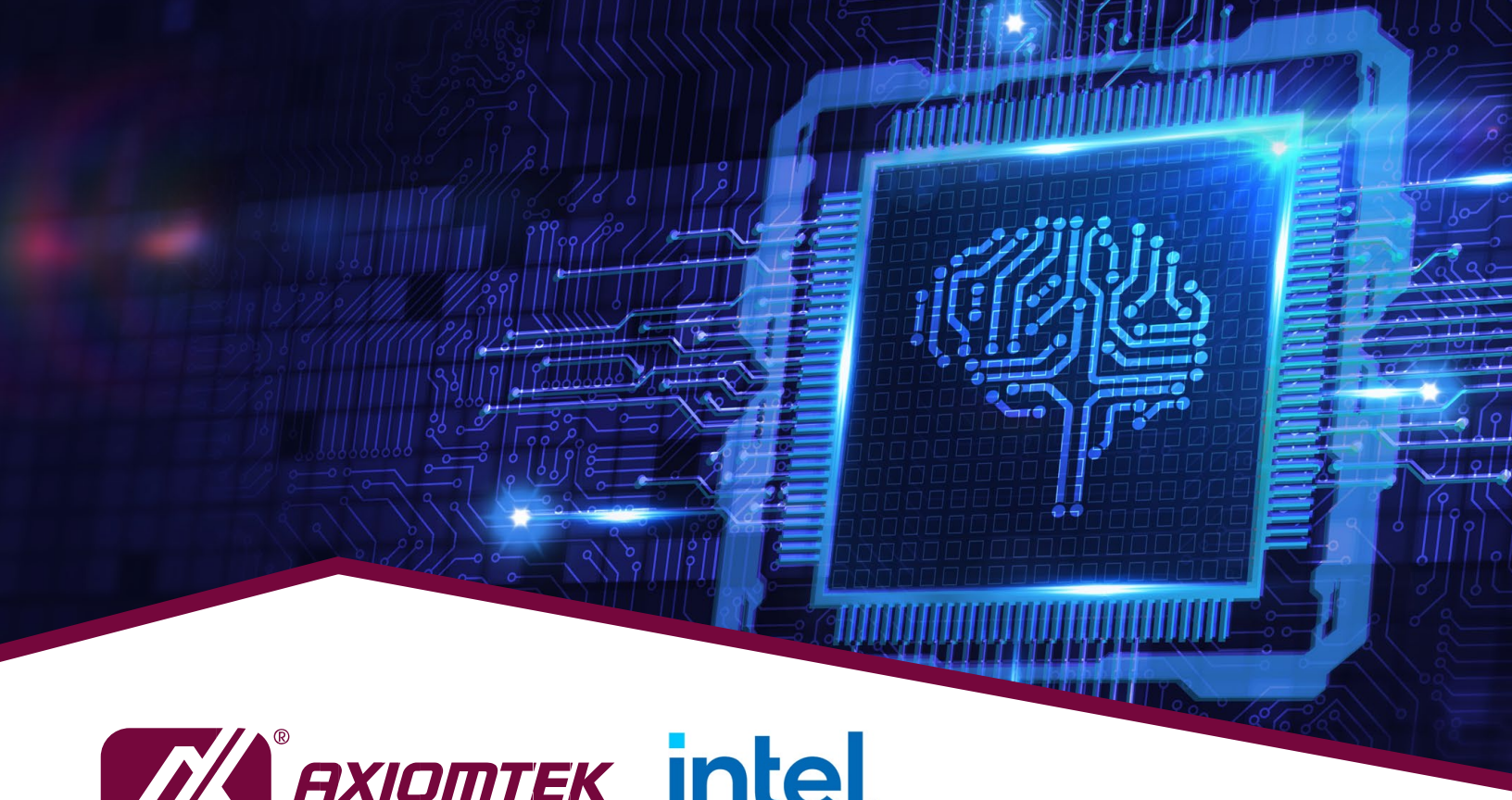

CHOOSING THE RIGHT EDGE AI HARDWARE ARCHITECTURE

By Abhishek Jadhav

Engineers selecting edge AI hardware often struggle with a mismatch between workload characteristics and hardware behavior. The challenge is meeting peak performance metrics and ensuring that the hardware can efficiently handle the specific mix of neural network inference, control logic, and data movement required by the application. In practice, system performance is shaped by how these parameters operate under real-world conditions, where memory bandwidth, data transfer overhead, and execution timing can become limiting factors.

An edge platform that delivers high TOPS can still fail due to memory bottlenecks, data movement overhead, and inability to meet latency requirements. Even over-provisioning compute can introduce unnecessary power consumption and thermal complexity. The result is a growing gap between theoretical capability and deployable system performance.

Therefore, choosing the right edge AI hardware architecture is a system design problem. It requires understanding how compute and data flow interact under deployment conditions. In this whitepaper, we analyze four different approaches that suit hybrid-compute workloads, rugged and power-constrained deployments, complex autonomous systems, and high-throughput deep learning. The analysis is also supported by measured inference benchmark data based on YOLOv8 model variants evaluated under consistent test configurations.

HYBRID COMPUTE PLATFORMS

The heterogeneous approach integrates CPU, GPU, and NPU, which allocates the right processor to each task. NPUs handle the bulk of neural network inference with low latency; GPUs handle complex-vision pre- or post-processing; and CPUs manage control logic

and orchestration. The result is a compute pipeline capable of meeting timing deadlines within a limited power and thermal budget.

Edge processors like the Intel® Core™ Ultra processor series introduce a tri-compute architecture. They combine (up to 16) high-performance x86 cores with an integrated Intel® Arc™ GPU and Intel® AI Boost NPU on a single SoC. The NPU delivers on the order of 10-13 TOPS for low-power AI inference, while the integrated Arc™ GPU provides hundreds of execution units capable of accelerating parallel workloads, such as vision preprocessing and general-purpose compute through Xe architecture.

This design allows concurrent execution of AI and non-AI workloads across multiple compute engines. Fully utilizing this architecture requires explicit workload partitioning through a compatible software stack, such as OpenVINO™ toolkit that supports both AUTO

device selection and heterogeneous execution modes to distribute inference across CPU, GPU, and NPU.

The Axiomtek eBOX630B edge system, built on an Intel® Core™ Ultra 7, serves as the reference platform for this architecture. The benchmark data below reflects measured inference throughput across YOLO model variants at 720p, batch size 1, across all three compute units. It leverages this hybrid compute architecture to boost AI at the edge.

Model variant	Precision	CPU (FPS)	GPU (FPS)	NPU (FPS)
YOLOv8n	INT8	62.19	200.96	130.22
YOLOv8n	FP16	13.27	169.59	131.96
YOLOv8n	FP32	10.01	168.34	132.19
YOLOv8s	INT8	21.25	134.46	84.81
YOLOv8s	FP16	2.55	81.34	68.99
YOLOv8s	FP32	2.27	85.01	68.87
YOLOv8m	INT8	5.34	63.38	48.49
YOLOv8m	FP16	0.82	35.29	31.99
YOLOv8m	FP32	0.82	34.04	32.05

Table 1: YOLOv8 inference performance (720p, batch size=1) evaluated individually on the CPU, GPU, and NPU of the eBOX630B featuring Intel® Core™ Ultra 7 155H

The performance data presented above is based on batch size = 1 configuration that shows latency-sensitive deployment scenarios such as processing a single video or camera stream. The setup prioritizes minimal end-to-end latency. In applications where latency constraints are relaxed, multiple input streams can be synchronized in software (batch size > 1) to improve throughput and hardware utilization.

The shared memory hierarchy also presents a key design advantage. When the compute units access the same system memory, data can be shared across them without incurring the latency and overhead. This enables zero-copy data movement that is important for latency-sensitive workloads, such as real-time video analytics. When the pipeline is structured to assign tasks based on each compute unit's memory access pattern and execution strengths, the architecture delivers a structural latency advantage over systems where compute and memory are separated across external interfaces.

RUGGED AND POWER-CONSTRAINED ENVIRONMENTS

In remote deployments where power consumption can affect the device performance, engineers want to prioritize on maximizing performance per watt. These platforms are often based around energy-efficient system-on-chips, paired with dedicated low-power AI accelerators. While instruction set architecture has more influence on the design choices, overall efficiency is determined by microarchitecture and model optimization.

These systems combine modest CPU cores with integrated NPUs or external accelerators to deliver efficient inference within a few watts. For instance, M.2-based accelerators such as those from Hailo and Axelera provide inference performance in the range of tens of TOPS while operating within sub-5W to low single-digit watt power consumption, depending on configuration.

This power efficiency is achieved in part through reduced precision computation, such as using INT8, which also lowers compute requirements and memory footprint. Quantization-aware model optimization improves performance per watt and size of the model weights to allow more complex models to fit within the limited on-chip SRAM of edge accelerators. These architectures are therefore well-suited for distributed and always-on applications, such as smart city sensors, environmental monitoring systems, precision agriculture deployments, and battery-powered edge devices.

From a design perspective, these platforms put an emphasis on conduction cooling over active cooling, with extended temperature operation, and wide input voltage ranges to withstand power fluctuation and support industrial environments. The reduced thermal footprint allows sealed enclosures to improve reliability in harsh conditions.

Axiomtek's AIM101 edge platform reflects this design philosophy by combining low-power processors with support for modular M.2 AI accelerators to enable flexible scaling of inference performance within a compact and energy-efficient design.

Model variant	Axelera (FPS)	DeepX (FPS)	MemryX (FPS)	Hailo (FPS)
YOLOv8n	423.29	249.72	79.97	311.28
YOLOv8s	353.89	237.17	75.26	277.33
YOLOv8m	179.70	137.93	35.09	66.38
YOLOv8l	139.30	92.85	-	36.11
YOLOv8x	-	50.72	-	18.08

Table 2: YOLOv8 inference performance with M.2 accelerators (720p, batch size=1)

The low power profile of the AIM101 becomes more impactful with larger deployment scales and works to reduce total cost of ownership (TCO). The difference between 5 W and 30 W per unit becomes significant when multiplied across large deployments. For example, in a scenario where 100 smart cameras are deployed, the impact is not only on electrical cost, but on other system design requirements since lower power helps enable passive cooling and reduce thermal stress on components. Over time, this contributes to improved system reliability and longer operational life in distributed edge environments.

HIGH-END CPUS FOR ROBOTICS AND AUTONOMY

In robotics, autonomous systems, and industrial automation, the primary challenge is to ensure deterministic execution and reliable system coordination. These workloads involve complex sensor fusion, real-time control loops, and decision-making processes that are inherently control-flow dominant.

High-performance CPUs, including Intel® Core™ i7/i9, remain important to these applications due to their strong single-thread performance, large-cache hierarchies, and advanced execution capabilities. These features enable efficient handling of tasks such as multi-sensor fusion involving cameras, LiDAR, and radar, and inertial sensors, as well as localization, mapping, and motion planning. In practice, these functions are implemented within the ROS 2 framework, which serves as the standard software environment for orchestrating real-time data exchange, synchronization, and decision-making in robotics and autonomous systems.

In specialized controllers for autonomous mobile robots, the CPU manages sensor synchronization, fusion, and decision-making, while offloading perception tasks to dedicated accelerators. This separation ensures that time-critical control tasks are not affected by the variable latency associated with AI inference. Achieving deterministic execution depends on real-time operating system (RTOS) support that allows predictable task scheduling and timing guarantees.

While CPUs offer flexibility and deterministic behavior, they are not optimized for large-scale neural network execution and therefore may be paired with GPUs or NPUs in a hybrid architecture as needed.

GPU PLATFORMS FOR DEEP LEARNING

For applications that require deep learning inference, GPU-based platforms are the way to go. These include scenarios such as multi-camera video analytics, large-scale vision processing, and complex neural network inference, where parallelism can be exploited. NVIDIA offers a range of edge AI platforms, including the Jetson™ family and discrete GPU-based systems for edge servers.

The NVIDIA® Jetson AGX Orin™ module features a 12-core Arm Cortex-A78AE CPU with an NVIDIA Ampere GPU to deliver 275 TOPS of AI performance. The figure is measured using INT8 sparse workloads and the real-world performance with dense models is lower and depends on model architecture and optimization. The platform is designed to support high-throughput inference across multiple 4K video streams and large neural network models.

One advantage of the NVIDIA platform is the mature software stack. Developers can train models on large servers and deploy them to Jetson™ devices using CUDA-compatible libraries. TensorRT provides inference optimization, and the JetPack™ SDK provides a full Linux-based environment with support for AI frameworks and hardware-accelerated multimedia processing. However, deploying on the Jetson™ platform introduces Arm architecture considerations, including cross-compilation and dependency management that can add complexity to development and deployment workflows.

For example, platforms like Axiomtek's AIE900A-A0, built on NVIDIA® Jetson AGX Orin™, are designed to handle high-density inference tasks such as smart surveillance NVRs analyzing many HD camera feeds, edge video analytics servers, advanced driver-assistance systems, or inspection systems that evaluate high-res images in real-time.

Model variant	AGX Orin™ 64GB (FPS)
YOLOv8n	573.57
YOLOv8s	426.73
YOLOv8m	222.13
YOLOv8l	164.81
YOLOv8x	108.91

Table 3: YOLOv8 inference performance on Jetson AGX Orin™ (720p, batch size=1)

While these GPU platforms deliver high AI throughput, they come with increased system costs. Power consumption can reach up to 60 W for modules like Jetson AGX Orin™, and significantly higher for discrete GPUs, requiring active cooling and thermal design. In addition, Arm-based deployment introduces software overhead in toolchain management and cross-compilation. At scale, these factors impact power provisioning, management complexity and cost.

The result is that GPU architectures are more suited for applications such as multi-camera analytics, large-scale vision processing, and real-time 4K stream analysis, where high throughput and model complexity justify these tradeoffs.

FROM ARCHITECTURE TO DEPLOYMENT

Selecting the right edge AI hardware architecture is the first part of the design process. The next challenge is converting the architecture into a working inference pipeline to meet latency, accuracy, and power constraints in real deployment.

Deploying a custom-trained AI model on edge hardware introduces practical problems with model size, precision, and memory footprint. All of these parameters must match with the capabilities of the target architecture. In many edge deployments, this involves converting models to reduce precision format to improve throughput and power efficiency, but requires calibration and validation to maintain accuracy.

On Intel-based edge platforms, toolchains like the OpenVINO™ toolkit and the Intel® Edge AI SDK provide a structured path from model training to deployment.

Models trained in standard frameworks like PyTorch and TensorFlow can be converted into an intermediate representation optimized for Intel architectures.

This approach is ideal for deployments that remain within an existing x86 ecosystem where adding an additional discrete GPU may not be possible due to cost or power constraints. It is also effective for single-stream or low-concurrency inference workloads where predictable latency and efficient CPU, GPU, and NPU utilization are more important than maximizing aggregate throughput.

For instance, a custom object detection pipeline can be partitioned across compute units based on the characteristics of each stage. For example, image decoding and preprocessing can be handled on the GPU, the neural network inference on the NPU, and post-processing tasks such as non-maximum suppression or tracking logic can run on the CPU. This type of pipeline-level optimization is more impactful than raw hardware performance, as it minimizes redundant data movement and ensures that each compute operates within its efficiency range.

	HYBRID COMPUTE (CPU + GPU + NPU)	RUGGED / POWER CONSTRAINED (LOW-POWER SOC + M.2 ACCELERATOR)	HIGH-END CPU (CONTROL CENTRIC ARCHITECTURE)	GPU-BASED (HIGH-THROUGHPUT PARALLEL COMPUTE)
PRIMARY FUNCTION	Workload partitioning across heterogeneous compute	Maximize performance-per-watt using optimized inference accelerators	Deterministic execution, control logic, and real-time system coordination	Massive parallel processing for deep learning inference and high-throughput workloads
APPLICATIONS	Real-time video analytics, industrial monitoring, edge AI pipelines with mixed workloads	Smart city sensing, agriculture and battery-powered edge devices, distributed remote deployments	Robotics, autonomous systems, ROS 2-based systems, sensor fusion,	Multi-camera video analytics, 4K video processing, large neural networks, edge servers
POWER EFFICIENCY	Moderate to high efficiency due to task specific execution and shared memory	Very high efficiency (sub-5W to low single-digit watts) with passive cooling	Low efficiency for AI workloads, but efficient for control-flow tasks	Low to moderate efficiency (high power draw about 60W+) and requires active cooling
AI BENCHMARK	Balanced performance across units. For YOLOv8n model INT8, CPU: 62 fps, GPU: 200 fps, NPU: 130 fps	Strong efficiency scaling. For YOLOv8n model, up to 423 fps (Axelera), 311 fps (Hailo) at low power	No strong AI benchmark advantage alone. For example, for YOLOv8m, 0.8 to 5 fps range on CPU. They are not optimized for neural networks	Highest raw throughput. For the YOLOv8n model on AGX Orin™ 64GB, 573 fps, and the YOLOv8x model has 108 fps. They are optimized for batch and multi-stream workloads

Software toolchain compatibility is as important a selection criterion as the hardware specifications themselves. Axiomtek provides practical reference workflows, including YOLO-based training guides and Intel platform-specific SDK documentation, to support deployment from dataset preparation through benchmarking on real hardware. In addition to Intel-based toolchains, Axiomtek also provides deployment samples and integration support for third-party accelerator SDKs to enable developers to work across a broad range of hardware platforms.

CONCLUSION

There is no one-size-fits-all edge hardware architecture. The right choice depends on a combination of latency requirements, throughput demands, power and thermal constraints, model characteristics, and system-level limitations of memory bandwidth and I/O capacity.

Effective design starts with an understanding of the workload, including data flow, timing requirements, and compute distribution. From there, tasks can be mapped to the most suitable compute units, and different platforms can be evaluated based on performance in real-world deployments. Equally important is the software stack, which plays a crucial role in enabling efficient hardware utilization.

By aligning workload requirements with the appropriate architecture, engineering teams can make informed design decisions grounded in real deployment constraints. The benchmark data presented throughout this whitepaper provides a practical starting point for evaluating how these architectures perform under representative workloads.

From there, the next step is to map application requirements to the appropriate hardware and software stack. It could be by leveraging platform-specific toolchains such as OpenVINO™ or accelerator SDKs or by working with deployment resources to validate performance in real-world conditions.

» Axiomtek offers a range of industrial edge AI platforms built around the architectures covered in this whitepaper — from Intel hybrid compute systems to low-power M.2 accelerator platforms and GPU-based inference hardware. Our U.S.-based engineering team can help you identify the right platform for your application and support customization, integration, and deployment.

Let's build something together. Connect with our team and reach out to us at **solutions@axiomtek.com**

AXIOMTEK®

intel prestige partner