

Easily speed up Deep Learning inference – Write once deploy anywhere!

Vladimir Kilyazov

AI Software Solutions Engineer

Notices and Disclaimers

- Performance varies by use, configuration and other factors. Learn more at www.Intel.com/PerformanceIndex
- Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.
- You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.
- The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.
- No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document, with the sole exception that a) you may publish an unmodified copy and b) code included in this document is licensed subject to the Zero-Clause BSD open source license (OBSD), <https://opensource.org/licenses/OBSD>. You may create software implementations based on this document and in compliance with the foregoing that are intended to execute on the Intel product(s) referenced in this document. No rights are granted to create modifications or derivatives of this document.
- No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document, with the sole exception that code included in this document is licensed subject to the Zero-Clause BSD open source license (OBSD), <http://opensource.org/licenses/OBSD>.
- No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.
- Code names are used by Intel to identify products, technologies, or services that are in development and not publicly available. These are not "commercial" names and not intended to function as trademarks.
- © Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.

System configuration

System board	Intel prototype, TGL U DDR4 SODIMM RVP	ASUSTek COMPUTER INC./Prime z370-a
CPU	11th Gen Intel® Core™ i5-1145G7 @ 2.6 GHz	8th Gen Intel® Core™ i5-8500t @ 3.0 GHz
Sockets/physical cores	1/4	1/6
Hyperthreading/turbo setting	Enabled/On	NA/On
Memory	2 x 8198 MB 3200 MT/s DDR4	2 x 16384 MB 2667 MT/s DDR4
OS	Ubuntu 18.04 LTS	Ubuntu 18.04 LTS
Kernel	5.8.0-050800-generic	5.3.0-24-generic
Software	Intel® Distribution of OpenVINO™ toolkit 2021.1.075	Intel Distribution of OpenVINO toolkit 2021.1.075
BIOS	Intel TGLIFUI1.R00.3243.A04.2006302148	AMI, version 2401
BIOS release date	June 30, 2020	July 12, 2019
BIOS setting	Load default settings	Load default settings, set XMP to 2667
Test date	September 9, 2020	September 9, 2020
Precision and batch size	CPU: int8, GPU: FP16-int8, batch size: 1	CPU: int8, GPU: FP16-int8, batch size: 1
Number of inference requests	4	6
Number of execution streams	4	6
Power (TDP link)	<u>28W</u>	<u>35W</u>
Price (USD) link on 02/25/2022 Prices may vary	<u>USD 312</u>	<u>USD 192</u>

1) Memory is installed such that all primary memory slots are populated.

2) Testing by Intel as of September 9, 2020.

Compounding effect of hardware and software configuration

[See the compounding effect](#)

System board	1) Purley E63448-400, Intel® Internal Reference System	2) Intel® Server Board S2600STB	3) Intel Server Board S2600STB	4) Intel® Internal Reference System
CPU	Intel® Xeon® Silver 4116 @ 2.1 GHz	Intel® Xeon® Silver 4216 CPU @ 2.10 GHz	Intel® Xeon® Silver 4216R CPU @ 2.20 GHz	Intel® Xeon® Silver 4316 CPU @ 2.30 GHz
Sockets, physical cores/socket	2, 12	2, 16	2, 16	2, 20
Hyperthreading/turbo setting	Enabled/On	Enabled/On	Enabled/On	Enabled/On
Memory	12x 16 GB DDR4 2400 MHz	12x 64 GB DDR4 2400 MHz	12x 32GB DDR4 2666 MHz	16 x32GB DDR4 2666 MHz
OS	UB-16.04.3 LTS	UB-18.04 LTS	UB-18.04 LTS	UB-20.04 LTS
Kernel	4.4.0-210-generic	4.15.0-96-generic	5.3.0-24-generic	5.13.0-rc5-intel-next+
Software	Intel® Distribution of OpenVINO™ toolkit R5 2018	Intel® Distribution of OpenVINO™ toolkit R3 2019	Intel® Distribution of OpenVINO™ toolkit 2021.2	Intel® Distribution of OpenVINO™ toolkit 2021.4.1
BIOS	PLYXCRB1.86B.0616.D08.2109180410	—	Intel Corporation SE5C620.86B.02.01. 0009.092820190230	WLYDCRB1.SYS.0020.P93.2103190412
BIOS release date	September 18, 2021	—	September 28, 2019	March 19, 2021
BIOS setting	Select optimized default settings, save, and exit	Select optimized default settings, save, and exit	Select optimized default settings, change power policy to "performance," save, and exit	Select optimized default settings, change power policy to "performance," save, and exit
Test date	October 8, 2021	September 27, 2019	December 24, 2020	September 6, 2021
Precision and batch size	FP32/Batch 1	int8/Batch 1	int8/Batch 1	int8/Batch 1
Workload: model/image size	MobileNet-SSD/300x300	MobileNet-SSD/300x300	MobileNet-SSD/300x300	MobileNet-SSD/300x300
Number of inference requests	24	32	32	10
Number of execution streams	24	32	32	10
Power (TDP link)	<u>170W</u>	<u>200W</u>	<u>250W</u>	<u>300W</u>
Price (USD) link on 02/25/2022 Prices may vary	<u>USD 2,024</u>	<u>USD 1,926</u>	<u>USD 2,004</u>	<u>USD 2,166</u>

AI is changing industries

■ Emergency response



Real-time emergency and crime response

■ Energy



Maximize production and uptime

■ Education



Transform the learning experience

■ Cities



Enhance safety, research, and more

■ Finance



Turn data into valuable intelligence

■ Health



Revolutionize patient outcomes

■ Industrial



Empower truly intelligent Industry 4.0

■ Media



Create thrilling experiences

■ Retail



Transform stores and inventory

■ Smart homes



Enable homes that see, hear, and respond

■ Telecom



Drive network and operational efficiency

■ Smart cities

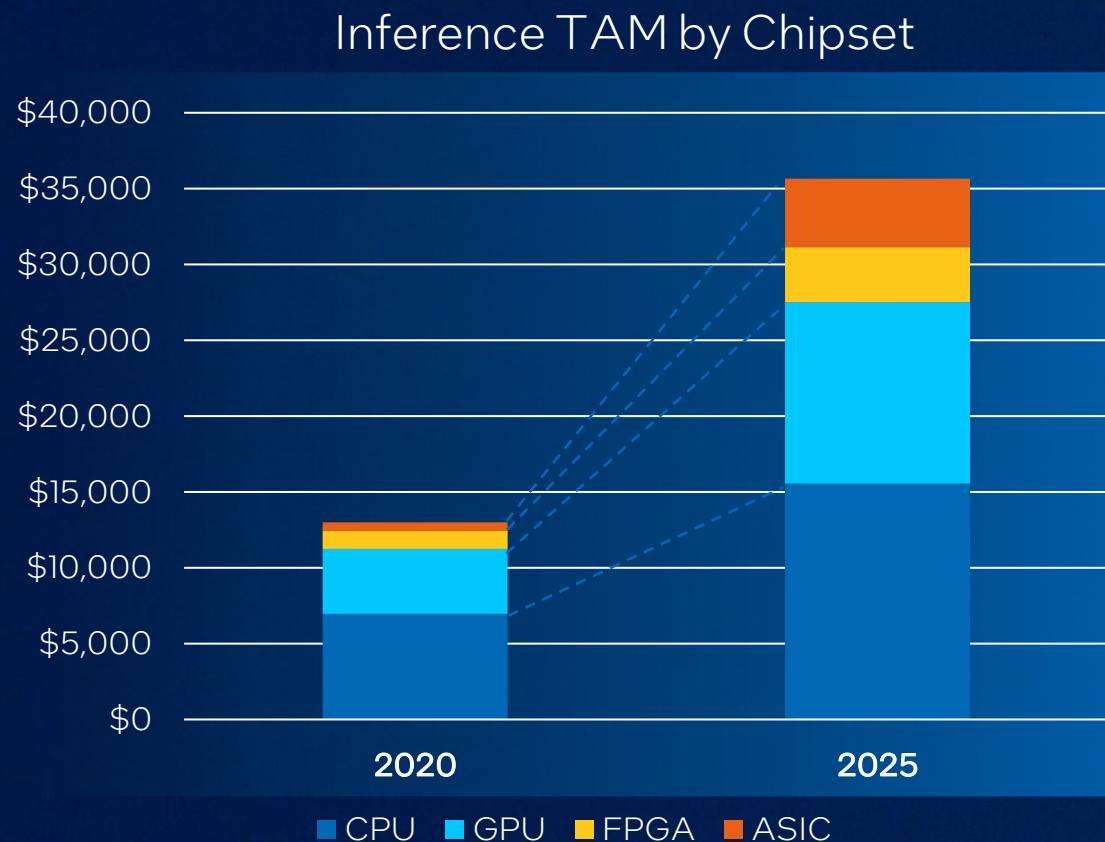


Efficient and robust traffic systems

Why deep learning

There is tremendous opportunity in the global growth in chipset revenue driven by AI inference

\$51.9B
by 2025

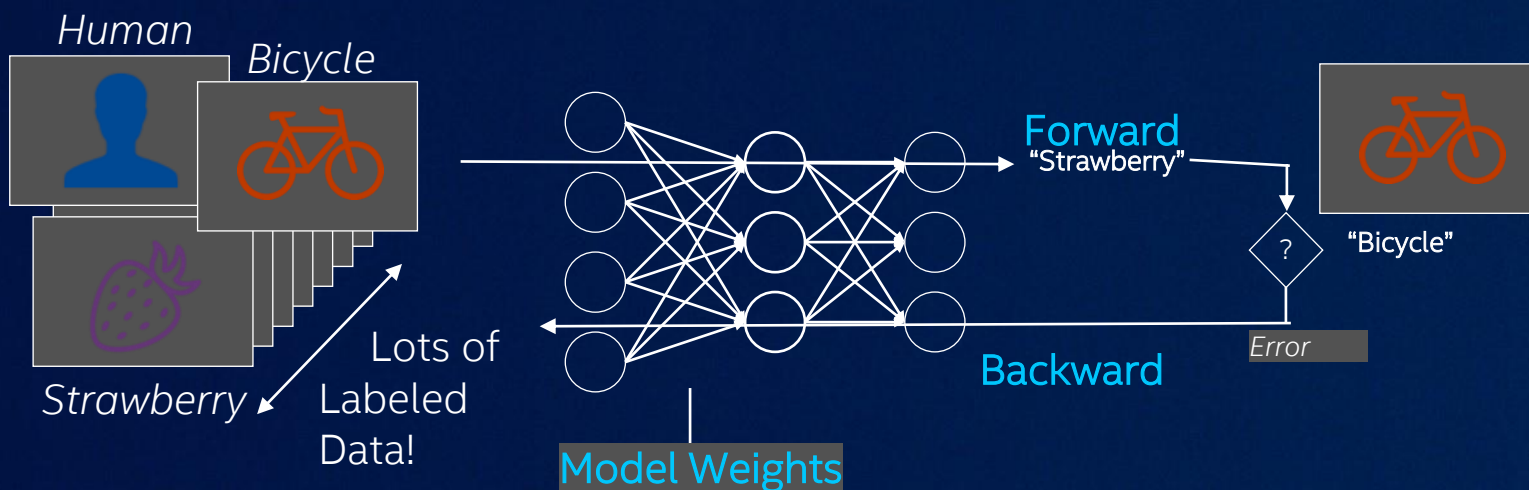


**Driven by
AI inference**

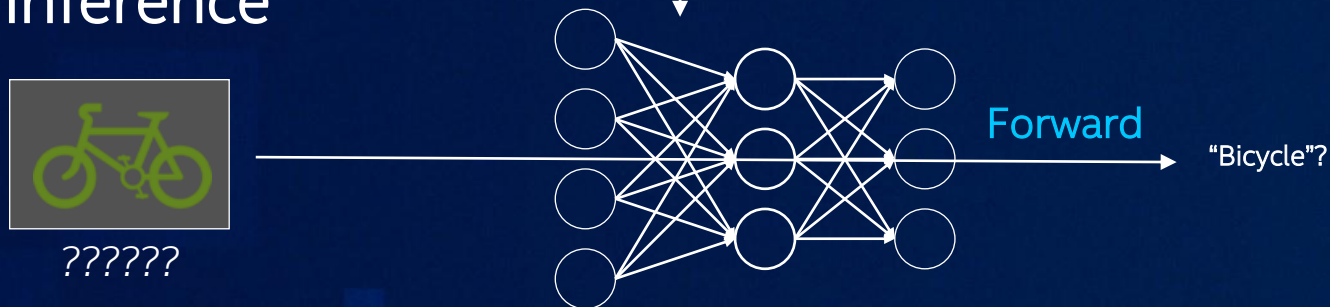
Source: Intel AI Market Model/Omdia Q2'20 AI Chipsets Report.
All figures are in US Dollars.

Deep Learning: Training vs. Inference

Training

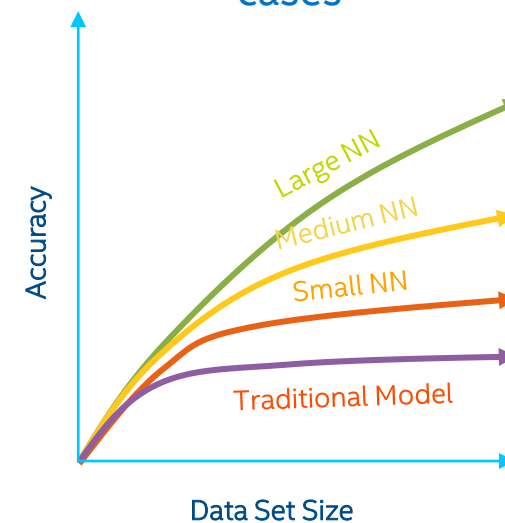


Inference



Did You Know?

Training requires a very large data set and deep neural network (many layers) to achieve the highest accuracy in most cases



Challenges in deep learning

Development and deployment challenges in deep learning



Unique inference needs

Gap in performance and accuracy between trained and deployed models

Low-performing, lower-accuracy models deployed



Integration challenges

No way to streamline end-to-end development workflow

Slow time to solution and time to market



No one size fits all

Diverse requirements for myriad use cases require unique approaches

Inability to meet use-case-specific requirements

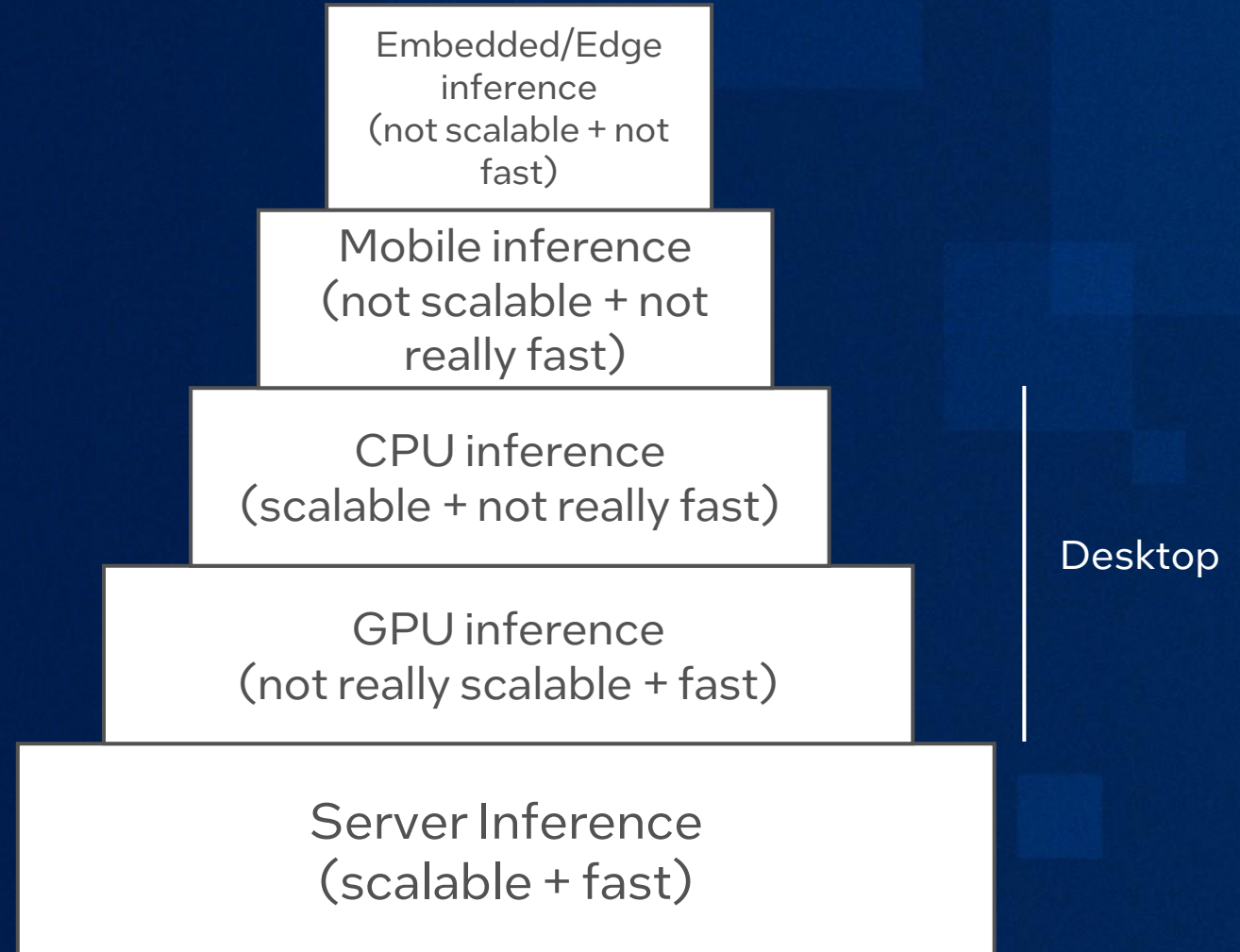
AI Inference target devices

Determinants of configuration selection:

- Performance
- Power consumption
- Price
- Size
- Location (availability and quality of Internet channel)
- Location conditions (temperature, dust and moisture protection)

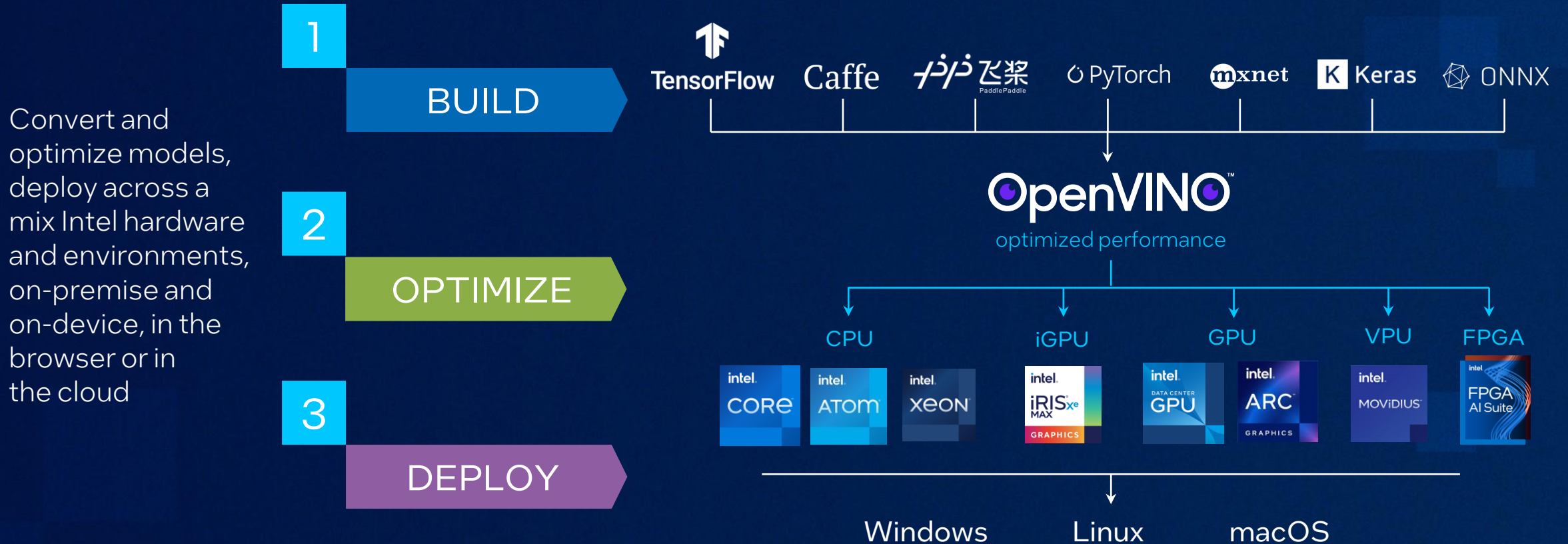
Server
focus

Edge
focus



Why Intel® Distribution of OpenVINO™ Toolkit

Fast, accurate real-world results with high-performance, deep learning inference



Object Detection + Intel® Xeon®: Compelling SW+HW AI Inference Performance Increases Over Time

Improvements mean exponential performance increase

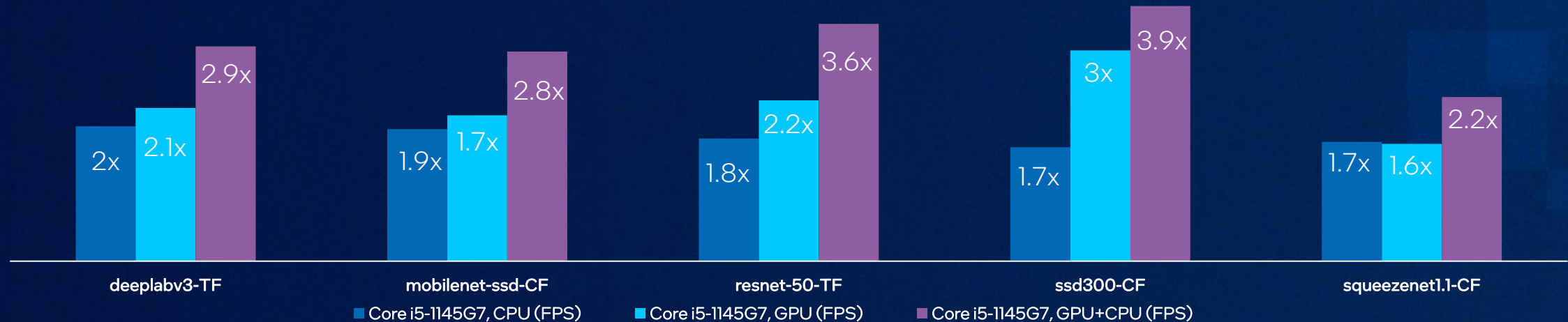


See [here](#) for workloads and configurations. Results may vary.
2018 R5 obtained on system configuration 1
2019 R3 obtained on system configuration 2
OV-2021.2 obtained on system configuration 3
OV-2021.4.1 and OV-2022.1 obtained on system configuration 4

Compounding effect of hardware and software

Use Intel® Iris® Xe graphics + CPU combined for maximum inferencing

Tiger Lake + Intel® Distribution of OpenVINO™ toolkit vs. Coffee Lake CPU



11th Gen Intel® Core™ i5-1145G7 (Tiger Lake) relative inference FPS compared to Intel® Core™ i5-8500 (Coffee Lake).



The above is preliminary performance data based on preproduction components. For more complete information about performance and benchmark results, visit intel.com/benchmarks. See backup for configuration details.

OpenVINO™ Toolkit Developer Journey

1

BUILD



Trained model



Open Model Zoo
280+ open sourced and optimized pretrained models available

2

OPTIMIZE



Model Optimizer
Converts and optimizes trained model using a supported framework

Read, Load, Infer



Post-Training Optimization Tool
Reduces model size into low-precision without retraining



Neural Network Compression Framework
Compression algorithms for quantization-aware training with PyTorch and TensorFlow frameworks



Deep Learning Workbench
Visually analyze and fine-tune

3

DEPLOY



Model Server
gRPC/ REST Server with C++ backend

OpenVINO Runtime (Inference Engine)
Common API that abstracts low-level programming for each hardware



1
oneAPI

Powered by oneAPI

The productive, smart path to freedom for accelerated computing from the economic and technical burdens of proprietary alternatives.

Open Model Zoo

Pretrained models



Supported frameworks and formats ▶ https://docs.openvino.ai/latest/openvino_docs_MO_DG_prepare_model_Supported_Frameworks_Layers.html#doxid-openvino-docs-m-o-d-g-prepare-model-supported-frameworks-layers

Convert model with Model Optimizer ▶ https://docs.openvino.ai/latest/openvino_docs_MO_DG_Deep_Learning_Model_Optimizer_DevGuide.html

Object
detection

Object
recognition

Gesture
recognition

Machine
translation

Text
recognition

Question
answering

Compressed
models

Human pose
estimation

Semantic
segmentation

300+
pretrained and
optimized models

Action
recognition

Image
retrieval

Image
processing

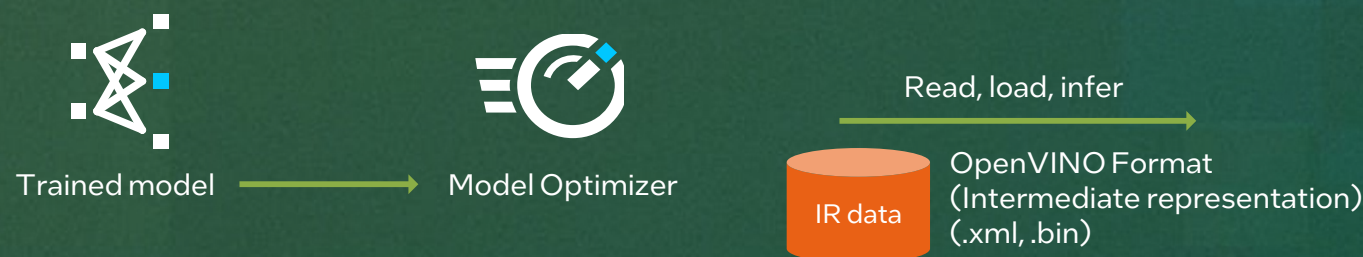
Instance
segmentation

Text
detection

Text
spotting

Model Optimizer

- A Python-based tool to import trained models and convert them to intermediate representation (IR)
- Optimizes for performance or space with conservative topology transformations
- Hardware-agnostic optimizations



Optimization techniques available are:

- Linear operation fusing
- Stride optimizations
- Group convolutions fusing

.xml – Describes the network topology
.bin – Describes the weights and biases binary data

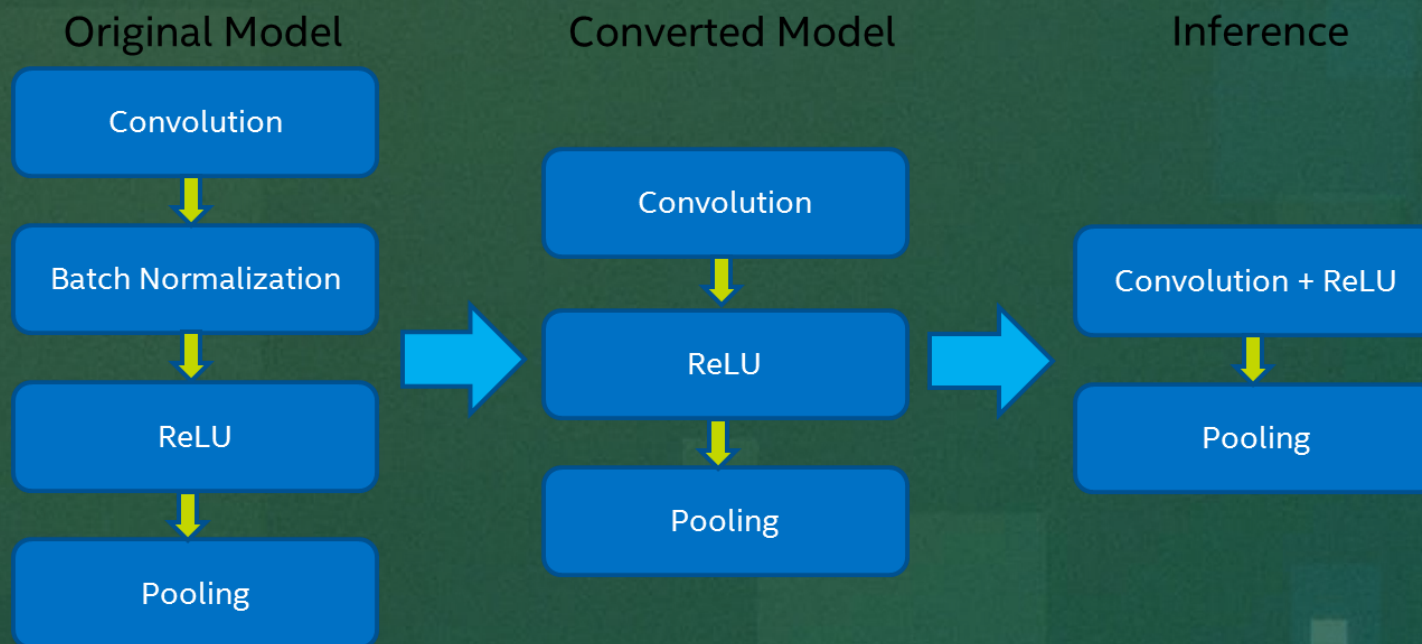
Note: Except for ONNX (.onnx model formats), all models have to be converted to an IR format to use as input to the Inference Engine.

Development guide ► https://docs.openvino.ai/latest/openvino_docs_MO_DG_Deep_Learning_Model_Optimizer_DevGuide.html

Model Optimizer: Linear Operation Fusing

Example

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



Development guide ► https://docs.openvino.ai/latest/openvino_docs_MO_DG_Deep_Learning_Model_Optimizer_DevGuide.html

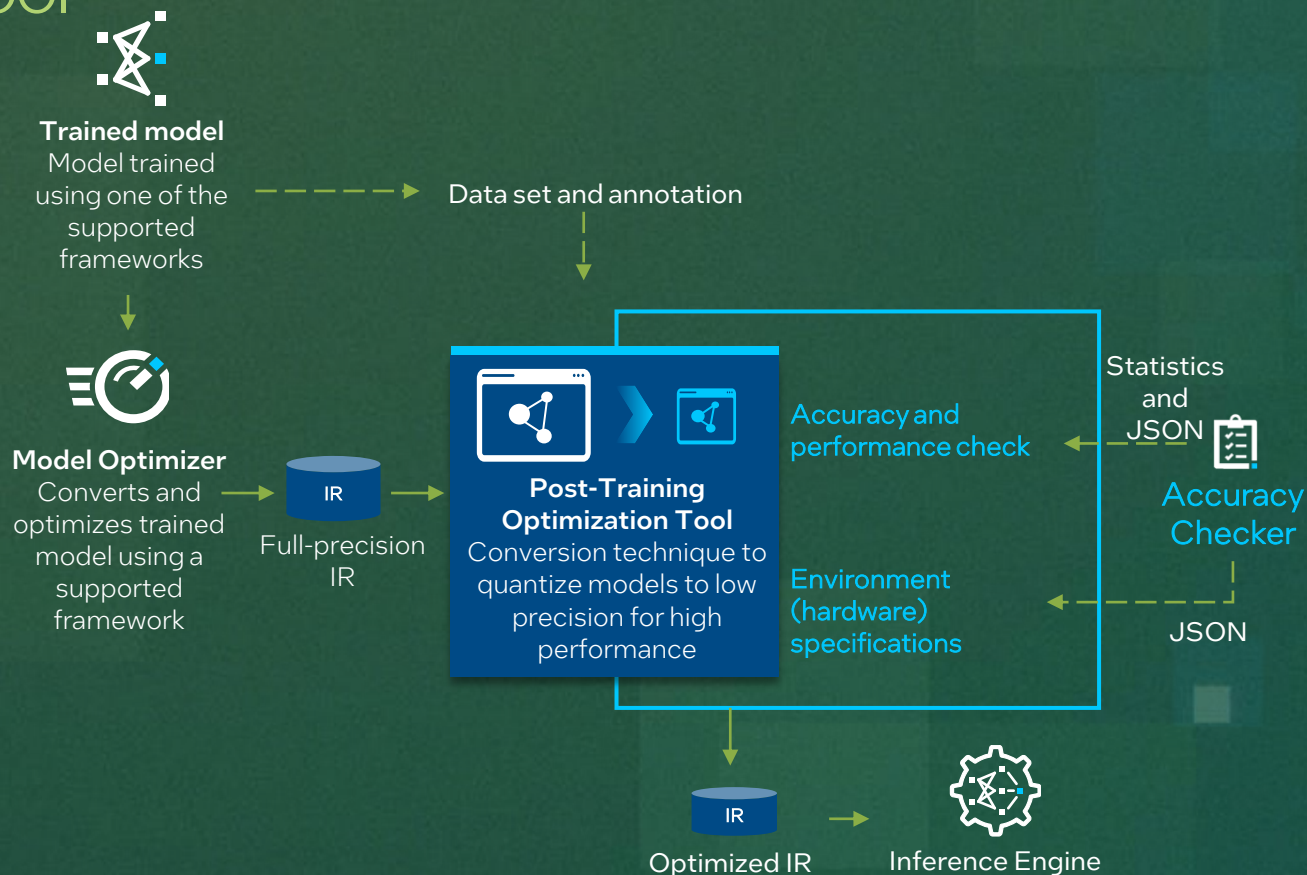
Post-Training Optimization Tool

Conversion technique that reduces model size into low precision without retraining

Reduces model size while also improving latency, with little degradation in model accuracy and without model retraining.

Different optimization approaches are supported: quantization algorithms, etc.

Available as a command line tool and API and inside Deep Learning Workbench.



Deep Learning Workbench

Web-based UI extension tool for model analyses and graphical measurements

- Visualize performance data for topologies and layers to aid in model analysis
- Automate analysis for optimal performance configuration (streams, batches, latency)
- Experiment with int8 or Winograd calibration for optimal tuning using the Post-Training Optimization Tool
- Provide accuracy information through Accuracy Checker
- Direct access to models from public set of Open Model Zoo
- Enable remote profiling, allowing the collection of performance data from multiple machines without any additional setup

Create Configuration

Select a model, dataset, and environment, then click Create to perform an inference. You can skip dataset selection and do it later.

Configuration Details

Model: densenet-121
Environment: Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz
Dataset: Imagenet_200_224x224

Configuration Tip
Environment depends on the model you select. Precision values differ across different targets. If your model and dataset are incompatible, you still can profile the configuration.

Model You can import model if you couldn't find model in the table.

Model Name	Date	Usage	Precisions	Size	Status	Actions
densenet-121	16/03/20, 07:01	Classification	FP16	16 Mb	✓	Details
squeezenet1.1	16/03/20, 07:00	Classification	FP16	3 Mb	✓	Details

Environment
Select Environment
Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz

Validation Dataset You can import validation dataset or generate dataset if you couldn't find dataset in the table.

Dataset Name	Date	Type	Task	Size	Status	Action
Imagenet_200_224x224	16/03/20, 07:09	ImageNet	Classification	5 Mb	✓	Details
COCO2000	16/03/20, 07:05	COCO	Object Detection, Instance Segmentation	329 Mb	✓	Details
VOC7_248_Mikhail	16/03/20, 07:05	VOC	Object Detection	12 Mb	✓	Details

Import/select model

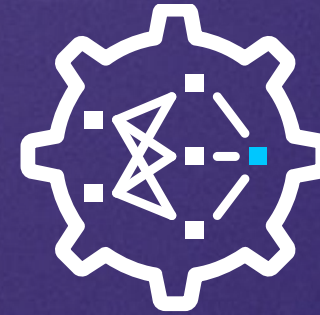
Import/select target

Import/select data set

Run

Runtime

- High-level C, C++, and Python inference runtime API
- Interface is implemented as dynamically loaded plugins for each hardware type
- Delivers superior performance for each type without requiring users to implement and maintain multiple code pathways



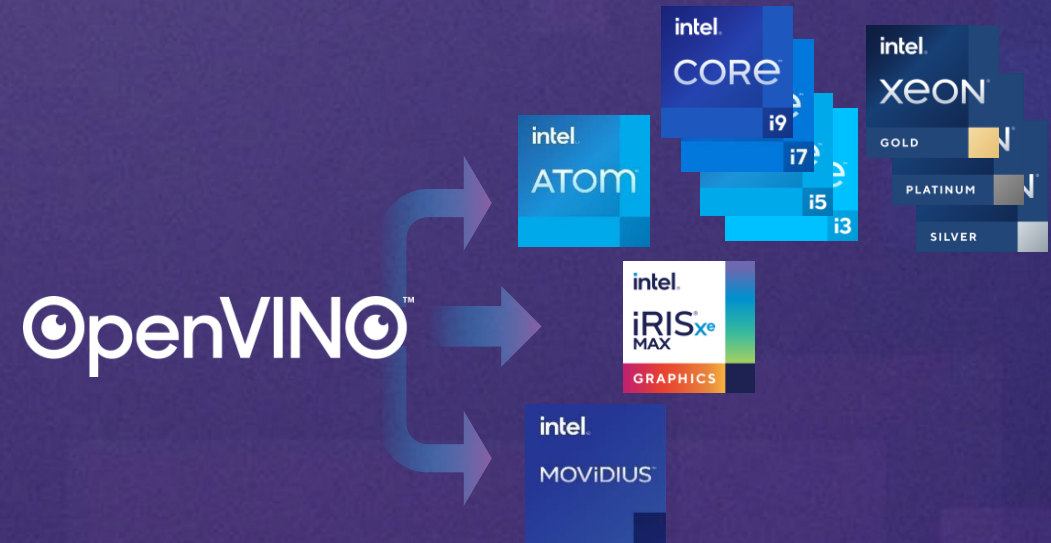
Development guide ► https://docs.openvino.ai/latest/openvino_docs_OV_UG_OV_Runtime_User_Guide.html#

Write once, deploy anywhere

Common high-level inference runtime for cross-platform flexibility

Write once, deploy across different platforms with the same API and framework-independent execution.

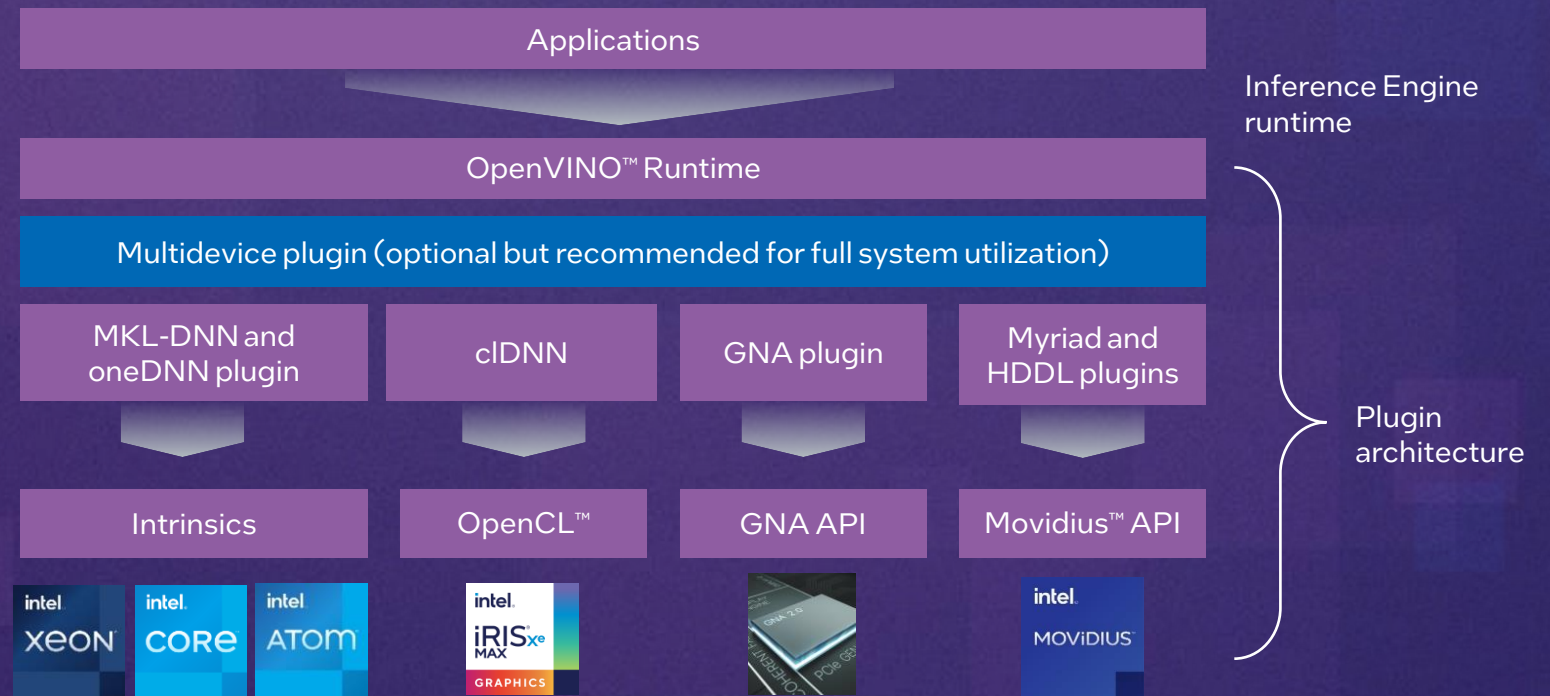
Full environment utilization, or multidevice plugin, across available hardware for superior performance results.



For more details on supported platforms see system requirements ► <https://www.intel.com/content/www/us/en/developer/tools/opencvino-toolkit/system-requirements.html>

Write once, deploy anywhere

Common high-level inference
runtime for cross-platform
flexibility



Additional tools and add-ons

Streamlined development experience and ease of use



- Provides an easy way of accessing a number of public models as well as a set of pretrained Intel models



- Generate an optimal, minimized runtime package for deployment
- Deploy with smaller footprint compared to development package



- Measure performance (throughput, latency) of a model
- Get performance metrics per layer and on overall basis



- Check model accuracy pre- and post-conversion using a known data set

[Computer Vision Annotation Tool](#)

This web-based tool helps annotate videos and images before training a model

[Deep Learning Streamer](#)

Streaming analytics framework to create and deploy complex media analytics pipelines

[OpenVINO™ Model Server](#)

Scalable inference server for serving optimized models over gRPC or REST API endpoints

[Dataset Management Framework](#)

Use this add-on to build, transform, and analyze data sets

[Neural Network Compression Framework](#)

Suite of compression algorithms for quantization-aware training with PyTorch and TensorFlow frameworks

[Training Extensions](#)

Trainable deep learning models for action recognition, segmentation, image classification, object detection, and text spotting

What's New with OpenVINO™ Toolkit 2022.1 Release

Updated, Cleaner API

The 2022.1 release **aligns with TensorFlow conventions**, streamlines optimization, and improves conversion performance.

- Performance has been significantly improved for model conversion on Open Neural Network Exchange (ONNX*) models
- Model Optimizer parameters have been reduced to minimize complexity

Broader Model Support

Tackle more applications including **natural language processing (NLP)**, **PaddlePaddle** models, and **anomaly detection**.

- OpenVINO 2022.1 adds Dynamic Input Shapes, more NLP models for training speech denoising and updates on speech recognition and speech synthesis
- Support for select PaddlePaddle* models
- Pretrained models for Anomaly Detection including anomaly segmentation for industrial inspection

Portability and Performance

The new AUTO plugin **discovers available devices** and **runs inference workloads dynamically** across CPU, GPU, and XPU

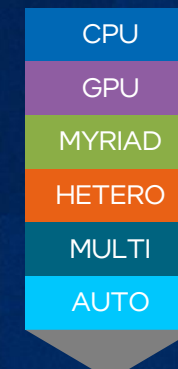
- The AUTO feature uses performance hints to define latency, throughput, and batching requirements—device and runtime logic configure themselves
- AUTO maximizes hybrid architectures for high performance inferencing on platforms with CPUs and integrated GPUs like 12th Gen Intel® Core™ processors (aka Alder Lake)

New feature in 2022.1: AUTO plugin

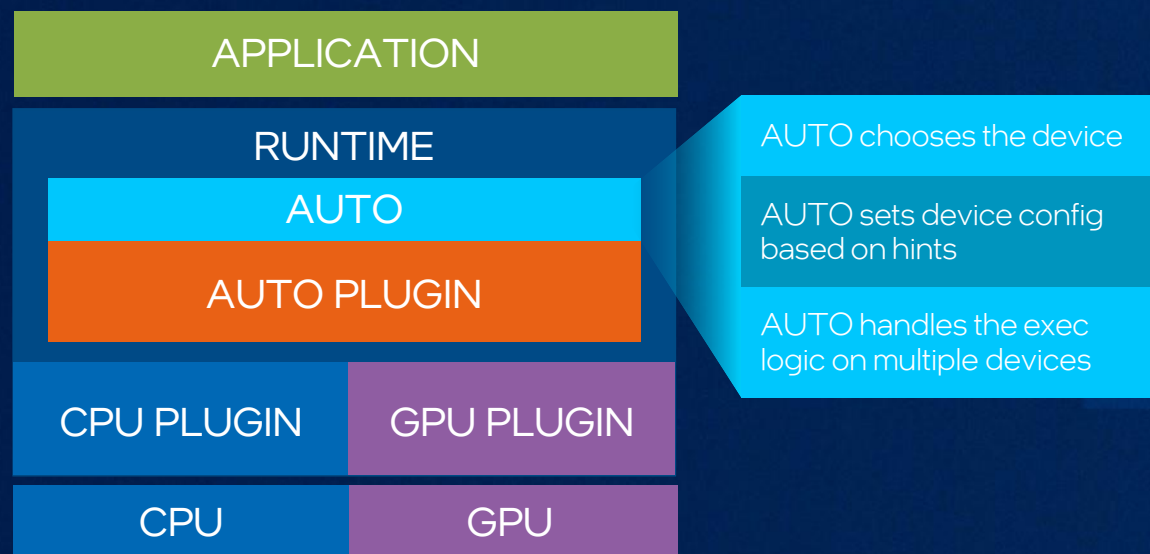
The AUTO plugin automatically detects processing resources and maximizes inference performance, which is ideal for hybrid architectures and container-based applications that land on unknown devices.

The AUTO plugin uses performance hints to select devices and configure workload logic. Performance hints reverse the direction of configuration by expressing a target scenario—for example, latency and throughput targets—with a single config key that then lets the device configure itself in response.

Learn more doc.openvino.ai



```
compiled_model = core.compile_model(model=model, device_name="AUTO")
```

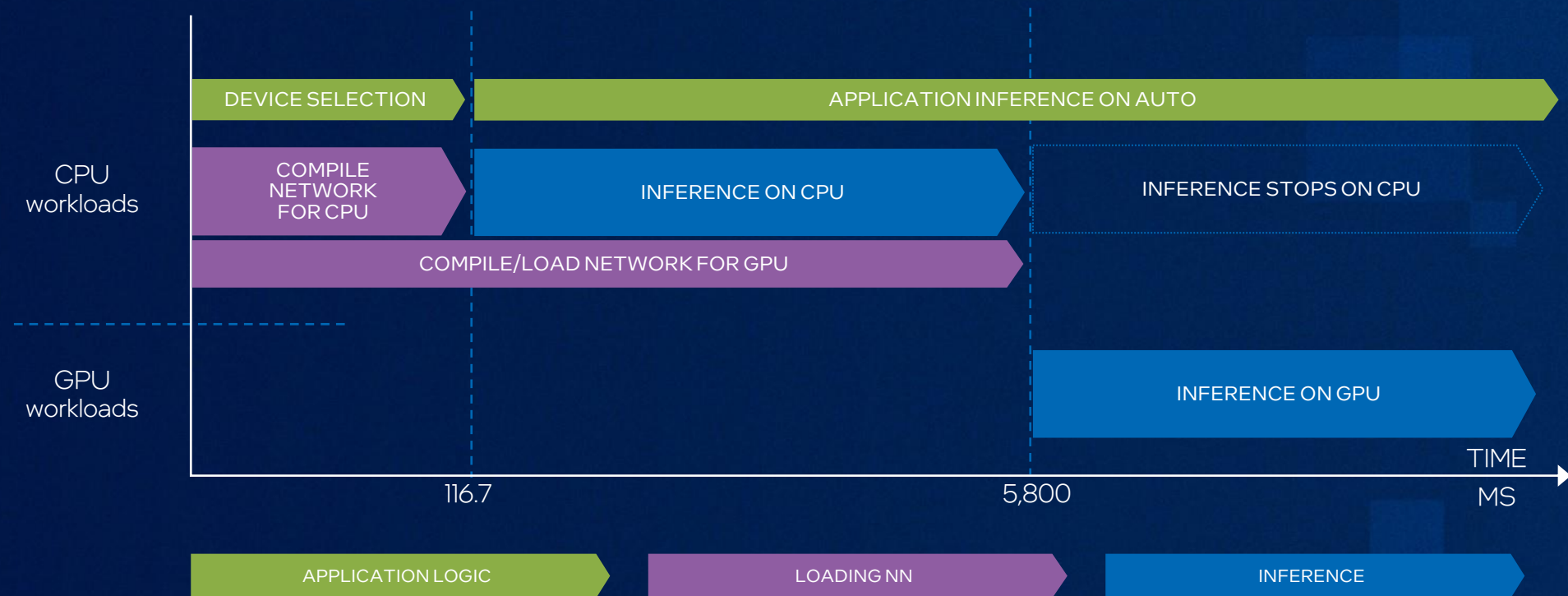


New feature in 2022.1: AUTO plugin—how it works

The AUTO plugin shifts workloads throughout runtime to maximize performance.

For example, the GPU may be the best device, but GPU initialization takes longer than the CPU because of the time it takes to compile the OpenCL™ kernel.

The AUTO plugin starts inference on the CPU while it loads and compiles for the GPU. When the GPU is ready, the AUTO plugin switches the device to the GPU and releases resources from the CPU.



New feature in 2022.1: Dynamic Input Shapes on CPU

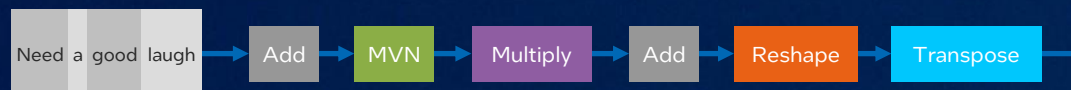
Words, phrases, and sentences come in many lengths, which means every input shape varies from sample to sample.

Dynamic Input Shapes reads and reshapes the model's network for each input automatically.

This allows sequence processing models, like BERT-based NLP models, to ingest, inference, and output results of varying lengths more efficiently.

In OpenVINO™ 2022.1, Dynamic Input Shapes run on CPUs only. Extending Dynamic Input Shapes to GPUs and VPUs is on the road map for a future dot release.

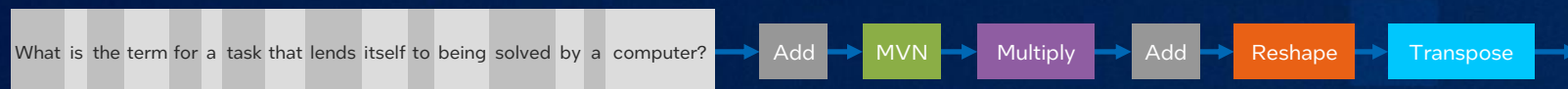
Need a good laugh



What is the weather going to be like today?



What is the term for a task that lends itself to being solved by a computer?



Edge Intelligence Solution = Video AI Box + AI Algo + Business App

Edge Intelligence Solution

- Unmanned parking
- FR payment
- Smart community
- Smart bank
- Industrial defect detection
- Forest vision fire alarm
- Smart farming
- And more....



Video AI Box

- Intel CPU
- Intel CPU with iGPU
- Intel CPU + 3rd Party Accelerator
- Validated Video AI Box from Intel partners



AI Algo

- Object detection
- Feature recognition
- Feature detection
- People counting
- Fire detection
- Classification
- Semantic segmentation

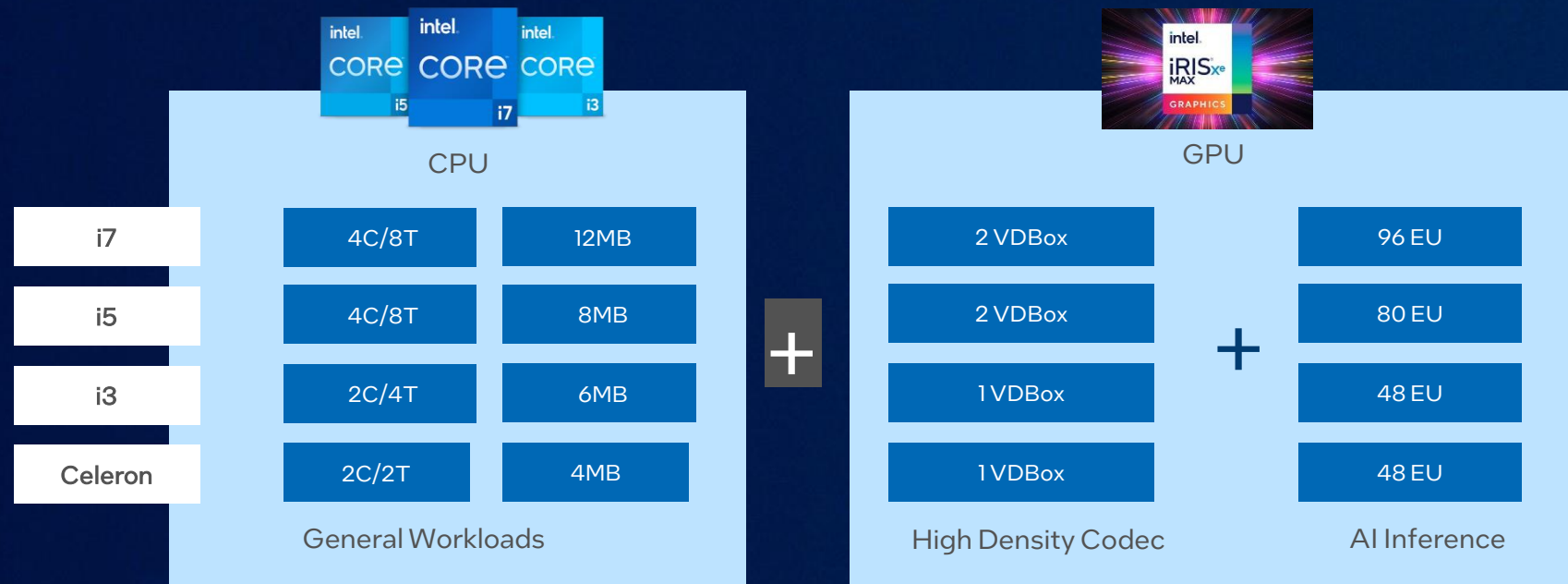


Business App

- VMS
- CRM System
- ERP System
- Customer UI
- Cloud Synergy
- Device Gateway
- Device Management
- Security

Intel® Video AI Box

“General Compute + Media + AI” Capability to Meet Diverse Requirements



11th Gen Intel® Core™ (Tiger Lake – UP3)
General Workloads + Codec + AI in one SoC

Intel® Video AI Box Platform

CPU+ iGPU platform to ease the development, improve the efficiency, lower the solution cost and reduce the operation efforts

Media and AI workloads on Integrated GPU with OpenVINO™ to accelerate the AI inference

OS, Business app and other general workloads consolidated on high performance X86 CPU

New Intel® Iris® Xe graphics

Powerful AI and Media Capability at Low Power



Up to 96 execution units providing 2.95x graphic performance improvement¹ and powerful AI computing capability



Up to 2 VDBox to high-density video decoding offloading EU and CPU resources



Intel® Deep Learning Boost to achieve accelerated INT8 AI inference via VNNI on CPU and DP4a instructions on iGPU



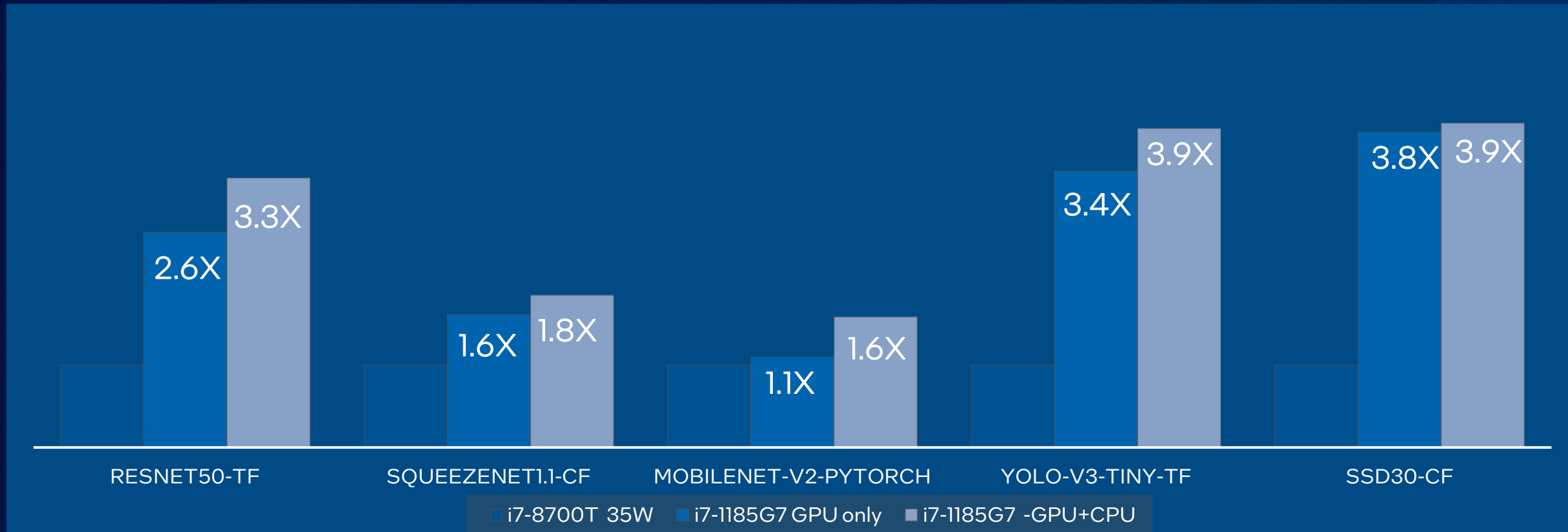
Leverage iGPU powerful media and AI capability to avoid additional discrete accelerator, reducing the cost for solutions and future maintenance

1. For more information and configuration, please visit intel.com/tigerlake-up3. [Workloads and configurations](#). Results may vary.

Intel® Video AI Box

11th Gen Intel® Core™ Processor Performance

Tiger Lake-UP3 + Intel® Distribution of OpenVINO™ toolkit vs Coffee Lake CPU



* For detailed configuration and AI inference benchmark results on more neural networks, please visit [Intel® Distribution of OpenVINO™ toolkit Benchmark Results](https://www.intel.com/PerformanceIndex). Performance varies by use, configuration and other factors. Learn more at www.intel.com/PerformanceIndex. Your costs and results may vary.

Demos and reference implementations

AI inferencing one-stop shop

Take advantage of prebuilt, open-source example implementations with step-by-step guidance and required components list

[Intruder Detector – C++](#)

[Machine Operator Monitor – C++](#)

[Motor Defect Detector – Python](#)

[Object Flaw Detector – C++](#)

[Object Size Detector – C++](#)

[Parking Lot Counter – C++](#)

[People Counter – C++](#)

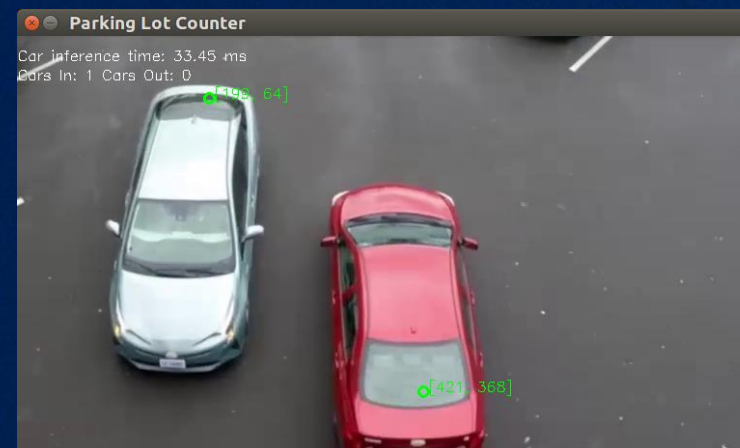
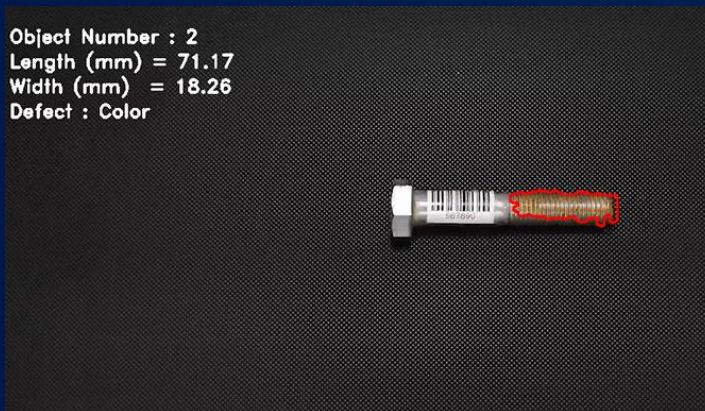
[Shopper Gaze Monitor – C++](#)

[Store Aisle Monitor – C++](#)

[Store Traffic Monitor – C++](#)

[Store Traffic Monitor – Python](#)

Object Number : 2
Length (mm) = 71.17
Width (mm) = 18.26
Defect : Color



New OpenVINO™ notebooks, demos and support for additional public models

OpenVINO™ notebook:

Ready-to-run Jupyter Notebooks with tutorials for converting TensorFlow and PyTorch models, image classification, segmentation, depth estimation, post-training quantization and more



Demos:

- Audio Noise Suppression & Time Series Forecasting demos

Additional Public Models:

- RCAN and IseeBetter (image super-resolution)
- Attention OCR (image text prediction)
- Tacotron 2 (text-to-speech)
- ModNet (portrait/image matting)

Launch an OpenVINO™ notebook



```
# Step 1: Create and Activate openvino_env Environment
python3 -m venv openvino_env
source openvino_env/bin/activate

# Step 2: Clone the Repository
git clone https://github.com/openvinotoolkit/openvino_notebooks.git
cd openvino_notebooks

# Step 3: Install and Launch the Notebooks
python -m pip install --upgrade pip
pip install -r requirements.txt --use-deprecated=legacy-resolver
python -m ipykernel install --user --name openvino_env

# Run the Notebooks
jupyter lab notebooks
```

OpenVINO™ notebooks: https://github.com/openvinotoolkit/openvino_notebooks
OpenVINO™ notebooks video: <https://youtu.be/JtRcFmXMdbg>

Ready to get started?

Download free directly from Intel

[Intel® Distribution of OpenVINO™ toolkit](#)

Also available from these sources:

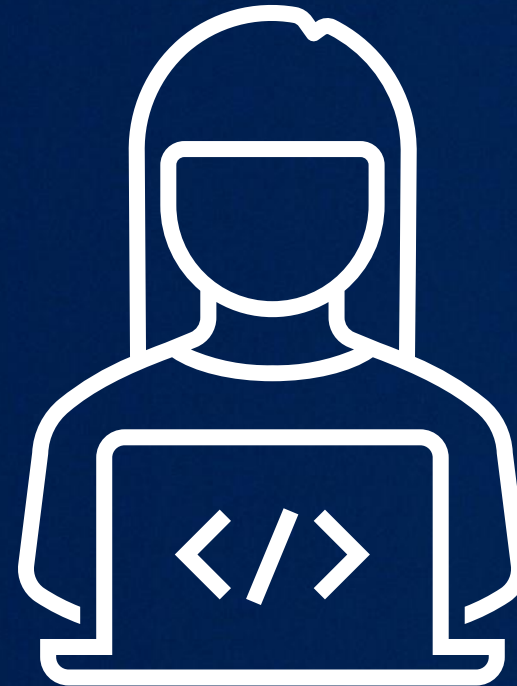
[Intel® DevCloud for the Edge](#) | [PIP](#)
[Docker Hub](#) | [Dockerfile](#)
[Anaconda Cloud](#) | [YUM](#) | [APT](#)

Build from source:

[GitHub](#) | [Gitee](#) (for China)



Demo





© Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.