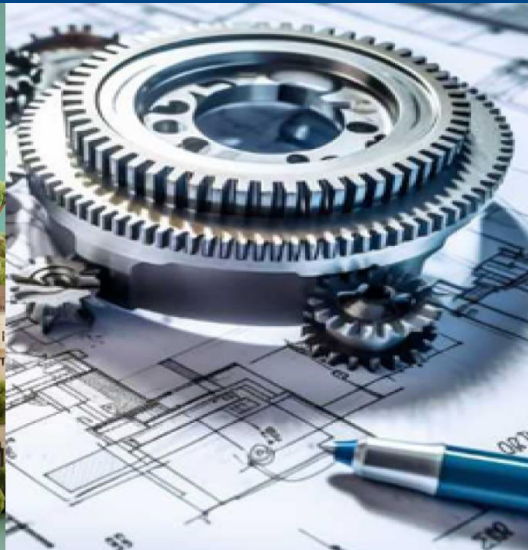
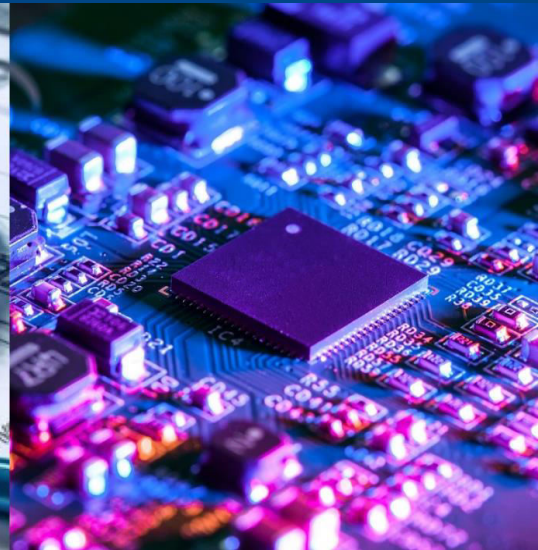
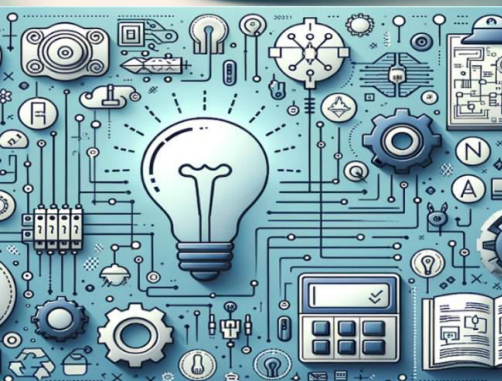




International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 9.864

Volume 9, Issue 10, October 2026



International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

GPU and CPU Performance Comparison for AI Applications

Quennie Marie U. Falcon

Graduate Studies, Surigao del Sur State University, Rosario, Tandag City, Philippines

qmufalcon@nemsu.edu.ph

ABSTRACT: The rapid growth of artificial intelligence (AI) applications, particularly deep learning, has intensified the demand for computing hardware capable of handling large-scale parallel computations efficiently. This study examines the performance differences between Central Processing Units (CPUs) and Graphics Processing Units (GPUs) when executing AI workloads, focusing on training time, inference speed, throughput, and power efficiency. Using a descriptive-comparative research design based on a structured review and synthesis of published benchmarking studies and hardware architecture specifications, the paper analyzes how the architectural differences between CPUs, which favor a small number of complex cores optimized for low-latency sequential execution, and GPUs, which favor thousands of simpler cores optimized for parallel throughput, translate into measurable performance gaps across representative deep learning tasks such as image classification and sequence modeling. The findings show that GPUs consistently outperform CPUs on workloads dominated by large matrix and convolution operations, with reported speedups ranging from roughly four to more than six times, while CPUs remain competitive, and in some optimized configurations superior, for workloads involving irregular computation, small batch sizes, and memory-intensive models. The study concludes that hardware selection for AI applications should be guided by workload characteristics rather than a uniform preference for GPU acceleration.

KEYWORDS: GPU architecture, CPU architecture, parallel computing, deep learning, artificial intelligence, performance benchmarking, computer organization, hardware acceleration.

I. INTRODUCTION

Artificial intelligence, and deep learning in particular, has become central to applications ranging from image recognition and natural language processing to autonomous systems and recommendation engines. These workloads rely heavily on operations such as matrix multiplication and convolution, which are computationally intensive but highly parallelizable. This characteristic has driven a shift in computer architecture away from relying solely on general-purpose Central Processing Units (CPUs) toward the use of Graphics Processing Units (GPUs) as accelerators for AI computation. CPUs are traditionally built around a small number of powerful cores designed for low-latency execution of sequential and control-heavy instructions, following a Multiple-Instruction-Multiple-Data (MIMD) style of execution. GPUs, in contrast, contain thousands of smaller cores organized to execute the same instruction across many data elements simultaneously, following a Single-Instruction-Multiple-Threads (SIMT) execution model. This architectural distinction is the foundation for the performance differences observed when the two platforms are applied to AI workloads.

Although GPUs have been widely adopted for AI acceleration, the actual performance advantage they provide is not uniform across all types of models, workloads, and configurations. Elmas and Bostanci [3] reported that a GPU was four to five times faster than a server-class CPU for recurrent neural network (RNN) classification, and Datamadness [2] observed that GPU training of a convolutional neural network (CNN) was more than six times faster than CPU-only training. In contrast, Aggarwal et al. [1] found that a CPU using an optimized math library outperformed baseline GPU configurations by 1.6 to 2.3 times on specific CNN training tasks, while Wang et al. [7] showed that CPUs achieve the highest floating-point utilization for recurrent models and can support the largest models because of their access to larger system memory. This inconsistency creates a practical problem for students, researchers, and practitioners who must decide which hardware architecture best fits a given AI application, especially under constraints of cost, power, and available infrastructure. This study is relevant to the field of Advanced Computer Organization and Architecture because it connects fundamental architectural concepts, such as instruction-level parallelism, memory hierarchy, and execution models, to real-world performance outcomes in a domain of major current importance. The results can guide students



International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

and practitioners in making informed hardware decisions, support instructors in illustrating architectural trade-offs with concrete examples, and provide a foundation for future primary research comparing emerging accelerator technologies.

II. OBJECTIVES OF THE STUDY

The general objective of this study is to compare the performance of CPUs and GPUs in executing AI workloads and to explain how their architectural differences translate into measurable performance outcomes. Specifically, this study aims to:

1. Compare the architectural characteristics of CPUs and GPUs that are relevant to AI computation.
2. Synthesize empirical performance data, such as training time, inference time, and throughput, reported in existing benchmarking studies.
3. Evaluate additional factors influencing hardware choice, including power consumption, cost-efficiency, and programmability.
4. Provide practical guidance on selecting appropriate hardware for different categories of AI applications.

In line with these objectives, the study seeks to answer the following research questions:

RQ1. How do the architectural differences between CPUs and GPUs affect their performance on AI workloads?

RQ2. What is the magnitude of the performance gap between CPUs and GPUs for representative deep learning tasks such as image classification and sequence modeling?

RQ3. Under what circumstances does the CPU remain competitive with, or superior to, the GPU?

RQ4. What trade-offs in power consumption and cost accompany the performance differences between the two architectures?

III. REVIEW OF RELATED LITERATURE

Hennessy and Patterson [5] provide the architectural foundation for this study, describing how processor design choices such as execution model, degree of parallelism, and memory hierarchy determine performance. Their quantitative approach explains why CPUs, which dedicate silicon to large caches and complex control logic for low-latency sequential execution, behave differently from GPUs, which dedicate silicon to many simple arithmetic units and high memory bandwidth for data-parallel throughput.

Elmas and Bostanci [3] compared an NVIDIA Tesla K80 GPU against an Intel Xeon Gold 6126 CPU on an RNN-based web page classification task. The GPU was four to five times faster than the CPU across all tests, demonstrating the advantage of massively parallel hardware for the matrix operations that dominate recurrent network training.

Datamadness [2] benchmarked TensorFlow 2 on an NVIDIA RTX 2080 GPU and an AMD Ryzen 2700X CPU using a CNN image classifier with approximately 58 million parameters. GPU training was more than six times faster than CPU-only training, consistent with the efficient mapping of convolution operations onto the GPU's SIMT architecture.

Aggarwal et al. [1] evaluated classical machine learning and deep learning algorithms on CPU and GPU platforms and found that a CPU configuration using an optimized math library (Intel MKL) outperformed baseline GPU configurations by 1.6 to 2.3 times for specific CNN training tasks in Caffe and Theano. Their findings show that software-level optimization can partially offset the CPU's architectural disadvantage in raw parallelism.

Wang et al. [7] benchmarked TPU, GPU, and CPU platforms using the ParaDnn suite of parameterized deep learning models. GPUs offered better flexibility for irregular computation, while CPUs achieved the highest floating-point operation utilization for RNNs and supported the largest models owing to their larger memory capacity.

Liu et al. [6] characterized the training of deep learning models on mobile devices using the NVIDIA Jetson TX2. During Inception V1 training, GPU utilization stayed near 100% while CPU core utilization ranged from 0 to 60%, and GPU utilization was sensitive to batch size, decreasing in efficiency as batches became smaller.

Gyawali [4] profiled CPU and GPU execution of deep learning models and reported that training a DenseNet121 model took roughly 13 hours on a CPU compared with about 2 hours on a GPU, while test accuracy remained similar on both



International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

platforms. The GPU's advantage was smaller for simpler networks, reinforcing that the speedup depends on model complexity.

IV. METHODOLOGY

A. Research Design: This study uses a descriptive-comparative research design implemented through a structured synthesis of secondary data. Rather than conducting new hardware experiments, the study systematically collects, organizes, and compares performance results, architectural specifications, and observations reported in peer-reviewed conference papers, archival preprints, and technical benchmarking reports. This approach is appropriate for an architecture-focused analysis because it allows performance outcomes to be interpreted directly in terms of underlying design principles, such as core count, execution model, and memory bandwidth.

B. Data Sources: Data were drawn from IEEE Xplore conference publications, arXiv preprints, and publicly available technical benchmarking reports published between 2018 and 2025. Sources were selected because they report controlled comparisons between CPU-only and GPU-accelerated execution of deep learning models, including convolutional neural networks (CNNs) and recurrent neural networks (RNNs), and because they document the specific hardware configurations used, such as processor model, GPU model, and memory specifications.

C. Data Collection Procedure: Candidate sources were identified using keyword searches combining terms such as "CPU," "GPU," "deep learning," "performance," and "benchmark." Sources were included if they reported quantitative performance metrics, such as training time, inference time, or hardware utilization, for at least one CPU configuration and one GPU configuration performing a comparable AI task. From each included source, the following information was extracted into a comparison table: hardware specifications, AI task and model type, performance metric reported, and the numerical or qualitative result of the comparison.

D. Data Analysis Method: The extracted data were organized using comparative tabulation to allow side-by-side interpretation of architectural specifications and performance outcomes. Where studies reported numerical training or inference times for both CPU and GPU configurations, relative speedup was calculated by dividing CPU execution time by GPU execution time. Qualitative observations regarding workload characteristics, such as batch size sensitivity and model type, were analyzed thematically to explain the conditions under which each architecture performs best.

V. RESULTS AND DISCUSSION

A. Architectural Comparison: Table 1 summarizes the core architectural differences between CPUs and GPUs that are most relevant to AI computation. The CPU's design, favoring fewer but more powerful cores with large caches, is well suited to tasks requiring complex branching and low-latency single-thread performance. The GPU's design, favoring a very large number of simpler cores and high memory bandwidth, is well suited to the large-scale, uniform, data-parallel operations that dominate deep learning workloads, particularly matrix multiplication and convolution.

Table 1. Architectural Comparison of CPU and GPU

Feature	CPU	GPU
Core count	Few, complex cores (typically 4–64)	Thousands of simpler cores (e.g., CUDA/stream cores)
Execution model	MIMD; optimized for low-latency, sequential/control-heavy execution	SIMT; optimized for high-throughput, data-parallel execution
Per-core clock speed	Higher (roughly 3–5 GHz)	Lower (roughly 1–2 GHz)
Cache hierarchy	Large multi-level cache (L1/L2/L3) to hide latency	Smaller cache per core; relies on massive memory bandwidth
Memory bandwidth	Lower (tens of GB/s)	Higher (hundreds of GB/s to several TB/s with HBM)
Best-suited workloads	Sequential logic, branching, irregular computation, small batch sizes	Large matrix/tensor operations, convolution, large-batch training



International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Note. MIMD = Multiple-Instruction-Multiple-Data; SIMT = Single-Instruction-Multiple-Threads; HBM = High Bandwidth Memory. Values are typical ranges compiled from architecture references and reviewed studies.

B. Performance Findings from the Literature: Table 2 summarizes representative quantitative and qualitative findings drawn from the reviewed studies. Across most benchmarks involving standard CNN and RNN training on non-optimized CPU configurations, GPUs demonstrated substantial speed advantages, generally in the range of four to six times faster than comparable CPU setups. This is consistent with the expectation that operations such as convolution and matrix multiplication map efficiently onto the GPU's parallel SIMT architecture, which can process thousands of data elements concurrently, whereas the CPU processes far fewer operations per cycle despite its higher per-core clock speed.

Table 2. Summary of Performance Findings from Reviewed Studies

Source	Hardware Compared	AI Task	Key Finding
Elmas & Bostanci [3]	Tesla K80 GPU vs. Intel Xeon Gold 6126 CPU	RNN-based web page classification	GPU was 4–5 times faster than CPU across all tests
Datamadness [2]	NVIDIA RTX 2080 GPU vs. AMD Ryzen 2700X CPU	CNN image classification (~58M parameters)	GPU training was more than 6 times faster than CPU-only training
Aggarwal et al. [1]	CPU with optimized math library (MKL) vs. baseline GPU configurations	CNN training (Caffe, Theano)	Optimized CPU outperformed GPU by 1.6–2.3 times in specific configurations
Wang et al. [7]	TPU, GPU, and CPU platforms (ParaDnn benchmark suite)	Multiple parameterized deep learning models	GPUs offer better flexibility for irregular computation; CPUs achieve the highest FLOPS utilization for RNNs and support the largest models due to memory capacity
Liu et al. [6]	NVIDIA Jetson TX2 (mobile CPU/GPU)	Inception V1 training	GPU utilization stayed near 100% while CPU core utilization ranged 0–60%; GPU utilization was sensitive to batch size

Note. CNN = Convolutional Neural Network; RNN = Recurrent Neural Network; MKL = Intel Math Kernel Library; FLOPS = floating-point operations per second; TPU = Tensor Processing Unit.

C. Influence of Workload Characteristics and Software Optimization: The results indicate that the performance gap between CPUs and GPUs is not fixed but depends heavily on workload characteristics and software optimization. When CPU execution was optimized using specialized math libraries, one reviewed study found that the CPU configuration outperformed baseline GPU configurations by a factor of 1.6 to 2.3 for certain CNN training tasks, illustrating that software-level optimization can partially offset the CPU's architectural disadvantage in raw parallelism. Similarly, benchmark suites that test a wide range of model types found that GPUs offer greater flexibility for irregular or small-batch computations, while CPUs achieve the highest floating-point operation utilization for recurrent architectures and can support larger models due to their access to larger system memory rather than the more limited memory available on many GPU accelerators.

D. Batch Size and Hardware Utilization: Observations from mobile and embedded deployments further show that GPU utilization during deep learning training tends to remain near maximum capacity, while CPU core utilization fluctuates and is generally lower, and that GPU utilization is notably sensitive to batch size, decreasing in efficiency as batch size shrinks. This finding is architecturally significant because it demonstrates that the GPU's throughput advantage depends on having a sufficiently large and uniform volume of parallel work; when batches are small or computation patterns are irregular, the overhead of coordinating thousands of GPU threads can erode the theoretical performance advantage, allowing the CPU's lower-latency, flexible execution model to remain competitive.



International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

E. Power Consumption and Cost Trade-offs: Beyond raw speed, the reviewed literature also points to trade-offs in power consumption and cost. GPUs typically draw substantially more power under sustained AI workloads than CPUs, which has implications for thermal management and operating cost in both data center and edge deployment contexts. These factors mean that the architecture offering the best raw training speed is not always the most practical choice once energy efficiency, hardware cost, and deployment environment are taken into account.

VI. CONCLUSION

This study compared CPU and GPU architectures in the context of AI applications by synthesizing architectural specifications and empirical performance findings from published research. The evidence shows that GPUs generally provide substantial speed advantages, typically four to six times faster than CPUs, for AI workloads dominated by large-scale, regular, parallel operations such as convolutional network training. However, CPUs remain a viable and, in some optimized or irregular-workload scenarios, superior option, particularly for models requiring large memory capacity, small-batch or irregular computation, or highly optimized software libraries.

These findings confirm that the underlying architectural principles taught in computer organization and architecture, including parallelism, execution model, and memory hierarchy, directly explain observed real-world AI performance outcomes. The overall contribution of this study is a structured, architecture-grounded comparison that can inform practical hardware selection decisions for AI applications rather than defaulting to GPU acceleration in every case.

VII. LIMITATIONS OF THE STUDY

1. The study relies on secondary data drawn from previously published experiments rather than controlled, primary benchmarking conducted by the researcher, which limits the ability to standardize hardware configurations across comparisons.
2. The reviewed studies used different hardware generations, software frameworks, and model architectures, which introduces variability that complicates direct numerical comparison across sources.
3. Hardware technology for AI acceleration evolves rapidly; findings based on studies from 2018–2025 may not fully reflect the performance characteristics of the newest CPU and GPU generations.
4. The study does not include an in-depth evaluation of specialized AI accelerators such as Tensor Processing Units (TPUs) or Neural Processing Units (NPUs), which are increasingly used alongside or in place of traditional CPUs and GPUs.

VIII. RECOMMENDATIONS

1. Future research should conduct primary, controlled benchmarking using standardized suites, such as MLPerf, to allow more direct and current comparisons between CPU and GPU performance on AI workloads.
2. Future studies should extend the comparison to include specialized accelerators such as TPUs and NPUs, as well as newer GPU architectures, to reflect the current AI hardware landscape.
3. Researchers should evaluate energy efficiency per unit of computation, such as performance per watt, in addition to raw speed, to better inform sustainable hardware deployment decisions.
4. Practitioners selecting hardware for AI applications should base their decision on workload characteristics: GPUs are recommended for large-batch, large-scale training of models dominated by matrix and convolution operations, while CPUs remain a reasonable choice for smaller-scale, memory-intensive, or irregular workloads, or where optimized CPU software libraries are available.
5. Future work could investigate heterogeneous CPU-GPU scheduling strategies that dynamically assign workload components to the architecture best suited for each computation stage.

IX. ACKNOWLEDGEMENTS

The author expresses sincere gratitude to the faculty and administration of the Graduate Studies program of Surigao del Sur State University for their guidance and institutional support throughout this research, particularly in the course MCS 202 (Advanced Computer Organization and Architecture). The author also thanks the researchers whose published benchmarking studies made this synthesis possible, and the colleagues and family members who provided



International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

encouragement. The author further acknowledges the use of an artificial intelligence writing assistance tool which supported the formatting, structuring, and editing of portions of this paper. All AI-assisted content was carefully reviewed, verified, and validated by the author to ensure factual accuracy, academic integrity, and full alignment with the objectives and findings of the study.

REFERENCES

- [1] Aggarwal, N., et al. (2025). CPU vs. GPU: Performance evaluation of classical machine and deep learning algorithms. In Proceedings of the IEEE Conference Publication. IEEE.
- [2] Datamadness. (2019). TensorFlow 2 – CPU vs GPU performance comparison. <https://datamadness.github.io/TensorFlow2-CPU-vs-GPU>
- [3] Elmas, O., & Bostanci, Y. (2019). Performance analysis and CPU vs GPU comparison for deep learning. In 2019 6th International Conference on Electrical and Electronics Engineering (ICEEE). Istanbul, Turkey: IEEE.
- [4] Gyawali, D. (2023). Comparative analysis of CPU and GPU profiling for deep learning models. arXiv preprint arXiv:2309.02521. <https://doi.org/10.48550/arXiv.2309.02521>
- [5] Hennessy, J. L., & Patterson, D. A. (2019). Computer architecture: A quantitative approach (6th ed.). Morgan Kaufmann.
- [6] Liu, J., Liu, J., Du, W., & Li, D. (2019). Performance analysis and characterization of training deep learning models on mobile devices. arXiv preprint arXiv:1906.04278. <https://doi.org/10.48550/arXiv.1906.04278>
- [7] Wang, Y. E., Wei, G.-Y., & Brooks, D. (2019). Benchmarking TPU, GPU, and CPU platforms for deep learning. arXiv preprint arXiv:1907.10701. <https://doi.org/10.48550/arXiv.1907.10701>
- [8] Wu, Y., Liu, L., Pu, C., Cao, W., Sahin, S., Wei, W., & Zhang, Q. (2018). A comparative measurement study of deep learning as a service framework. arXiv preprint arXiv:1810.12210. <https://doi.org/10.48550/arXiv.1810.12210>



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com