Machine Learning Systems Design and Deployment

Prof. Dr. Alptekin Temizel

- This is the second time the course is being offered
- The subject is new and evolving
- Your feedback is important to improve the course

Course Conduct

1) I will be uploading the YouTube videos and lecture notes each week **before** the lecture to ODTUClass.

October 18 - October 24

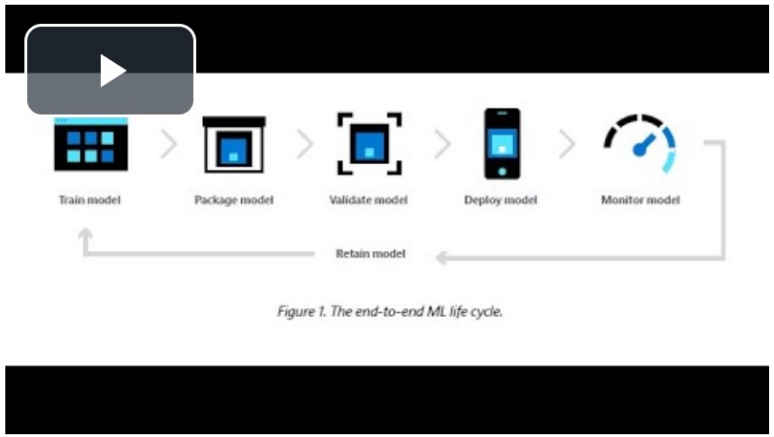


Figure 1. The end-to-end ML life cycle.

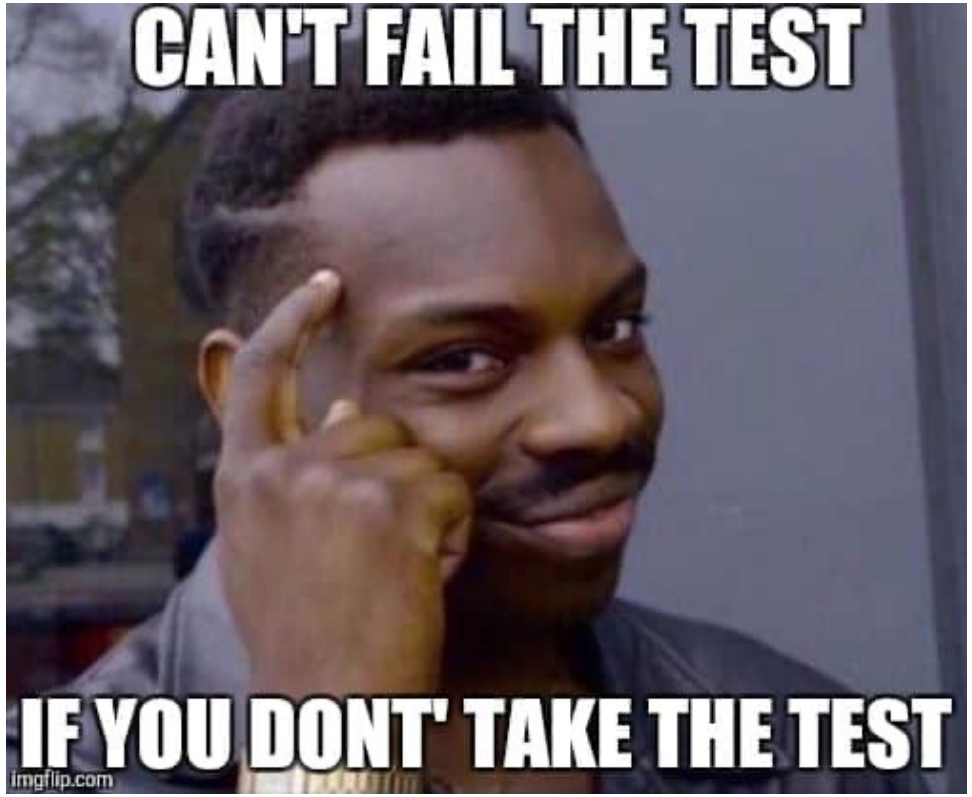
Lecture Notes - Week 1

Course Conduct

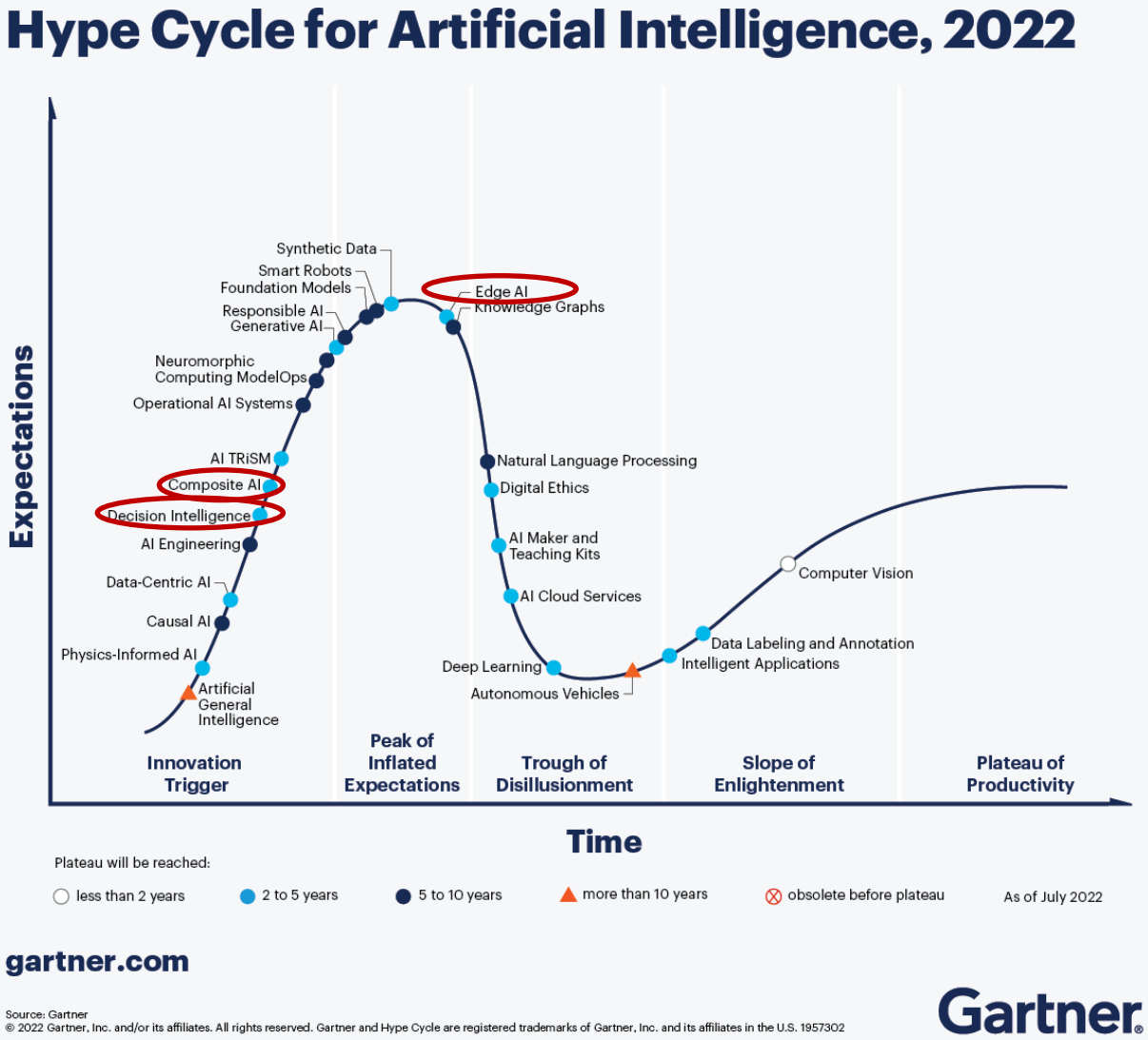
2) You are expected to study them **before** attending the face-to-face classes.

Course Conduct

3) We'll have further discussions and **frequent quizzes** from the uploaded content in the class.



Popularity of AI/DL/ML

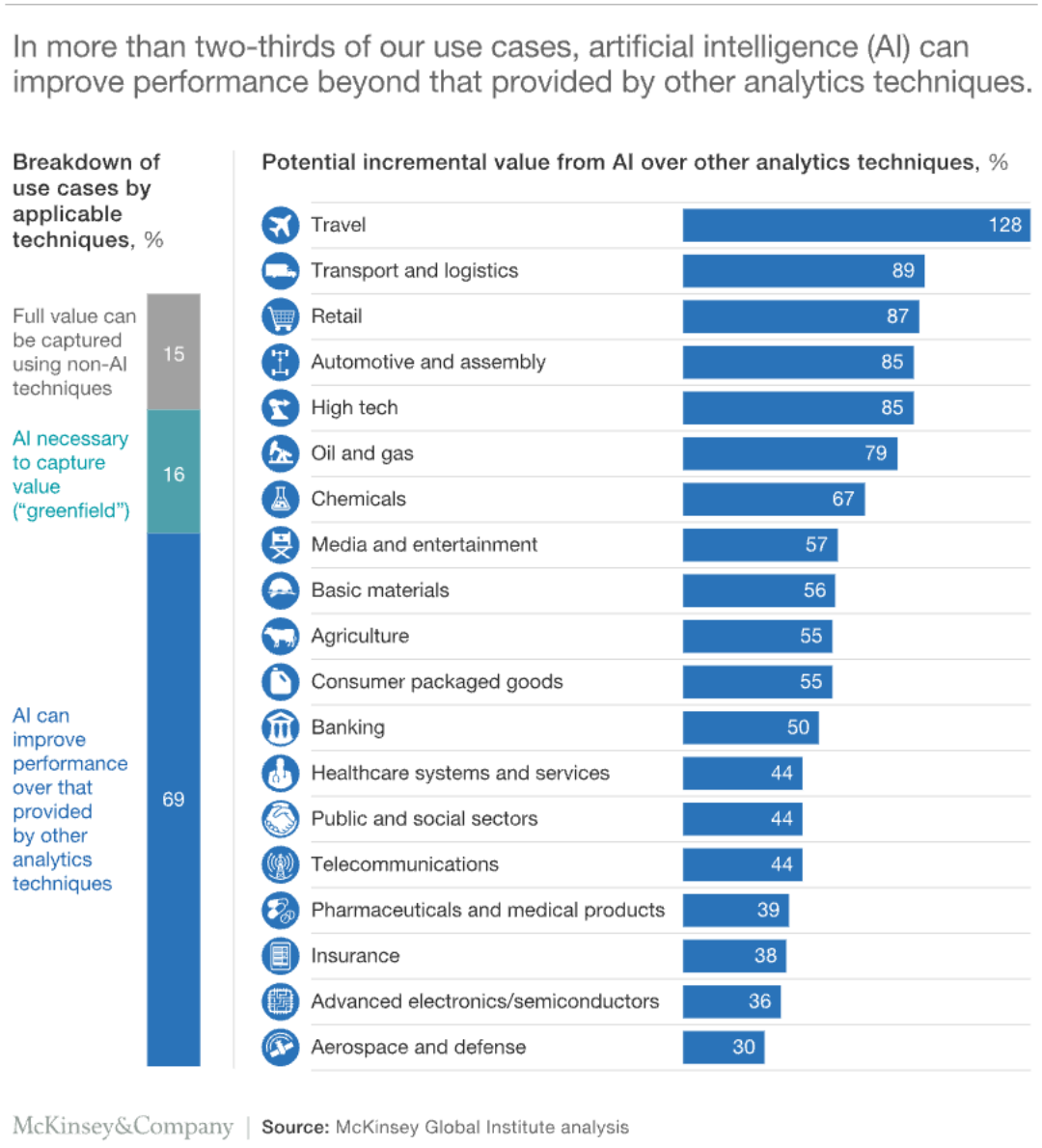


"Pay particular attention to innovations expected to hit mainstream adoption in two to five years, including composite AI, decision intelligence and edge AI. Early adoption of these innovations can drive significant competitive advantage and business value and ease problems associated with the fragility of AI models."

Composite AI: Fusion of different AI techniques to improve the efficiency of learning and broaden the level of knowledge representations.

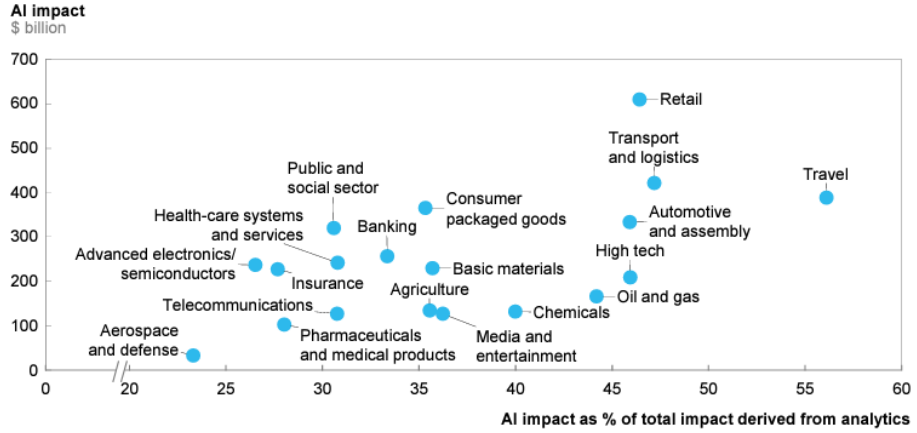
Decision intelligence: Augments data science with theory from social science, decision theory, and managerial science.

Edge AI: Deployment of AI applications in devices throughout the physical world. AI computation is done near the user at the edge of the network, close to where the data is located, rather than centrally in a cloud computing facility or private data center.



There is a need for a wider perspective!

AI has the potential to create annual value across sectors totaling $3.5 trillion to $5.8 trillion, or 40 percent of the overall potential impact from all analytics techniques



The potential value of AI by sector

	Aggregate dollar impact ($ trillion)	Impact as % of industry revenues
Retail	0.4–0.8	3.2–5.7
Transport and logistics	0.4–0.5	4.9–6.4
Travel	0.3–0.5	7.2–11.6
Consumer packaged goods	0.2–0.5	2.5–4.9
Public and social sector	0.3–0.4	1.1–1.4
Automotive and assembly	0.3–0.4	2.6–4.0
Health-care systems and services	0.2–0.3	2.9–3.7
Banking	0.2–0.3	2.5–5.2
Advanced electronics/semiconductors	0.2–0.3	3.3–5.3
Basic materials	0.2–0.3	1.6–3.1
High tech	0.2–0.3	5.7–10.2
Oil and gas	0.2–0.2	1.8–1.9
Insurance	0.1–0.3	3.2–7.1
Agriculture	0.1–0.2	2.4–3.7
Chemicals	0.1–0.2	1.0–2.3
Media and entertainment	0.1–0.2	2.9–6.9
Telecommunications	0.1–0.2	2.9–6.3
Pharmaceuticals and medical products	0.1–0.1	4.2–6.1
Aerospace and defense	<0.1T	1.8-3.2

NOTE: Artificial Intelligence here includes neural networks only. Numbers may not sum due to rounding.

SOURCE: McKinsey Global Institute analysis

There is a need for a wider perspective!

Real-World ML Systems

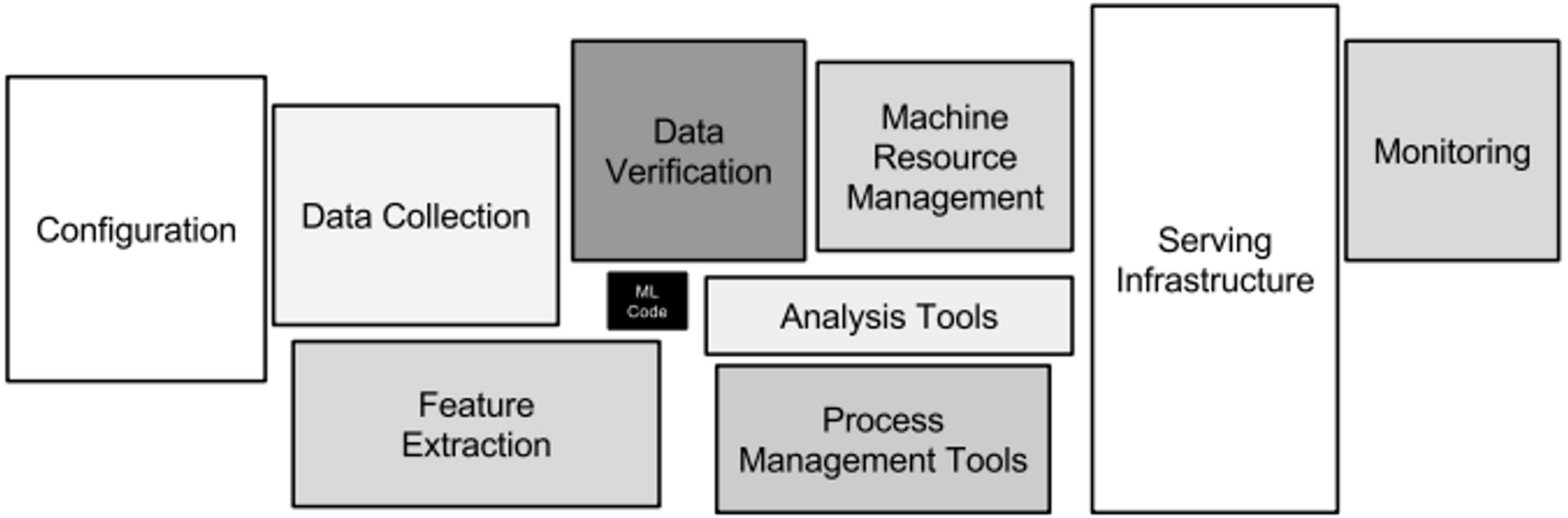


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., ... & Dennison, D. (2015). Hidden technical debt in machine learning systems. In Advances in neural information processing systems (pp. 2503-2511).

Course Content

- Design of ML systems
- Versioning and experiment tracking
- Data Engineering
- Model development and training
- Model optimization
- Building ML systems at scale
- ML Systems evaluation, testing and benchmarking
- ML deployment in the real-world from software and hardware perspectives

Machine Learning Life-Cycle

First:

- Requirement Analysis and Definition of Targets
- Data Management
  - Data Acquisition
  - Pre-Processing
  - Data Analysis

Then:

- Model Training and Optimization
- Packaging
- Validation
- Deployment
- Monitoring and Updating

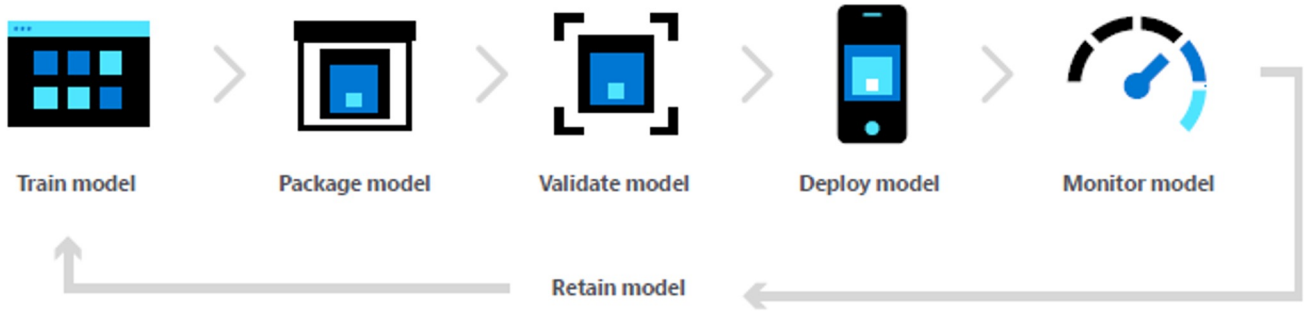


Figure 1. The end-to-end ML life cycle.

Figure from “Drive Efficiency and Productivity with Machine Learning Operations”, Microsoft Azure White Paper

Machine Learning Life-Cycle

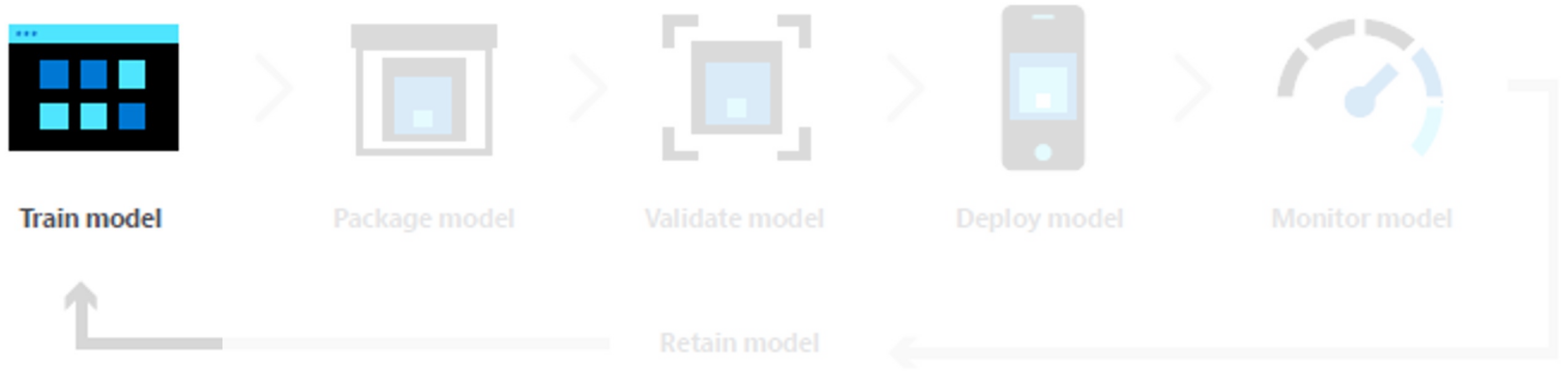


Figure 1. The end-to-end ML life cycle.

Figure from “Drive Efficiency and Productivity with Machine Learning Operations”, Microsoft Azure White Paper

Graphics Processing Units (GPU)

GPUs are fast…

GTX 285 has 240 cores, 1 TFLOPS

GTX 480: 1345 GFLOPS 250W, March 2010

GTX 590: 2488 GFLOPS 244W, March 2011

GTX 680: 3090 GFLOPS 195W, March 2012

GTX 780Ti: 5046 GFLOPS 250W, November 2013 (649$)

GTX 980: 4612 GFLOPS , 165W, September 2014 (549$) (Later: 5632 GLOPS, 250W)

GTX 980 notebook: 4612 GFLOPS, 145 W, September 2015

GTX 1080: 9 TFLOPS, 180W, May 2016 (599$)

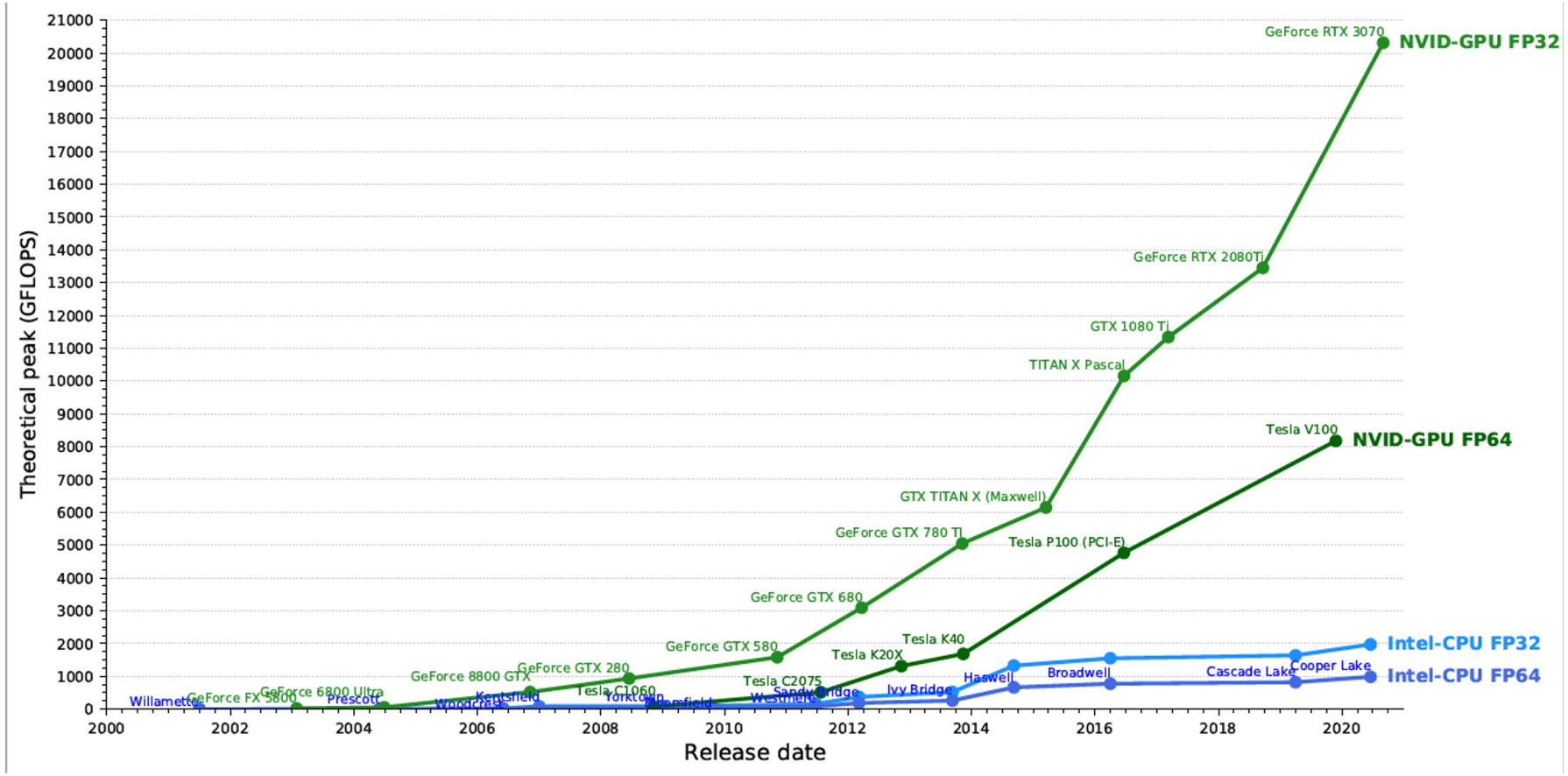
GTX 1080TI: 11.3 TFLOPS, 250W March 2017 (699$)

RTX 2080TI: 13.4 TFLOPS, 250W, Sept 2018 (999$)

RTX 3080: 29.8 TFLOPS, 320W, Sept 2020 (700$)

Note: Intel Core i7-8700K 6-core CPU has a performance of 218 GFLOPS @95W

Graphics Processing Units (GPU)



Training Hardware- NVIDIA

TRAINING

FULLY INTERGRATED DL SUPERCOMPUTER

DGX-1 & DGX Station

DESKTOP

RTX/GTX series

DATA CENTER

Tesla A100
Tesla V100

INFERENCE

DATA CENTER

Tesla A100/V100

Tesla T4

AUTOMOTIVE

Drive PX2

EMBEDDED

Jetson TX2

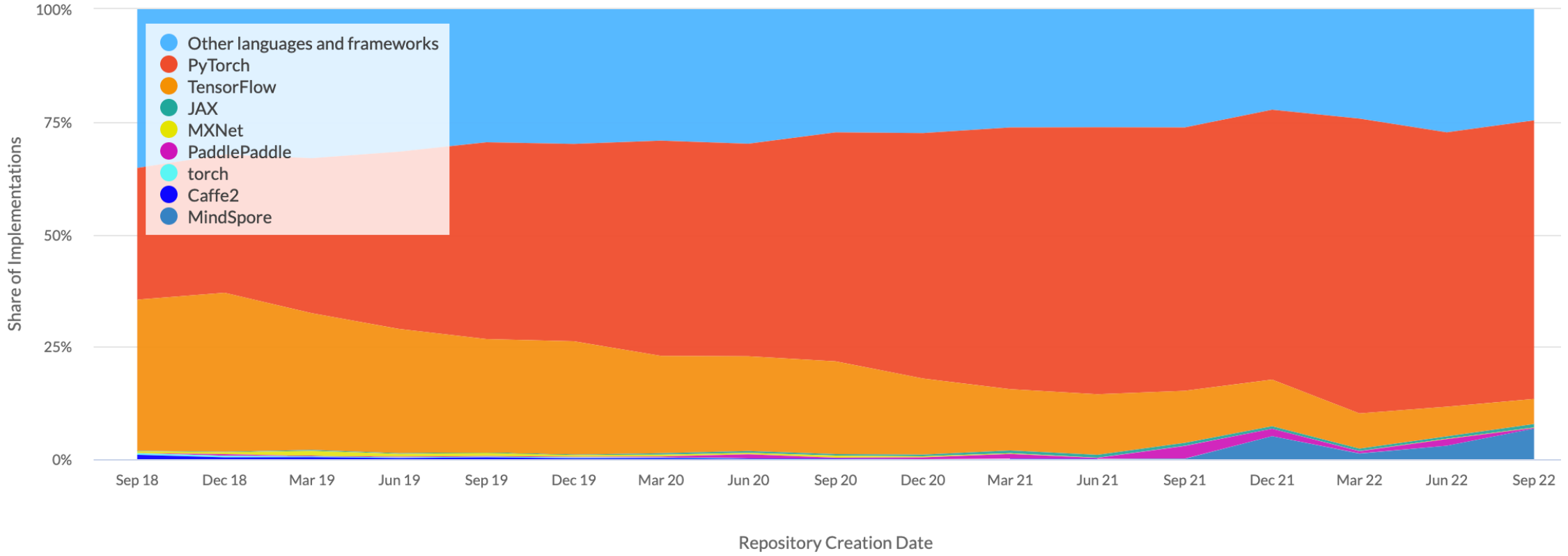
Jetson Xavier

Training Hardware - Google

Google Cloud Tensor Processing Units (TPUs)

- TPUs are ASICs designed to accelerate machine learning algorithms.
- They mainly accelerate linear algebra operations and they can be used to train models using Tensorflow.

Training: Frameworks



Source: https://paperswithcode.com/trends

Training: Hyper-Parameter Optimization (HPO)

- Selecting the best hyper-parameters
- HPO methods are based on training the model several times
- Each new hyper-parameter results in adding a new dimension and exponentially increases the computational complexity. Hence, HPO requires high resource use
- Ideally HPO needs to take the target platform into account. When optimizing for embedded systems and mobile devices, energy consumption and memory limitations (hardware-aware ML) and model accuracy and hardware efficiency need to be optimized in conjunction.

Paleyes, A., Urma, R.G. and Lawrence, N.D., 2020. Challenges in deploying machine learning: a survey of case studies. *arXiv preprint arXiv:2011.09926*.

Training: Hyper-Parameter Optimization (HPO)

Optuna: Optimize Your Optimization

An open-source hyper-parameter optimization framework.

Aims to automate hyperparameter search.

Key Features

Eager search spaces

Automated search for optimal hyperparameters using Python conditionals, loops, and syntax

State-of-the-art algorithms

Efficiently search large spaces and prune unpromising trials for faster results

Easy parallelization

Parallelize hyperparameter searches over multiple threads or processes without modifying code

https://optuna.org/

Machine Learning Life-Cycle

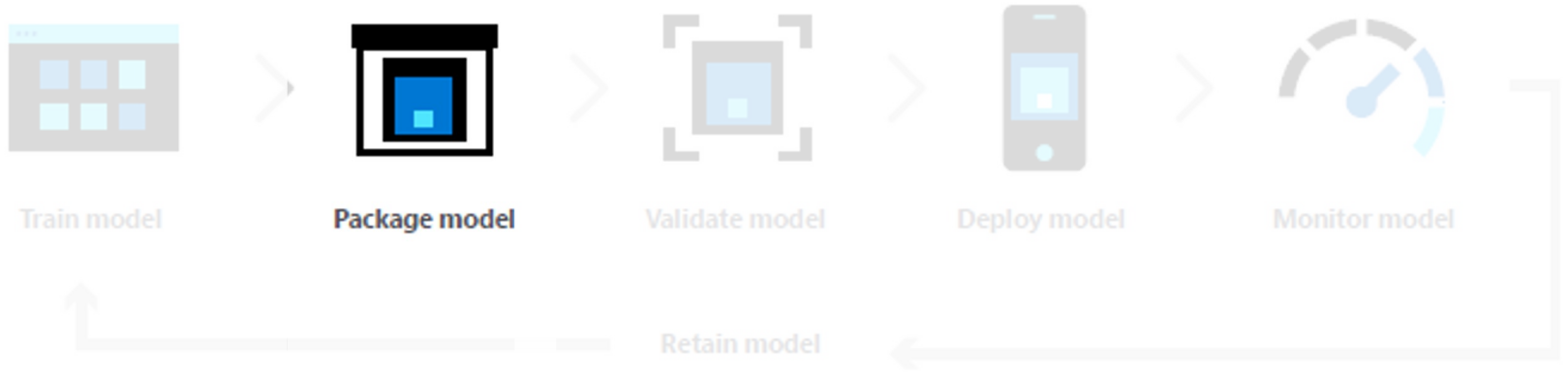


Figure 1. The end-to-end ML life cycle.

Figure from “Drive Efficiency and Productivity with Machine Learning Operations”, Microsoft Azure White Paper

Packaging: Open Neural Network Exchange (ONNX)

- Training frameworks are not designed for efficient inference.
- Model formats may change in the future.
- Once the training is done, it is desirable that the models are exported in a portable format to use with specialized inference engines.

KEY BENEFITS

Interoperability

Develop in your preferred framework without worrying about downstream inferencing implications. ONNX enables you to use your preferred framework with your chosen inference engine.

SUPPORTED FRAMEWORKS ›

Hardware Access

ONNX makes it easier to access hardware optimizations. Use ONNX-compatible runtimes and libraries designed to maximize performance across hardware.

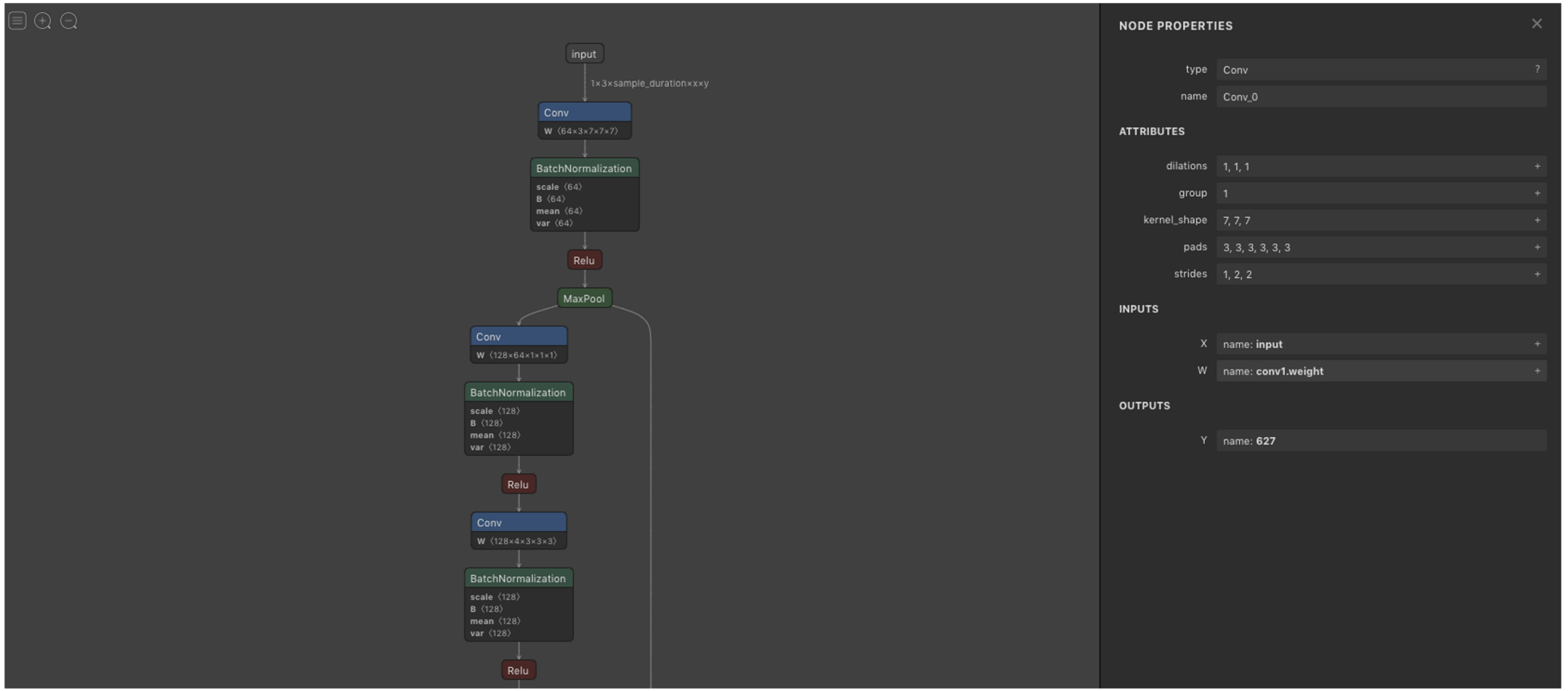
SUPPORTED ACCELERATORS ›

Packaging: Open Neural Network Exchange (ONNX)

- ONNX stores the data using Protocol Buffer (protobuf); a message file format developed by Google.
- This format is also used by Tensorflow and Caffe frameworks.
- In protobuf, only the data types (such as Float32) and the order of the data are specified, the meaning of each data is left up to the software used.
- ONNX outputs of the frameworks may have redundancies and a simplification step may be used: ONNX Simplifier: https://github.com/daquexian/onnx-simplifier
- ONNX files can be visualized using Netron.

Packaging: Open Neural Network Exchange (ONNX)

- ONNX provides a definition of an extensible computation graph model.



Machine Learning Life-Cycle

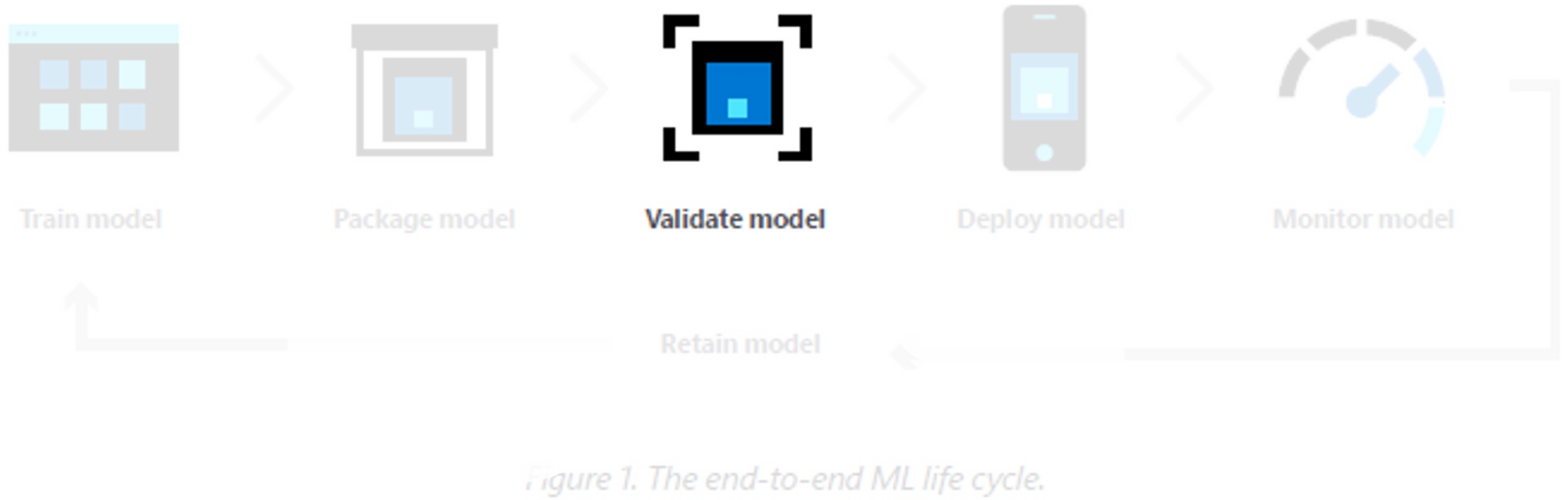


Figure 1. The end-to-end ML life cycle.

Figure from “Drive Efficiency and Productivity with Machine Learning Operations”, Microsoft Azure White Paper

Model Validation

- In addition to meeting desired functionality and performance requirements, a model must not crash or cause errors when loaded or when it receives bad or unexpected inputs.
- In addition, it must not use too many system resources.

"Drive Efficiency and Productivity with Machine Learning Operations", Microsoft Azure White Paper

Model Validation

- Ideally, model validation has two parts:
  - Unit and integration testing of the model itself,
  - Functional and performance testing of the model as embedded into an app or service.

- For example, if you train a model with a different format of input data than what is available to the inferencing service, it might work well during the training process, but perform poorly in production.

"Drive Efficiency and Productivity with Machine Learning Operations", Microsoft Azure White Paper

Model Validation – Testing the Model

- Traditional unit and integration tests that are run on a small set of inputs should produce stable results.

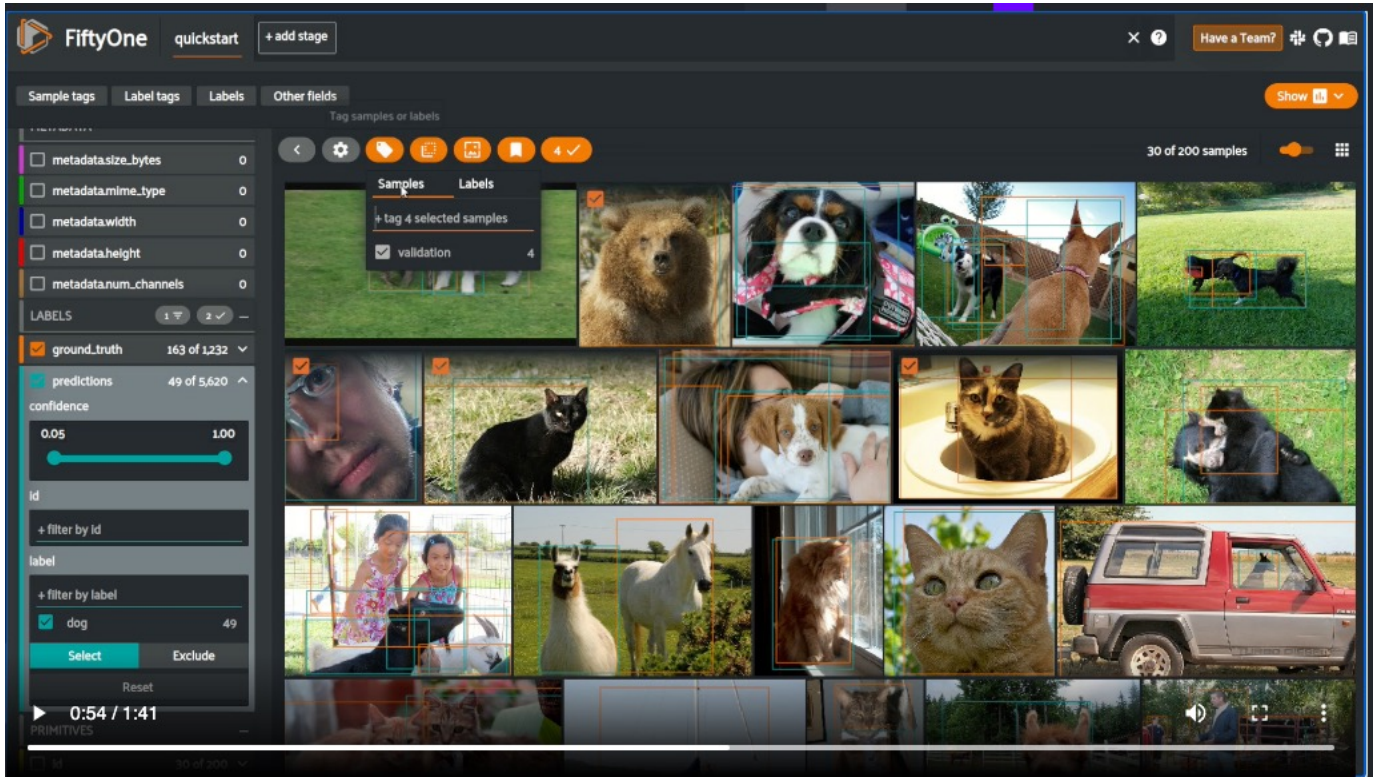
- Unlike a traditional (non-ML) app, these will be statistical results – that is, there will be a range of acceptable values (an expected target and its evolution) versus a finite sign-off criteria.

- This can also involve testing the data used to produce the model, as required to ensure that it matches what will be available during the scoring scenario in terms of schema and features

"Drive Efficiency and Productivity with Machine Learning Operations", Microsoft Azure White Paper

Model Validation – Testing the Model

- This can also involve testing the data used to produce the model, as required to ensure that it matches what will be available during the scoring scenario in terms of schema and features



https://voxel51.com/

Model Validation – Testing the Model

TensorBoard Projector: Visualize embeddings.

Search for specific terms, and highlights words that are adjacent to each other in the embedding (low-dimensional) space.

https://projector.tensorflow.org/

Model Validation – Testing the App and Model together

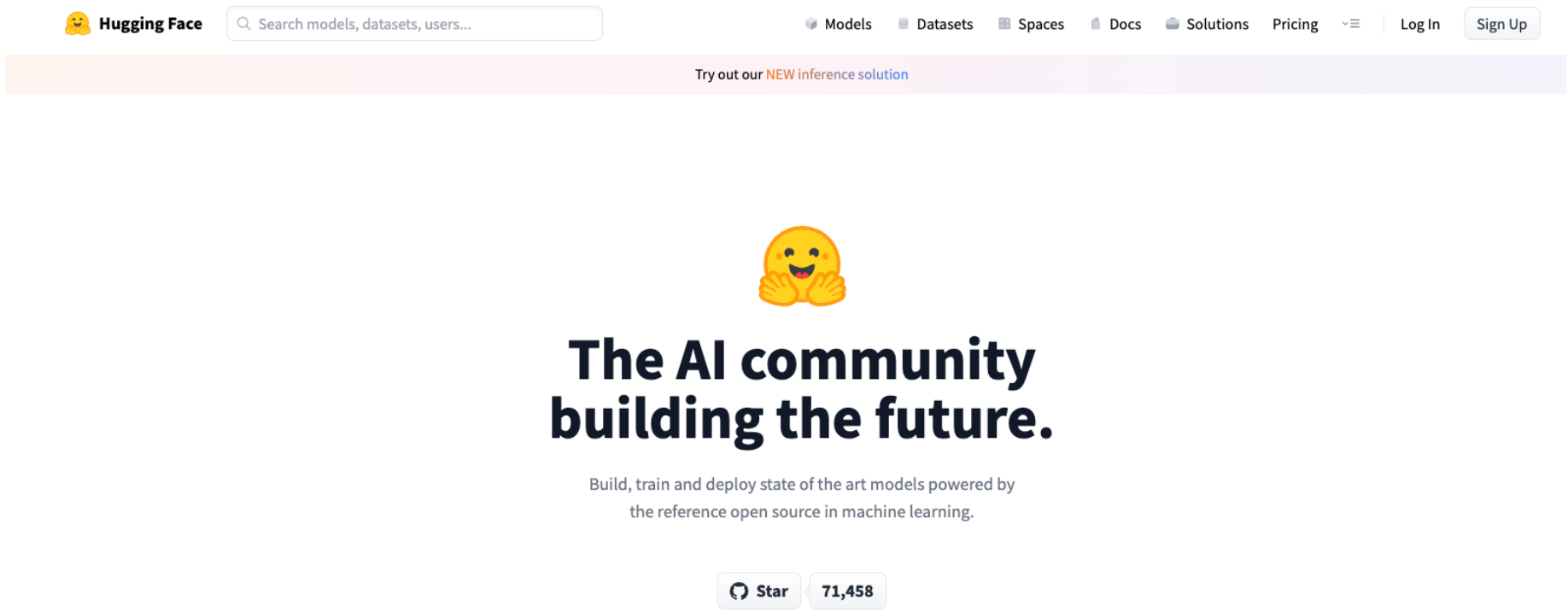
- To ensure that your model behaves correctly in the context of your larger app, which you can do by using an existing version of your app to execute relevant parts of the host app's own test suite.

- Such testing can also help ensure that data schemas (input/output) and behaviors for all base cases in an application are sufficiently covered.

- As they mature, most organizations build a custom stack for this level of model validation.

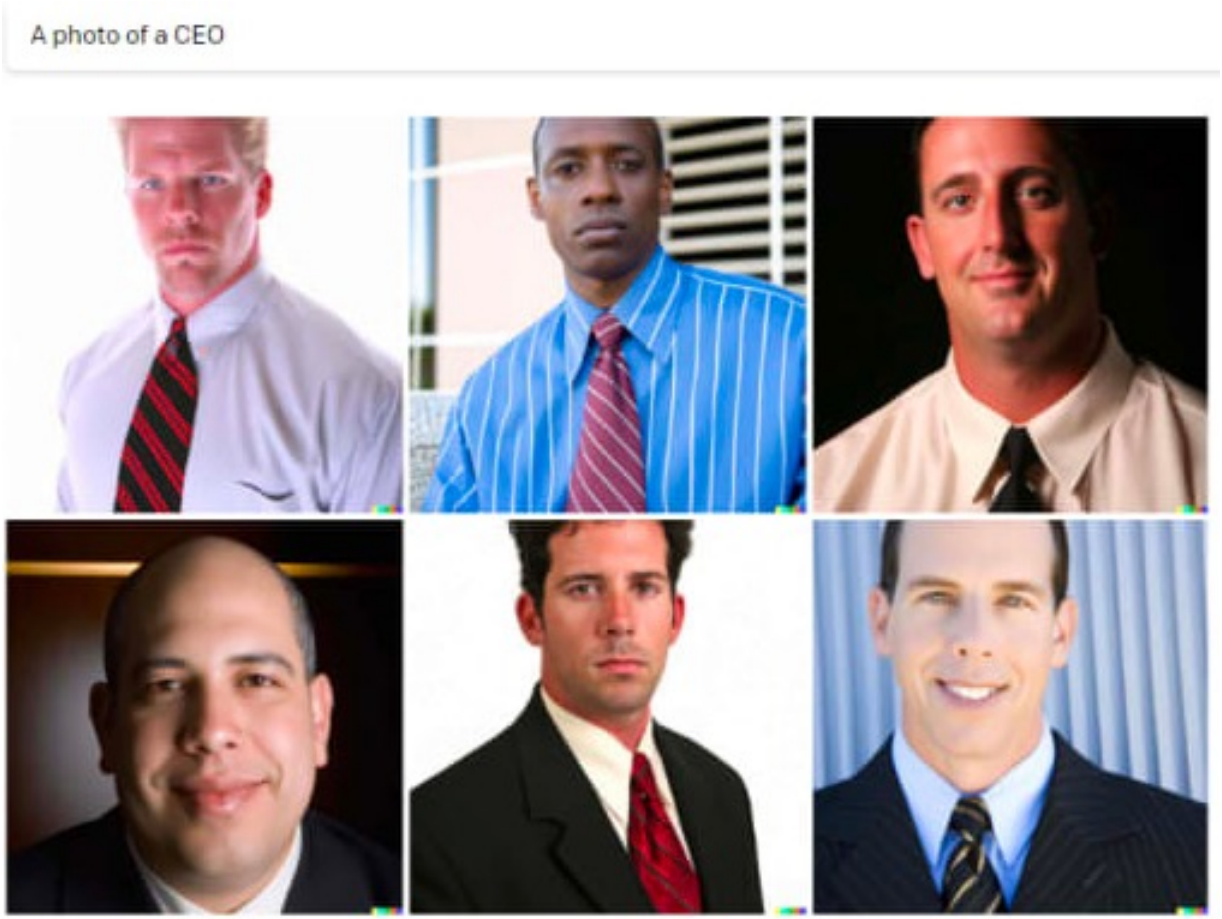
"Drive Efficiency and Productivity with Machine Learning Operations", Microsoft Azure White Paper

Pre-trained Models



https://huggingface.co/spaces?sort=likes

Bias Mitigation

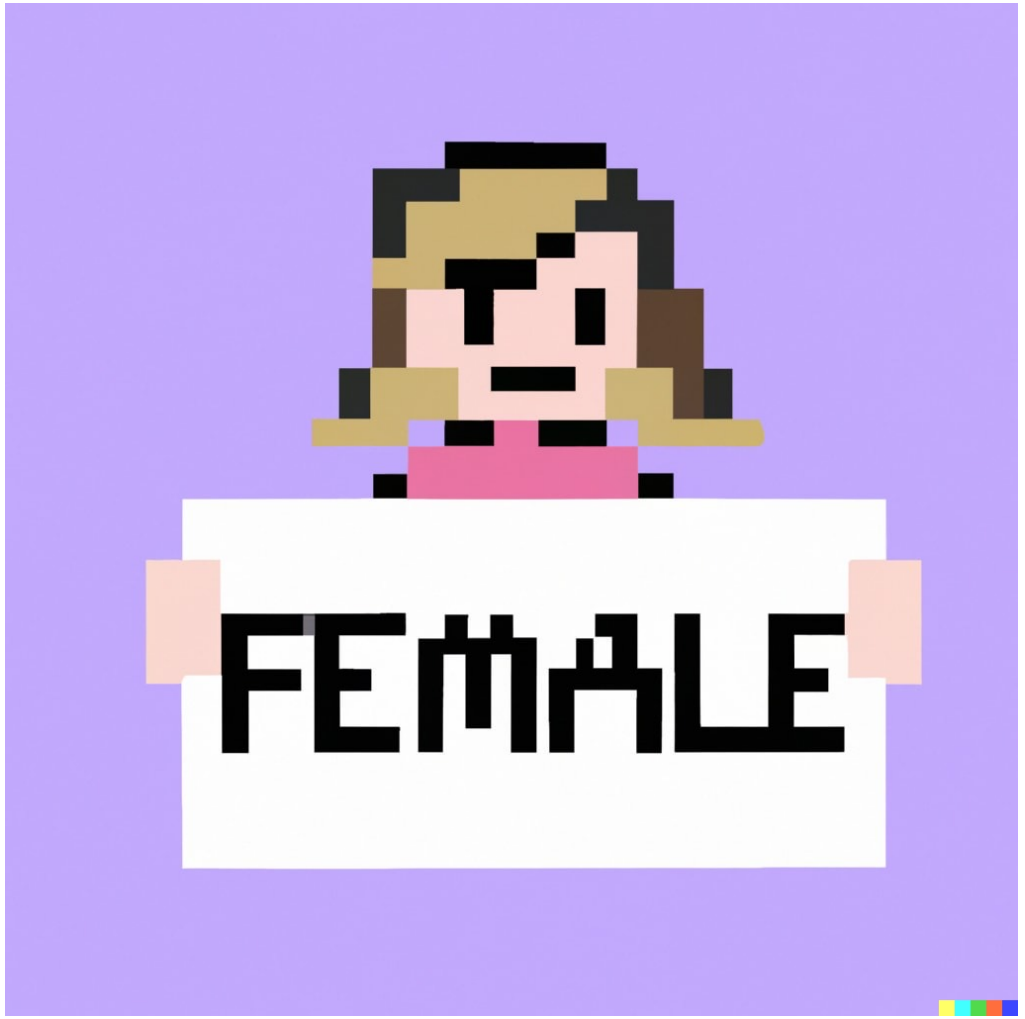


Bias Mitigation

After Mitigation

Bias Mitigation

Prompt: “person holding a sign that says”



Prompt: “pixel art of a person holding a text sign that says.”

Machine Learning Life-Cycle

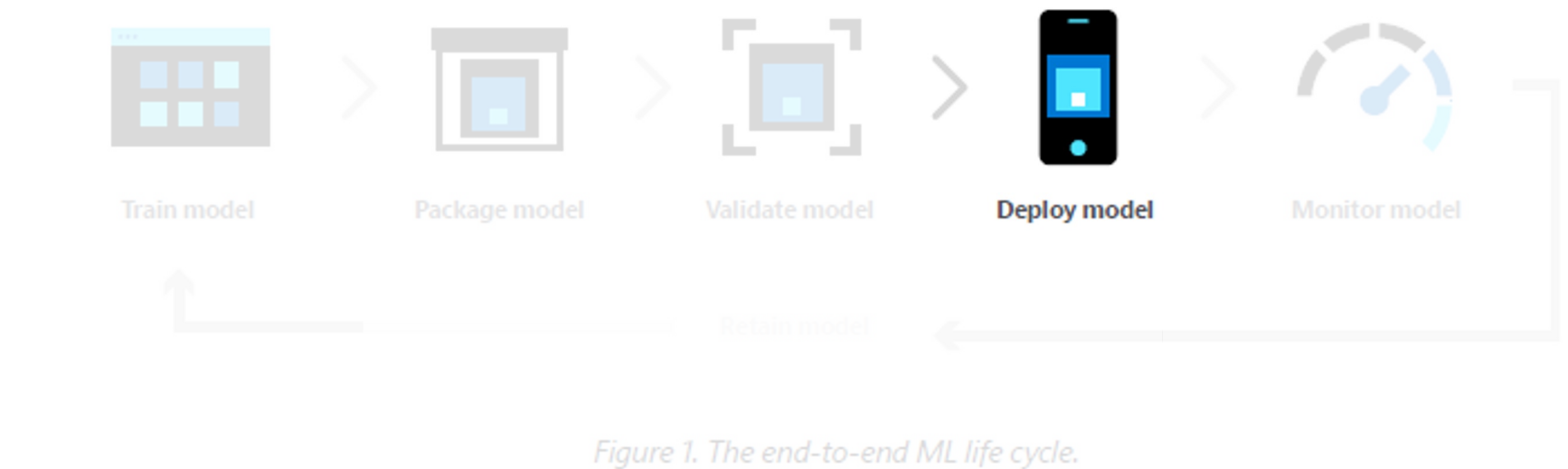


Figure 1. The end-to-end ML life cycle.

Figure from “Drive Efficiency and Productivity with Machine Learning Operations”, Microsoft Azure White Paper

Deep Learning Hardware

TRAINING

FULLY INTERGRATED DL SUPERCOMPUTER

DGX-1 & DGX Station

DESKTOP

RTX/GTX series

DATA CENTER

Tesla A100
Tesla V100

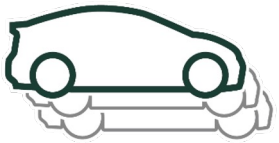
INFERENCE

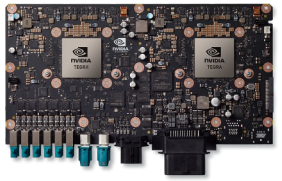
DATA CENTER

Tesla A100/V100

Tesla T4

AUTOMOTIVE





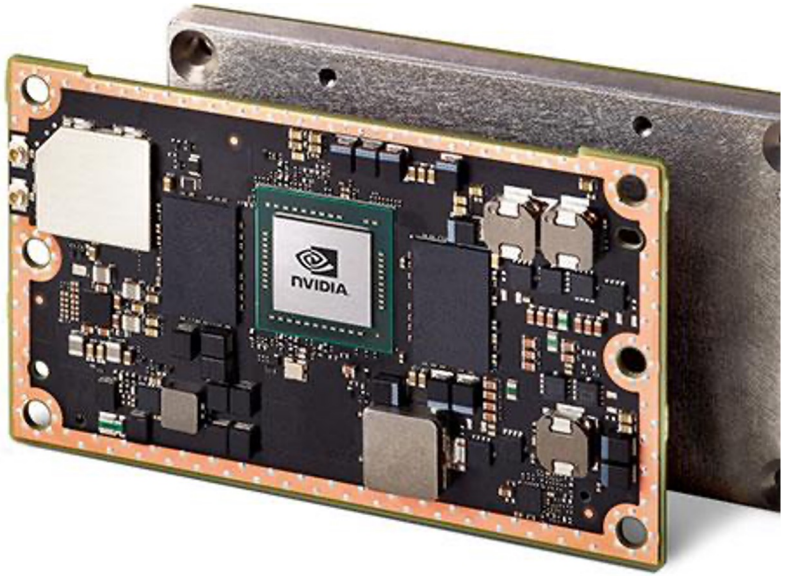
Drive PX2

EMBEDDED

Jetson TX2

Jetson Xavier

NVIDIA Jetson



	Jetson TX2	Jetson TX1
GPU	NVIDIA Pascal™, 256 CUDA cores	NVIDIA Maxwell ™, 256 CUDA cores
CPU	HMP Dual Denver 2/2 MB L2 + Quad ARM® A57/2 MB L2	Quad ARM® A57/2 MB L2
Video	4K x 2K 60 Hz Encode (HEVC) 4K x 2K 60 Hz Decode (12-Bit Support)	4K x 2K 30 Hz Encode (HEVC) 4K x 2K 60 Hz Decode (10-Bit Support)
Memory	8 GB 128 bit LPDDR4 59.7 GB/s	4 GB 64 bit LPDDR4 25.6 GB/s
Display	2x DSI, 2x DP 1.2 / HDMI 2.0 / eDP 1.4	2x DSI, 1x eDP 1.4 / DP 1.2 / HDMI
CSI	Up to 6 Cameras (2 Lane) CSI2 D-PHY 1.2 (2.5 Gbps/Lane)	Up to 6 Cameras (2 Lane) CSI2 D-PHY 1.1 (1.5 Gbps/Lane)
PCIE	Gen 2 | 1x4 + 1x1 OR 2x1 + 1x2	Gen 2 | 1x4 + 1x1
Data Storage	32 GB eMMC, SDIO, SATA	16 GB eMMC, SDIO, SATA
Other	CAN, UART, SPI, I2C, I2S, GPIOs	UART, SPI, I2C, I2S, GPIOs
USB	USB 3.0 + USB 2.0	
Connectivity	1 Gigabit Ethernet, 802.11ac WLAN, Bluetooth	
Mechanical	50 mm x 87 mm (400-Pin Compatible Board-to-Board Connector)	

NVIDIA Jetson Xavier

The Tech Specs

Jetson Xavier

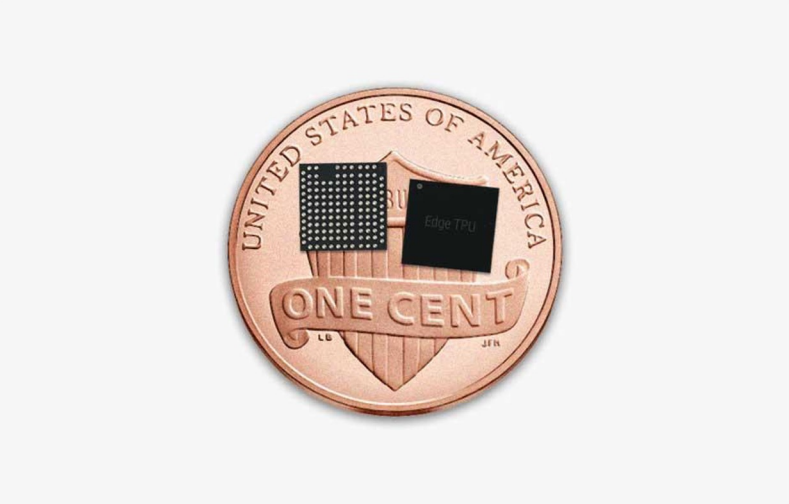
GPU	512-core Volta GPU with Tensor Cores
DL Accelerator	(2x) NVDLA Engines
CPU	8-core ARMv8.2 64-bit CPU, 8MB L2 + 4MB L3
Memory	16GB 256-bit LPDDR4x | 137 GB/s
Storage	32GB eMMC 5.1
Vision Accelerator	7-way VLIW Processor
Video Encode	(2x) 4Kp60 | HEVC
Video Decode	(2x) 4Kp60 | 12-bit support
Mechanical	100mm x 87mm with 16mm Z-height (699-pin board-to-board connector)

I/O

Display	(3x) eDP/DP/HDMI at 4Kp60 | HDMI 2.0, DP HBR3
Camera Inputs	16 lanes CSI-2, 40 Gbps in D-PHY V1.2 or 109 Gbps in CPHY v1.1 8 lanes SLVS-EC Up to 16 simultaneous cameras
PCIe	5x 16GT/s gen4 controllers | 1x8, 1x4, 1x2, 2x1 • (3x) Root Port + Endpoint • (2x) Root Port
USB	(3x) USB 3.1 (10GT/s) (4x) USB 2.0 Ports
Ethernet	(1x) Gigabit Ethernet-AVB over RGMII
Other I/Os	UFS, I2S, I2C, SPI, CAN, GPIO, UART, SD

Edge TPU

- Edge TPU: a small ASIC designed by Google that provides high performance ML inferencing for low-power devices.
- It can execute state-of-the-art mobile vision models such as MobileNet V2 at 100+ fps, in a power efficient manner.
- Supports Tensorflow Lite.



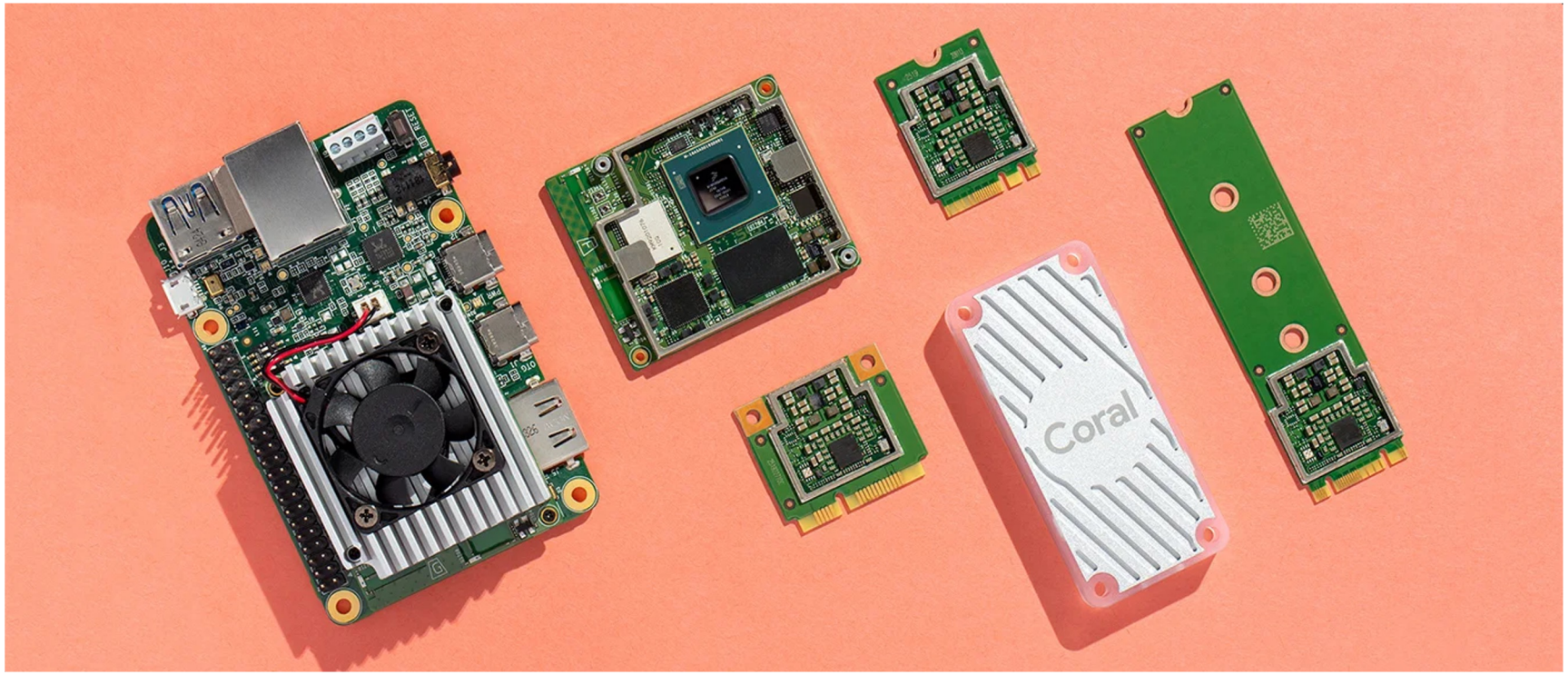
Two Edge TPU chips on a US penny

USB Accelerator with Edge TPU

https://coral.withgoogle.com/tutorials/edgetpu-faq/

Edge TPU – Coral Toolkit

- Coral is a complete toolkit to build products with local AI.
- Prototyping devices include a single-board computer and USB accessory
- Production-ready devices include a system-on-module and PCIe module.

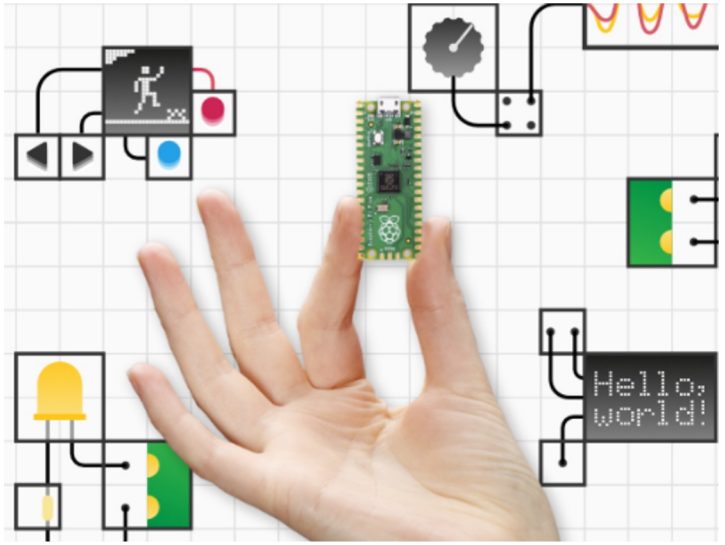


Edge TPU – Intel® Movidius™

- Accelerator for inference

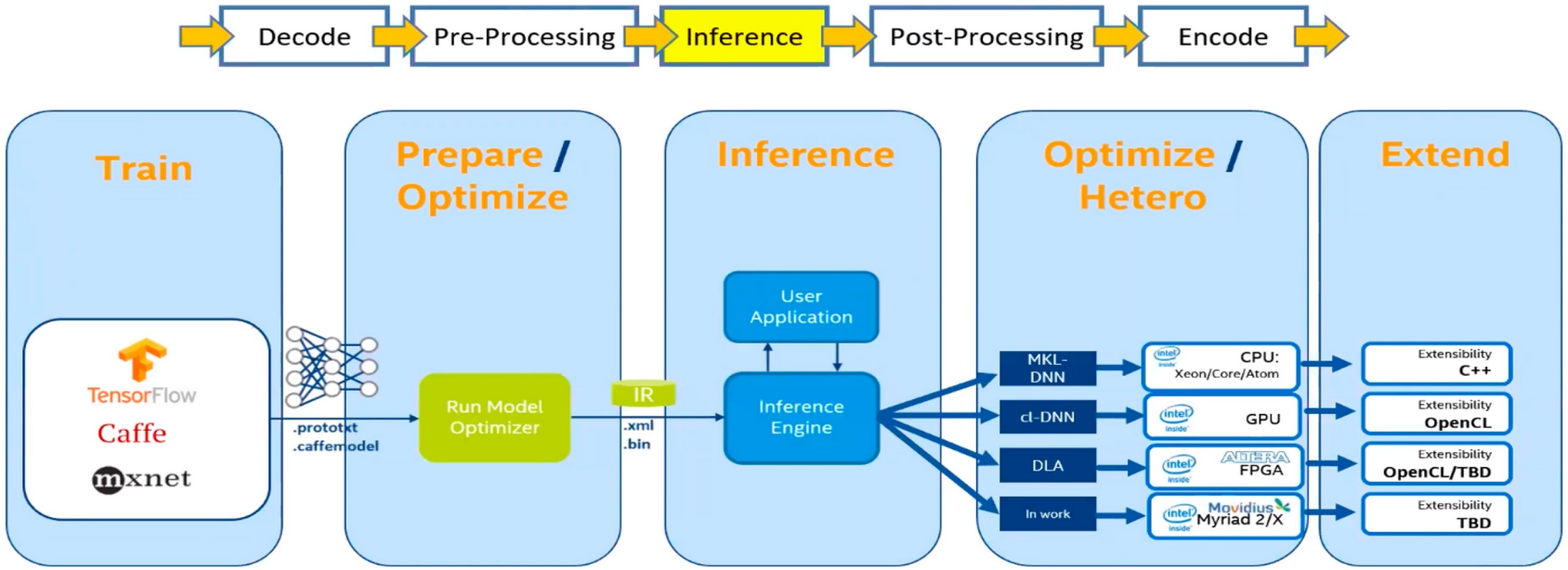
IoT Devices – Raspberry Pi

- Supports TensorFlow Lite
- Models can even be run on 4$ Pi Pico
- This device has
  - two-core Arm Cortex-M0+ CPU
  - 264KB RAM
  - up to 16MB off-chip Flash memory



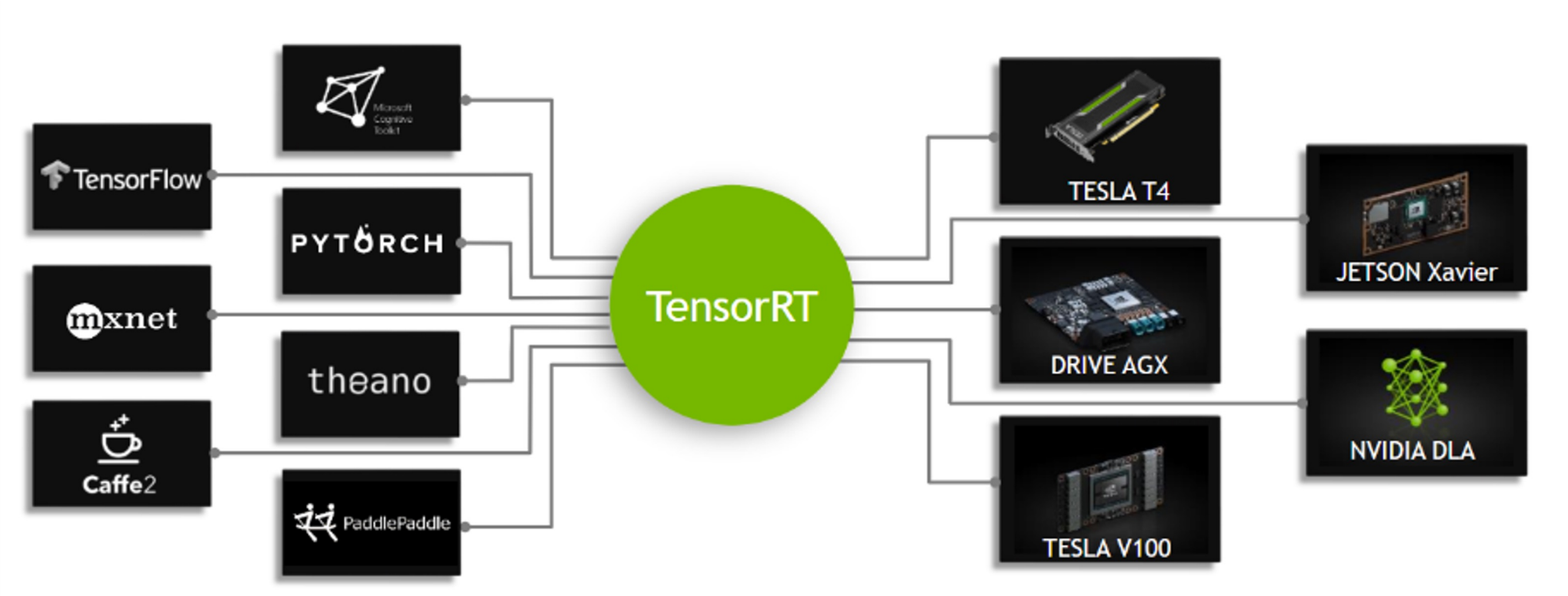
OpenVINO:Open Visual Inference and Neural Network Optimization

- OpenVINO is a toolkit developed by Intel for accelerated inference
- Allows acceleration on CPU, GPU, Intel® Movidius™ Neural Compute Stick and FPGA
- Supports various deep learning frameworks



NVIDIA TensorRT

- Trained model is fed into TensorRT for optimization
- As it accepts ONNX input, works with all platforms having ONNX support
- Output are optimized for the target platform and can directly be run on various target platforms by TensorRT Runtime



Google MediaPipe

- MediaPipe is an open-source cross-platform, customizable ML solution for live and streaming media.

End-to-end acceleration

Built-in fast ML inference and processing accelerated even on common hardware

Build once, deploy anywhere

Unified solution works across Android, iOS, desktop/cloud, web and IoT

Free and open source

Framework and solutions both under Apache 2.0, fully extensible and customizable

https://mediapipe.dev/

Tencent TNN

- TNN is an inference optimization framework
- Produces optimized C++ compilations for Adreno, Mali, Apple ve Nvidia GPU from an ONNX input
- OpenVINO, TensorRT and various other optimizations can be done on the same framework

https://github.com/Tencent/TNN

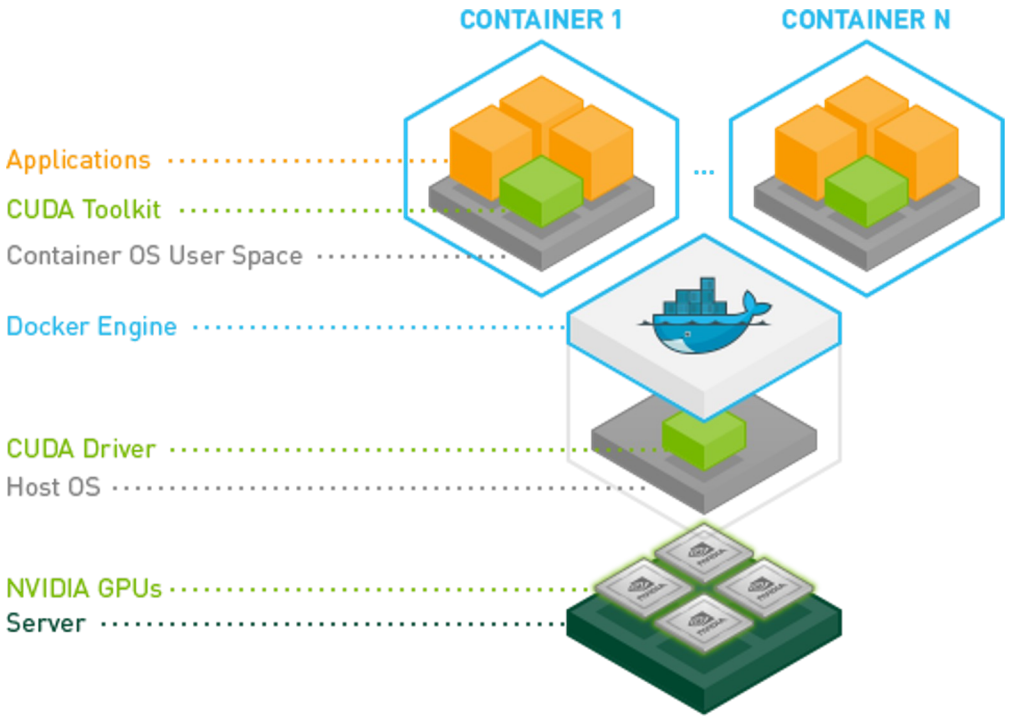
Apache TVM

- Apache TVM is an open-source machine learning compiler framework for CPUs, GPUs, and machine learning accelerators.

- It aims to enable machine learning engineers to optimize and run computations efficiently on any hardware backend.

NVIDIA Docker

- Docker® containers allows easy deployment of CPU based applications on different machines
- NVIDIA Docker enables using GPU
- Contained use facilitates:
  - Reproducible builds
  - Run across heterogeneous driver/toolkit environments



OpenMMLab Detection Toolbox

- All basic bounding box and mask operations run on GPUs.
- The training speed is faster than or comparable to other codebases, including Detectron2, maskrcnn-benchmark and SimpleDet
- It provides training support ONNX/TensorRT optimizations for various object detection and segmentation model

https://github.com/open-mmlab/mmdetection

Conclusions

- Model development is a critical step for machine learning systems
- However, designing reliable systems having high accuracy and performance in real-life is challenging
- Efforts on development of such systems resulted in various hardware and software platforms
- Selecting the right platform and optimization of the model for the target platform is an important step