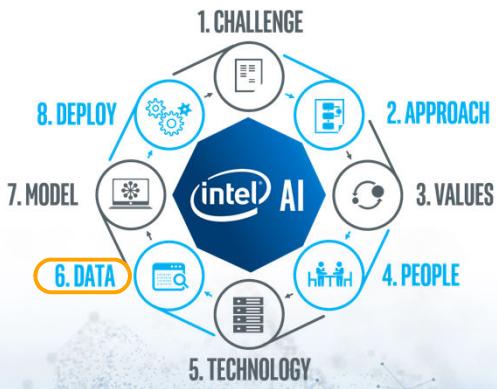
4. Enter the previously copied username and password to access your jupyter notebook account.
5. Click on the 'New' menu on the right side of the page and select the 'Terminal' to access the terminal
6. Now you are connected to your DevCloud account terminal via jupyter notebook. You can always return 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

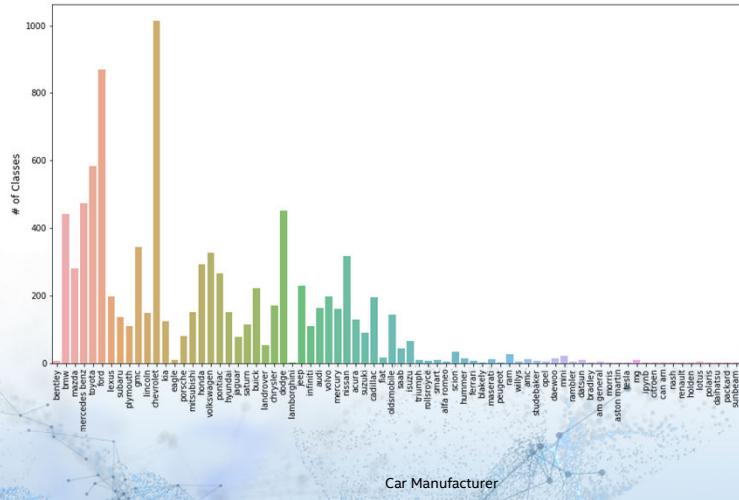
OBTAI N A STARTER DATASET

- 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** (<http://vmmrdb.cecsresearch.org/>)

INITIAL ASSESSMENT OF THE DATASET

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

- <http://vmmrdb.cecsresearch.org/>
 - Large in scale and diversity
 - Images are collected from Craigslist
 - Contains 9170 classes
 - Identified 76 Car Manufacturers
 - 291,752 images in total
 - Manufactured between 1950-2016



DATASET FOR THE STOLEN CARS CHALLENGE

Hottest Wheels: The Most Stolen New And Used Cars In The U.S. (<https://www.forbes.com/sites/jimgorzelany/2018/09/18/hottest-wheels-the-most-stolen-new-and-used-cars-in-the-u-s/#3e9577545258>)

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 # indicates number of stolen cars in each model in 2017
- 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

The problem we are trying to solve is based on the hottest wheels – most stolen cars.

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](#)

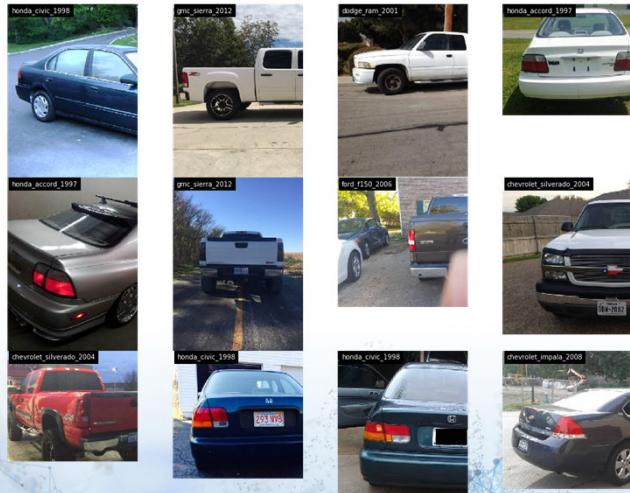
INSPECT THE DATASET

- **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)



We wanted a category of everything but the top 10 most stolen. After experimentation we discovered that it had too many similarities to the other 10 categories and we ended retraining without the others category and we got significant improvement in prediction.

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



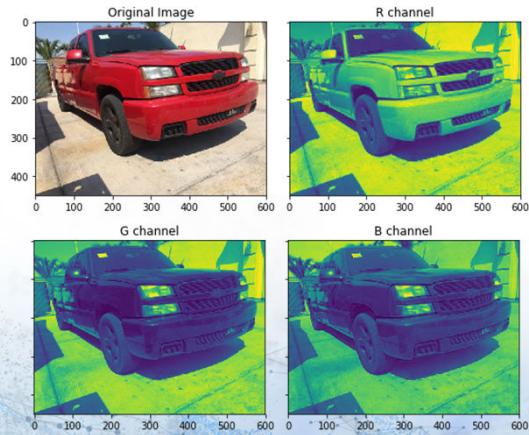
SAMPLE STD
NORMALIZATION



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)
```



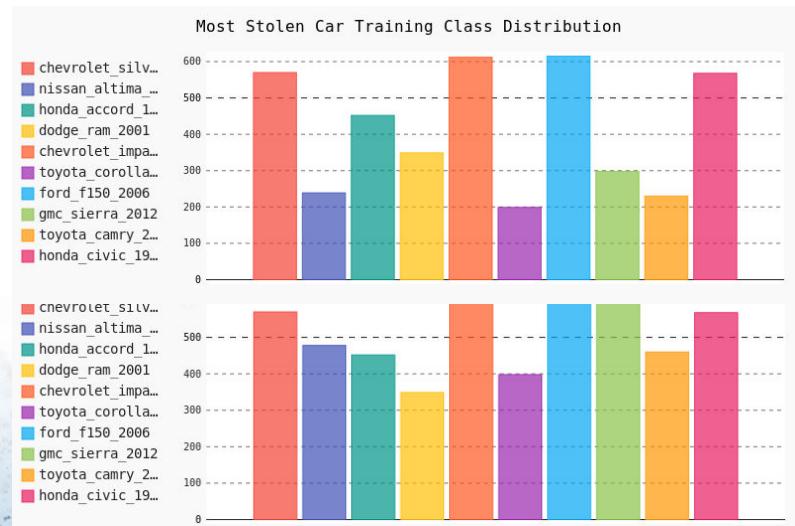
DATA AUGMENTATION

- Oversample Minority Classes in Training

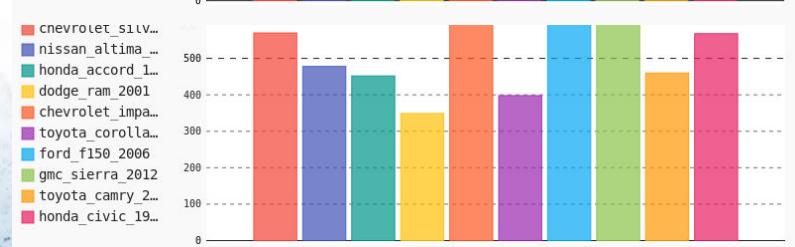


SUMMARY

Before Preprocessing



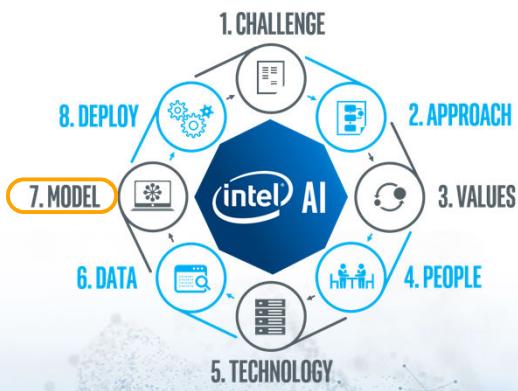
After Preprocessing





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



TensorFlow™



mxnet



Caffe2
PyTorch



BigDL™
FOR
Apache Spark*

More under optimization:



PaddlePaddle and more.

GET STARTED TODAY AT [HTTPS://SOFTWARE.INTEL.COM/EN-US/Frameworks](https://software.intel.com/en-us/frameworks)

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

intel AI | 72

How do you unleash all that deep learning performance on the Intel® Xeon® Processor? Well, you need to install an Intel-optimized framework to get started.

Intel aims to ensure that all major DL frameworks and topologies will run well on Intel Architecture, and customers are free to choose whichever framework(s) best suit their needs. We've been directly optimizing the most popular AI frameworks for Intel Architecture (based on market demand) and producing huge speedups. We intend to enable even more frameworks in the future through the Intel® nGraph™ Compiler. Please note that each of these frameworks have a varying degree of optimization and configuration protocols, so visit ai.intel.com/framework-optimizations/ for full details. Of special note is the BigDL framework that's been getting a LOT of traction lately with customers who want an easy way to achieve high-performance deep learning on their existing big data/analytics infrastructure. BigDL is a distributed deep learning library for Spark that can run directly on top of existing Spark or Apache Hadoop* clusters with support for Scala or Python programming languages.

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 Inferred 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).

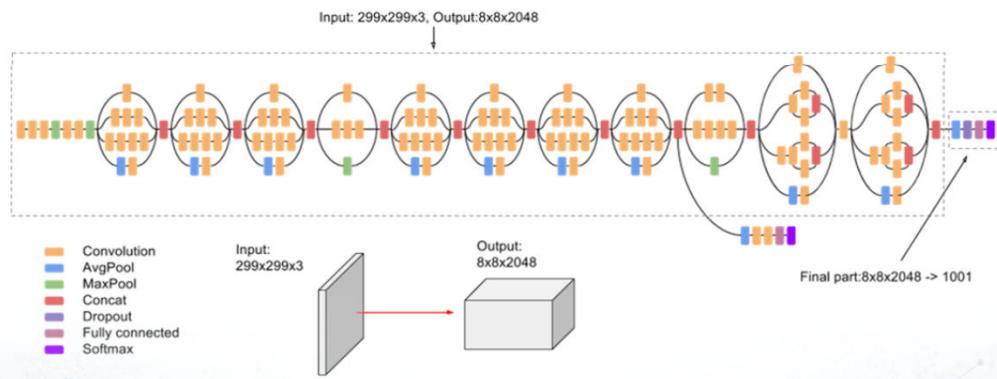
The original paper* was trained on ResNet-50. However, it is not supported currently on Intel® Movidius™ Neural Compute Stick.

The supported networks that we trained the model on:

- Inception v3
- VGG16
- MobileNet

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

INCEPTION V3



ImageNet 2015

Szegedy, et al. 2014

Idea: network would want to use different receptive fields

Want computational efficiency

Also want to have sparse activations of groups of neurons

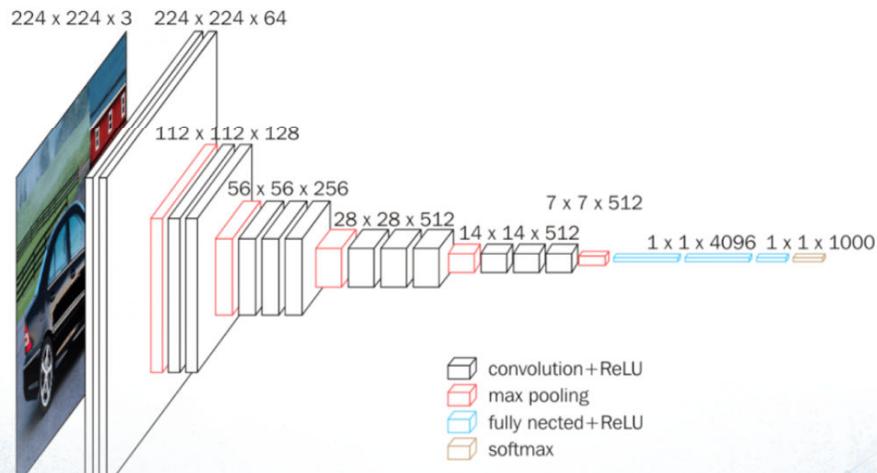
Hebbian principle: “Fire together, wire together”

Solution: Turn each layer into branches of convolutions

Each branch handles smaller portion of workload

Concatenate different branches at the end

VGG16



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

One of the first architectures to experiment with many layers (more is better approach)
Uses multiple 3x3 convolutions to simulate larger kernels with fewer parameters

two 3x3 convolutions are equal to one 5x5

three 3x3 convolutions are equal to one 7x7

$$3 \times 3 \times c \times c = 9c^2$$

$$7 \times 7 \times c \times c = 49c^2$$

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	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5x Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
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 \times 1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

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

Picked initially due to it's small nature

Uses global hyperparameters that efficiently tradeoff between latency and accuracy

These hyper-parameters allow the model builder to choose the right sized model for their application based on the constraints of the problem

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)

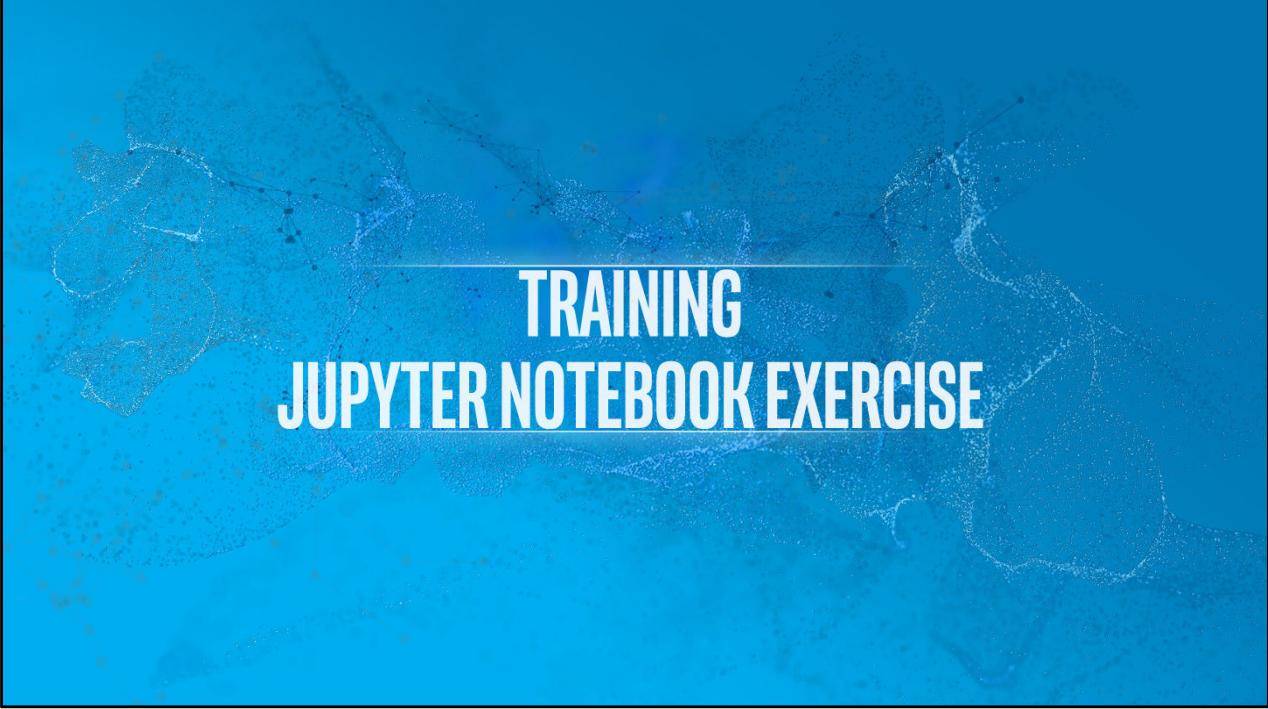
As you will see in the hands on section your results will be similar

SUMMARY

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

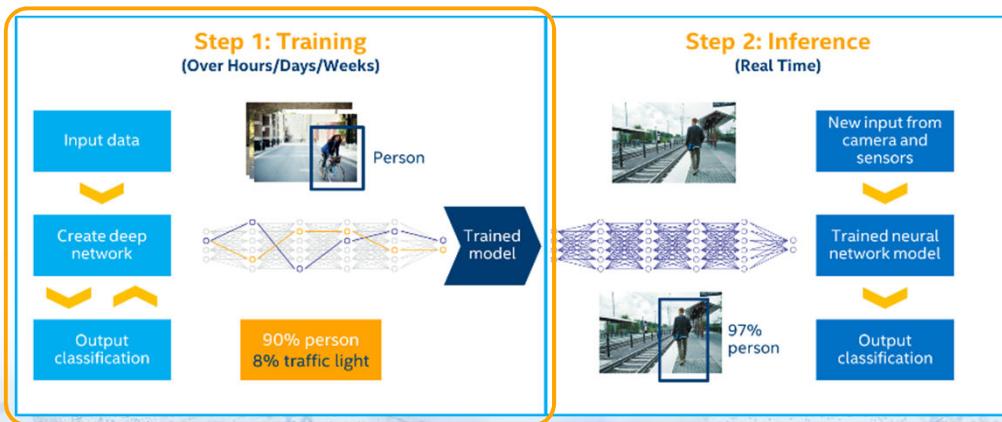
- Time to train
- Size of the model
- Inference speed
- Acceptable accuracy

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



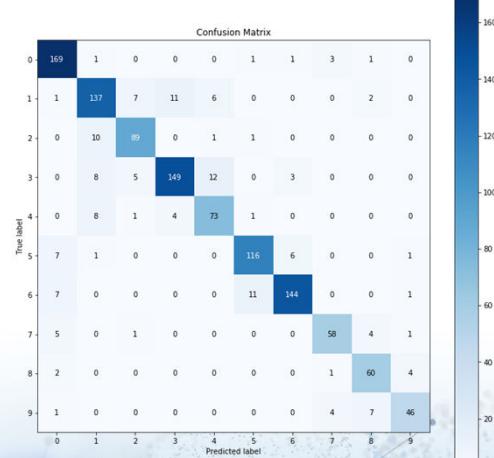
Complete Notebook : Part2-Training_InceptionV3.ipynb

(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

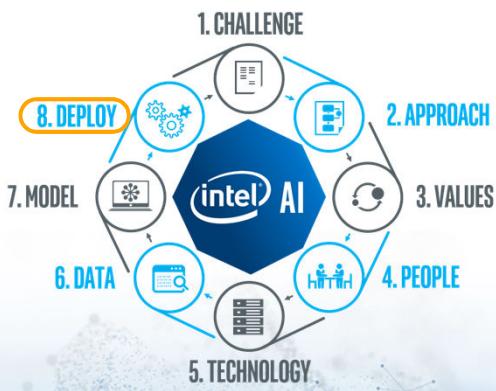
- Understand how to interpret the results of the training by analyzing our model with different metrics and graphs
 - Confusion Matrix
 - Classification Report
 - Precision-Recall Plot
 - ROC Plot
- [Complete Notebook – Part3-Model_Analysis.ipynb](#)





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?



1. Inference on the PC is the process of performing computations on custom or specialized trained AI models in systems where limitations for size, power, and real-time performance are required to ensure success.

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

- <https://towardsdatascience.com/deep-learning-on-the-edge-9181693f466c>
- **1. Bandwidth and Latency**
- **2. Security and Decentralization**
- **3. Job Specific Usage (Customization)**



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

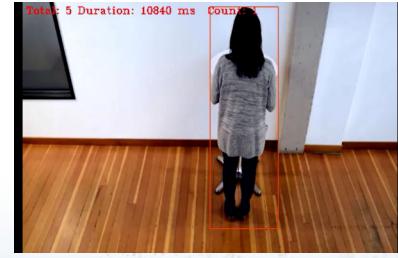
Video stored locally

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 emotions 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



<https://towardsdatascience.com/background-removal-with-deep-learning-c4f2104b3157>



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.

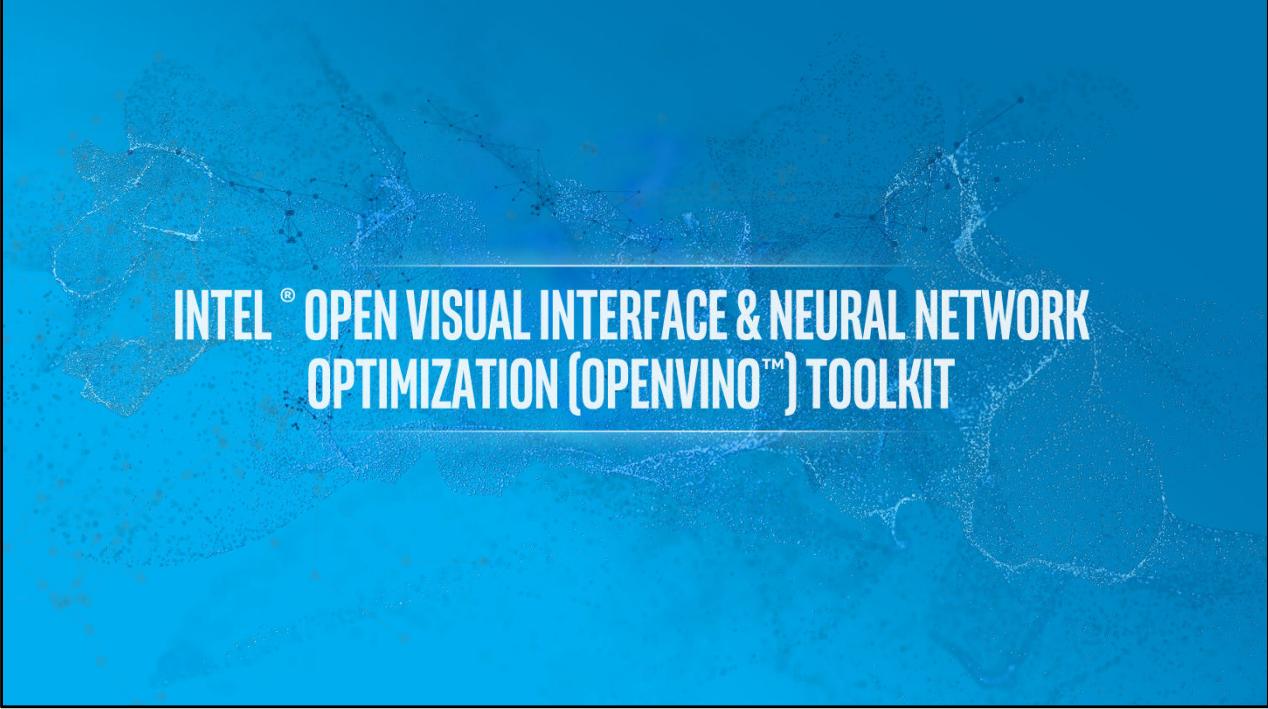
- Object Detection
- Standard & Pipelined Image Classification
- Security Barrier
- Object Detection for Single Shot Multibox Detector (SSD) using Asynch API
- Object Detection SSD
- Neural Style Transfer
- Hello Infer Classification
- Interactive Face Detection
- Image Segmentation
- Validation Application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

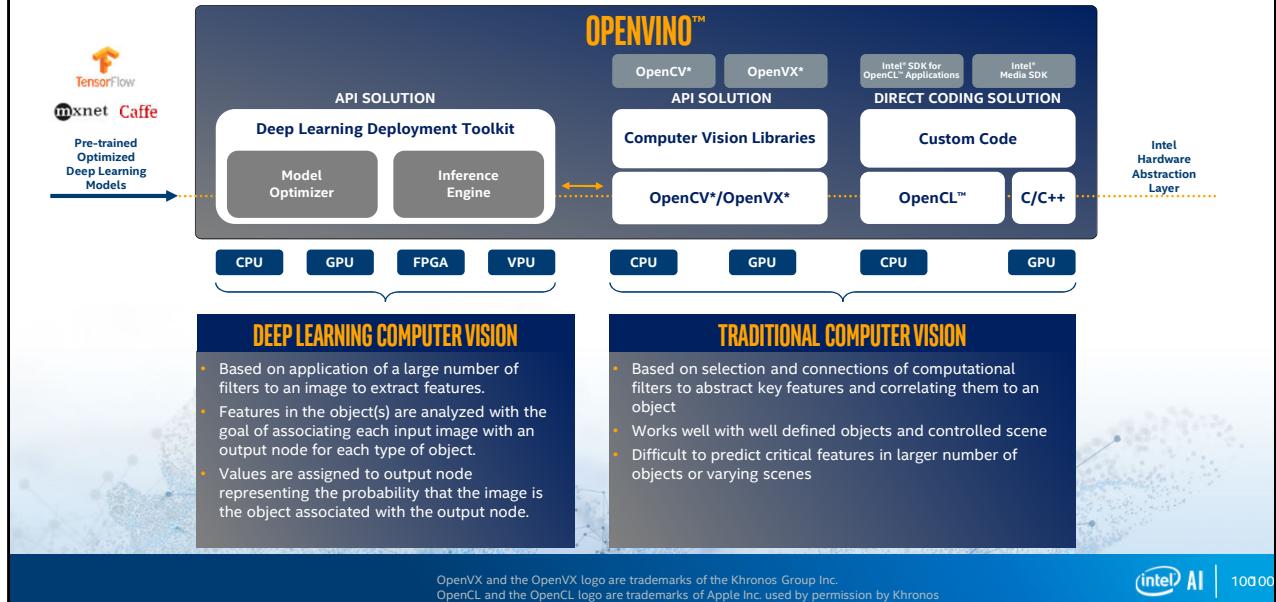
Sharpen the difference



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



Key Takeaways

- **OpenVINO™** has tools for both Traditional and Deep Learning CV
- Multiple Intel tools (Media SDK, **OpenVINO™**, ISS) work together to provide a complete CV pipeline optimization solution
- Using **OpenVINO™** allows developer to maximize HW performance by using common API without having to go to the Metal
- Easy to incorporate deep learning with the Deep Learning Deployment Toolkit
- Trad. And DL are not mutually exclusive

OpenCL is used for:

- required to run with GPU target (cIDNN) using Intel® Processor Graphics
- custom kernels
- other kernels can be used for other non-inference pipeline stages, such as color conversions

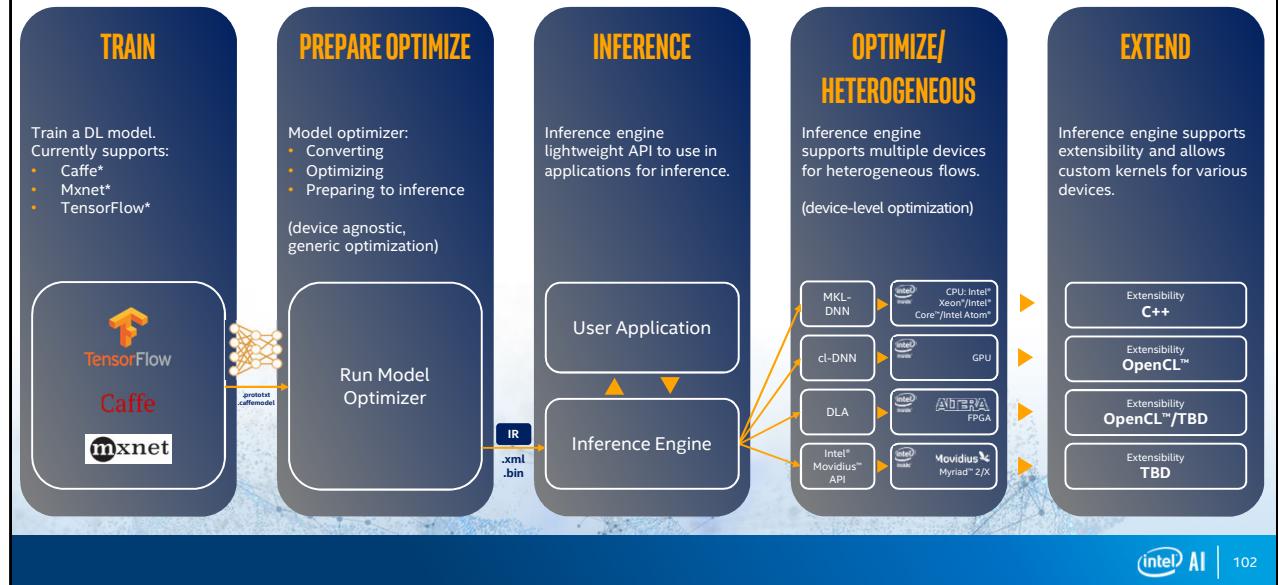
Media SDK – API to access intel Quick Sync Video – hw accelerated encoding, decoding and processing

- H.265, H.264, MPEG-2 and more
- Resize, scale, deinterlace, color conversion,, composition, denoise, sharpen and more
- Outstanding perf., rich API to tune pipeline, support new proc. w/o code change



USAGE MODEL

INTEL® DEEP LEARNING DEPLOYMENT TOOLKIT



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

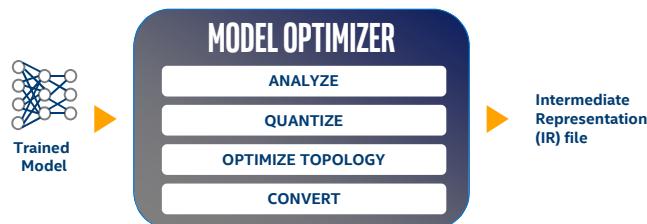
3. The Model Optimizer provides tools to convert a trained model to a frozen graph in the event it is not already done.



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

intel AI | 105

The **redesigned Model Optimizer software** is implemented as Python code, replaces the previous solution entirely and offers new features:

- entirely new workflow, at the same time simplified and not requiring User to rebuild Caffe, etc.
- Windows support;
- Caffe is not required to generate IRs for models consisting of Standard Layers, OR when user already provides his custom layers;
- fallback to original framework is possible in case of unsupported layers (then framework is required);
- additional optimizations generalized from existed in the old MO;
- improved usability, stability and diagnostics capabilities; [no analyzer cap]
- total ~110 public models supported for Caffe, MXNet and TensorFlow frameworks – list is available on request;

The Model Optimizer is easier to install, and easier to use for optimizations
Improved performance and output

deep learning

written in easy Python language, more efficient workflow
using standard layers, get faster performance without the overhead of
frameworks

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.

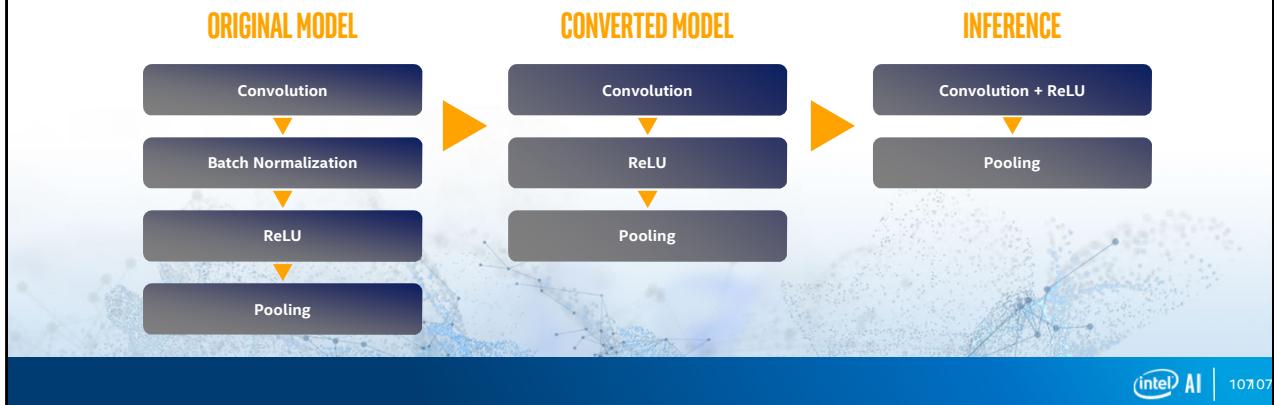
The **redesigned Model Optimizer software** is implemented as Python code, replaces the previous solution entirely and offers new features:

- entirely new workflow, at the same time simplified and not requiring User to rebuild Caffe, etc.
- Windows support;
- Caffe is not required to generate IRs for models consisting of Standard Layers, OR when user already provides his custom layers;
- fallback to original framework is possible in case of unsupported layers (then framework is required);
- additional optimizations generalized from existed in the old MO;
- improved usability, stability and diagnostics capabilities; [no analyzer cap]
- total ~110 public models supported for Caffe, MXNet and TensorFlow frameworks – list is available on request;

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.



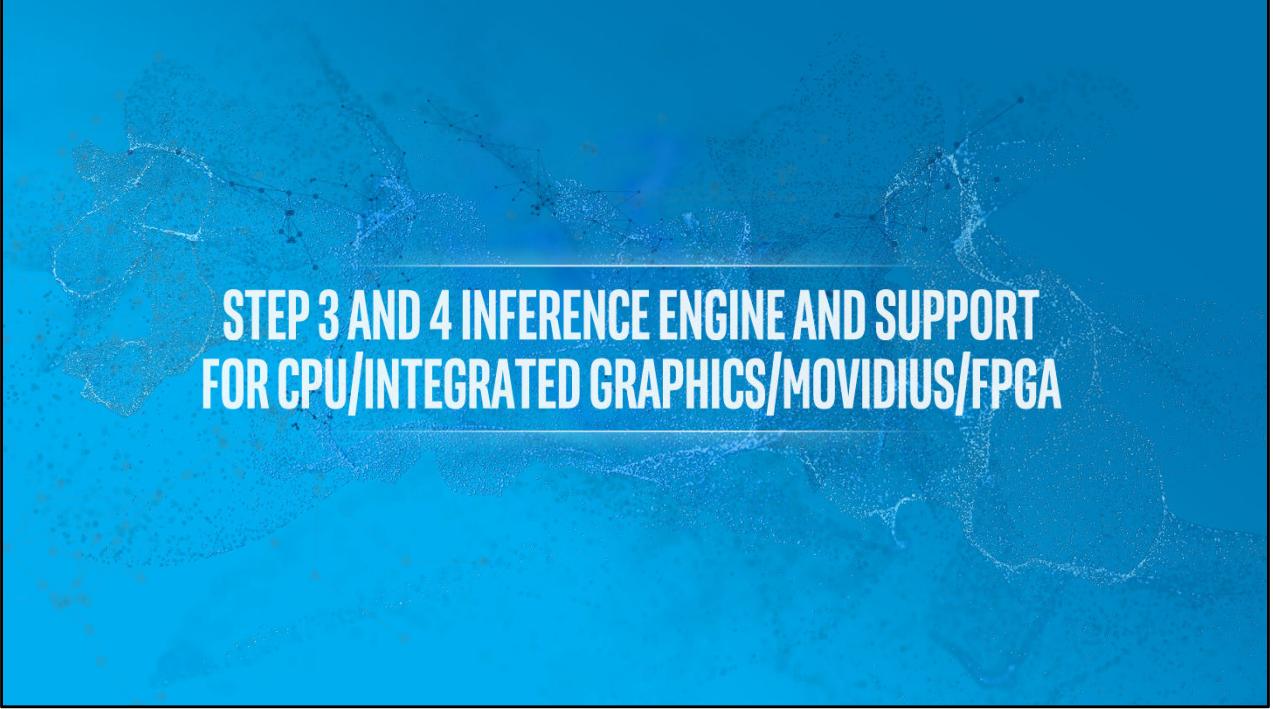
- The Model Optimizer is easier to install, and easier to use for optimizations
- Improved performance and output
- deep learning
- written in easy Python language, more efficient workflow
- using standard layers, get faster performance without the overhead of frameworks

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://docs.openvinotoolkit.org/latest/_docs_MO_DG_prepare_model_convert_model_Convert_Model_From_Caffe.html
 - Tensorflow: https://docs.openvinotoolkit.org/latest/_docs_MO_DG_prepare_model_convert_model_Convert_Model_From_TensorFlow.html
 - MxNet: https://docs.openvinotoolkit.org/latest/_docs_MO_DG_prepare_model_convert_model_Convert_Model_From_MxNet.html

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

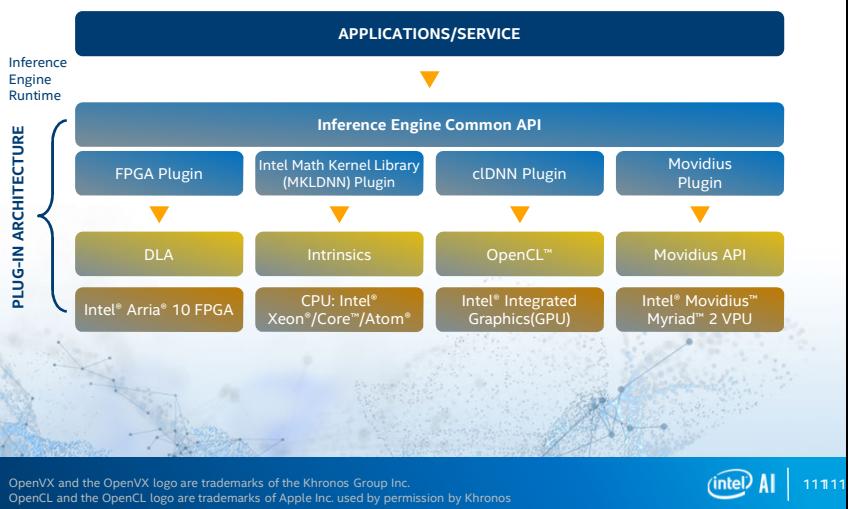


STEP 3 AND 4 INFERENCE ENGINE AND SUPPORT FOR CPU/INTEGRATED GRAPHICS/MOVIDIUS/FPGA

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



OpenVX and the OpenVX logo are trademarks of the Khronos Group Inc.
OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission by Khronos

intel AI | 11111

Heterogeneity - Device affinities.

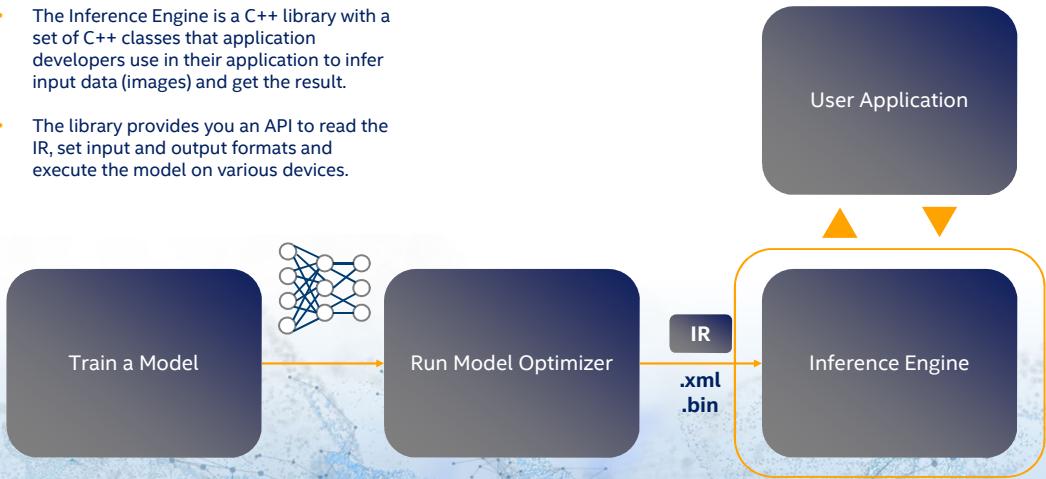
User can specify (example) "HETERO: FPGA,CPU" to fallback to CPU for layers that FPGA does not support.

This can also be used for CPU+GPU cases when User has custom layers implemented on CPU only and wants to execute rest of topology on GPU without obligation to rewrite the custom layer for GPU

Async API usage improves overall frame-rate of the application, allowing to do other things (like next frame decoding), while accelerator is busy with inference of the current frame.

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 – cIDNN Plugin**
 - Supports FP32 and FP16 (recommended for most topologies)
 - Supports Gen9 and above graphics architectures (<https://github.com/01org/cIDNN>)
- **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



113

Layers in mkldnn and cldnn and extension layers



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  
. . . . .
```



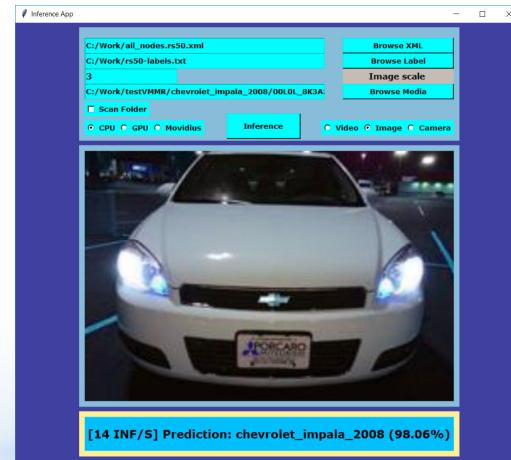
Note: The Model Optimizer does not revert input channels from RGB to BGR by default. Manually specify the command-line parameter to perform this reversion: -
-reverse_input_channels



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.

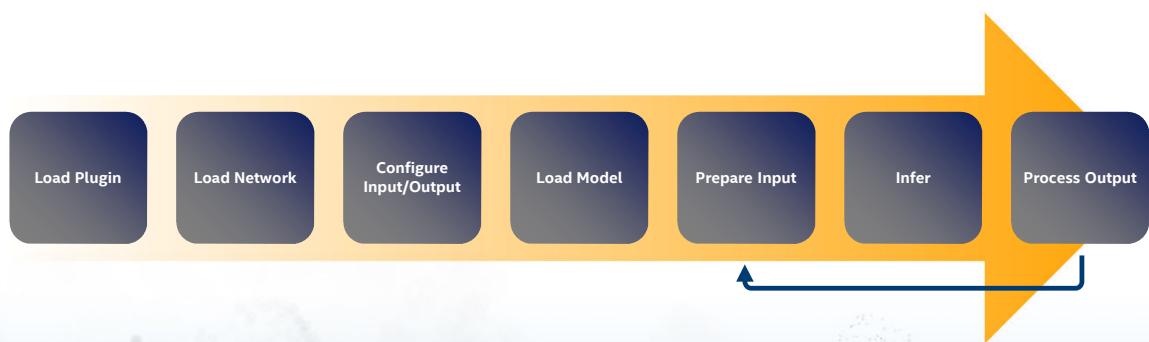
The model is based on the modified Inception_V3 network that was derived from a checkpoint trained for ImageNet with 1000 categories. For purposes of this exercise the model was modified in the last layer to only account for the 10 categories of most stolen cars.

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]
```

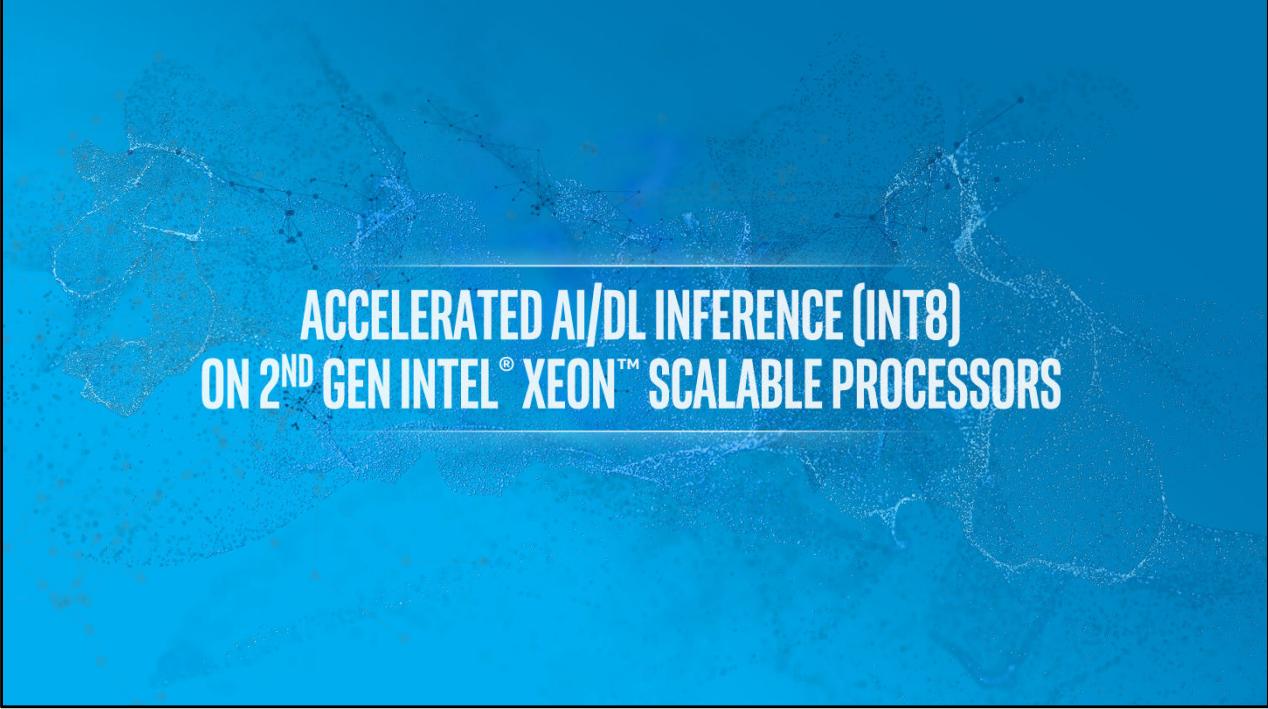
7. Process Output

```
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
throughput¹
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 Cafe ResNet-50 inference throughput performance on Intel® Xeon® Scalable Processor. See Configuration Details 3. Performance results are based on testing, as of 7/17/2019 (11/08/2018) (15.7x), 2/28/2019 (14x) and 2/28/2019 (30x) and may not reflect all publicly available software updates. No product can be absolutely secure. See configuration disclosure for details. Optimized for performance only on Intel microprocessors. Not optimized for non-Intel microprocessors. No product can be absolutely secure. See configuration disclosure for details. Optimized for performance only on Intel microprocessors. Not optimized for non-Intel microprocessors. The optimizations include SSE2, SSE3, SSE4, 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 microprocessors are described in Intel’s publications. Performance tests, such as SPEC CPU® Reference Performance, are measured using specific computer system configurations, components, software, operating systems, 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>



| 126

New 2nd Generation Intel® Xeon® processor scalable family, which is drop-in compatible with the previous Intel® Xeon® Scalable processor platform.

You can use it to:

Achieve the deep learning performance you need thanks to built-in acceleration with Intel DL boost, optimized DL SW frameworks, and the ability to efficiently scale up to hundreds of nodes

Lower TCO/increase utilization by sharing resources between data center and AI workloads, with even more agility thanks to new features like IMT/ADQ/SST

Confidently analyze your sensitive data with hardware-enhanced security including new features like Intel SecL and TDT

And so much more...

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****



intel AI | 127

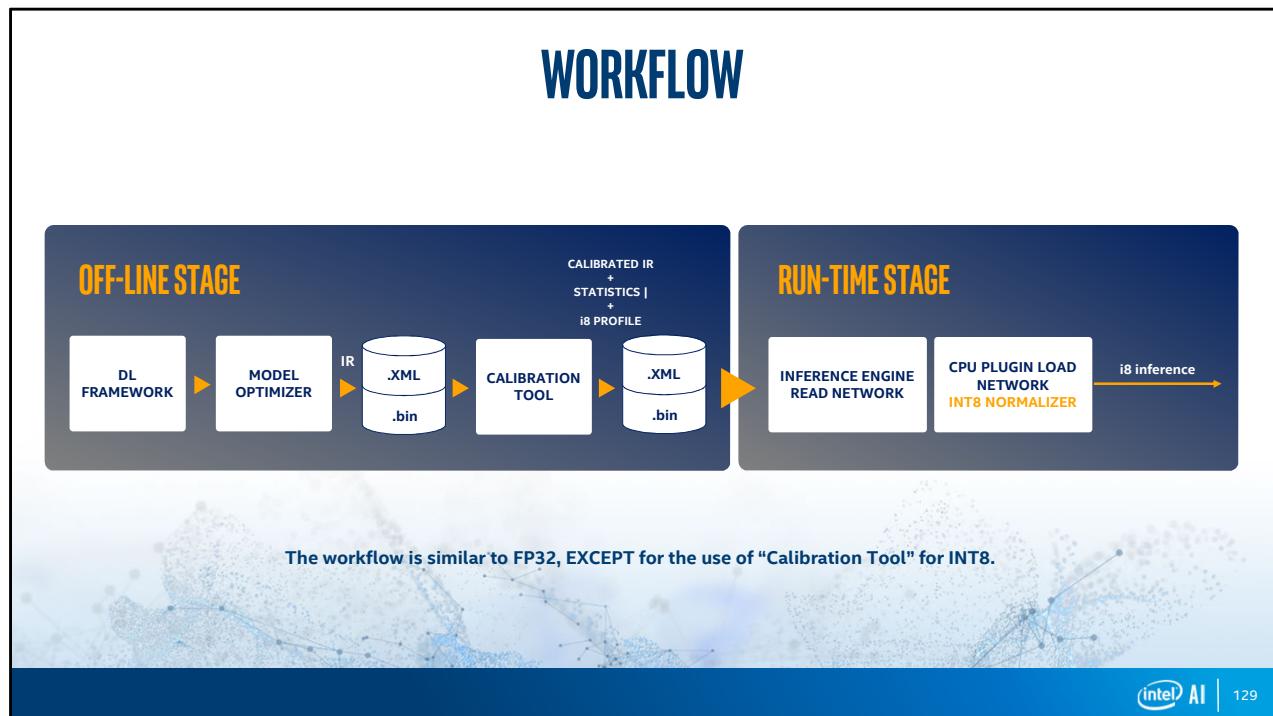
Intel® Deep Learning Boost (VNNI) on the 2nd Generation Intel® Xeon® Scalable processor is designed to deliver significant, more efficient Deep Learning (Inference) acceleration.

- Intel® DL Boost (VNNI) is a new Intel® Advanced Vector Extension (Intel® AVX-512) instruction
- It is a fused multiply-add instruction, which is often used in matrix manipulations as part of deep learning inference
- The new VNNI instruction combines what were three separate instructions into a single processor instruction, saving clock cycles on the processor.
- VNNI can help to speed up image classification, speech recognition, language translation, object detection and more



INT8 INFERENCE USING THE INTEL® DISTRIBUTION OF OPENVINO™ TOOLKIT

WORKFLOW



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** (<https://software.intel.com/en-us/articles/OpenVINO-ModelOptimizer>). This will output the model in Intermediate Representation (IR) format.
- Perform model calibration using the **calibration tool** (http://docs.openvino-toolkit.org/2019_R1/_inference_engine_samples_calibration_tool_README.html) 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.



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:** https://intel.az1.qualtrics.com/jfe/form/SV_9EIVi2JXNF1ViiV





LEARN MORE ABOUT INTEL'S AI OFFERINGS

RESOURCES

- **Intel® Distribution of OpenVINO™ Toolkit**
 - <https://software.intel.com/en-us/openvino-toolkit>
 - **Reinforcement Learning Coach**
 - <https://github.com/NervanaSystems/coach>
 - **NLP Architect**
 - http://nlp_architect.nervanasys.com/
 - **Nauta**
 - <https://www.intel.ai/nauta/>
 - **BigDL**
 - <https://software.intel.com/en-us/ai/frameworks/bigdl>
 - **Intel Optimizations to Caffe***
 - <https://software.intel.com/en-us/ai/frameworks/caffe>
 - **Intel Optimizations to TensorFlow***
 - <https://software.intel.com/en-us/ai/frameworks/tensorflow>
- Learn more through the AI webinar series:**
- <https://software.seek.intel.com/AIWebinarSeries>
- AI Courses:**
- Introduction to AI
 - <https://software.intel.com/en-us/ai/courses/artificial-intelligence>
 - Machine Learning
 - <https://software.intel.com/en-us/ai/courses/machine-learning>
 - Deep Learning
 - <https://software.intel.com/en-us/ai/courses/deep-learning>
 - Applied Deep Learning with Tensorflow*
 - <https://software.intel.com/en-us/ai/courses/tensorflow>

