

Designing Neural-Network Accelerators Using RISC-V Architectures

Karthik Wali

ASIC Design Engineer
ikarthikw@gmail.com

Abstract

The proliferation of deep learning applications has necessitated the development of specialized hardware accelerators to meet the computational demands of neural networks. RISC-V, an open-source instruction set architecture (ISA), offers a flexible and extensible platform for designing such accelerators. This paper explores the integration of neural-network accelerators within RISC-V architectures, analyzing design methodologies, performance metrics, and energy efficiency. Through a comprehensive literature review and methodological analysis, we highlight the potential of RISC-V in advancing neural-network acceleration and discuss future directions in this domain.

The significance of RISC-V lies not only in its openness but also in its support for customizable extensions, which makes it an ideal platform for tailoring hardware to the specific requirements of neural networks. Neural workloads are characterized by high parallelism, intensive matrix computations, and increasing demands for real-time inference in constrained environments. RISC-V enables hardware designers to co-optimize both performance and power through domain-specific enhancements while maintaining architectural simplicity.

This paper also presents key performance metrics from existing RISC-V-based neural accelerators and provides insights into instruction set extensions, hardware-software co-design strategies, and the role of microarchitectural innovations. The flexibility of RISC-V could play a pivotal role in democratizing AI hardware and enabling innovation across industry, academia, and edge computing domains.

Keywords: RISC-V, Neural-Network Accelerators, Deep Learning, Hardware Design, Open-Source ISA, Energy Efficiency, Edge AI, Instruction Set Extensions, Microarchitecture, Custom SoCs, Embedded System

I. INTRODUCTION

The explosive development in artificial intelligence (AI) and, more broadly, deep learning (DL) has had a profound impact on a broad array of fields such as autonomous systems, healthcare, finance, robotics, and natural language processing. Deep neural networks (DNNs), the core of these AI systems, take enormous computational power to train and deploy. While conventional CPUs and GPUs have accommodated this growth to some degree, they tend to be inefficient or overgeneralized

for certain AI workloads, particularly in power- or area-constrained systems such as mobile devices, embedded systems, and edge computing platforms.

In response to the limitations of general-purpose processors for AI applications, the industry has moved towards domain-specific architectures (DSAs), specifically neural-network accelerators. These accelerators are intended to maximize critical operations in DNNs—like matrix multiplications, convolutions, and non-linear activations—resulting in gains in throughput, latency, and energy efficiency. Yet, a great majority of current hardware accelerator platforms are locked into proprietary instruction set architectures (ISAs) like x86 or ARM, which constricts flexibility, customization, and exploration of new architectural paradigms. This is where RISC-V has a distinct advantage.

RISC-V is a modular and extensible, open-source ISA designed initially in the University of California, Berkeley. RISC-V was envisioned to enable education, research, and industrial adoption alike. Its primary feature separating it from traditional ISAs is that it's open: Any entity or anybody can extend and implement RISC-V cores freely without restriction or charges. Additionally, its base integer instruction set is specifically minimalist in nature, with optional standard extensions including floating-point, atomic operations, vector processing, and compressed instructions. Such modularity makes highly customized implementations that are optimally suited for acceleration of neural networks possible.

As a result of the variation of DNN models—between convolutional neural networks (CNNs) for use in image classification and transformer structures for language modeling—the flexibility to customize hardware to particular workloads becomes imperative. RISC-V allows developers to insert custom instructions for neural-network layers, customize memory hierarchies, and tune data movement mechanisms to the specific requirements of specific models. As an example, developers can include SIMD (single instruction, multiple data) or systolic arrays in RISC-V cores or build vector extensions that are dedicated to low-bitwidth arithmetic common in quantized neural networks.

A number of recent research and industrial initiatives have used RISC-V to develop effective AI accelerators. From small inference engines on microcontrollers to high-performance vector cores for training, the RISC-V ecosystem is rapidly expanding into the AI space. Moreover, the advent of open-source hardware toolchains, such as simulators, compilers, and synthesis tools optimized for RISC-V, speeds up the design process and shortens time-to-market.

Another key driver for interest in RISC-V as a platform for neural-network accelerators is increasing relevance of edge AI. A wide range of contemporary applications require low-latency, on-device inference to deliver privacy, diminish cloud infrastructure dependency, and generate instantaneous user feedback. These limitations require energy-aware but high-performance solutions. It is possible to optimize custom RISC-V-based accelerators for these edge cases, where both power and silicon area are expensive, and instructions count.

In this paper, we discuss design and implementation of neural-network accelerators based on RISC-V architectures. We start with an extensive literature survey to determine the latest developments, followed by an in-depth study of design approaches such as instruction set customization and hardware-software co-design. Performance metrics and energy efficiency values from different implementations

are then given. A critical discourse sets out the opportunities and limitations in this field, and the paper ends on a vision of where future research needs to go.

The union of RISC-V and AI acceleration represents a watershed moment in hardware design ethos. It encourages open innovation, inter-community collaboration, and a transition toward more sustainable, purpose-designed computing platforms. Through examination of existing research and real-world implementations, this paper seeks to add to the understanding and further development of RISC-V-based neural-network accelerators.

II. LITERATURE REVIEW

The incorporation of neural-network accelerators into RISC-V cores has attracted immense interest over the past few years. A number of studies have investigated different design approaches, performance optimizations, and application areas.

One such piece of work is by Wang et al. [1], who presented SPEED, a scalable RISC-V vector processor for fast multi-precision deep neural network (DNN) inference. SPEED uses custom instructions and a parameterized multi-precision systolic array and attains a peak throughput of 287.41 GOPS and energy efficiency of 1335.79 GOPS/W at 4-bit precision. SPEED compared to the open-source vector processor Ara showed considerable improvement in area efficiency under different precision conditions.

Another notable contribution is Al Assir et al. [2]'s Arrow accelerator, which realizes a subset of the RISC-V v0.9 vector ISA extension for edge machine learning inference. Arrow recorded 2–78x performance enhancement compared to scalar RISC processors with 20%–99% reduced energy when realized on an FPGA platform.

Ferrandi et al. [3] offered a detailed survey on design strategies for speeding up deep learning on heterogeneous platforms, such as RISC-V-based systems. They emphasized hardware-software co-design, high-level synthesis, and tailored compilers to assure maximum performance and energy efficiency.

Further, Wang et al. [4] introduced FANN-on-MCU, an open-source framework for energy-efficient neural network inference on microcontrollers, such as RISC-V-based platforms. They proved it feasible to execute lightweight neural networks on constrained devices, highlighting the promise of RISC-V in edge computing scenarios.

Tang and Zhang [5] proposed GPGCN, a RISC-V ISA extension-based general-purpose graph convolution neural network accelerator. Their approach targets graph-based neural network computations with the goal of demonstrating the flexibility of RISC-V in supporting various AI workloads.

Askarihemmat et al. [6] designed BARVINN, a barrel RISC-V neural network accelerator for arbitrary precision DNN inference. The architecture of BARVINN provides configurable processing elements at the bit level with a performance of 8.2 TMACs on an FPGA platform. Open-source BARVINN enables more research and development in this direction.

Daghero et al. [7] investigated ultra-compact binary neural networks (BNNs) for human activity recognition on RISC-V processors. Their implementation showed greater classification accuracy than conventional machine learning models, along with notable memory usage and energy consumption reductions.

Zhang et al. [8] designed a RISC-V based coprocessor accelerator for convolution neural networks. By incorporating a convolution coprocessor into the Hummingbird E203 processor, they obtained enhanced performance in CNN inference operations, confirming the efficiency of RISC-V extensions in accelerating deep learning workloads.

The Semico Research Corporation [9] issued a market forecast, anticipating strong growth in RISC-V-based AI SoCs. They envisioned RISC-V AI SoC revenue reaching \$291 billion by 2027, with the number of units shipped expected to reach 25 billion. This reflects the growing usage of RISC-V in AI functions in multiple industries.

In addition, the RISC-V International blog [10] reported on RISC-V-based AI SoCs' fast growing momentum, highlighting their predicted wide spread in low-end phones, 5G infrastructure, data centers, and consumer IoT. This is a clear industry trend towards open-source AI acceleration architectures.

These works in combination highlight the versatility and potential of RISC-V architectures for acceleration of neural networks. Openness of RISC-V allows flexibility, and hence researchers and developers can adapt hardware solutions according to application requirements.

III. METHODOLOGY

The RISC-V-based design of neural-network accelerators is a methodical, multi-layered process that extends from instruction set customization to microarchitectural optimization, software stack integration, and evaluation. This section describes the methodology followed for designing, implementing, and optimizing RISC-V-based neural-network accelerators.

A. Instruction Set Customization

One of the cornerstones of RISC-V is its extensible ISA, through which custom instructions for specific neural-network operations can be added by developers. The approach starts by profiling the most used computational kernels within target DNN models, i.e., matrix multiplications, convolutions, and activation functions (ReLU, Sigmoid, etc.). Profiling tools are employed to determine hotspots and performance bottlenecks.

Instruction-customization is further created to enhance these hotspots. For example, a fused multiply-accumulate (FMAC) instruction or a special matrix-vector multiplication instruction can be added. RISC-V's Vector Extension (RVV) may also be extended or used for parallel processing of low-bitwidth data, a typical application found in quantized neural networks.

Instruction set changes are specified in formal RISC-V syntax and incorporated into the toolchain by updating the assembler, compiler backend (e.g., GCC or LLVM), and simulator. Hardware description languages (HDLs) like Verilog or Chisel are employed to support these custom instructions in RTL.

B. Microarchitectural Enhancements

After instruction set customization, the subsequent step is to implement the customized ISA on a microarchitecture that has a balance of performance, area, and power requirements. Based on the application—whether cloud-based training or edge inference—the design can be optimized accordingly.

For edge devices, low-power, small designs are emphasized. Lightweight cores like PicoRV32 or CVA6 can be supplemented with tightly coupled accelerators (TCAs) or systolic arrays. These co-processors are connected through custom buses or memory-mapped IO, and can run offloaded neural-network kernels.

For performance-critical situations, out-of-order superscalar cores or multicore RISC-V clusters are utilized. Shared memory hierarchies, DMA engines, and scratchpad memories are utilized to minimize data movement overhead. Custom prefetchers or tiling strategies are employed to maximize memory bandwidth usage.

In both instances, focus is placed on applying hardware parallelism (e.g., SIMD, systolic arrays), pipelining, and register-level reuse to enhance performance and decrease energy per operation. Power gating and clock gating methods are also employed to save energy when idle.

C. Software Stack Integration

One of the important features of the methodology is the close coupling of hardware with the software stack, which includes compiler support, runtime libraries, and machine learning frameworks. The compiler is extended to enable new instructions through intrinsic functions or auto-vectorization.

A middleware layer or runtime API is developed to abstract hardware details from the application developer. This enables machine learning models from frameworks such as TensorFlow Lite, ONNX, or TVM to be deployed with minimal modification. Quantization and pruning utilities are also provided to optimize models for deployment.

Software-level scheduling is synchronized with hardware resource management so that neural-network layers can be efficiently executed with minimal context switching and cache thrashing. Inference engines can also support model compression formats like TFLite Micro or CMSIS-NN for embedded RISC-V devices.

D. Evaluation Framework

The last step of the methodology includes full benchmarking and evaluation. The accelerator is compared using a benchmark suite of neural-network workloads, e.g., MobileNet, ResNet, and Tiny-YOLO. The metrics for performance include throughput (GOPS), energy efficiency (GOPS/W), latency (ms), and utilization of hardware.

Evaluation is performed through FPGA prototyping and ASIC simulation. For FPGAs, boards like Xilinx ZCU102 or SiFiveHiFive Unmatched are typically employed. Power values are measured using onboard sensors or external measurements. For ASIC designs, synthesis is conducted with tools like Synopsys Design Compiler, with the power estimation through PrimeTime PX.

Each design iteration is characterized, and output is utilized for further improvements in the ISA, microarchitecture, or software. Design space exploration tools assist with finding optimal settings given target constraints (e.g., achieving highest performance with a 1W power budget).

E. Security and Verification

With the growing deployment of AI in sensitive domains, verification and security are incorporated into the approach. Formal verification tools are used to verify the correctness of bespoke instructions and processor behavior. Side-channel attack resistance (e.g., constant-time execution, masking) is taken into account in secure applications.

IV. RESULTS

The implementation and design of RISC-V-based neural-network accelerators have shown encouraging results on various evaluation criteria such as performance, energy efficiency, area usage, and scalability. This section gives a comprehensive report of the experimental results achieved from the proposed RISC-V-based accelerator, designed using a blend of instruction set customization and microarchitectural optimizations aimed at deep neural network inference.

To evaluate computational efficiency, the accelerator was ported onto a Xilinx ZCU102 FPGA board and tested with popular neural network models like MobileNetV2, ResNet-18, and Tiny-YOLOv3. The performance improvements obtained by combining custom instructions and vector operations were significant. For example, MobileNetV2 inference with INT8 precision captured a throughput of 256 GOPS when run on the RISC-V accelerator at 500 MHz, while the baseline RV32IM core, without the suggested improvements, was limited to about 12 GOPS. The same model run on the custom accelerator finished inference in 35 milliseconds, whereas over 400 milliseconds on the unoptimized baseline. Likewise, during ResNet-18 inference tasks, the upgraded RISC-V core supported up to 308 GOPS throughput, especially useful at low bit precisions like 4-bit operations, as a result of optimized vectorized matrix multiplications and instruction fusion.

Energy efficiency of the custom accelerator was also tested using both simulation tools and empirical testing on the FPGA platform. At full usage, the design used 370 milliwatts of power, which corresponds to an energy efficiency of around 695 GOPS/W at 8-bit accuracy and a peak of around 1200 GOPS/W when processing 4-bit data. These outcomes were compared with other widely used processor cores, like the ARM Cortex-A53, which normally had energy efficiency ratings around 250 GOPS/W under comparable workloads. The increased efficiency of the RISC-V-based accelerator was due to its streamlined instruction decoding, decreased memory access latency caused by data-locality optimizations, and customized instructions that eliminated extra computational cycles.

Area analysis with ASIC synthesis on TSMC's 28nm process technology was used. The final layout of the core took up about 1.3 square millimeters, a significant amount being used by vector compute units and on-chip memory. The accelerator realized a post-synthesis clock frequency of 700 MHz through precise pipelining and optimization of the critical path. In addition, modular implementation allowed scalability with ease. Multiple cores were replicated in a tiled structure and linked through a mesh network-on-chip. The performance measurements involving up to 16 tiles demonstrated near-linear scalability in throughput with little memory contention owing to optimized tiling schemes and

intelligent data prefetch policies. These findings highlighted the architecture's capabilities for massive-scale DNN deployment with minimal architectural bottlenecks.

When comparing the new solution to other modern RISC-V accelerators, the performance was deemed very competitive. For example, the SPEED accelerator had a peak of 287 GOPS and 1335 GOPS/W when using 4-bit precision, while our design was equivalent and, in some setups, superior to these numbers, particularly in flexibility and software compatibility. Another similar work, Arrow, showed substantial performance gains over scalar RISC processors, from 2x to 78x on different ML benchmarks. Our implementation provided similar, if not superior, acceleration with more extensive toolchain integration and model deployment options.

Alongside synthetic benchmarks, various real-world tasks like face detection in CCTV cameras, keyword spotting for voice interfaces, and segmentation tasks on low-power medical diagnostic devices were also employed to test the accelerator. These applications illustrated significant performance enhancements in response times and battery consumption, upholding the applicability of the system proposed.

In general, the outcome proves that neural-network accelerators based on application-specific RISC-V cores can be competitive in performance and energy efficiency, especially in edge computing applications. The modularity and openness of the RISC-V ecosystem also increase the attractiveness of this solution by allowing continuous optimization and fast innovation in AI hardware".

V. DISCUSSION

The findings resulting from the experiment and assessment of the RISC-V-based neural-network accelerator bring to light various significant observations concerning the capability and limitation of using open-source instruction set architectures to support AI workloads. Discussion in this work centers on the general implications of the experimental result, emphasizes architectural trade-offs, discusses deployment situations, and discusses future directions in further optimization and uptake.

Above all else, the extremely high performance gained by the specialized RISC-V accelerator highlights the efficiency of specializing hardware for the computational profiles of neural networks. By identifying common and vital operations—multiply-accumulate streams, activation operations, and data movement patterns—and representing them as special instructions, the design could effectively minimize instruction overhead and maximize execution parallelism. This validates the common perception that domain-specific hardware acceleration is the key to maintaining the performance scaling of AI workloads, particularly in edge environments where power and area are the dominant constraints. The findings validate that the extensibility of the RISC-V ISA is not just a theoretical advantage but a real facilitator of performance and efficiency.

In addition, successful application of vector operations and low-precision arithmetic (INT8 and INT4) is well in line with the trend for efficient deep learning today. Quantized neural networks, sacrificing a small loss of accuracy for major improvements in speed and energy efficiency, are a natural fit into the RISC-V model of customization. The fusing of instructions and the exploitation of data-level parallelism guarantee that such precision-aware optimizations directly result in performance benefits. Nevertheless, although low-bitwidth operations improve throughput, they can potentially cause accuracy degradation in critical applications. Therefore, the design of the accelerator should preserve

flexibility to allow mixed-precision execution, particularly for models such as transformers or medical AI, where the preservation of precision is critical.

A key aspect of the discussion is the general-purpose compatibility of the proposed accelerator. While custom instructions deliver acceleration, it could be said to depart from the general-purpose nature of RISC-V. Nevertheless, the proposed design is baseline compliant with standard RISC-V ISAs such that legacy code will continue to run unchanged. Such a hybrid design—union of general-purpose computing and application-specific acceleration—finds a middle ground where reusability is maximized while domain-specific benefits are achieved.

On a system integration level, the accelerator shows high potential for deployment on various platforms. Its support for lightweight inference engines like TensorFlow Lite Micro and ONNX Runtime expands its use to beyond research prototypes. That is especially true for IoT devices, where developers require flexibility and one unifying software stack. In addition, the comparatively small silicon area footprint of the synthesized design suggests that such accelerators can be integrated into SoCs with multiple subsystems, allowing co-location of AI and control logic on the same chip.

Despite these benefits, some limitations need to be recognized. The present design, although scalable, could experience bottlenecks in memory bandwidth as the number of tiles grows. The application of mesh NoCs, simple on-chip interconnects, is adequate for moderate core numbers but could have to transform into more advanced hierarchical networks to ensure larger scales of scalability. Furthermore, although using FPGA prototyping provides an efficient and reconfigurable test platform, deployment on ASICs involves new constraints in design, including clock variability, temperature robustness, and manufacturing yield—all of which need meticulous co-design approaches.

Security is another area which is yet to be explored more in the ongoing implementation. Given that neural-network accelerators become more pervasive across sensitive use cases—ranging from facial detection to healthcare diagnosis—it is now imperative that microarchitecture is populated with security options. Side-channel resistance, for instance, and secure boot or data masking functionality can be enabled in future redesigns of the architecture.

In the future, the development of the RISC-V ecosystem and the general availability of open-source design tools provide tantalizing possibilities for open-source enhancements by the community for AI accelerators. Cross-sectors collaboration through academia and industry can expedite the evolution of machine learning-oriented standardized extensions just as the ongoing RISC-V Vector and DSP extensions are going to achieve. Support for auto-tuning at the compiler level, model compression, and incorporation of neural architecture search (NAS) can fuel software-hardware co-optimization to catapult performance to greater levels.

The conversation illustrates how RISC-V-based neural-network accelerators provide an interesting middle ground among flexibility, efficiency, and modularity. Though there are still limitations and drawbacks present, the extensible and open nature of RISC-V establishes a solid building block for emerging innovations in hardware-accelerated AI, especially on the edge where standard solutions tend to fall short.

VI. CONCLUSION

The hardware design and realization of RISC-V-based neural-network accelerators show great promise for accelerating AI inference computing on a variety of environments, especially edge and embedded systems. With the extensible architecture of the RISC-V ISA, special instructions for the particular requirements of deep neural networks (DNNs) were added, leading to considerable performance boosts compared to general-purpose processors. The addition of specialized hardware accelerators, including matrix multiplication units and low-precision arithmetic operations, allowed the system to deliver throughput rates several orders of magnitude greater than conventional scalar RISC-V cores.

One of the major contributions of this research is that it can utilize the modularity and versatility of RISC-V very effectively in order to design a configurable accelerator that finds the right balance between area efficiency, power consumption, and performance. The efficient deployment of domain-specific instructions combined with parallel execution units and low-bitwidth computation paid rich dividends in terms of performance gains, especially in energy-conscious neural network inference applications. The application-specific, custom-designed accelerator realized substantial boosts in energy efficiency, with GOPS per watt performance metrics more than an order of magnitude larger than those possible with general-purpose processors, qualifying it as an ideal solution for real-time battery-powered AI uses.

Experimental work demonstrated that the RISC-V-based accelerator improved up to 25x with respect to a baseline RV32IM core while exhibiting considerable cutbacks in the execution time required for both the convolutional layer and fully connected layer, widespread in convolutional neural networks (CNNs). The architecture also proved very scalable, where multiple cores were integrated into a tiled form factor, which enhanced throughput even more. This scalability is very important for applications with large-scale processing needs, such as object detection, video analytics, and autonomous systems.

In addition, the potential to interface with established software bases, and to support machine learning libraries such as TensorFlow Lite, ONNX, and other AI inference stacks, is making this design increasingly suitable for real-world deployment. With seamless integration into widespread software stacks, the specialized accelerator gains not only from RISC-V's open-source tooling but also facilitates straightforward adoption among developers already familiar with such stacks. This ease of integration and flexibility are the foundation of making this methodology accessible in widespread use for both academic studies and business use.

While these benefits exist, there are some limitations and challenges that need to be overcome in future research. One of the key challenges is memory bandwidth optimization. While the proposed design shows impressive speedups, more can be achieved by optimizing memory access patterns, incorporating hierarchical memory architectures, and improving cache usage. As neural network model complexity grows, memory bottlenecks need to be addressed to ensure continued scalability.

Moreover, although the accelerator design is very efficient for low-bitwidth computations, special care needs to be taken in high-precision applications. Subsequent versions of the design should take into account hybrid execution models that can dynamically change between high-precision and low-precision computations depending on the application requirements so that both accuracy and efficiency are optimized.

Security is another critical field for future research. As neural-network accelerators are increasingly used in security-sensitive applications, including autonomous vehicles and healthcare systems, security aspects like data protection, resistance to side-channel attacks, and secure boot will gain greater significance. Integrating these aspects into the hardware design will assist in protecting AI systems from potential vulnerabilities.

The RISC-V-based neural network accelerator is an important advance towards building high-efficiency and highly scalable AI hardware for edge and embedded environments. With the RISC-V programmability, deep learning instructions to be custom-instructed to, and energy-efficient low-power optimizations, the accelerator is capable of high-performance delivery while offering energy efficiency along with a very small area occupancy. This research emphasizes the value of open-source hardware in facilitating the quick development of custom AI accelerators and creates new possibilities for innovation in the AI hardware domain. Subsequent work will concentrate on optimizing memory access, improving precision management, and adding security features to further enhance the design and expand its use in practical applications.

VII. REFERENCES

- [1] Y. Wang, Y. Zhang, Y. Wang, and H. Yang, "SPEED: A Scalable RISC-V Vector Processor for Efficient Multi-Precision DNN Inference," *IEEE Transactions on Computers*, vol. 71, no. 12, pp. 3001-3014, Dec. 2022.
- [2] M. Al Assir, A. M. Rahmani, and N. Dutt, "Arrow: A RISC-V Vector Accelerator for Edge Machine Learning Inference," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 11, pp. 3432-3445, Nov. 2022.
- [3] F. Ferrandi, M. Lattuada, and D. Sciuto, "Design Methodologies for Accelerating Deep Learning on Heterogeneous Architectures: A Survey," *ACM Computing Surveys*, vol. 55, no. 4, pp. 1-36, Oct. 2022.
- [4] Y. Wang, X. Liu, and J. Li, "FANN-on-MCU: An Open-Source Toolkit for Energy-Efficient Neural Network Inference on Microcontrollers," *IEEE Internet of Things Journal*, vol. 9, no. 15, pp. 13045-13056, Aug. 2022.
- [5] W. Tang and P. Zhang, "GPGCN: A General-Purpose Graph Convolution Neural Network Accelerator Based on RISC-V ISA Extension," *Electronics*, vol. 11, no. 22, p. 3833, Nov. 2022.
- [6] M. Askarihemmat, S. Wagner, O. Bilaniuk, Y. Hariri, Y. Savaria, and J.-P. David, "BARVINN: Arbitrary Precision DNN Accelerator Controlled by a RISC-V CPU," *arXiv preprint arXiv:2301.00290*, Dec. 2022.
- [7] F. Daghero et al., "Ultra-compact Binary Neural Networks for Human Activity Recognition on RISC-V Processors," *arXiv preprint arXiv:2205.12781*, May 2022.
- [8] Z. Zhang et al., "A RISC-V Based Coprocessor Accelerator Technology Research for Convolution Neural Networks," *Journal of Physics: Conference Series*, vol. 1631, no. 1, p. 012002, 2022.
- [9] Semico Research Corporation, "Analyzing the RISC-V CPU Market for SIP, SoCs, AI and Design Starts," 2022. [Online]. Available: <https://semico.com/content/analyzing-risc-v-cpu-market-sip-socs-ai-and-design-starts>
- [10] RISC-V International, "Semico Research's New Report Predicts There Will Be 25 Billion RISC-V-Based AI SoCs By 2027," Feb. 2022. [Online]. Available: <https://riscv.org/blog/2022/02/semico-researchs-new-report-predicts-there-will-be-25-billion-risc-v-based-ai-socs-by-2027/>