University of Maribor Press
DOI: https://doi.org/10.18690/um.fov.5.2026.2
ISBN: 978-961-299-160-9

Edge AI: Small Language Models on the Go

Joachim J. Włodarz

Abstract
The proliferation of edge devices, ranging from smartphones and various wearable devices, up to industrial sensors or autonomous vehicles, gives an opportunity to leverage the power of AI-based methods directly at the point where data is acquired or generated. However, deploying traditional Large Language Models (LLMs) on resource-constrained edge devices becomes impractical due to substantial computational and memory requirements. In this contribution, the rapidly evolving field of Edge AI is explored, specifically focusing on the development and deployment of Small Language Models (SLMs), optimized for edge environments. The various challenges and opportunities associated with SLMs are indicated, together with a review of the current techniques for model compression and optimization. An outline of future research and development is also given.

Keywords: edge AI; small language model; model compression; quantization; pruning and federated learning; on-device AI; resource-constrained devices

Joachim J. Włodarz
University of Silesia, Faculty of Science and Technology, Bankowa 14, 40-007 Katowice, Poland
e-mail: joachim.wlodarz@us.edu.pl

1 Introduction

The proliferation of Internet of Things (IoT) devices [1], coupled with the growing demand for real-time and personalized user experiences, has significantly accelerated the advancement of edge computing. Edge AI, which involves executing artificial intelligence applications directly on edge devices, offers notable advantages when compared to traditional networked or cloud-based solutions [2]. These advantages include reduced latency, enhanced data privacy and improved reliability - especially in environments with limited connectivity - as well as a decrease in bandwidth consumption. It also makes it possible to work completely offline, without access to any network.

Historically, the computational requirements of advanced AI models, particularly Large Language Models (LLMs) [3], have constrained their deployment on edge devices. However, recent breakthroughs in model compression techniques and the emergence of Small Language Models (SLMs) [4] have stimulated a major shift within this landscape.

Ultra-compact AI models are increasingly deployed on edge devices with minimal resources, even with just a few kilobytes of memory onboard [5]. It makes it possible to process data locally on commodity devices such as sensors, wearables, and home appliances. This innovative approach yields considerable benefits in energy efficiency, enabling such devices to function for years on a single battery. Additionally, it offers near-instantaneous response times and enhanced privacy, as data processing occurs on-device without the need to transmit information externally.

In this contribution, it will be shown that SLMs are a vital enabler for Edge AI, facilitating the incorporation of robust language capabilities across a myriad of applications that were once considered impractical. The specific challenges and opportunities inherent in this field will be explored, offering a comprehensive overview of current solutions and highlighting potential future directions.

2 LLMs and resource constraints

Traditional LLMs, like the GPT model series introduced by OpenAI [6], have demonstrated impressive capabilities, but they are also characterized by immense size – hundreds of billions of parameters - and huge computational complexity. Deploying these models on edge devices presents several significant hurdles. Namely, their memory footprint often exceeds by orders of magnitude the capacity of edge devices. Moreover, the computational cost of inference demands significant processing power, leading to high energy consumption and resulting in slow response times on resource-constrained hardware. Furthermore, transferring data between the edge devices and the cloud can be bandwidth-intensive and unreliable, not to mention that sending sensitive data to the cloud for processing raises privacy and security concerns.

The recent shortage of RAM chips has significantly altered the Edge AI landscape. This scarcity is prompting a crucial architectural shift, as developers increasingly adopt “DRAM-less” hardware designs, such as the Hailo-8 and Hailo-8L accelerators, which

enable the execution of AI models entirely on-chip. In addition to hardware adaptations, this shortage has made model optimization a necessity, accelerating further the transition from LLMs to more efficient SLMs. The availability of currently used LPDDR5X and DDR5 chips for edge systems is anticipated to remain critically constrained at least through the next year. In consequence, the AI scene becomes effectively bifurcated into the "high end" part enjoying priority, and the "everyone else adapts" part, where repurposed or even refurbished hardware and also various microcontroller based designs become more and more attractive for Edge AI projects.

3 Small language models

Small Language Models (SLMs) are typically defined as models with only a few billion parameters, offering a compelling alternative to traditional LLMs for Edge AI applications. While they may exhibit reduced performance on certain tasks compared to their larger counterparts, their smaller size and much lower computational requirements make them significantly more suitable for deployment on edge devices.

Several SLMs have emerged recently, each demonstrating a balance between performance and efficiency. They are the primary driver for Edge AI, enabling "intelligence" on devices like smartphones, IoT sensors, or medical equipment. The following models are widely used for on-device tasks:

- Google Gemma 3 / 3n [7]: A multimodal family (text, image, audio) with variants like Gemma 3n 1B that can reach speeds over 2,500 tokens/second on mobile GPUs.
- Microsoft Phi-4 Mini / Phi-4 [8, 9]: The 3.8B parameter Mini is highly optimized for reasoning and coding, while Phi-4 (14B) pushes the upper limit of what "small" entails.
- Meta Llama 3.2 (1B & 3B) [10]: Specifically designed for mobile and edge performance with high accuracy-to-size ratios.
- Qwen 2.5 (1.5B) [11]: An Alibaba-developed model popular for multilingual edge applications.
- TinyLlama 1.1B [12]: Compact and fast open-source model.
- Shakti Family (100M–500M) [13]: Specialized ultra-compact models designed for domain-specific tasks like legal or medical analysis on tiny hardware.
- SmolLM2 (135M & 360M & 1.7B) [14]: A new state-of-the-art family of SLMs.

4 Model compression and optimization

To further enhance the suitability of SLMs for Edge AI, various model compression and optimization techniques are being employed. For example, quantization reduces the precision of model weights and activations, e.g. from 32-bit to 4-bit integers, significantly lowering the memory footprint and improving inference speed [15, 17]. Instead of applying one precision level to the whole model, Mixed-Precision Quantization or

Activation-Aware Quantization methods could be used [17, 18, 19]. These methods assign higher precision to "sensitive" layers that impact accuracy the most, and lower the precision (e.g., 2-bit or 4-bit) to others. AWQ is a preferred method for SLMs because it protects the most important weights during quantization, maintaining at the same time a higher reasoning accuracy than the older methods like GPTQ. Other techniques like Post-Training Quantization (PTQ) or Quantization-Aware Training (QAT) are also commonly used [20].

The model size and computational complexity could be substantially reduced by pruning, the removal of redundant or less important connections in the neural network [21]. Structured pruning is often preferred for more effective hardware acceleration. Rather than removing individual weights, which can be expensive for hardware to optimize, structured pruning removes entire blocks or layers [22]. Algorithms like SparseGPT [23] and Wanda [24] are usually used to prune models after training, allowing a model to retain its "intelligence" while shrinking its footprint by 20–50%.

Knowledge distillation, where a smaller "student" model mimics the behavior of a larger "teacher" model, allows the developed model to inherit the knowledge of the teacher model while maintaining a significantly smaller size [25]. Some models use iterative refinement [26, 27], a "self-distillation" procedure to "teach themselves", to improve performance without requiring an external teacher model and a separate distillation procedure.

Low-rank factorization [28, 29] decomposes weight matrices into lower-rank approximations, reducing the number of parameters, and hardware-aware optimization tailors model architectures and optimization techniques to the specific hardware capabilities of the edge device.

Speculative decoding (SD) is another inference acceleration technique that uses a tiny "draft" model alongside the primary SLM [30]. The tiny draft model quickly predicts several future tokens. The primary SLM verifies then these tokens in a single parallel step, rather than generating them one by one. Frameworks like SLED (Speculative LLM Decoding) [31] can increase system throughput by up to 2.8x on popular SMB hardware like Raspberry Pi 5 or NVIDIA Jetson.

These techniques are often combined into a single pipeline to fit models into memory-constrained environments. Modern SLMs are no longer just "shrunken" versions of their "big" counterparts, being rather designed specifically for the respective edge hardware (Hardware-Aware Architecture) [32].

5 Typical applications, use cases and benefits

Recently, Small Language Models (SLMs) and their applications have transitioned from experimental pilot designs to industry standards, with roughly 80% of AI inference now occurring locally on edge devices according to industry estimates [34]. Their ability to provide real-time, private, and offline intelligence resulted in diverse use cases.

In healthcare, portable and wearable devices use SLMs to analyze heart rhythms and lung sounds locally, alerting emergency services when a verified anomaly is detected [35]. Devices like digital stethoscopes transcribe sounds and flag potential abnor-

malities instantly, enabling clinical workflow acceleration. Direct on-device diagnostic [36, 37] eliminates concerns about patient data confidentiality by processing sensitive information locally within the hospital's secure network.

In industrial environments, the use of Edge AI enables predictive maintenance, e.g. sensors can analyze vibrations and temperatures in real-time, predicting imminent equipment failures before they occur and reducing unplanned downtime significantly [38]. High-speed vision systems using local SLMs could be used for inspection on production lines in real time, ensuring immediate rejection of faulty items when necessary.

Millisecond-level navigation decisions are crucial in the case of autonomous vehicles [39]. Edge AI enables dynamic traffic management, where edge-native AI analyzes local sensor data at intersections to adjust signals in real-time, reducing congestion and emissions. Autonomous vehicles process terabytes of sensor data locally to support safety features, such as pedestrian detection and truck platooning, for which cloud-access-related latency could be fatal.

The environmental sector faces immense challenges, requiring constant monitoring, analysis, and informed decision-making. Traditional cloud-based AI solutions often struggle to meet the specific needs of environmental applications due to latency, bandwidth limitations, and power constraints. Edge AI, particularly leveraging SLMs, offers a substantially improved approach [40]. Acoustic monitoring in wildlife reserves, for instance, can utilize SLMs on low-power devices to identify species presence, or detect illegal activities without transmitting large audio files. Similarly, camera trap analysis can be performed locally, automatically identifying the animals and detecting unusual behaviors [41].

Precision agriculture could benefit from SLMs analyzing crop health in real-time, enabling targeted action and reducing pesticide use [42]. Air quality monitoring stations, equipped with sensors and SLMs, can analyze data and provide alerts when pollutant levels exceed safe thresholds. The ability to perform these analyses on-device eliminates the need for constant cloud connectivity and reduces power consumption, making Edge AI with SLMs a powerful tool for sustainable environmental management. Such devices may also be used for real-time monitoring of air and water quality, triggering instant alerts when spikes of hazardous concentrations are detected.

In the case of consumer electronics, SLMs enable "always-on" virtual assistance, that handle tasks like document summarization, email drafting, and real-time translation entirely offline [43]. Smart Home devices like thermostats can use local SLMs to learn user behavior and habits, optimizing energy usage without sending personal routines to the cloud. On a larger scale, SLMs could be also seen as enablers of locally tuned and democratically aligned intelligence that can better serve urban equity and efficiency goals [44].

By processing data at the source, Edge AI provides a "Privacy-by-Design" architecture [45], essential for meeting the data regulations in many countries. Sensitive information, such as medical records or financial transactions, never leaves the device, eliminating the risk of interception during cloud transit. Since data is processed locally, there is no centralized data center for hackers to target, significantly lowering the impact of large-scale breaches. Local processing by SLMs also eliminates the 50–200ms "round-trip" delay typical of cloud servers. Usually SLMs can deliver sub–5ms latency,

which is very important for time-sensitive applications. Additionally, the performance then remains stable, regardless of network congestion or server load.

Shifting tasks from cloud LLMs to local SLMs can reduce the inference costs by up to 90% (from ~$0.50 in the cloud to ~$0.05 on-device) [46, 47]. Local processing can also drastically reduce the volume of raw data sent over networks, lowering recurring telecommunication and data storage expenses. It could also ensure more reliability in environments where internet connectivity is impossible or unreliable, e.g. within coal mines.

Specialized SLMs could be optimized for standard hardware, allowing companies to scale AI without expensive and power hungry GPU clusters. Local inference reduces the massive energy demands and carbon footprints associated with running large-scale data centers. SLMs consume much less power than un-optimized much bigger models, also allowing battery-powered IoT devices to operate for days rather than hours. It is especially important in field operations, enabling robust performance in remote areas, e.g. for emergency responders in disaster zones, or industrial workers on factory floors with spotty Wi-Fi. Smart home devices and medical wearables could therefore continue to function during internet outages, maintaining essential safety and monitoring services.

6 Hardware platforms for edge AI with SLMs

While the advancements in software and in the algorithms for SLMs are undoubtedly noteworthy, it is essential to recognize that the hardware platforms supporting these technologies hold equal, if not greater, importance. This is particularly true given the rapid progression in hardware capabilities, which often outpaces significantly the improvements made in software [48]. The choice of a device depends on various factors such as computational requirements, power constraints, cost, and environmental conditions, with options ranging from very low-power microcontrollers to high-performance GPUs and specialized NPUs. The decentralized model of operation eliminates the necessity for continuous cloud connectivity, thereby providing advantages such as reduced latency, improved privacy, and decreased bandwidth expenses. Single-Board Computers (SBCs), hosting different CPUs and sometimes equipped with GPUs or NPUs, often serve as Edge AI platforms, facilitating the execution of artificial intelligence algorithms on compact and energy-efficient devices. Another choice could be repurposed Thin-Client Computers (TCCs), equipped often with surprisingly efficient hardware at a fraction of the cost of specialized industrial buildups. Refurbished older models typically cost only a fraction of the price of similar new equipment.

In resource-constrained embedded AI systems, microcontrollers (MCUs) are increasingly employed. Quite often they are enhanced with hardware accelerators such as dedicated Neural Processing Units (NPUs). These components enable the local execution of complex tasks, including vision and voice recognition. Unlike conventional MCUs, AI-optimized variants maintain a delicate balance between low power consumption and high computational capability, facilitating the operation of lightweight models such as TinyML [5]. Typically, MCUs exhibit a higher level of specialization, making

them more ”tailored” for specific applications when compared to single-board computers (SBCs) or Thin-Client Computers (TCCs).

6.1 Edge AI SBCs

The Raspberry Pi SBCs [49], particularly the Raspberry Pi 4 and Raspberry Pi 5, together with the respective Computing Modules (CM), have emerged as a popular choice for Edge AI buildups due to their affordability, versatility, and extensive community support [50]. These devices can easily serve as a cornerstone for accessible Edge AI, ranging from simple hobbyist projects to platforms capable of handling generative AI workloads. It is worth mentioning that the dot product (DotProd) and half-precision floating point (FP16) arithmetic instructions, introduced in the ARMv8.2 CPU family ISAs [51], could provide the mathematical efficiency needed for modern transformers running e.g. on Raspberry Pi 5. The usage of FP16 allows the CPU to process twice as many data points per clock cycle when compared to FP32, while using the same SIMD (Single Instruction, Multiple Data) registers. Moreover, since FP16 weights are half the size of FP32, one can fit twice as many weights into the CPU cache, drastically reducing the time CPU spends waiting for data to be processed. Unlike 8-bit integers, FP16 maintains enough dynamic range to run models like Llama 3.2 or Gemma 3 with virtually zero accuracy loss when compared to their cloud-based versions. The DotProd instruction performs a 4-way 8-bit integer dot product and accumulates the result into a 32-bit integer in a single instruction. For quantized models (INT8), DotProd provides a theoretical 4x speedup over standard integer math. This is the primary reason why a Raspberry Pi 5 (ARMv8.2) is significantly faster than a Raspberry Pi 4 (ARMv8.0) for AI tasks at similar clock speeds. Because the hardware executes these operations in fewer cycles, it also reduces the ”energy-per-token,” which is especially important for battery-operated edge devices. For example, this allows the Raspberry Pi 5 to run 4-bit quantized SLMs like Phi-3.5 Mini at 10–15 tokens per second, making the difference between ”laggy” text (~2 tps) and “real-time” text (10+ tps) generation.

While early SLM use on Raspberry Pi relied solely on the CPU, the platform can now utilize dedicated Neural Processing Units (NPUs) to achieve better performance. The newest Raspberry Pi AI HAT+ 2, launched in January 2026, features the Hailo-10H chip, delivering up to 40 TOPS of performance [52]. Unlike previous versions, the AI HAT+ 2 includes 8GB of dedicated RAM, allowing it to handle SLMs and Vision-Language Models (VLMs) independently of the main Raspberry Pi board. However, intensive SLM workloads would require an active cooler to prevent thermal throttling.

The NVIDIA Jetson Nano SBC offers a significant step up in processing power, particularly for GPU-accelerated tasks, though at a higher cost and power consumption [53, 54]. The NVIDIA Jetson Nano is considered a foundational, but nowadays legacy platform for edge AI. While it still offers valuable learning opportunities for computer vision, its limited memory and outdated software stack make it impractical for running modern SLMs effectively. Even when highly optimized and quantized, they require nowadays more memory for the model weights, tokenizer, and KV cache, often leading to system crashes or severe performance degradation (swap thrashing). The NVIDIA

Jetson Nano SBC uses a 128-core NVIDIA Maxwell™ GPU architecture from 2019. This architecture lacks the Tensor Cores found in newer generations (like Orin or Thor) that are essential for accelerating the matrix multiplication operations that dominate SLM inference. NVIDIA Jetson Orin Nano / Orin Nano Super are the official successors, offering significantly more AI performance and memory options (8GB/16GB). Through a "Super Mode" software update in JetPack 6.2, it delivers up to 67 TOPS (up from 40 TOPS) and a 1.7x performance boost for SLMs [55].

NVIDIA Jetson AGX Thor is the current high-end platform for advanced robotics, providing massive compute power for complex agentic AI workflows. Launched in late 2025/early 2026, it is now the premier platform for robotics and humanoid AI. Powered by the NVIDIA Blackwell™ GPU architecture, it delivers up to 2070 FP4 TFLOPS of AI compute over 7.5x the performance of the previous Orin generation [56], but at the cost of a disproportionate price increase.

6.2 Edge AI TCCs

Repurposing PC terminal hardware, including thin client computers (TCCs) and point-of-sale systems, is increasingly recognized as a sustainable and cost-effective alternative to single-board computers (SBCs) or specialized industrial computers [57]. These devices often feature surprisingly capable processors and substantial amounts of RAM, making them well-suited for Edge AI deployments. This approach not only extends the devices' lifespan but also contributes to reducing electronic waste.

The trend of utilizing such devices as "Edge AI Gateways" has gained traction, particularly as they are commonly available in corporate surplus markets. Typically fanless and durable, these systems offer greater performance than microcontrollers while remaining more affordable than NVIDIA Jetson kits. For an extensive overview, including the devices indicated below, please refer to the ParkyTowers online service [58].

Thin clients from reputable brands such as HP (t-series), Dell (Wyse), or Lenovo (e.g., the ThinkCentre M-series) are exceptionally well-suited for the requirements of SLMs. It is advisable to opt for newer models that feature either the AMD Ryzen Embedded or Intel Elkhart Lake or newer chip families, as these processors incorporate advanced instruction sets, including AVX2, which significantly enhance the matrix calculations necessary for SLM applications. Unlike many Single Board Computers (SBCs), thin clients generally include SODIMM slots, allowing for straightforward expansion up to 16GB or even 32GB of RAM. This capability facilitates the execution of more substantial 7B or 14B models, such as Phi-4 or Llama 3.2 3B, which may otherwise fail to run on platforms like the Raspberry Pi 5. Additionally, thin clients typically offer a wider array of I/O interfaces compared to SBCs, further enhancing their versatility and functionality in professional settings.

For instance, the HP Pro t640 Thin Client is powered by the AMD Ryzen™ Embedded R1505G processor, which boasts 2 cores and 4 threads with a maximum boost frequency of 3.3 GHz, and is also equipped with Radeon™ Vega 3 Graphics. It offers exceptional performance, particularly when upgraded to 16GB or 32GB of RAM (DDR4). This compact, fanless device is designed for continuous operation, provid-

ing enterprise-grade durability. Furthermore, the default M.2 flash memory can be upgraded to a higher-capacity NVMe SSD, allowing for the storage of multiple models and vector databases. The integrated AMD Radeon™ Vega 3 Graphics also facilitates light GPU acceleration for AI applications, enhancing overall processing capabilities. Two of the four integrated USB ports support speeds up to 10 Gbps (USB 3.2 Gen 2).

The newer HP Pro t550 Thin Client from the same family features an Intel Celeron™ J6412 processor, a part of the Intel Elkhart Lake family, which includes four cores and four threads, capable of reaching a burst frequency of up to 2.6 GHz. This model is enhanced by the integration of Intel GNA 2.0 (Gaussian & Neural Accelerator), an ultra-low-power AI co-processor specifically designed for continuous, "always-on" background operations [59].

Distinct from traditional GPUs and CPUs, the GNA is optimized for low-precision integer arithmetic, rendering it highly efficient for particular Small Language Model components. In the context of the HP Pro t550, the GNA functions primarily as a "gatekeeper" or "pre-processor", effectively conserving energy and CPU resources. It is capable of monitoring for specific trigger phrases or recognizing human speech patterns with minimal power consumption, utilizing microwatts. The more power-intensive Celeron™ CPU cores are then only activated when a valid command is identified.

Additionally, the GNA can serve as a neural filtering hardware device, adept at eliminating background noise, such as that generated by fans or traffic, to deliver a "clean" signal to the model, subsequently enhancing accuracy. Another potential application includes continuous biometric monitoring, where the GNA can execute compact neural networks for tasks like speaker identification or heart-rate analysis directly from sensors without placing undue stress on the main system. It is important to note that the GNA 2.0 requires models to be quantized to INT8 format. The Intel OpenVINO™ Neural Network Compression Framework (NNCF) can be utilized to convert models like Phi-3 Mini or TinyLlama into this optimal format [60].

6.3 Edge AI MCUs

Microcontrollers have emerged as essential components in the realm of Edge AI, facilitating the execution of artificial intelligence algorithms on low-power devices [5]. This capability for localized processing enables immediate decision-making with minimal latency, enhancing user privacy and lowering energy consumption in comparison to traditional cloud-based solutions.

In practical applications, these devices are proficient in managing specialized functions such as predictive maintenance. For example, by analyzing data related to vibrations or temperature, they can identify potential equipment failures in industrial settings before they escalate. Furthermore, microcontrollers support computer vision applications, enabling gesture recognition and defect detection through the utilization of optimized models like MobileNet [62]. In the consumer sector, they are an integral part of voice-activated keyword spotting in smart home devices and the monitoring of vital signs in healthcare wearables, which can detect various irregularities, such as heart arrhythmias. Agriculture also reaps the benefits of these technologies, as microcontrollers

facilitate the monitoring of soil quality and the identification of pests via field-deployed cameras.

The hardware supporting these diverse applications varies based on performance requirements and also budget considerations. Products like the STMicro STM32 series are designed for high-performance image processing and demanding machine learning tasks, typically incorporating dedicated hardware accelerators [63]. For more economically sensitive IoT initiatives, the ESP32-S3 [64] provides AI-extended instructions at a more accessible price point, while Texas Instruments (TI) chips prioritize low-latency performance for high-speed controls [65].

To effectively bridge the divide between AI models and MCU hardware, specialized development tools are essential. Frameworks such as LiteRT for Microcontrollers [67], known also as TensorFlow Lite for Microcontrollers, empower developers to compress and quantize models, ensuring compatibility with the limited memory constraints of microcontrollers. Additionally, manufacturer–specific utilities such as the STM32Cube AI Studio [66], facilitate the direct conversion of neural networks into optimized C code, simplifying the deployment of advanced intelligence even on the most compact hardware setups.

When choosing the right hardware for AI on microcontrollers, the decision largely depends on the specific requirements of the project. Factors such as the need for high-performance vision capabilities, integrated wireless connectivity, or ultra-low power consumption for extended battery life usually play pivotal roles here.

For tasks demanding intensive processing, such as real-time image recognition and video processing, the STMicroelectronics STM32N6 [68] stands out as a premier option due to its dedicated Neural-ART Accelerator, with an impressive 600 GOPS of computing power. Likewise, the Renesas RA8M85, equipped with a Cortex-M85 core and Helium extensions, efficiently manages complex mathematical operations inherent to machine learning, surpassing older conventional designs [69]. For projects with a particular focus on gesture recognition or low-power vision applications, the Grove Vision AI Module V2, which integrates an ARM Cortex M55 alongside a micro NPU [70], excels in delivering high frame rates while consuming minimal power.

In the case of applications to smart home technology or mutually connected IoT devices, the Espressif ESP32-S3 presents a commendable balance of cost-effectiveness and AI vector instructions, making it a preferred choice within the Arduino community. More specialized IoT solutions, such as the Silicon Labs EFR32MG24, come equipped with an integrated Matrix Vector Processor tailored for AI workloads and seamless compatibility with modern smart home protocols like Matter [71]. For high-performance wearables and audio processing applications, the Nordic nRF54 Series [72] offers a robust multi-core architecture capable of handling demanding AI tasks without sacrificing stable Bluetooth connectivity.

In scenarios where prototyping or the development of battery-operated industrial sensors is required, the Raspberry Pi RP2350 MCU [73], utilized in the Pico 2 board, provides extensive community support and a low-cost entry point for exploring TinyML. For more specialized industrial or medical applications where devices need to function for years on a single charge, the STM32U5 Series remains the benchmark for ultra-low power consumption while delivering the necessary performance for fundamental anomaly detection.

The STM32 family and the Raspberry Pi Pico series MCUs exemplify two distinct tiers of embedded AI development. The STM32 ecosystem is expansive, encompassing a range of products from low-power microcontrollers to high-performance processors such as the STM32H7. Many of these higher-end models are equipped with dedicated Neural Processing Units and hardware floating-point units designed specifically to enhance the performance of complex inference tasks. ST Microelectronics complements this robust hardware with advanced software tools, that automate the conversion of standard AI models into optimized code suitable for production-grade applications.

In contrast, the Raspberry Pi Pico MCU series is aimed at creators and educational audiences, focusing on simplicity and affordability. The original Pico's capabilities are somewhat limited for AI applications, primarily due to its RP2040 chip lacking a hardware floating-point unit. As a result, it must rely on slower software emulation for math-intensive AI tasks. However, the introduction of the RP2350 MCU chip in the newer Pico 2 board significantly addresses this limitation by incorporating hardware floating-point support and digital signal processing instructions, allowing for much more efficient handling of machine learning tasks compared to its predecessor.

In summary, professionals engaged in high-speed vision or industrial sensor development will find the STM32 to be a more suitable choice due to its extensive hardware scalability and comprehensive development tools. Conversely, the Pico and Pico 2 serve as excellent platforms for rapid prototyping, hobbyist projects, and educational contexts, where ease of use through MicroPython and cost-effectiveness are prioritized over maximum processing performance.

6.4 Edge AI smartphones

Last but not least, the market is witnessing a growing availability of AI-native smartphones [61], intended for the consumer sector. These devices have evolved from traditional flagship models with supplementary smart features to those where artificial intelligence serves as the core architecture. This transformation is marked by the seamless integration of specialized hardware, including Neural Processing Units (NPUs), which enable sophisticated generative models to operate directly on the device. By processing data locally at the "edge", rather than depending on cloud servers, these smartphones deliver significantly enhanced performance and improved privacy. This capability also facilitates offline functionality, proving invaluable in situations where network connectivity is limited or unavailable.

7 Future directions

The future of Edge AI is defined by the ongoing shift from “connected devices” to “autonomous devices”. The convergence of 2-bit quantization, specialized NPU hardware, and agentic workflows is gradually moving the majority of AI workloads from massive data centers to the palm of the user’s hand. Recent breakthroughs allow SLMs to run with significantly less memory. This makes it possible to run a high-reasoning parameter model on a device with little RAM, like a smartphone or a mid-range IoT gateway. These hyper-efficient models allow for “always-on” AI that consumes less power than a standard LED bulb, enabling solar-powered environmental sensors to “think” for months without needing a recharge.

It has to be emphasized that computer architecture related issues play a crucial role in the effectiveness of Edge AI, by harmonizing rapid processing capabilities with the inherent limitations of local hardware. Given that, these devices do not possess the extensive resources available to cloud servers, but only specialized components such as Neural Processing Units (NPUs) or Application-Specific Integrated Circuits (ASICs) that are designed to perform complex AI computations in parallel. This design significantly reduces latency and enables real-time decision-making in applications such as robotics and medical sensors.

In cases when energy efficiency is paramount, architectural innovations like Computation-in-Memory (CiM) technology [74] could lower power consumption, by decreasing the energy-intensive transfer of data between processors and memory, which enhances the battery longevity in portable electronics. In addition to speed and power considerations, these architectural designs address the “memory wall’’ challenge by incorporating localized caches and high-bandwidth memory solutions that facilitate seamless data transfer without overloading the device. This emphasis on local execution also provides a security advantage, safeguarding sensitive information from potential exposure, by ensuring it remains on the device, thereby mitigating vulnerabilities to external cyber threats. Contemporary trends, such as heterogeneous computing and the adoption of flexible instruction sets like RISC-V [75], empower engineers to customize hardware for specific applications. This ensures that whether the device is a smart camera or an industrial sensor, its architecture would be optimized to meet the demands of its specific AI workload.

The primary benefit of RISC-V in the artificial intelligence sector is its modular instruction set architecture. This design empowers developers to eliminate unnecessary complexity and incorporate custom instructions specifically optimized for tensor operations and matrix multiplication. Given that modern AI workloads demand substantial resources, traditional fixed architectures often result in power wastage due to general-purpose features that do not contribute to deep learning. RISC-V addresses this inefficiency through a “building blocks’’ approach, allowing developers to combine a fundamental integer set with the RISC-V Vector Extension to achieve remarkable parallel throughput, all without the licensing limitations and stringent design constraints inherent in proprietary solutions.

In high-performance data centers, RISC-V frequently functions as the management layer within expansive AI accelerators, coordinating data flow between memory and

specialized computing units. Companies such as NVIDIA leverage these cores to execute complex scheduling tasks, ensuring that their primary processing units remain fully operational. Concurrently, at the edge of the network, RISC-V is slowly becoming the standard for efficient, low-power inference in devices like smart sensors and wearables. In these applications, integrating specific AI algorithms directly into the silicon delivers real-time image recognition and voice processing capabilities while maintaining a minimal power budget that larger architectures struggle to match.

The transition toward RISC-V is also driven by a quest for architectural independence, enabling global technology firms and research institutions to innovate in AI hardware development without dependence on a single vendor's direction. This shift has resulted in a rapidly growing ecosystem, in which software frameworks like TensorFlow Lite or PyTorch are being fine-tuned also for RISC-V instructions. Nowadays, we are witnessing the advent of the first RISC-V AI PCs, designed to compete with established players by offering comparable tops-per-watt performance for executing large language models locally.

It is worth mentioning that also the dataflow architecture proposed more than fifty years ago by Jack Dennis [76] seems to be a significant advancement in hardware designs intended to accelerate AI workloads. This architecture is characterized by computations that are initiated by the availability of data rather than a predetermined sequence of instructions [77]. In contrast to traditional processors that operate according to a linear program counter, dataflow systems "conceptualize'' in a sense the AI models as directed graphs in which operations are executed automatically upon the arrival of the respective inputs. This architecture provides remarkable efficiency, particularly for neural networks, as it facilitates the simultaneous processing of data across multiple layers of a model through extensive parallelization. By maintaining a continuous flow of data through a pre-configured pipeline, this approach effectively mitigates the performance bottlenecks and excessive energy consumption associated with the well-known "memory wall'' phenomenon prevalent in conventional CPUs and GPUs. The Hailo AI accelerators mentioned above in the context of Raspberry Pi SBCs are the prominent real-world implementation of a successful structure-driven dataflow architecture designed specifically for Edge AI [78, 79].

The revival of analog computing [80, 81] could provide a critical evolution in hardware designed to overcome the energy and data-transfer limitations of traditional digital systems [82]. Unlike standard digital processors, which must constantly move information between separate memory and logic units, analog devices utilize analog signals related to physical phenomena which are used to model the problem being solved. This approach effectively eliminates the "von Neumann bottleneck", allowing for the processing of artificial intelligence tasks with much better energy efficiency and significantly lower latency than conventional GPUs or CPUs. It is also possible to use various metamaterials to build analog computing devices [84], even as reconfigurable metastructures able to perform complex calculations [85]. While digital systems remain superior for high-precision tasks, modern hybrid analog/digital AI chips have reached a level of accuracy sufficient for complex pattern recognition [83], making them a cornerstone of the decentralized AI movement where 80% of inference is projected to happen locally on devices. This shift not only promises to preserve battery life but also enhances

data privacy by ensuring that sensitive information is processed at the source rather than being transmitted to the cloud.

While significant progress has been made, several challenges and opportunities still remain. Improving SLM performance through novel architectures and training techniques is paramount, as is developing federated learning approaches for training SLMs collaboratively across multiple edge devices while preserving privacy [86]. As privacy concerns grow, the way AI models are trained is changing. Instead of sending user data to the cloud, devices perform "on-device training" to personalize the model. Then the device sends only the mathematical updates (gradients) back to improve the global model, ensuring Zero-Knowledge Privacy.

Running language models on microcontrollers necessitates a focus on highly compressed TinyML architectures instead of general-purpose SLMs. The most effective strategy seems to be here to utilize hardware specialized models, like the STM32 Model Zoo [87], or develop custom models using frameworks like TensorFlow Lite, which can subsequently be converted into optimized C code via specialized toolchains, like the STM32Cube.AI toolchain in the case of STM32 hardware. For applications involving voice and text processing, developers often employ grammar-based models or specialized audio event detection families that are suitable for the constrained SRAM available on these chips. High-performance models, such as the STM32H7 series, are usually favored for their speed here, while the more recent STM32N6 series features a dedicated hardware NPU that enhances the efficiency of neural network execution. Additionally, tools like NanoEdge AI Studio [88] facilitate on-device learning and anomaly detection, empowering users without requiring extensive data science expertise.

The future also involves devices talking to each other to solve problems. If an edge device doesn't have the compute power for a complex task, it can "lease" NPU cycles from a nearby AI PC or a smart vehicle via ultra-low-latency 6G or Wi-Fi 7 connections. Groups of federated devices can then share their local SLM insights to form a high-resolution "global view" or "collaborative edge". It enables "swarm sensing" by a group of sensors covering a large area - both from above (drones) and from stationary installed sensors, and then to develop a "swarm intelligence" to deal e.g. with problems in a massive industrial complex [89]. Allowing SLMs to adapt to new data and tasks without forgetting previously learned knowledge represents also a significant frontier.

Explainable AI (XAI) for Edge SLMs is crucial for fostering trust and accountability, and hardware-software co-design can further improve performance and energy efficiency [90]. Nowadays, AI is having a tremendous impact on many aspects of our life, including healthcare and engineering, where intelligent systems cannot be considered as "black boxes" or "infallible oracles". Explainability could be also treated as an edge-AI system service [91], in contrast to being a specific model property implemented locally.

8 Conclusions

Edge AI powered by SLMs represents a transformative shift in the landscape of artificial intelligence. By bringing language understanding and generation capabilities directly to the edge, we can unlock a new era of personalized, real-time, and privacy-preserving applications. The potential for sustainable environmental monitoring and action, coupled with the growing feasibility of repurposing existing hardware, underscores the immense promise of this field. While challenges remain, ongoing research and development efforts are steadily pushing the boundaries of what is possible, paving the way for a future where intelligent devices are seamlessly integrated into our daily lives.

References

1. Domínguez-Bolaño, T., Campos, O., Barral, V., Escudero, C. J., & García-Naya, J. A. (2022). An overview of IoT architectures, technologies, and existing open-source projects. *Internet of Things*, *20*, Article 100626. https://doi.org/10.1016/j.iot.2022.100626
2. Gauttam, H., Nain, G., Pattanaik, K. K., & Mendes, P. (2026). Edge-AI: A systematic review on architectures, applications, and challenges. *Journal of Network and Computer Applications*, *245*, Article 104375. https://doi.org/10.1016/j.jnca.2025.104375
3. Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., & Mian, A. (2025). A comprehensive overview of large language models. *ACM Transactions on Intelligent Systems and Technology*, *16*(5), Article 106. https://doi.org/10.1145/3744746
4. Wang, F., Zhang, Z., Zhang, X., Wu, Z., Mo, T., Lu, Q., Wang, W., Li, R., Xu, J., Tang, X., He, Q., Ma, Y., Huang, M., & Wang, S. (2025). A comprehensive survey of small language models in the era of large language models: Techniques, enhancements, applications, collaboration with LLMs, and trustworthiness. *ACM Transactions on Intelligent Systems and Technology*, *16*(6), Article 145. https://doi.org/10.1145/3768165
5. Tsoukas, V., Gkogkidis, A., Boumba, E., & Kakarountas, A. (2024). A review on the emerging technology of TinyML. *ACM Computing Surveys*, *56*(10), Article 259. https://doi.org/10.1145/3661820
6. OpenAI. (2026). *Open models by OpenAI* [Website]. Retrieved February 15, 2026, from https://openai.com/open-models/
7. Gemma Team. (2025). Gemma 3 technical report. arXiv. https://arxiv.org/abs/2503.19786
8. Microsoft. (2024). *Phi-4 technical report*. Microsoft Research. https://www.microsoft.com/en-us/research/wp-content/uploads/2024/12/P4TechReport.pdf
9. Microsoft. (2025). Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-LoRAs. arXiv. https://arxiv.org/abs/2503.01743
10. Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., …Schrijver, R. (2024). The Llama 3 herd of models. arXiv. https://arxiv.org/abs/2407.21783
11. Qwen Team. (2025). Qwen2.5 technical report. arXiv. https://arxiv.org/abs/2412.15115
12. Zhang, P., Zeng, G., Wang, T., & Lu, W. (2024). TinyLlama: An open-source small language model. arXiv. https://arxiv.org/abs/2401.02385
13. Aralimatti, R., Shakhadri, S. A. G., Kruthika, K. R., & Angadi, K. B. (2025). Fine-tuning small language models for domain-specific AI: An edge AI perspective. In K. Arai (Ed.), *Intelligent systems and applications: Proceedings of the IntelliSys 2025 conference* (Lecture Notes in Networks and Systems, Vol. 1554, pp. [xx–xx]). Springer. https://doi.org/10.1007/978-3-031-99965-9_31

14. Allal, L. B., Lozhkov, A., Bakouch, E., von Werra, L., & Wolf, T. (2025). SmolLM2: When smol goes big – data-centric training of a small language model. arXiv. https://arxiv.org/abs/2502.02737
15. Li, S., Nguyen, H., Zheng, B., Nguyen, H.-T., Yao, Y., Zhou, Y., Qin, Z., Zhang, H., Han, X., Hu, S., Chen, W., & [remaining authors]. (2024). Evaluating quantized large language models. In *Proceedings of the 41st International Conference on Machine Learning* (Vol. 235, pp. 28480–28524). JMLR.org. https://doi.org/10.5555/3692070.3693214
16. Wang, Y., Huang, L., Zhang, J., & [remaining authors]. (2024). Art and science of quantizing large-scale models: A comprehensive overview. arXiv. https://arxiv.org/abs/2409.11650
17. Rakka, M., Fouda, M. E., Khargonekar, P., & Kurdahi, F. (2024). A review of state-of-the-art mixed-precision neural network frameworks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *46*(12), 7793–7812. https://doi.org/10.1109/TPAMI.2024.3394390
18. Qin, T., Luo, J., Cheng, C., & [remaining authors]. (2025). Mixed-precision quantization based on information entropy. *Scientific Reports*, *15*, Article 12974. https://doi.org/10.1038/s41598-025-91684-8
19. Lin, J., Tang, J., Tang, H., Yang, S., Xiao, G., & Han, S. (2025). AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. *GetMobile: Mobile Computing and Communications*, *28*(4), 12–17. https://doi.org/10.1145/3714983.3714987
20. Zhao, X., Xu, R., & Guo, X. (2023). Post-training quantization or quantization-aware training? That is the question. In *Proceedings of the 2023 China Semiconductor Technology International Conference (CSTIC)* (pp. 1–3). IEEE. https://doi.org/10.1109/CSTIC58779.2023.10219214
21. Hou, B., Wu, Q., Hao, Y., & [remaining authors]. (2025). Instruction-following pruning for large language models. arXiv. https://arxiv.org/abs/2501.02086
22. Guo, J., Chen, X., Tang, Y., & Wang, Y. (2025). SlimLLM: Accurate structured pruning for large language models. arXiv. https://arxiv.org/abs/2505.22689
23. Frantar, E., & Alistarh, D. (2023). SparseGPT: Massive language models can be accurately pruned in one shot. In *Proceedings of the 40th International Conference on Machine Learning* (Vol. 202, pp. 10323–10337). JMLR.org. https://doi.org/10.5555/3618408.3618822
24. Yu, P., Wang, J., Sui, X., Ling, N., Wang, W., & Jiang, W. (2026). Efficient post-training pruning of large language models with statistical correction. arXiv. https://arxiv.org/abs/2602.07375
25. Moslemi, A., Briskina, A., Dang, Z., & Li, J. (2024). A survey on knowledge distillation: Recent advancements. *Machine Learning with Applications*, *18*, Article 100605. https://doi.org/10.1016/j.mlwa.2024.100605
26. Zhang, L., Bao, C., & Ma, K. (2022). Self-distillation: Towards efficient and compact neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(8), 4388–4403. https://doi.org/10.1109/TPAMI.2021.3067100
27. Shenfeld, I., Damani, M., Hübotter, J., & Agrawal, P. (2026). Self-distillation enables continual learning. arXiv. https://arxiv.org/abs/2601.19897
28. Sainath, T. N., Kingsbury, B., Sindhwani, V., Arisoy, E., & Ramabhadran, B. (2013). Low-rank matrix factorization for deep neural network training with high-dimensional output targets. In *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2013)* (pp. 6655–6659). IEEE. https://doi.org/10.1109/ICASSP.2013.6638949
29. Hsu, Y.-C., Ting, H., Chang, S., Lou, Q., Shen, Y., & Jin, H. (2022). Language model compression with weighted low-rank factorization. arXiv. https://arxiv.org/abs/2207.00112
30. Leviathan, Y., Kalman, M., & Matias, Y. (2023). Fast inference from transformers via speculative decoding. In *Proceedings of the 40th International Conference on Machine Learning* (Vol. 202, pp. 19274–19286). JMLR.org. https://doi.org/10.5555/3618408.3619203
31. Yan, M., Agarwal, S., & Venkataraman, S. (2025). Decoding speculative decoding. arXiv. https://arxiv.org/abs/2402.01528
32. Marculescu, D., Stamoulis, D., & Cai, E. (2018). Hardware-aware machine learning: Modeling and optimization. In *Proceedings of the 2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD '18)* (pp. 1–8). ACM. https://doi.org/10.1145/3240765.3243479

33. Rhea, S., Dohan, D., Firat, O., & [remaining authors]. (2024). HW-GPT-Bench: Hardware-aware architecture benchmark for language models. *Advances in Neural Information Processing Systems*, *37*, 60776–60834. https://doi.org/10.52202/079017-1944
34. Korolov, M. (2026, January 8). CES 2026: AI compute sees a shift from training to inference. *Computerworld*. Retrieved February 15, 2026, from https://www.computerworld.com/article/4114579/ces-2026-ai-compute-sees-a-shift-from-training-to-inference.html
35. Gupta, S., & Chaudhary, A. (Eds.). (2026). *Artificial intelligence in healthcare: Trends, applications, and future directions*. Apple Academic Press.
36. Nguyen, M. H., Shen, Y., Liao, J., & [remaining authors]. (2025). On-device diagnostic recommendation with heterogeneous federated BlockNets. *Science China Information Sciences*, *68*, Article 140102. https://doi.org/10.1007/s11432-024-4162-2
37. Villalobos-Quesada, M., Ho, K., Chavannes, N. H., & Talboom-Kamp, E. P. (2023). Direct-to-patient digital diagnostics in primary care: Opportunities, challenges, and conditions necessary for responsible digital diagnostics. *European Journal of General Practice*, *29*(1), Article 2273615. https://doi.org/10.1080/13814788.2023.2273615
38. Artiushenko, V., Lang, S., Lerez, C., Reggelin, T., & Hackert-Oschätzchen, M. (2024). Resource-efficient edge AI solution for predictive maintenance. *Procedia Computer Science*, *232*, 348–357. https://doi.org/10.1016/j.procs.2024.01.034
39. Uprety, I., & Zhao, X. (2025). Edge-deployable LLMs for autonomous vehicle intelligence. In *Proceedings of the Tenth ACM/IEEE Symposium on Edge Computing (SEC '25)* (pp. 1–7). ACM. https://doi.org/10.1145/3769102.3774639
40. Rahman, M. A., Dewan, M. A. A., Hasan, M., & [remaining authors]. (2025). A scalable framework for deploying AI-powered wildlife monitoring in resource-limited field environments. *IEEE Access*, *13*, 145023–145041. https://doi.org/10.1109/ACCESS.2025.3598927
41. Boscoe, B., Johnson, S., Osborn, A., Campbell, C., & Mager, K. (2025). GreenCrossingAI: A camera-trap/computer-vision pipeline for environmental science research groups. In *Proceedings of the Practice and Experience in Advanced Research Computing 2025 (PEARC '25)* (pp. 1–8). ACM. https://doi.org/10.1145/3708035.3736003
42. Chen, S., Li, D., Liu, J., & Wei, R. (2024). Raspberry Pi-based intelligent greenhouse IoT platform. In *Proceedings of the 2023 5th International Conference on Internet of Things, Automation and Artificial Intelligence (IoTAAI '23)* (pp. 787–791). ACM. https://doi.org/10.1145/3653081.3653213
43. Dai, X., & Yao, W. (2025). Research on development strategies for edge-AI-based smart-home devices. In *Proceedings of the 2nd Guangdong-Hong Kong-Macao Greater Bay Area International Conference on Digital Economy and Artificial Intelligence (DEAI '25)* (pp. 1474–1481). ACM. https://doi.org/10.1145/3745238.3745469
44. Tiwari, A. (2026). A proposal to localise urban AI: A conceptual shift from generalist LLMs to task-specific SLMs. *Computational Urban Science*, *6*, Article 11. https://doi.org/10.1007/s43762-026-00241-0
45. Sachdev, R. (2020). Towards security and privacy for edge AI in IoT/IoE-based digital marketing environments. In *Proceedings of the 2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC 2020)* (pp. 341–346). IEEE. https://doi.org/10.1109/FMEC49853.2020.9144755
46. Mathai, A. (2026, January 17). Cut AI costs by 90%: Why smart companies are downsizing to small language models (SLMs). *Mindster*. Retrieved February 15, 2026, from https://mindster.com/mindster-blogs/small-language-models-slm-cost-efficiency/
47. Dong, Z., Sharma, H., O'Toole, E., Champati, J. P., & Wu, K. (2026). Pay for hints, not answers: LLM shepherding for cost-efficient inference. arXiv. https://arxiv.org/abs/2601.22132
48. Scott, K. (2001). *On Proebsting's law* (Technical Report). University of Virginia, Department of Computer Science. https://doi.org/10.18130/V33Z0W
49. Raspberry Pi Ltd. (2026). *Raspberry Pi computer hardware*. Retrieved February 15, 2026, from https://www.raspberrypi.com/documentation/computers/raspberry-pi.html
50. Włodarz, J. (2024). Computer chips and social economy: The impact of affordable computing. In Z. Wittine, S. Franc, & A. Barišić (Eds.), *International Scientific Conference "Empowering*

Change: Fostering Social Entrepreneurship for a Sustainable Future" (pp. 50–56). University of Zagreb. https://doi.org/10.1007/978-3-031-12345-6_6

51. ARM Ltd. (2025, December). *The Armv8.2 architecture extension*. ARM Developer. Retrieved February 15, 2026, from https://developer.arm.com/documentation/109697/2025_12/Feature-descriptions/The-Armv8-2-architecture-extension
52. Raspberry Pi Ltd. (2026). *Raspberry Pi AI HAT+ 2 product brief*. Retrieved February 15, 2026, from https://pip-assets.raspberrypi.com/categories/1319-raspberry-pi-ai-hat-2/documents/RP-009655-MM-4-raspberry-pi-ai-hat-plus-2-product-brief.pdf
53. Chavan, S. R., Gavande, P., & Mhaske, M. D. (2025). A comprehensive review of Nvidia Jetson Nano module. *International Research Journal of Modernization in Engineering Technology and Science*, *7*(3), 1378–1386. https://doi.org/10.56726/IRJMETS84923
54. Maslekar, A., Suryavanshi, A., & Gavande, P. (2025). Catalyst for intelligent edge: A comprehensive analysis of the Nvidia Jetson Nano – architecture, performance, benchmarking and comparative standing in the AIoT landscape. *International Research Journal of Modernization in Engineering Technology and Science*, *7*(10), 2999–3008. https://doi.org/10.56726/IRJMETS84050
55. Maheshwari, S., & Su, C. (2025, January 16). NVIDIA JetPack 6.2 brings super mode to NVIDIA Jetson Orin Nano and Jetson Orin NX modules. *NVIDIA Developer Blog*. Retrieved February 15, 2026, from https://developer.nvidia.com/blog/nvidia-jetpack-6-2-brings-super-mode-to-nvidia-jetson-orin-nano-and-jetson-orin-nx-modules/
56. NVIDIA Corp. (2026). *NVIDIA Jetson Thor: The ultimate platform for physical AI and robotics*. Retrieved February 15, 2026, from https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-thor/
57. Caruso, J. E. (2025, December 10). AI at the edge – Intelligent systems operating where data is generated. *Syslogic Technical Articles*. Retrieved February 15, 2026, from https://www.syslogic.com/blog/ai-at-the-edge
58. ParkyTowers. (2026). *Thin clients*. Retrieved February 15, 2026, from https://www.parkytowers.me.uk/thin/
59. Fatkina, A., Kozlov, A., Shevelev, A., & [remaining authors]. (2019). GNA: New framework for statistical data analysis. *EPJ Web of Conferences*, *214*, Article 05024. https://doi.org/10.1051/epjconf/201921405024
60. Intel Corp. (2026). *OpenVINO toolkit documentation*. Retrieved February 15, 2026, from https://docs.openvino.ai
61. Hirsch, M., Mateos, C., & Majchrzak, T. A. (2025). Exploring smartphone-based edge AI inferences using real testbeds. *Sensors*, *25*(9), Article 2875. https://doi.org/10.3390/s25092875
62. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv. https://arxiv.org/abs/1704.04861
63. Hentati, M., Hentati, R., Gouiaa, Y., & Abid, M. (2025). Edge AI-based classification: A case study on STM32 microcontroller. In *Proceedings of the 2025 International Conference on Advanced Systems and Emergent Technologies (IC_ASET 2025)* (pp. 1–6). IEEE. https://doi.org/10.1109/IC_ASET65966.2025.11231782
64. Espressif Systems Co., Ltd. (2026). *ESP32-S3 technical reference manual*. Retrieved February 15, 2026, from https://www.espressif.com/documentation/esp32-s3_technical_reference_manual_en.pdf
65. May, A. (2026). *Increasing intelligence at the edge with embedded processors* [White paper]. Texas Instruments. Retrieved February 15, 2026, from https://www.ti.com/lit/wp/spry349a/spry349a.pdf
66. STMicroelectronics NV. (2022). *STM32Cube ecosystem overview: Making STM32 development easier* [Product presentation]. Retrieved February 15, 2026, from https://www.st.com/resource/en/product_presentation/stm32cube_ecosystem_overview.pdf
67. Google AI for Developers Community. (2026). *LiteRT for microcontrollers*. Retrieved February 15, 2026, from https://ai.google.dev/edge/litert/microcontrollers/overview

68. STMicroelectronics NV. (2026). *STM32N6x5xx STM32N6x7xx datasheet*. Retrieved February 15, 2026, from https://www.st.com/resource/en/datasheet/stm32n657a0.pdf
69. Renesas Electronics Co. (2023). *An introduction to Renesas Advanced (RA) MCU kits* [Presentation]. Retrieved February 15, 2026, from https://www.renesas.com/en/document/ppt/introduction-renesas-advanced-ra-mcu-kits
70. Seeed Studio, Inc. (2026). *Grove Vision AI module V2* [Product wiki]. Retrieved February 15, 2026, from https://wiki.seeedstudio.com/grove_vision_ai_v2/
71. Silicon Laboratories, Inc. (2026). *EFR32MG24 wireless SoC family datasheet*. Retrieved February 15, 2026, from https://www.silabs.com/documents/public/data-sheets/efr32mg24-datasheet.pdf
72. Nordic Semiconductor ASA. (2026). *nRF54L series technical documentation*. Retrieved February 15, 2026, from https://docs.nordicsemi.com/category/nrf-54L-series
73. Raspberry Pi Ltd. (2025). *RP2350 datasheet: A microcontroller by Raspberry Pi*. Retrieved February 15, 2026, from https://pip-assets.raspberrypi.com/categories/1214-rp2350/documents/RP-008373-DS-2-rp2350-datasheet.pdf
74. Sun, Z., Kvatinsky, S., Si, X., Alhajj, R., Kang, J., & [remaining authors]. (2023). A full spectrum of computing-in-memory technologies. *Nature Electronics*, *6*, 823–835. https://doi.org/10.1038/s41928-023-01053-4
75. Borade, S. A., Bansod, S., Hati, A. J., & Singh, S. K. (2025). AI edge processor using RISC-V instruction set architecture design. In *Proceedings of the 2025 Global Conference in Emerging Technology (GINOTECH 2025)* (pp. 1–8). IEEE. https://doi.org/10.1109/GINOTECH63460.2025.11076631
76. Dennis, J. B. (1974). First version of a data flow procedure language. In G. Goos & J. Hartmanis (Eds.), *Programming symposium: Proceedings, colloque sur la programmation* (Lecture Notes in Computer Science, Vol. 19, pp. [xx–xx]). Springer.
77. Veen, A. H. (1986). Dataflow machine architecture. *ACM Computing Surveys*, *18*(4), 365–396. https://doi.org/10.1145/27633.28055
78. Hailo Technologies Ltd. (2025). *Bringing generative AI to the edge: LLM on Hailo-10H*. Retrieved February 15, 2026, from https://hailo.ai/blog/bringing-generative-ai-to-the-edge-llm-on-hailo-10h/
79. Krispin-Avraham, I., Orfaig, R., & Bobrovsky, B.-Z. (2024). Real-time 3D object detection using InnovizOne LiDAR and low-power Hailo-8 AI accelerator. arXiv. https://doi.org/10.48550/arXiv.2412.05594
80. Ulmann, B. (2022). *Analog computing*. De Gruyter Oldenbourg. https://doi.org/10.1515/9783110787740
81. Ulmann, B. (2026). *Analog computing: Development, programming, applications, and future directions*. De Gruyter. https://doi.org/10.1515/9783112210178
82. Haensch, W., Gokmen, T., & Puri, R. (2019). The next generation of deep learning hardware: Analog computing. *Proceedings of the IEEE*, *107*(1), 108–122. https://doi.org/10.1109/JPROC.2018.2871057
83. Guo, N., Huang, Y., Mai, T., & [remaining authors]. (2016). Energy-efficient hybrid analog/digital approximate computation in continuous time. *IEEE Journal of Solid-State Circuits*, *51*(7), 1514–1524. https://doi.org/10.1109/JSSC.2016.2543729
84. Włodarz, J. (2019). Analog computing and (meta)materials. In *Proceedings of the 1st Polish-Chinese Conference "From Molecular Modeling to Nano- and Biotechnology"* (pp. 1–46).
85. Tzarouchis, D. C., Edwards, B., & Engheta, N. (2025). Programmable wave-based analog computing machine: A metastructure that designs metastructures. *Nature Communications*, *16*, Article 908. https://doi.org/10.1038/s41467-025-56019-1
86. Wang, Z., Wu, F., Yu, F., Zhou, Y., Hu, J., & Min, G. (2024). Federated continual learning for edge-AI: A comprehensive survey. arXiv. https://arxiv.org/abs/2411.13740
87. STMicroelectronics NV. (2026). *AI model zoo for STM32 devices* [Software repository]. Retrieved February 15, 2026, from https://github.com/STMicroelectronics/stm32ai-modelzoo
88. STMicroelectronics NV. (2026). *NanoEdge AI Studio: Automated machine learning (ML) tool for STM32 developers*. Retrieved February 15, 2026, from https://www.st.com/en/development-tools/nanoedgeaistudio.html

89. Sah, D. K., Vahabi, M., & Fotouhi, H. (2025). Federated learning at the edge in industrial internet of things: A review. *Sustainable Computing: Informatics and Systems*, *46*, Article 101087. https://doi.org/10.1016/j.suscom.2025.101087
90. Górriz, J. M., Ramírez, J., Ortíz, A., & [remaining authors]. (2023). Computational approaches to explainable artificial intelligence: Advances in theory, applications and trends. *Information Fusion*, *100*, Article 101945. https://doi.org/10.1016/j.inffus.2023.101945
91. Singh, S. K., & Roy, J. (2026). Scalable explainability-as-a-service (XaaS) for edge AI systems. arXiv. https://arxiv.org/abs/2602.04120

About author

Joachim J. Włodarz has been active in academia since the early 1980s, primarily in the fields of quantum chemistry/physics and computer science. He is currently a university professor at the Faculty of Science and Technology, University of Silesia in Katowice, Poland.