UNIVERSITATEA LUCIAN BLAGA DIN SIBIU

HPI Hasso Plattner Institut

CLUJ IT

Sibiu IT Cluster

SID 2025

Sibiu Innovation Days

06 - 07 November, Sibiu - RO

EMERGING DISRUPTIVE TECHNOLOGIES:

Balancing Innovation, Risks, and Societal Impact

Artificial Intelligence On Devices With Limited Capabilities

PhD Eng. Marian-Valentin Bănică

Prof. Anamaria Rădoi

Agenda

1. Introduction to Edge AI
2. Benefits of Edge AI for Resource-Constrained Devices
3. Constraints and Challenges in Edge AI Deployment
4. Application Scenarios for Edge AI
5. Model Pruning Techniques for Edge AI
6. Quantization Techniques for Edge AI
7. Knowledge Distillation for Edge Models
8. Hardware Alternatives for Edge AI Deployment
9. Compiler Toolchains for Edge AI Optimization
10. Benchmarking Edge AI Models and Platforms
11. Benchmarking Edge AI Models and Platforms
12. Future Trends and Case Studies in Edge AI

Introduction to Edge AI

- Edge AI combines artificial intelligence with edge computing enabling smart data processing and decision-making directly on edge devices.
- Edge AI contrasts with traditional AI by performing computations locally rather than relying on centralized cloud servers.
- This local processing supports real-time, efficient, and resilient AI applications, especially as the number of connected devices grows rapidly.

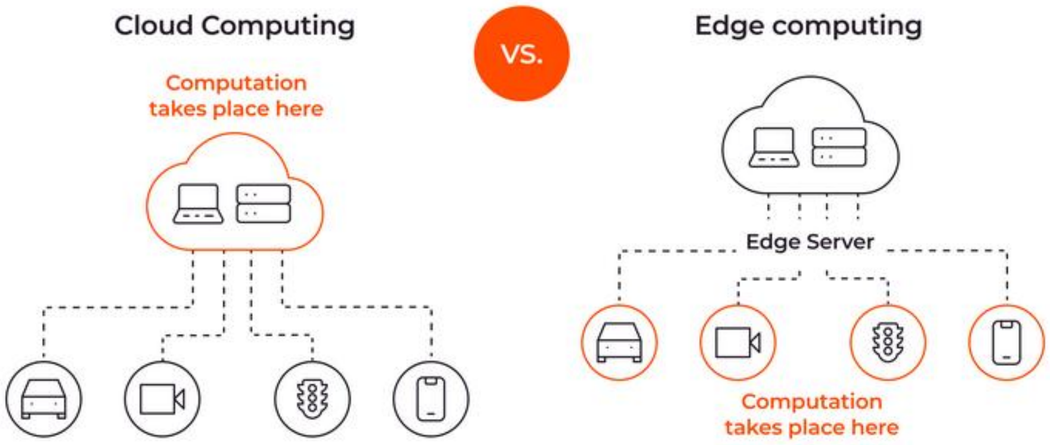


Fig1. Cloud vs Edge computing from Business Benefits of AI Inference at the Edge

Benefits of Edge AI for Resource-Constrained Devices

Benefit	**Example**
⚙ **Reduced Latency & Real-Time Inference**	Local processing removes cloud delays — essential for safety-critical applications like autonomous driving.
🔒 **Enhanced Data Security & Privacy**	Sensitive information (e.g., health or surveillance data) stays on-device, reducing exposure risks and ensuring compliance.
💰 **Reduced Bandwidth & Lower Costs**	Only processed data is sent to the cloud, lowering data transfer needs — beneficial in low-connectivity or high-cost regions.
♻ **Low Power Consumption / Energy Efficiency**	TinyML models enable milliwatt-level power use, ideal for battery-powered devices.
🌐 **Scalability and Local Operation**	Edge devices continue working even when cloud connectivity is limited or unavailable.

Constraints and Challenges in Edge AI Deployment

Challenge Area	Implication / Example
🧠 **Model Size / Complexity**	Requires drastic model compression and simplification.
💻 **Hardware Constraints**	Typical microcontrollers can’t run large models efficiently.
💾 **Memory Limitations**	Necessitates aggressive quantization and model reduction.
⚖️ **Optimization Need**	Demands tailored pruning, quantization, and distillation.
🔧 **Optimization Implementation**	Sparse models lack hardware/software support, increasing engineering effort.

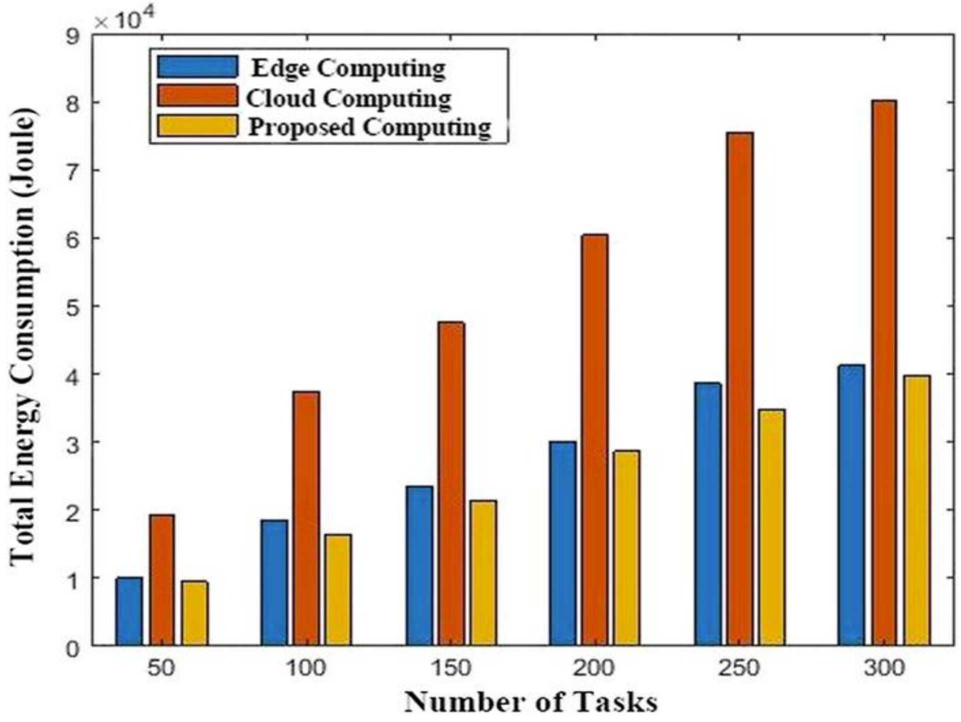


Fig2. Cloud vs Edge energy consumption from: Transactions on Emerging Telecommunications Technologies

Application Scenarios for Edge AI

Healthcare Devices: Patient monitoring, anomaly detection, and on-device diagnostics.

Smart Cities: Traffic management, public safety monitoring, and energy optimization.

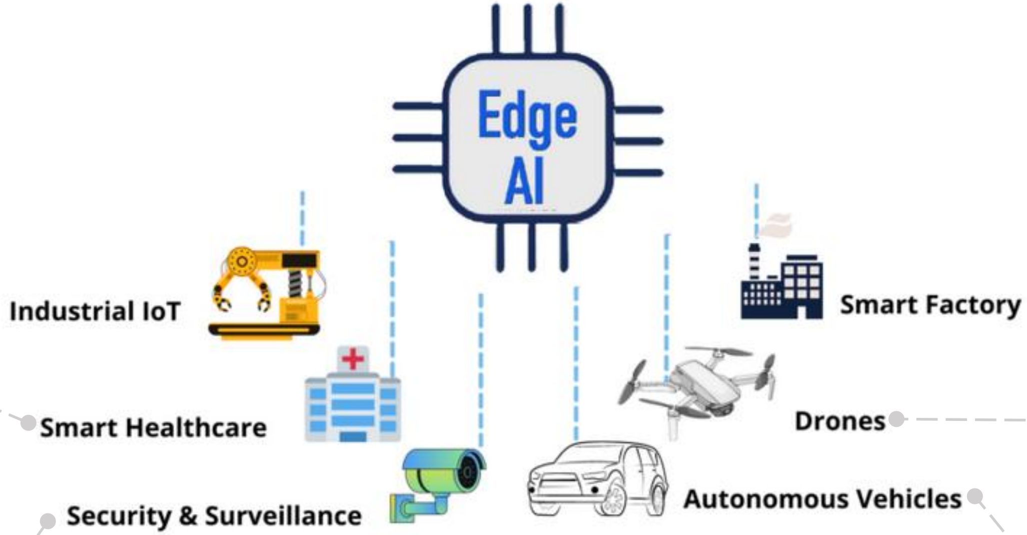


Fig3. Edge AI applications from What are the top edge AI chips of 2025?

Smart Manufacturing: Real-time quality inspection, predictive maintenance, and worker safety.

Autonomous Vehicles & Drones: Obstacle avoidance, route optimization, and sensor fusion.

Autonomous Vehicles & Drones: Obstacle avoidance, route optimization, and sensor fusion.

Model Pruning Techniques for Edge AI

- Pruning: Reduces model complexity by removing unimportant weights, neurons, or even layers.
- Unstructured Pruning: Arbitrarily removes individual weights.
- Structured Pruning: Removes entire neurons, channels, or filters.
- Dynamic Pruning: Adapts pruning during training or inference.
- Benefits: Smaller model size, reduced computational demands, faster inference, lower energy use.

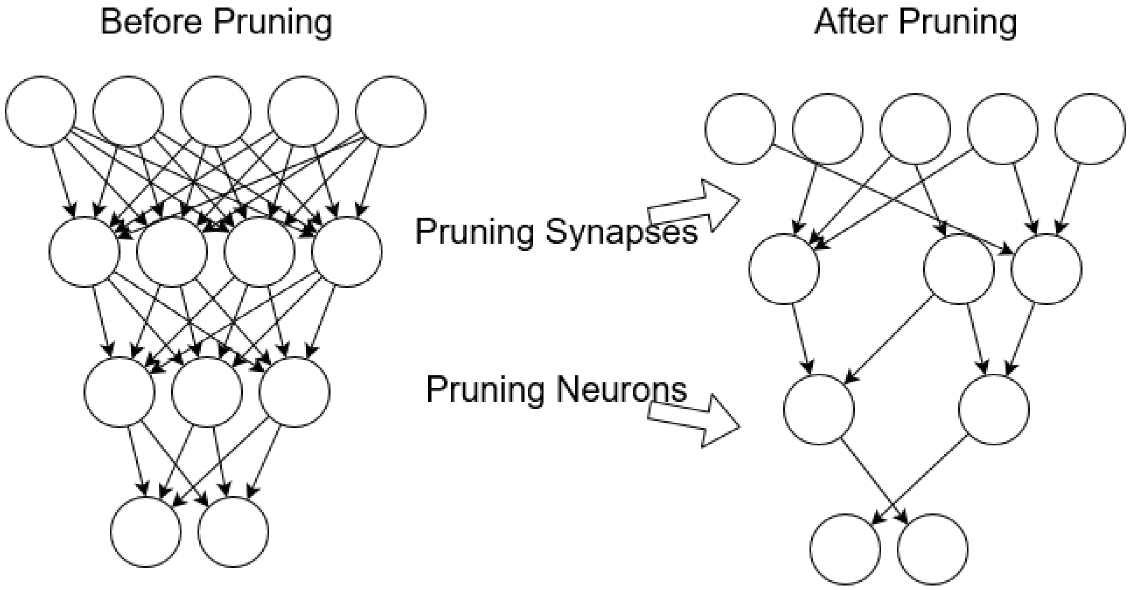
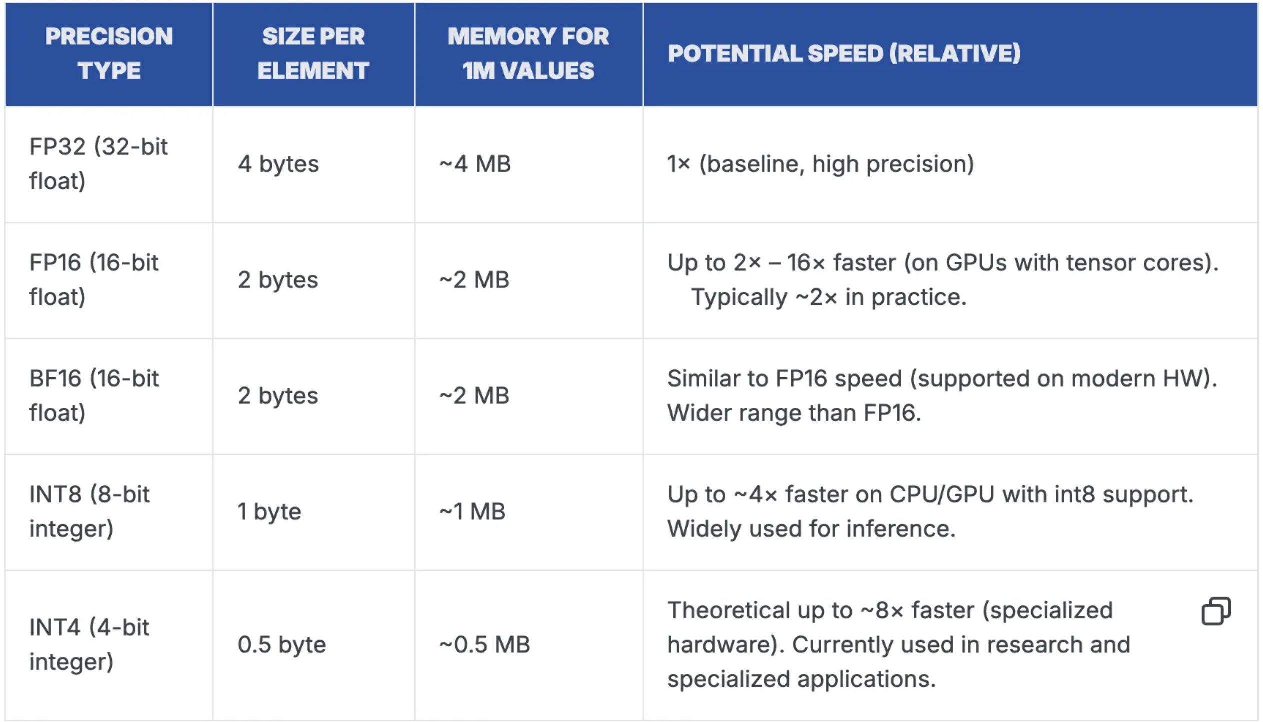


Fig4. Edge AI applications from What are the top edge AI chips of 2025?

Quantization Techniques for Edge AI

- Quantization: Converts model weights and activations from high-precision (e.g., 32-bit float) to lower-precision formats (e.g., 8-bit integer).
  - Post-Training Quantization
  - Quantization-Aware Training
  - Dynamic/Mixed Precision Quantization
- Benefits: Reduces memory usage, speeds up inference, and lowers power consumption, sometimes with negligible accuracy loss.



PRECISION TYPE	SIZE PER ELEMENT	MEMORY FOR 1M VALUES	POTENTIAL SPEED (RELATIVE)
FP32 (32-bit float)	4 bytes	~4 MB	1× (baseline, high precision)
FP16 (16-bit float)	2 bytes	~2 MB	Up to 2× – 16× faster (on GPUs with tensor cores). Typically ~2× in practice.
BF16 (16-bit float)	2 bytes	~2 MB	Similar to FP16 speed (supported on modern HW). Wider range than FP16.
INT8 (8-bit integer)	1 byte	~1 MB	Up to ~4× faster on CPU/GPU with int8 support. Widely used for inference.
INT4 (4-bit integer)	0.5 byte	~0.5 MB	Theoretical up to ~8× faster (specialized hardware). Currently used in research and specialized applications.

Table5. Memory spaced used by 1M values

Knowledge Distillation for Edge Models

- Knowledge Distillation: Transfers knowledge from a large teacher model to a compact student model.
- Approach: Train the student with soft labels (probabilities) produced by the teacher.
- Objective: Maintain high accuracy in a smaller, faster, and more efficient model.
- Benefits: Student models generalize better, often outperforming direct training under tight resource limits.

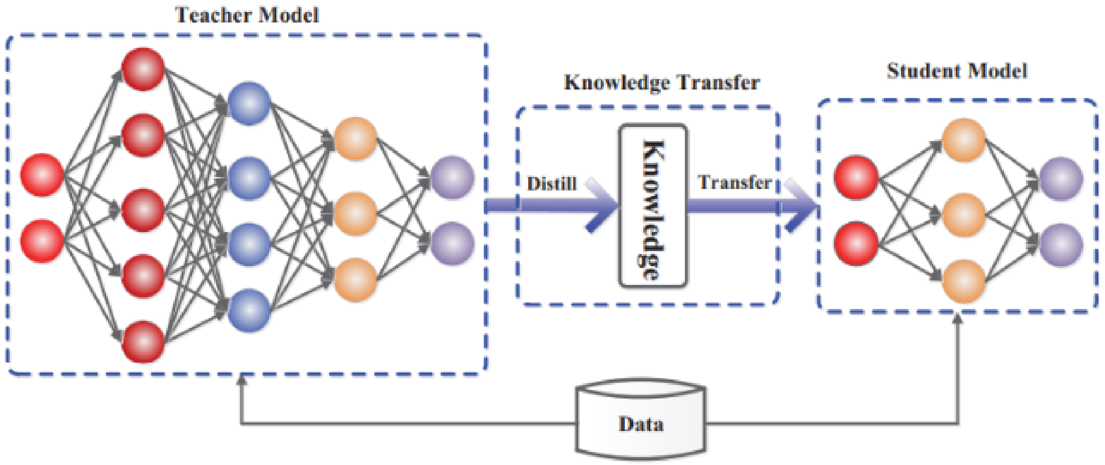


Fig6. Transfer of knowledge from teacher model to a student model from The Big Impact of Small AI: A Guide to Small Language Models

Hardware Alternatives for Edge AI Deployment

- Edge AI Hardware: Includes CPUs, GPUs, NPUs, FPGAs, and custom ASICs (e.g., NVIDIA Jetson, Google Edge TPU, Qualcomm Snapdragon).
- Selection Criteria: Performance (TOPS), power consumption, size, and supported accelerators.
- Trends: Growth in heterogeneous SoC integration for multi-modal processing.

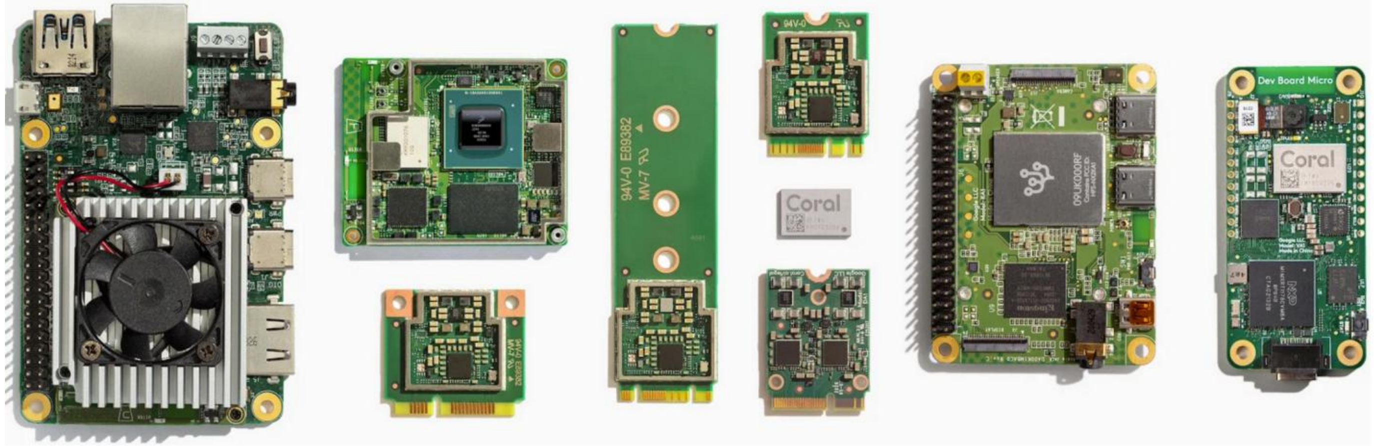


Fig7. Edge AI devices

Compiler Toolchains for Edge AI Optimization

- Compiler Toolchains: Enhance model execution by converting, optimizing, and deploying AI models for target edge hardware.
- Examples: TensorFlow Lite, ONNX Runtime, Apache TVM, vendor-specific compilers.
- Optimization: Quantization, operator fusion, graph pruning, hardware delegation.
- Role: Ensures models leverage vendor-specific accelerators and meet deployment constraints.

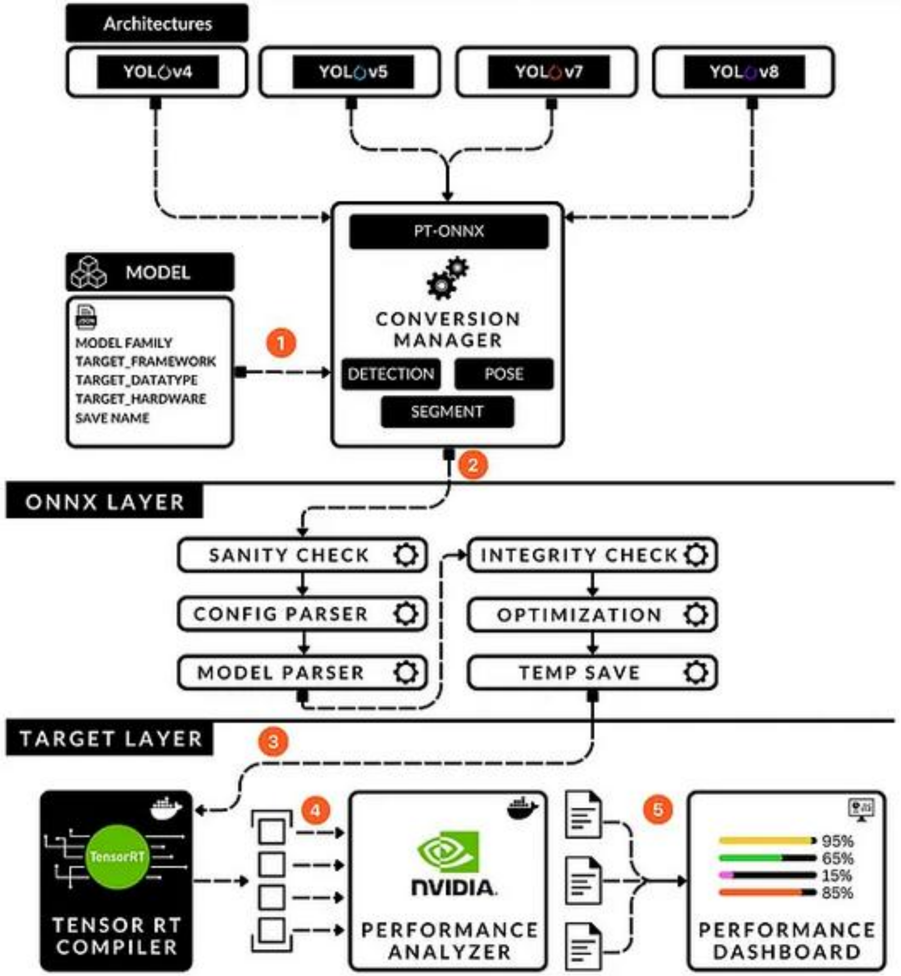


Fig8. Stages of a model from The end-to-end PyTorch to TensorRT pipeline for YOLO models

Benchmarking Edge AI Models and Platforms

- Benchmarking: Essential for evaluating inference latency, memory usage, throughput, and energy consumption on actual edge platforms.
- Metrics: Initialization time, memory usage, peak power draw, steady-state inference time.
- Key Tools: Vendor SDKs, open-source toolkits, custom scripts, Edge Impulse Studio.
- Importance: Guides design and optimization by exposing real-world device performance and limitations.

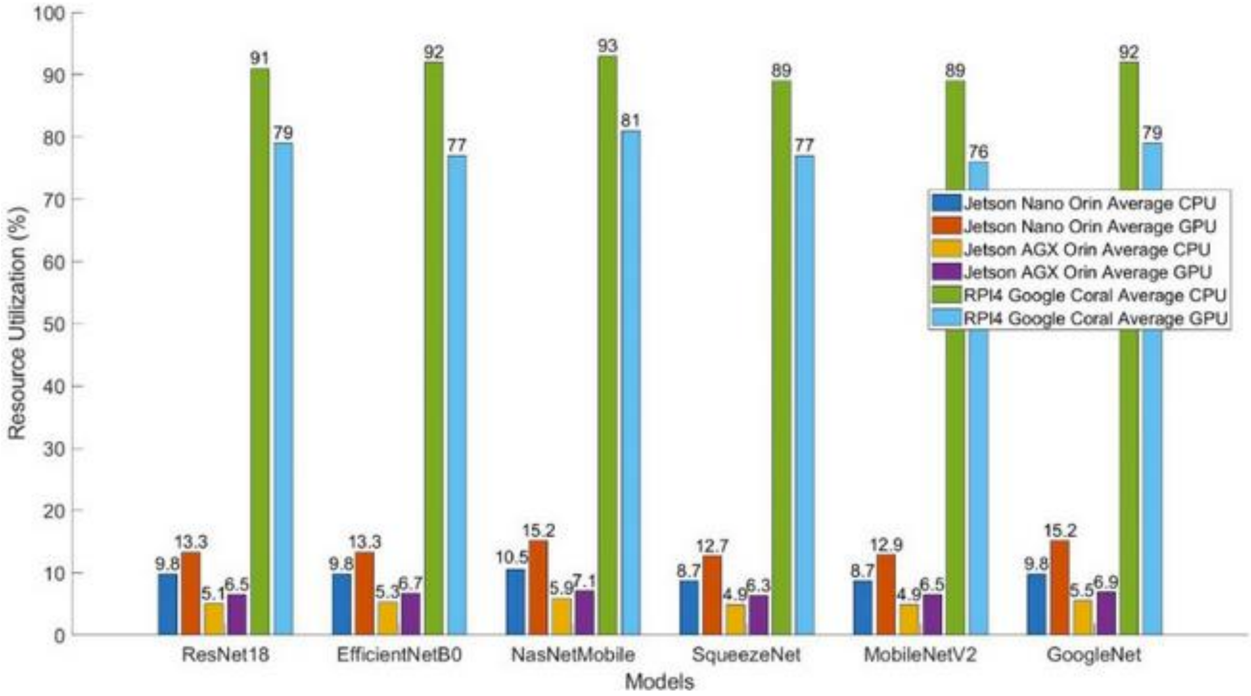


Fig9. Resource utilization on different platforms from Standalone edge AI-based solution for Tomato diseases detection

Edge AI Frameworks and Development Kits

- Popular Frameworks: TensorFlow Lite, PyTorch (w/ TorchScript), ONNX Runtime, Apache TVM, Edge Impulse.
- Features: Model conversion, quantization, hardware abstraction, cross-platform deployment.
- Ecosystem: Libraries, developer tools, model zoos, documentation, and community support.
- Emergence of Vendor SDKs: Chip vendors provide optimized SDKs for faster development and integration.

Future Trends and Case Studies in Edge AI

- Emerging Trends: Federated learning, privacy-first AI, self-optimizing edge models, neuromorphic chips, autonomous multi-agent systems.
- Case Studies:
  - Autonomous Drones: Real-time adaptation with compact CNNs onboard.
  - Smart Hospitals: Edge AI supporting privacy and adaptive intelligence in patient care.
  - Industrial Robots: Predictive maintenance using streamlined transformers.

Q
A