Edge Intelligence I: Foundations & The Substrate

Chapter 4

Topic: What is Edge Intelligence | Hardware | Software

Outline

- ❑ What is Edge Intelligence
  - AI meets the edge
  - Training vs inference
  - Why intelligence moves out
- ❑ Computing Models
  - CPU & GPU baseline
  - ASIC, FPGA, Jetson
  - Performance metrics
- ❑ Software
  - Frameworks & runtimes
  - Lite & mobile engines
  - Development stack
- ❑ Summary & Practice
  - Summary
  - Practice

Edge Intelligence: Motivation

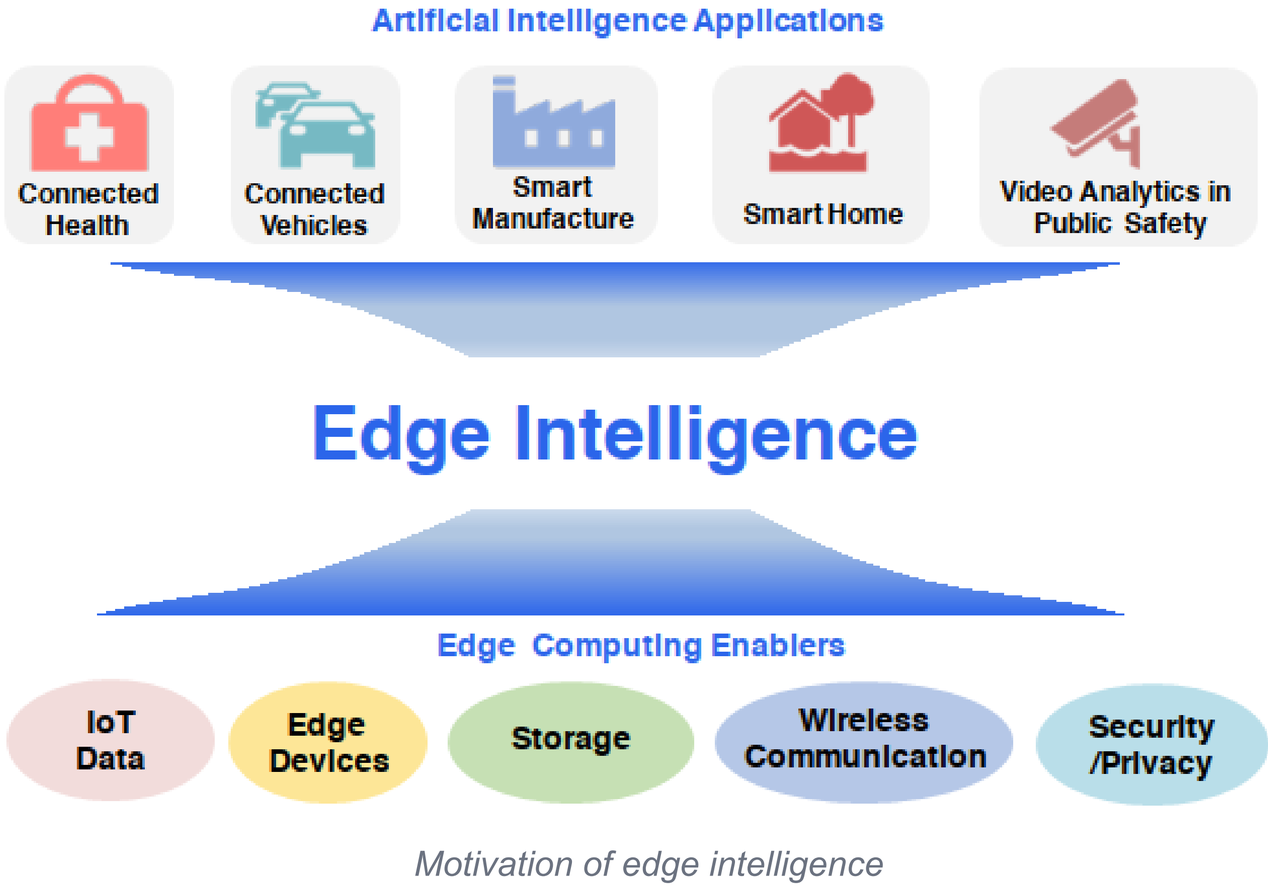
- **Pushed by edge computing**

IoT data, edge devices, storage, wireless links and on-device security make it feasible to run AI where data is born.

- **Pulled by AI applications**

Connected health, vehicles, smart manufacturing, smart homes and video analytics all demand real-time, on-site inference.

The edge handles video, speech, time-series and sensor data with no round-trip to the cloud.



Motivation of edge intelligence

Edge Intelligence: Definition

Edge intelligence is the capability of edge devices to execute artificial-intelligence algorithms locally, acquiring, storing and processing data with ML at the network edge.

- Because hardware differs, every edge has a different EI capability — captured by the ALEM tuple:

Accuracy

Quality of the model's output (e.g. mAP, BLEU). Compression trades some of it away.

Latency

Inference time per task — the measure of edge performance under load.

Energy

Extra power drawn while running inference on the device.

Memory

Footprint of the model in memory during execution.

International Electrotechnical Commission. Edge Intelligence. https://www.iec.ch/basecamp/e dge-intelligence. Accessed: 2024-05-27. 2024

Edge Intelligence: Collaboration

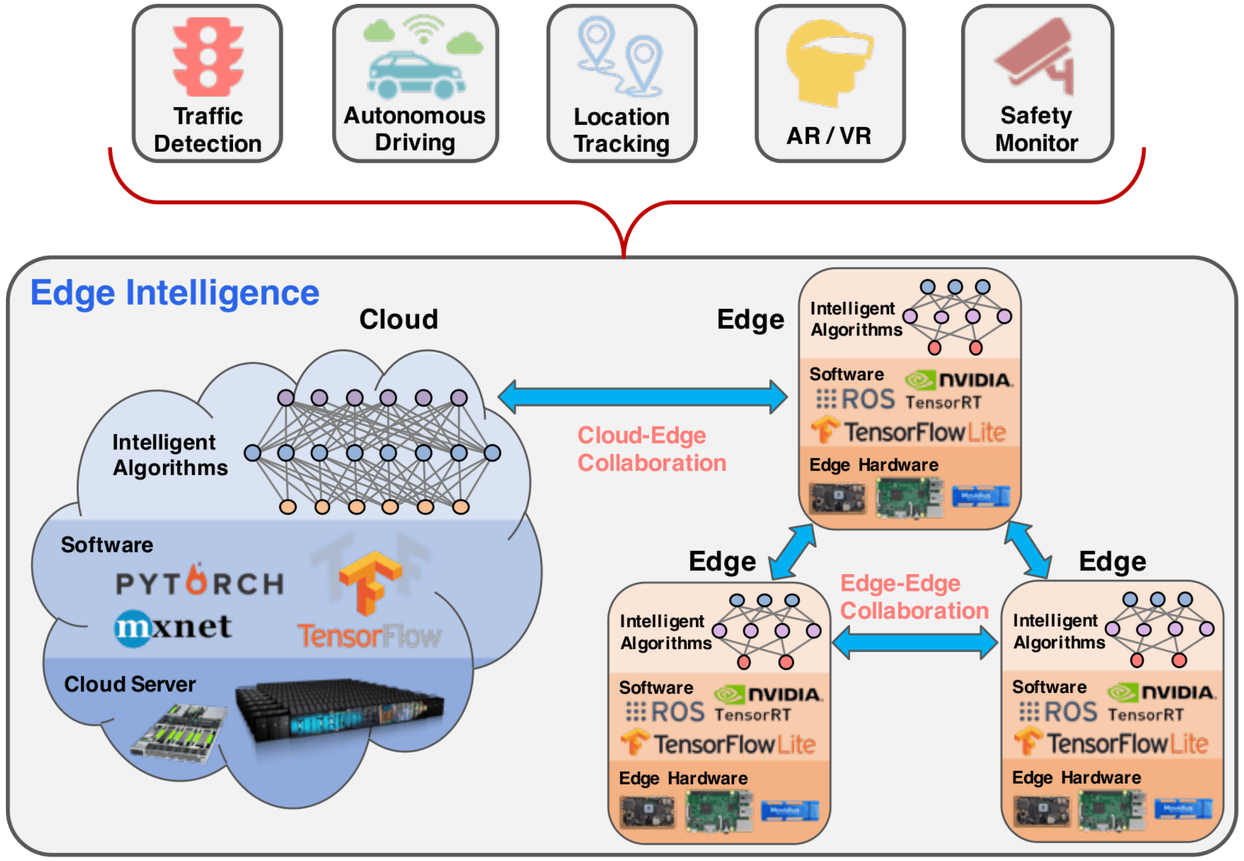
Two ways Edges cooperate

Cloud – Edge

Train in the cloud, download to the edge for inference. Edges may retrain locally via transfer learning and push updates back to a global model — e.g. DDNN spans cloud and edge.

Edge – Edge

Edges split one heavy job by compute power or divide a task by context. A phone predicts you're nearly home and cues the thermostat: hard alone, easy in concert.



Edge Intelligence

Edge Intelligence: Data Flows

Three ways to use edge data

Cloud training

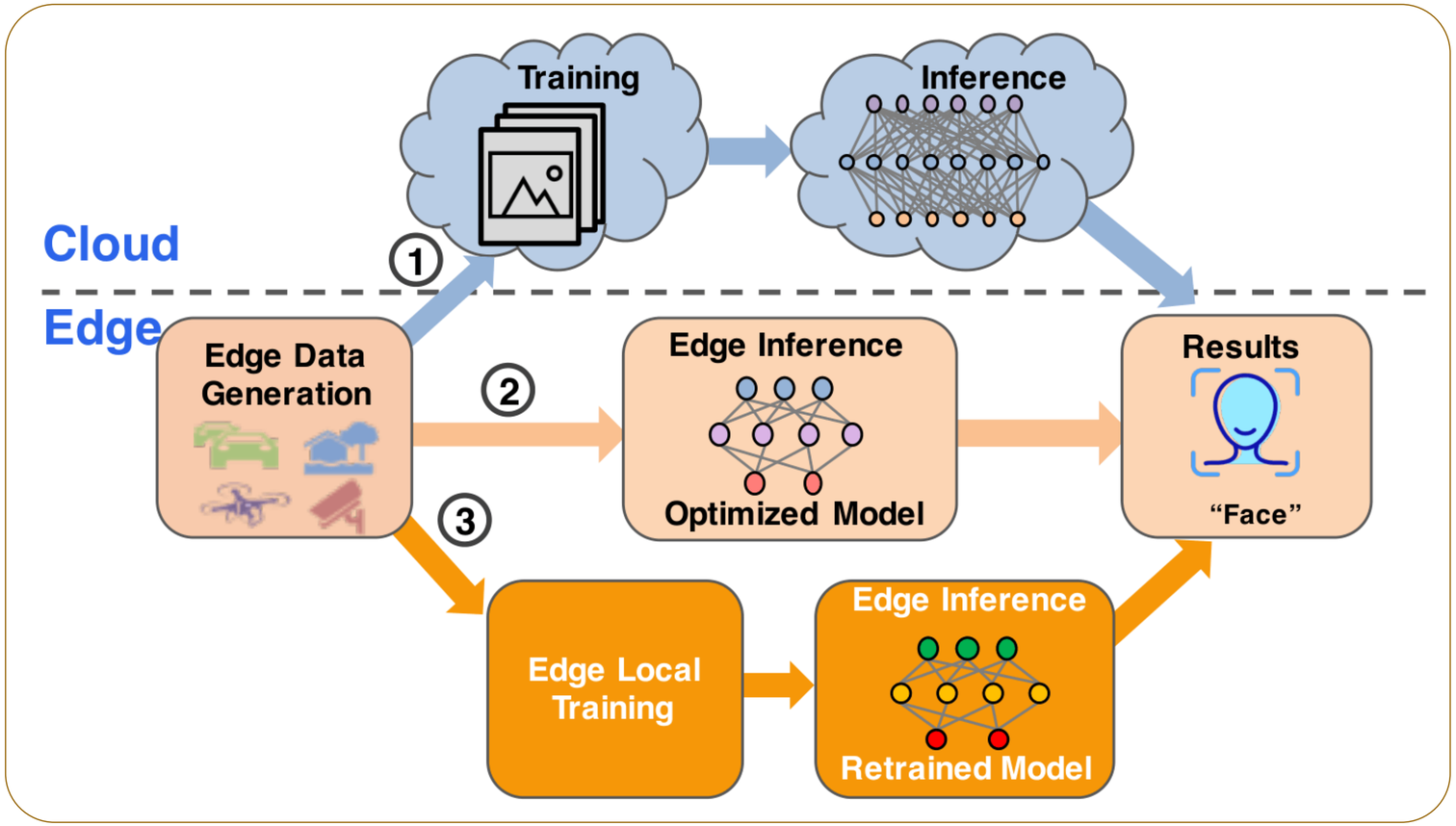
Upload to cloud, train on multi-source data, infer, return result. Traditional machine intelligence.

Edge inference

Run the cloud-trained model on-device against local data. The current EI dataflow.

Edge local training

Retrain on-device with transfer learning to build a personalized model. The future of EI.



Dataflow of edge intelligence

Edge Intelligence: Case For EI

Why intelligence moves to the edge

Latency

Decisions happen next to the data. No cloud round-trip for face ID, speech, or driving.

Bandwidth

Raw video and sensor streams stay local; only results or model updates travel.

Privacy

Sensitive data is processed on-device, lowering the risk of interception or leakage.

Reliability

Works offline and near the source. Resilient when the network is weak or down.

Hardware: Case For EI

Unlike cloud data centers, edge devices run under tight compute and power budgets. Five families cover the field.

CPU

Flexible, everywhere

General control & light inference

GPU

Massively parallel

Deep learning, autonomous systems

FPGA

Reconfigurable

Custom, adaptive AI workloads

ASIC

Fixed-function

Peak efficiency: TPU, VPU, NPU

Neuromorphic

Brain-inspired

Spiking, sensory, cognitive

Hardware: CPU & GPU

CPU

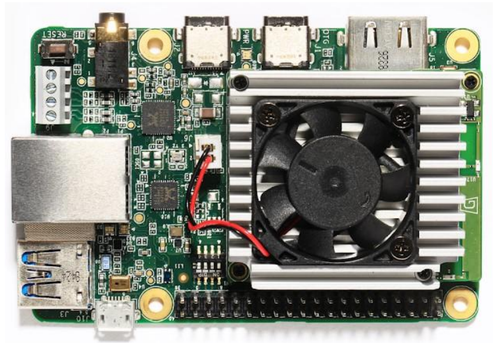
- Serial, general-purpose control logic
- Runs the OS, orchestration, glue code
- Handles light or infrequent inference
- Ubiquitous — every edge device has one
- Poor throughput on dense tensor math

GPU

- Thousands of cores for parallel math
- The workhorse for deep-learning training
- Strong on batched, high-throughput inference
- Jetson brings GPU compute to the edge
- Costs the most power of the edge options

Hardware: ASIC

An ASIC is hard-wired for a specific task, sacrificing flexibility for the best accuracy, latency and energy in its niche.



TPU

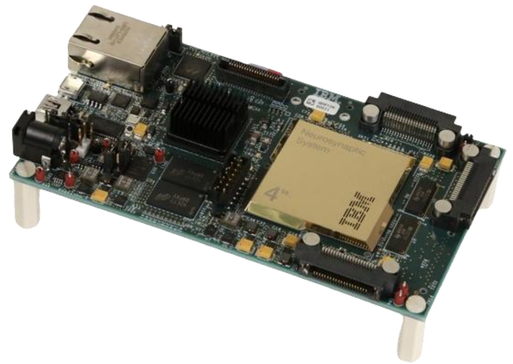
Google Edge TPU

Tensor Processing Unit tuned for TensorFlow inference — image classification & object detection at very low power.

VPU

Intel Movidius Myriad X

16 SHAVE cores + a neural engine, >1 TOPS. Vision-centric: facial recognition, visual SLAM, low-power detection.



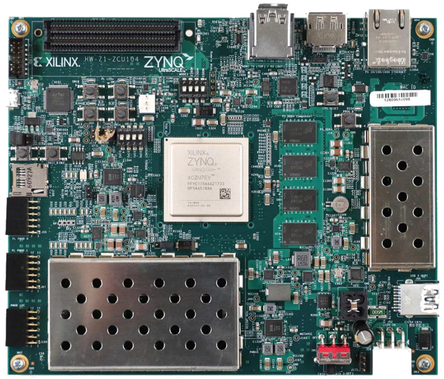
Neuromorphic

IBM TrueNorth · Intel Loihi

Brain-inspired, highly parallel & efficient — real-time pattern recognition, robotics, cognitive computing.

Hardware: FPGA & GPU

FPGA



- Xilinx (Alveo, MPSoC) & Intel (Agilex, Stratix 10)
- Arm cores + programmable logic → real-time, low-power
- ESE ran sparse LSTM faster & greener than CPU/GPU
- Studies find FPGA well-suited to EI workloads

Jetson

- Jetson Nano — 472 GFLOPs at just 5–10 W
- AGX Xavier — 512-core Volta GPU + 8-core CPU
- CUDA + TensorRT accelerate robotics & AV workloads
- Jetson Orin (2022) scales to complex edge AI

Hardware: Comparison of Hardware Types for Edge AI Applications

Type	Sub-type	Examples	AI performance	Power eff.	Latency	Key applications
ASIC	**TPU**	Google TPU, Google Edge TPU	Excellent for AI inference, optimized for TensorFlow	⚡⚡⚡	**Very low**	Real-time AI inference, image classification, object detection
	VPU	Intel Movidius Myriad X	High for vision-centric AI tasks	⚡⚡⚡	**Low**	Facial recognition, visual SLAM, low-power object detection
	Neuromorphic chip	IBM TrueNorth, Intel Loihi	Efficient for brain-inspired AI models	⚡⚡⚡⚡	**Very low**	Cognitive computing, robotics, sensory processing
FPGA	—	Xilinx UltraScale+, Intel Stratix	High, customizable for specific AI workloads	⚡⚡	**Low to moderate**	Custom AI models, adaptive AI tasks
GPU	—	Jetson Nano, AGX Xavier	High, particularly in parallel processing tasks	⚡	**Medium**	Deep learning, autonomous systems, AI research

Power efficiency ⚡ = high ⚡⚡ = moderate ⚡⚡⚡ = very high ⚡⚡⚡⚡ = extremely high

Software: Frameworks

Cloud Frameworks Don’t Fit the Edge

Cloud-first

Built to train at scale

- TensorFlow, Caffe, MXNet, PyTorch
- Large datasets, learn millions of weights
- Run on GPU / CPU / FPGA / TPU clusters
- Resource-hungry — a poor fit for edges

Edge-optimized

Built to infer on-device

- Same inference, far fewer resources
- Small-scale, real-time input data
- Installed on phones, Pi, laptops, servers
- Optimized kernels & quantization

TensorFlow Authors. *TensorFlow Lite*. https://www.tensorflow.org/lite. Accessed: 2024-05-21. 2024.

Tianqi Chen et al. “MxNet: A flexible and efficient machine learning library for heterogeneous distributed systems”. In: arXiv preprint arXiv:1512.01274 (2015).

Software: Lite/Mobile Runtime

TensorFlow Lite
Google's mobile/edge runtime — fused, quantized kernels cut latency.

ONNX Runtime
Cross-platform engine (Meta + Microsoft); runs on Jetson, FPGA, NPU, AGX.

Core ML
Apple's on-device package — minimal memory & power

QNNPACK / XNNPACK
Meta & Google low-level kernels for quantized & float inference.

OpenVINO
Intel toolkit — converts TF/PyTorch models, deploys across Intel HW.

Vitis AI · FINN
Xilinx FPGA stacks — optimize, compile & quantize for the fabric.

Azure IoT Edge
Microsoft — deploy & manage cloud-native AI workloads on devices.

SageMaker Neo · TensorRT
AWS & NVIDIA — compile once, accelerate inference on target HW.

Amazon Web Services, Inc. Amazon SageMaker Neo: Train Once, Run Anywhere. https://aws.amazon.com/sagemaker/neo/. Accessed: 2024-08-21. 2024.

AMD Xilinx. Vitis AI. https://www.xilinx.com/products/design-tools/vitis/vitis-ai.html. 2024.

Apple Inc. Core ML Documentation. https://developer.apple.com/documentation/coreml. Accessed: 2024-05-20. 2024.

Google. XNNPACK. https://github.com/google/XNNPACK. GitHub repository. 2024.

Intel. OpenVINO Toolkit Overview. https://www.intel.com/content/www/us/en/developer/tools/openvino-toolkit/overview.html. Accessed: 2024-07-10. 2024.

Dukhan Marat, W. Yiming, and L. Hao. QNNPACK: Open source library for optimized mobile deep learning. 2018.

Microsoft. Azure IoT Edge. https://azure.microsoft.com/en-us/products/iot-edge. Accessed: 2024-07-10. 2024.

NVIDIA Corporation. TensorRT. https://developer.nvidia.com/tensorrt. Accessed: 2024-05-20. 2024.

ONNX Runtime Developers. ONNX Runtime. https://onnxruntime.ai/. 2021.

Yaman Umuroglu et al. "FINN: A framework for fast, scalable binarized neural network inference". In: Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. 2017, pp. 65–74.

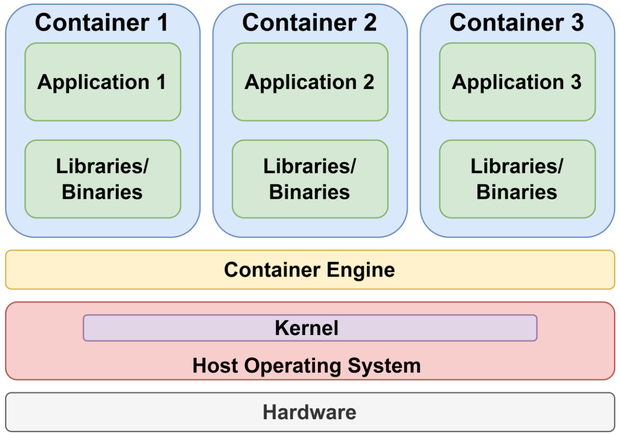
Software: Deployment

Runtimes & Containers: The Stack

Running Environments

- TinyOS — event-driven OS for sensor networks
- ROS / ROS 2 — robotics; DDS messaging, no master
- AWS IoT Greengrass — deploy AI to fleets of devices
- OpenVDAP · EdgeOS_H — vehicle & smart-home stacks
- SageMaker Edge — compile & manage models on devices
- PYNQ — Python on Xilinx Zynq FPGA + Arm SoCs

Containers



Container architecture over a shared kernel

- Bundle code, runtime, libraries & deps into one portable unit.
- Share the OS kernel — small, resource-efficient, easy to migrate.
- **Engines: Docker, RKT, LXD, Kubernetes.**

Summary

- Edge intelligence is the capability of edges to run AI algorithms locally, measured by ALEM (Accuracy, Latency, Energy, Memory).
- Cloud-edge and edge-edge collaboration split training and inference across the continuum.
- CPU, GPU, FPGA, ASIC (TPU/VPU) and neuromorphic chips trade flexibility for efficiency.
- Lite runtimes, edge OSs and containers turn silicon into portable, efficient intelligence.

Practice

1. How does edge hardware differ from traditional data-center hardware?

2. Discuss the advantages and challenges of implementing machine learning at the edge?

3. What are the key considerations in choosing an edge application development framework?

References

- International Electrotechnical Commission. Edge Intelligence. https://www.iec.ch/basecamp/e dge-intelligence. Accessed: 2024-05-27. 2024.
- TensorFlow Authors. TensorFlow Lite. https://www.tensorflow.org/lite. Accessed: 2024-05-21. 2024.
- Tianqi Chen et al. “MxNet: A flexible and efficient machine learning library for heterogeneous distributed systems”. In: arXiv preprint arXiv:1512.01274 (2015).
- Amazon Web Services, Inc. Amazon SageMaker Neo: Train Once, Run Anywhere. https://aws.amazon.com/sagemaker/neo/. Accessed: 2024-08-21. 2024.
- AMD Xilinx. Vitis AI. https://www.xilinx.com/products/design-tools/vitis/vitis-ai.html. 2024.
- Apple Inc. Core ML Documentation. https://developer.apple.com/documentation/coreml. Accessed: 2024-05-20. 2024.
- Google. XNNPACK. https://github.com/google/XNNPACK. GitHub repository. 2024. Intel.
- OpenVINO Toolkit Overview. https://www.intel.com/content/www/us/en/developer/tools/openvino-toolkit/overview.html. Accessed: 2024-07-10. 2024.
- Dukhan Marat, W. Yiming, and L. Hao. QNNPACK: Open source library for optimized mobile deep learning. 2018.
- Microsoft. Azure IoT Edge. https://azure.microsoft.com/en-us/products/iot-edge. Accessed: 2024-07-10. 2024.
- NVIDIA Corporation. TensorRT. https://developer.nvidia.com/tensorrt. Accessed: 2024-05-20. 2024.
- ONNX Runtime Developers. ONNX Runtime. https://onnxruntime.ai/. 2021.
- Yaman Umuroglu et al. "FINN: A framework for fast, scalable binarized neural network inference". In: Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. 2017, pp. 65–74.