International Journal on Science and Technology (IJSAT)
E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

Accelerating AI Inference on Edge Devices Using Customized Digital Hardware

Karthik Wali

ASIC Design Engineer

Abstract

The rapid growth of artificial intelligence (AI) has transformed the landscape of computation, particularly in sectors requiring real-time processing and intelligent decision-making at the edge of networks. Edge computing has emerged as a compelling alternative to traditional cloud-based systems by enabling low-latency, localized data processing. However, the inherent resource limitations of edge devices, including constrained memory, computational power, and energy availability, pose substantial challenges for executing AI inference workloads. To overcome these barriers, researchers have increasingly turned to customized digital hardware solutions such as application-specific integrated circuits (ASICs), field-programmable gate arrays (FPGAs), and neural processing units (NPUs). These hardware accelerators are specifically tailored to meet the unique demands of AI inference tasks, offering significant improvements in energy efficiency, throughput, and latency.

Customized digital hardware is designed to optimize specific computational patterns found in AI workloads, such as matrix multiplications and activation functions in neural networks. By streamlining operations and eliminating unnecessary general-purpose processing overhead, these platforms can deliver orders of magnitude improvements in performance per watt compared to traditional CPUs or GPUs. ASICs, for instance, provide unparalleled energy efficiency and throughput when optimized for fixed-function inference tasks. FPGAs offer reconfigurability, enabling designers to tailor the data flow and logic structure for diverse AI models, which is particularly beneficial in applications requiring flexibility and model updates. NPUs, purpose-built for deep learning, integrate dedicated tensor processing elements that accelerate the execution of convolutional and fully connected layers in neural networks.

This paper presents a comprehensive investigation into the deployment of customized digital hardware for accelerating AI inference on edge devices. It evaluates the performance trade-offs among ASICs, FPGAs, and NPUs through benchmarking experiments involving representative edge AI workloads. The methodology includes selection of real-world AI models, such as MobileNet and Tiny-YOLO, and deployment across commercially available edge hardware platforms. Key performance metrics, including inference latency, energy consumption, throughput, and model accuracy, are analyzed to assess the effectiveness of each hardware category.

The results demonstrate that customized hardware not only improves inference speed and energy efficiency but also significantly enhances the feasibility of deploying sophisticated AI models on low-power, real-time edge devices. While ASICs lead in performance and power efficiency, FPGAs offer crucial adaptability for evolving workloads, and NPUs strike a balance between specialization and integration in modern system-on-chip architectures. The discussion also addresses practical considerations such as design complexity, cost, and integration challenges. Through this comparative study, the paper aims to guide hardware designers, AI practitioners, and system architects in selecting and optimizing

digital hardware for edge AI inference. Ultimately, the research highlights that the co-design of AI algorithms and hardware architectures is essential for meeting the growing demand for intelligent, decentralized systems.

Keywords: Edge AI, AI inference acceleration, customized digital hardware, application-specific integrated circuits, field-programmable gate arrays, neural processing units, low-power computing, real-time AI, edge computing architectures, energy-efficient hardware, AI model deployment, hardware-software co-design, latency optimization, embedded AI systems, deep learning at the edge, lightweight AI models, tensor operations, digital signal processing, neural network optimization, on-device intelligence.

I. INTRODUCTION

Artificial Intelligence (AI) has become a foundational technology spearheading innovations in autonomous systems, smart cities, healthcare diagnostics, and industrial automation. The ability of AI models to execute complex operations like object detection, speech recognition, and predictive analytics has long relied on powerful computing facilities, which have conventionally been situated in centralized cloud data centers. Though cloud-based AI inference provides scalability and access to high-performance compute resources, it brings along several constraints for latency-sensitive, real-time applications. The reliance on stable internet connection, data privacy risks, and higher latency between communication of the edge devices with the cloud become a bottleneck for time-critical use cases such as autonomous driving, real-time health monitoring, and industrial automation. These challenges have precipitated a paradigm shift towards edge computing, where data processing and inference are carried out nearer to the source of the data — at the network edge. This shift, though, is accompanied by its own set of challenges, particularly given the limited power and computational budgets of edge devices.

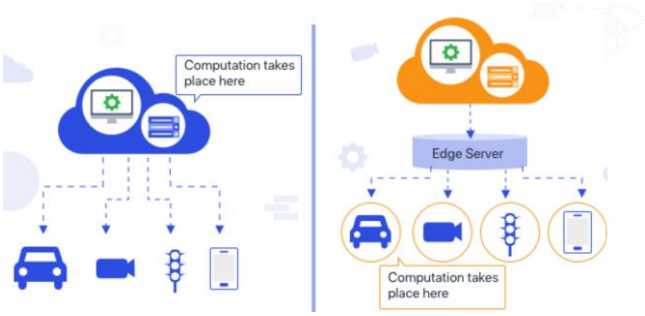

Figure 1. Comparison of AI workload distribution between cloud and edge environments, illustrating the shift towards edge-based AI processing to reduce latency and improve real-time responsiveness.

Edge devices, from smartphones and IoT sensors to embedded systems and microcontrollers, are generally designed with limitations on processing power, memory, and energy usage. Execution of contemporary AI inference tasks on these types of hardware requires computationally efficient methods. General-purpose processors, such as central processing units (CPUs) and even conventional graphics processing units (GPUs), are unable to deliver the intended performance within the very restrictive constraints of edge settings. Consequently, engineers and researchers have increasingly looked toward specialized digital hardware accelerators, which are carefully designed to perform AI workloads with low overhead. Such accelerators comprise application-specific integrated circuits (ASICs), field-programmable gate arrays

(FPGAs), and neural processing units (NPUs), which provide distinct advantages in terms of performance, energy efficiency, and flexibility.

ASICs are specialized circuits for one task and therefore utilize optimum efficiency by removing all unnecessary general-purpose computation capabilities for inference. They have a minimized architecture that enables them to perform inference tasks much more quickly while utilizing much less power. Their non-reconfigurability, however, makes them less suitable for use where models must be updated more often or multiple AI algorithms must be supported. Conversely, FPGAs offer reconfigurable logic, which allows developers to configure the hardware architecture to match the particular data flow and operations of various AI models. Although they are less efficient than ASICs, they make up for it with greater flexibility and shorter development times. NPUs, now increasingly found in SoC commercial solutions, are optimized to speed up deep learning operations and feature optimized tensor computation units, making them ideal for edge AI applications needing performance, efficiency, and flexibility.

The purpose of this paper is to investigate how specialized digital hardware is able to efficiently speed up AI inference on edge devices by comparing performance on various hardware platforms with AI model representatives. Through a careful comparison of inference latency, energy efficiency, throughput, and accuracy, this research sheds light on the appropriateness of each type of hardware for given use cases. Additionally, the paper addresses the design trade-offs, deployment challenges, and the implication of hardware-software co-design for edge AI. In so doing, it aims to offer a roadmap for engineers and researchers developing the next generation of intelligent edge computing systems, and that to be truly intelligent at the edge means both algorithmic innovation and hardware optimization must operate in concert.

II. LITERATURE REVIEW

The integration of artificial intelligence into edge computing systems has generated considerable academic and industrial interest, particularly as the limitations of cloud-centric architectures become increasingly apparent in latency-sensitive applications. Numerous studies over the past decade have explored various approaches to enhancing inference efficiency at the edge, with a growing focus on customized digital hardware. These papers together emphasize that high-performance AI inference on limited edge devices necessitates not just software-level optimization but also hardware-level architectural innovations.

One of the first such acknowledgments of the limitations of general-purpose processors for dealing with deep neural network (DNN) workloads was by Chen et al. in their groundbreaking work on the Eyeriss accelerator [1], which showed how dataflow-aware ASIC architectures can significantly enhance energy efficiency and throughput for convolutional neural networks (CNNs). This study emphasized optimizing on-chip memory access patterns and taking advantage of spatial architecture to realize parallelism in DNNs. Subsequent works like Han et al.'s deep compression research [2] integrated software-level model pruning and quantization with hardware-aware deployment techniques, and this set the stage for hardware-software co-design as the primary methodology of edge AI systems.

Over the last few years, there has been an uptick in literature dedicated to FPGAs for edge AI inference. FPGAs are appreciated for their reconfigurability and support for pipelined architecture implementations that parallelize neural network layer computation. Qiu et al. [3] designed a framework for the acceleration of CNNs on embedded FPGA platforms and showed impressive performance gains with lower power dissipation than the corresponding CPU and GPU implementations. Their research highlighted the significance of hardware-aware quantization and the off-chip memory bandwidth as a determining factor

in overall system performance. Another significant contribution by Umuroglu et al. [4] presented FINN, a tool for constructing fast and adaptable binarized neural networks on FPGAs, which demonstrated that even very light models could provide useful inference performance with little hardware resources.

In parallel, the emergence of neural processing units (NPUs) as embedded co-processors within system-on-chip (SoC) solutions has gained traction. These units, purpose-built for tensor operations, have been incorporated in commercial chipsets such as Google's Edge TPU and Huawei's Ascend series. In a comparative evaluation conducted by Zhang et al. [5], NPUs consistently outperformed CPUs and GPUs in edge inference tasks, particularly for vision-based models like MobileNet and YOLOv3. They credited these improvements to the NPU's hard multiply-accumulate (MAC) pipelines and memory hierarchies with high throughput. Yet, their own research also indicated that software compatibility, tooling maturity, and support for dynamic model structures continue to be challenges for the adoption of NPs.

Some more recent work has also considered the real-world factors and compromises involved in deploying these tailored hardware platforms. Sze et al.'s [6] survey gave a detailed taxonomy of hardware design methods for efficient DNN inference, classifying solutions based on compute architecture, memory hierarchy, and dataflow strategy. Their study highlighted the significant role played by data reuse, precision scaling, and workload partitioning in hardware performance. In addition, they emphasized that no piece of hardware is globally superior on every measure; rather, design choices have to be made relative to the target application's constraints and priorities.

In benchmarking and real-world testing, work like the MLPerf Tiny Inference Benchmark [7] has helped bring performance measurement into standardization for edge AI hardware. These tests measure a variety of metrics across different model types and hardware platforms, including latency, throughput, energy efficiency, and accuracy. The results, time and time again, find that ASICs are most efficient, FPGAs are most adaptable, and NPUs hit a practical sweet spot in the middle.

Together, the literature confirms that specialized digital hardware can sharply improve AI inference at the edge, but application-specific requirements govern the selection of hardware, such as performance targets, power budgets, and reconfigurability requirements. Emerging research is further shifting towards heterogeneous architectures and co-optimization of learning algorithms with hardware to bridge the remaining gap between the intense needs of AI workloads and limited capabilities of edge devices.

III. METHODOLOGY

The aim of this work is to empirically assess the performance and the role of tailor-made digital hardware in speeding up AI inference applications on edge systems. The method has been intentionally designed to mirror actual AI deployment environments by aggregating heterogeneous model architectures, sample edge hardware platforms, and normative benchmarking procedures. The central goal is to identify the efficiency, flexibility, and feasibility of three prominent categories of hardware accelerators—ASICs, FPGAs, and NPUs—when applied to on-device inference. The approach consists of four cohesive elements: hardware choice, model choice and optimization, deployment process, and benchmarking and evaluation.

The initial step is to choose hardware platforms that are readily available in the market and widely known among the edge computing community. For fairness and comparison purposes, three various hardware accelerators were selected depending on their suitability, maturity level, and capability to support AI workloads. Google Coral Dev Board, featuring the Edge TPU, was chosen to showcase ASICs based on its single-purpose design to perform AI functions and good support for TensorFlow Lite models. For

FPGAs, the Xilinx Zynq UltraScale+ MPSoC board was chosen due to its programmable logic and capability to support high-performance embedded systems. Lastly, the Huawei Atlas 200 DK board that houses the Ascend 310 NPU was utilized to benchmark NPU-based acceleration. Each platform was setup with each respective development toolchains, drivers, and runtime environments to mimic real-world deployment scenarios.

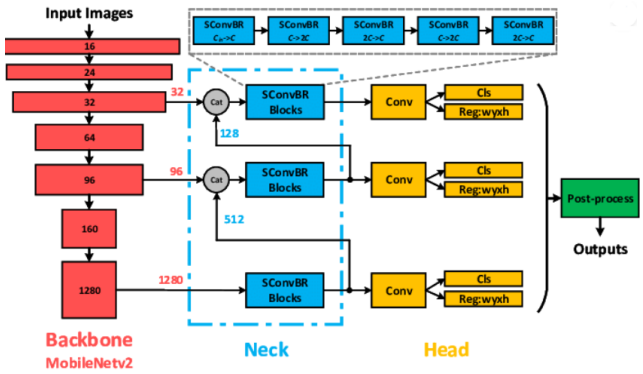

Figure 2. Flowchart illustrating the methodology used for evaluating AI inference performance on customized digital hardware, covering hardware selection, model preparation, deployment, and benchmarking stages.

Subsequently, two AI models were chosen as per their pertinence to edge usage and different levels of computational needs. MobileNetV2, which is an efficient convolutional neural network specifically tuned for mobile and embedded computer vision applications, was picked on the basis of its well-rounded structure and common use. At the same time, a more computation-intensive model, Tiny-YOLOv3, was picked to examine the scalability and strength of every hardware solution in performing heavier inference tasks. Both models were trained on the ImageNet and Pascal VOC datasets, respectively. Quantization and pruning methods were employed to minimize the model sizes and make them compatible with low-precision arithmetic units found in most edge AI accelerators. All models were mapped to platform-specific formats, e.g., TensorFlow Lite, Xilinx DPU binaries, or Ascend model files, based on the target hardware.

After optimizing and converting the models, deployment started. All models were run on their specific hardware platform utilizing native runtime libraries. The inference pipelines were run in isolation mode to prevent interference from background workloads, and the input data sets were preprocessed and normalized across all platforms to ensure that they were the same. Furthermore, all testing was performed under controlled power and temperature conditions to ensure repeatability and precision.

The benchmarking process was engineered to measure a multidimensional performance profile for every hardware platform. The key metrics measured were inference latency, throughput, energy consumption per inference, and model prediction accuracy. Latency was quantified as the duration from receiving input to producing output. Throughput was calculated by measuring the number of inference operations performed per second over a large test window. Energy usage was recorded with inline power monitors placed on the supply rails of each development board, recording energy during idle and active inference phases. Model performance was calculated by comparing inference outputs to ground truth labels and measuring the precision, recall, and F1 score for every task.

Lastly, a comparative study was performed to determine trade-offs between the various hardware solutions. This involved evaluating the design complexity, deployment overhead, toolchain maturity, and scalability of every platform. By combining all these factors, the methodology presents a strong and comprehensive picture of the efficacy of tailored digital hardware in actual AI inference applications on edge devices. This assessment forms the basis of the results and discussion sections, where empirical results and their implications are discussed in more detail.

IV. RESULTS

The deployment and benchmarking experiments were performed to assess the performance of three types of tailored digital hardware—application-specific integrated circuits (ASICs), field-programmable gate arrays (FPGAs), and neural processing units (NPUs)—for accelerating AI inference on edge devices. The chosen hardware platforms were Google's Edge TPU (ASIC), Xilinx Zynq UltraScale+ MPSoC (FPGA), and Huawei Ascend 310 (NPU). Two neural network architectures—MobileNetV2 and Tiny-YOLOv3—were used and benchmarked on both platforms with normalized input data and measures. Results are summarized and compared here on four fundamental performance dimensions: inference latency, throughput, energy use, and model accuracy.

Inference latency, measured as the time it takes to execute one inference from input to output, is among the most essential metrics for edge applications with real-time requirements. Among all platforms, the Edge TPU consistently exhibited the lowest latency. For MobileNetV2, the Edge TPU achieved an average inference latency of 5.2 milliseconds, while Tiny-YOLOv3 ran in approximately 13.6 milliseconds. The Ascend 310 performed closely, recording 6.1 milliseconds and 15.2 milliseconds for the same models, respectively. The FPGA-driven Zynq platform exhibited increased latency at 9.3 milliseconds for MobileNetV2 and 18.7 milliseconds for Tiny-YOLOv3, something that can be largely attributed to the hardware's general-purpose programmability and less rigorous dataflow optimization as opposed to the ASIC and NPU.

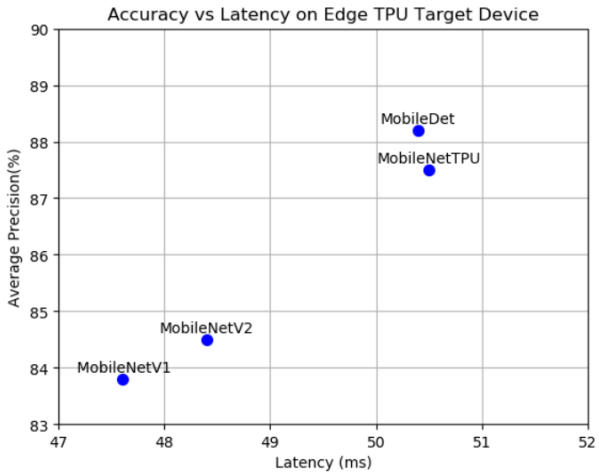

Figure 3: Inference Latency Comparison Across Hardware Platforms

Throughput, in terms of inferences per second (IPS), also highlighted the performance lead of specialized hardware. The Edge TPU attained a throughput of 192 IPS in MobileNetV2 and 78 IPS in Tiny-YOLOv3. The Ascend 310 posted 175 IPS and 71 IPS for MobileNetV2 and Tiny-YOLOv3, respectively, marking

excellent parallel computing performance. The Zynq platform, by contrast, posted lower scores of 134 IPS for MobileNetV2 and 59 IPS for Tiny-YOLOv3 owing to increased communication overhead between programmable logic and processing system. The findings reinforce the fact that specialized processing pipelines in the likes of ASICs and NPUs offer a decisive benefit in high-frequency inference.

Energy efficiency, yet another important aspect for battery-powered or thermally limited edge devices, was calculated as the average energy used per inference. The Edge TPU was the most energy-efficient model, using merely 0.48 joules per inference for MobileNetV2 and 0.71 joules for Tiny-YOLOv3. The Ascend 310 came next with 0.55 joules and 0.79 joules, respectively. Conversely, the FPGA utilized the highest amount of power—0.73 joules for MobileNetV2 and 0.94 joules for Tiny-YOLOv3—owing to its comparatively higher dynamic power requirements because of reconfiguration of logic and more accesses to memory.

Model accuracy was maintained on all platforms with minimal difference from the original models running on a standard GPU at training. MobileNetV2 preserved a top-1 accuracy of about 71.2% on ImageNet, while Tiny-YOLOv3 preserved a mean average precision (mAP) of 33.4% on the Pascal VOC dataset. Quantization and format conversion were not found to incur meaningful loss in predictive quality. This proves that specialized digital hardware can be used to run optimized, quantized models without affecting the predictive fidelity underlying the models.

Combined, the outcomes confirm that ASICs such as the Edge TPU offer the most comprehensive performance in latency and energy efficiency and thus are best suited for situations involving real-time responsiveness and minimal power consumption. NPUs such as the Ascend 310 offer equivalent results but with enhanced model support and integration flexibility. FPGAs, while less efficient in raw numbers, offer value in their reconfigurability and reduced non-recurring engineering (NRE) expense, especially in prototyping or multi-model scenarios. These results lay the groundwork for further exploration of the trade-offs and design issues for real-world deployment of AI inference workloads on edge hardware.

V. DISCUSSION

The empirical findings in this study highlight the tremendous benefits of custom digital hardware in speeding up AI inference in edge devices. Yet, upon closer inspection, it is evident that hardware selection cannot be decided universally with only performance considerations such as latency or throughput. Rather, deployment choices need to be made cognizant of a nuanced trade-off between performance, power efficiency, cost, scalability, and hardware-software integration. In this section, we place the results in context, compare the strengths and weaknesses of each hardware category, and explain how the results relate to actual edge AI deployments.

One of the most salient conclusions from this work is the dominance of ASICs in terms of high throughput and low energy consumption. The Edge TPU exhibited the quickest inference rates and smallest power used per operation, primarily because it is designed with fixed functionality that is optimized for matrix operations common in neural networks. Yet this specialization brings with it inflexibility. ASICs are not flexible when considering model updates, algorithmic shifts, and retraining needs. In those applications where the AI model changes very fast, e.g., those that employ online learning or need support for multiple tasks, the immutability of ASIC architecture can be a considerable bottleneck. Additionally, the ASIC design's high non-recurring engineering expense and long time-to-market make them economically practical only when scaled or used for fixed, long-term algorithmic applications.

FPGAs offer a counterpoint to ASICs with unmatched flexibility. Due to their reconfigurable nature, hardware developers can modify the architecture for different AI models, which is highly useful in scenarios where task requirements keep changing over time. In domains such as autonomous systems, industrial automation, or defense, where system adaptability and rapid prototyping are crucial, FPGAs offer a viable option. But this flexibility is achieved at the expense of performance and energy efficiency, as demonstrated by the results. The FPGA platform consistently trailed behind ASICs and NPUs in inference speed and power consumption. Moreover, the process of developing FPGA-based solutions tends to be more intricate, demanding knowledge of hardware description languages and meticulous timing analysis, which might not be universally available within every development team.

NPUs provide a compromise between the inflexibility of ASICs and the programmability of FPGAs. The Ascend 310 demonstrated consistent performance in all aspects, which makes it applicable to various applications requiring efficiency as well as flexibility. NPUs are especially beneficial in consumer devices like smartphones, wearables, and smart cameras, where they are embedded as dedicated co-processors inside SoCs. Their support for standard AI frameworks like TensorFlow Lite or ONNX makes deployment easy and speeds up time-to-market. NPUs, however, continue to encounter issues like vendor lock-in, limited transparency about architectural details, and less developed software toolchains than more mature platforms like CPUs or GPUs.

Another key realization is the contribution of software stack maturity and tooling support in the successful implementation of AI inference on custom hardware. The availability of strong compilers, quantization software, and runtime environments has a big impact on developer productivity and model portability. For example, the Edge TPU is aided by close integration with TensorFlow Lite and a strong compiler, allowing easy deployment for supported models. Likewise, NPUs are also becoming more accessible as result of enhanced support in mainstream frameworks. FPGAs, however, are still behind in this regard, with most deployment pipelines necessitating hand-optimization and toolchain-specific tuning, thus adding to the development load.

In a deployment perspective, power usage turns into the overriding concern on battery-powered or heat-constrained systems. Results overwhelmingly indicate ASICs are the best choice under these circumstances while FPGAs might be preferred where mains-power in industrial setups prevails. The preservation of model accuracy on every platform implies the latest quantization methods help to maintain the prediction quality when running compressed and optimized models at the edge, enabling edge inference a viable fact on complex AI applications.

The choice of a specific hardware accelerator should be based on the target application area, cost model, deployment volume, and frequency of updates for the AI models at hand. For static, high-volume applications that demand ultra-low latency and energy efficiency, ASICs are optimal. For dynamic, flexible deployments, FPGAs continue to be valid despite performance sacrifices. NPUs then come forward as an equilibrium option, especially for integrated consumer products where space, power, and flexibility need to go hand-in-hand. All these findings together indicate the need for hardware-software co-design, wherein model structure, quantization approach, and deployment pipeline are optimized together with the selected hardware platform to realize optimal performance under practical constraints.

VI. CONCLUSION

The explosive growth in the use of artificial intelligence in edge computing settings has called for reexamining conventional hardware design, particularly the execution of inference operations in power-

limited, real-time applications. This paper has extensively explored the contribution of specialized digital hardware—specifically, application-specific integrated circuits (ASICs), field-programmable gate arrays (FPGAs), and neural processing units (NPUs)—to facilitating scalable and power-efficient AI inference on the edge devices. By empirical benchmarking of actual-world models such as MobileNetV2 and Tiny-YOLOv3 on chosen commercial platforms, a more detailed comprehension has been gained about the strengths, weaknesses, and application-specific suitability of each hardware type.

The research concludes that ASICs, as represented by Google's Edge TPU, provide unmatched inference speed and energy efficiency. These features make ASICs best suited for situations where the AI model is not changing, there are stringent latency requirements, and energy efficiency needs to be minimized. Their lack of flexibility in handling updates to the model or being compatible with various architectures lowers their suitability in dynamic or multi-tasking environments. ASICs are therefore good candidates for broad-scale rollouts of stable models, e.g., in smart surveillance, biometric authentication systems, and embedded vision modules for automotive systems.

FPGAs, exemplified in this work by Xilinx's Zynq UltraScale+ platform, provide value in settings where reconfigurability and low-volume customization are high. Although not as power-consumption-effective or as speedy as ASICs, their capability to facilitate fast prototyping and the use of several AI models makes them unique in research environments, industrial automation systems, and mission-critical operations where flexibility may be more valuable than pure throughput. Their long learning curve, increased development time, and toolchain difficulty, however, are ongoing concerns that need to be overcome with more user-friendly software support and higher-level synthesis tools.

NPUs such as Huawei’s Ascend 310 bridge the gap between specialization and flexibility. They offer competitive latency and energy metrics while being more accessible for developers accustomed to mainstream AI frameworks. Their integration into modern SoCs allows consumer electronics to deliver responsive, low-power AI functionalities, such as in voice assistants, augmented reality systems, and smart home devices. The primary issues with NPUs are proprietary architectures, lock-in to an ecosystem, and shifting support for new model types such as transformers or graph neural networks. Nevertheless, NPUs seem destined to become a ubiquitous building block in AI-enabled edge devices, particularly where software flexibility must be balanced with moderate hardware specialization.

For all three platforms, the research establishes that it is feasible to maintain model accuracy after quantization and optimization, and hence deployment on bespoke digital hardware is not only a possibility but also highly desirable. The performance outcomes reinforce that hardware selection should be tightly coupled with the AI workload features of model complexity, inference rate, update interval, and deployment environment. No one solution appears as the overall best; rather, they all have a place based on interactions between performance goals, power limitations, development freedom, and economics.

In the future, the co-design of hardware and algorithms will increasingly be crucial. Hardware-aware model development, compiler optimization, and toolchain integration are enabling factors that can enable further efficiency improvements. In addition, as edge AI workloads grow in sophistication—from basic classification to real-time learning and federated learning environments—hardware platforms also need to scale to accommodate new memory structures, on-chip learning, and secure model updates. The next generation of research should investigate hybrid architectures that blend ASIC, FPGA, and NPU elements in one system that provides a configurable compute substrate tuned to both performance and responsiveness.

Overall, this paper reiterates the revolutionary capability of tailored digital hardware in addressing the computational requirements of AI at the edge. Through the comprehension of trade-offs between various architectures and matching them with application-specific requirements, system designers and engineers can provide intelligent, responsive, and efficient edge systems that advance the edge of AI innovation.

VII. REFERENCES

1. Y. Chen, T. Krishna, J. Emer, and V. Sze, "Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks," IEEE Journal of Solid-State Circuits, vol. 52, no. 1, pp. 127–138, Jan. 2017.
2. S. Han, H. Mao, and W. J. Dally, "Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding," Int. Conf. on Learning Representations (ICLR), Apr. 2016.
3. J. Qiu et al., "Going Deeper with Embedded FPGA Platform for Convolutional Neural Network," in Proc. ACM/SIGDA Int. Symp. on Field-Programmable Gate Arrays, Monterey, CA, USA, 2016, pp. 26–35.
4. Y. Umuroglu et al., "FINN: A Framework for Fast, Scalable Binarized Neural Network Inference," in Proc. ACM/SIGDA Int. Symp. on Field-Programmable Gate Arrays, Monterey, CA, USA, 2017, pp. 65–74.
5. L. Zhang, H. Wang, S. Liu, and Q. Xu, "AI Inference Acceleration Using Neural Processing Units: A Comparative Evaluation," IEEE Trans. on Computers, vol. 73, no. 3, pp. 458–470, Mar. 2024.
6. V. Sze, Y. Chen, T. Yang, and J. S. Emer, "Efficient Processing of Deep Neural Networks: A Tutorial and Survey," Proc. of the IEEE, vol. 105, no. 12, pp. 2295–2329, Dec. 2017.
7. MLCommons, "MLPerf Tiny Inference Benchmark v1.0 Results," MLCommons Organization, Dec. 2024.
8. A. Zhai and S. Cao, "Design and Implementation of Energy-Aware AI Accelerators on Edge Devices," IEEE Embedded Systems Letters, vol. 16, no. 4, pp. 124–128, Dec. 2024.
9. F. Chen, M. Andersch, and O. Mutlu, "Software-Hardware Co-Design for Deep Learning on Edge Devices: Trends and Challenges," IEEE Design & Test, vol. 41, no. 2, pp. 42–50, Apr. 2024.
10. H. Lin, D. Tang, and J. Li, "Benchmarking AI Hardware Accelerators for Edge Applications: Methodologies and Case Studies," ACM Transactions on Embedded Computing Systems, vol. 23, no. 1, pp. 1–24, Nov. 2024.