The American Journal of Interdisciplinary Innovations and Research
ISSN 2642-7478 Volume 08 - 2026

Improving Time Efficiency of Machine Learning Algorithms Through GPU Parallelization

Fayzullo Fozilov
Tashkent University of Information Technologies named after Muhammad al-Khwarizmi, Uzbekistan

Murodjon Abdusadikov
Tashkent University of Information Technologies named after Muhammad al-Khwarizmi, Uzbekistan

Khurshid Turaev
Tashkent University of Information Technologies named after Muhammad al-Khwarizmi, Uzbekistan

Nozima Atadjanova
Tashkent University of Information Technologies named after Muhammad al-Khwarizmi, Uzbekistan

Indira Tursinkulova
Tashkent University of Information Technologies named after Muhammad al-Khwarizmi, Uzbekistan

Received: 26 Mar 2026 | Received Revised Version: 20 Apr 2026 | Accepted: 03 May 2026 | Published: 31 May 2026

Volume 08 Issue 05 2026 | Crossref DOI: 10.37547/tajiir/Volume08Issue05-09

Abstract

This paper discusses the application of parallel computing technologies in artificial intelligence and machine learning processes. The study focuses on heterogeneous computing systems based on CPUs and GPUs, as well as the use of CUDA technology for parallel data processing. Experimental results show that GPU-based parallelization significantly improves computational speed and reduces execution time compared to traditional CPU-based processing. The research confirms the effectiveness of GPUs in accelerating machine learning algorithms and other computationally intensive tasks.

Keywords: Parallel processing algorithms, artificial intelligence, machine learning, heterogeneous computing systems, CUDA technology.

Cite This Article: Fayzullo Fozilov, Murodjon Abdusadikov, Khurshid Turaev, Nozima Atadjanova, & Indira Tursinkulova. (2026). Improving Time Efficiency of Machine Learning Algorithms Through GPU Parallelization. The American Journal of Interdisciplinary Innovations and Research, 8(05), 78–84. https://doi.org/10.37547/tajiir/Volume08Issue05-09

1. Introduction

It is well known that, as technology continues to advance, the processing speed of computers is increasing rapidly day by day. In the implementation of various problems and algorithms, time efficiency has become one of the most critical factors. Today, many fields require high-performance computing capabilities that can only be achieved through parallel computing technologies. This,

in turn, plays a significant role in improving time efficiency, especially in machine learning processes [1, 2]. Machine Learning (ML) is a branch of Artificial Intelligence (AI) that enables systems to automatically learn from experience and improve their performance through specific algorithms without being explicitly programmed. Machine learning is considered one of the scientific approaches aimed at developing computer systems capable of making decisions and predictions independently [3, 4]. Thanks to modern computational technologies, machine learning applications have become increasingly widespread and practical across different industries.

However, many machine learning algorithms rely on complex mathematical computations and require processing massive amounts of data automatically. These operations demand substantial computational resources and considerable execution time. In order to accelerate machine learning tasks and improve processing efficiency, both Central Processing Units (CPUs) and Graphics Processing Units (GPUs) are widely utilized. The combined use of these processors significantly enhances computational speed and performance, leading to the development of heterogeneous computing systems.

HETEROGENEOUS COMPUTING SYSTEMS

Heterogeneous computing systems refer to architectures that utilize multiple types of processors or cores working together to perform computational tasks efficiently. Such systems combine the strengths of CPUs and GPUs, where CPUs handle sequential operations while GPUs execute large-scale parallel computations simultaneously. This collaboration allows machine learning algorithms to process data faster, optimize resource utilization, and reduce overall training time (Figure 1).

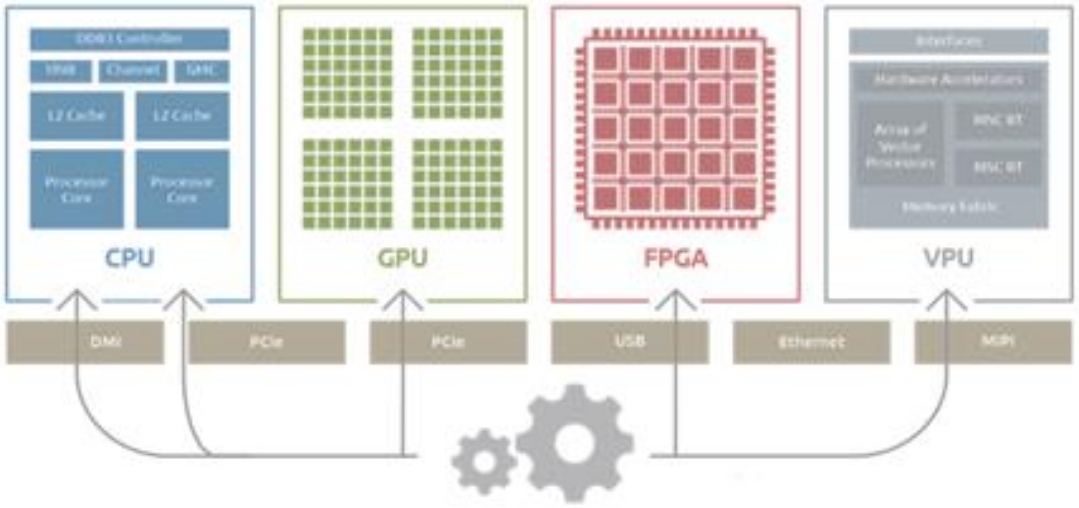

Figure 1. Heterogeneous computing systems

These systems improve performance and energy efficiency not only by increasing the number of identical processors, but also by integrating different types of coprocessors within a single architecture. In other words, heterogeneous computing systems enable tasks to be solved in a parallel manner, allowing multiple computational operations to be executed simultaneously [5, 6].

Parallel processing refers to the distribution of algorithms and computational tasks across multiple processors, where several operations are carried out at the same time within a computer system. Instead of executing instructions sequentially on a single processor, parallel computing divides workloads into smaller parts that can run concurrently on different processing units. This approach significantly increases computational speed, optimizes resource utilization, and reduces the total execution time required for complex tasks (Figure 2).

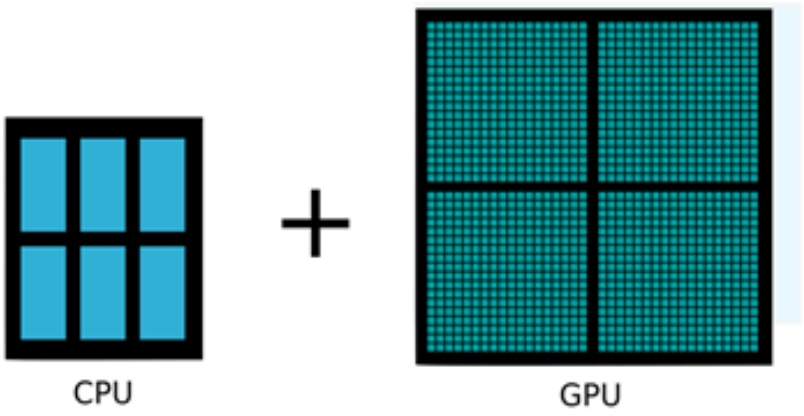

Figure 2. Using CPU and GPU processors in parallelization

GPU-Based Parallel Processing

Typically, this approach refers to distributed processing, where a conventional machine learning algorithm performs an enormous number of computations on very large datasets. Machine learning tasks such as training neural networks, data analysis, image recognition, and predictive modeling often require extensive mathematical operations that can become extremely time-consuming when executed sequentially [7].

If we examine Figure 2, it can be observed that a computer's Graphics Processing Unit (GPU) contains significantly more processing cores compared to a Central Processing Unit (CPU). While a CPU usually consists of a limited number of highly optimized cores designed for sequential processing, a GPU is built with hundreds or even thousands of smaller cores capable of executing many operations simultaneously. As a result, parallel processing on GPUs provides substantially better and more efficient performance than processing tasks solely on CPUs.

A GPU (Graphics Processing Unit) is a specialized processor similar to the computer's central processor, but it is specifically designed for handling graphical computations and parallel operations. GPUs were originally developed to accelerate graphics rendering for applications such as video games, visual simulations, and multimedia processing. Today, however, GPUs are widely used in scientific computing, artificial intelligence, and machine learning due to their exceptional capability to process large amounts of data in parallel. By utilizing GPUs, it becomes possible to efficiently manage and accelerate complex graphical applications, large-scale software systems, video processing, image rendering, and computationally intensive machine learning algorithms. Their architecture allows high-speed execution of repetitive mathematical calculations, making GPUs one of the most important hardware components in modern AI and high-performance computing systems [8].

As mentioned above, parallel computing refers to the process of executing multiple processors or program operations simultaneously. In general, it is a type of computing architecture in which large and complex problems are divided into smaller, independent, and often similar tasks that can be processed concurrently. These tasks are distributed across multiple processors connected through shared memory, and after execution, the results are combined to produce the final output. This approach is highly effective for solving computationally intensive problems because it distributes workloads among several processors, significantly reducing execution time and increasing overall system efficiency [9].

When implementing parallelization on Graphics Processing Units (GPUs), a certain portion of the task is usually assigned to the computer's Central Processing Unit (CPU), while the remaining computationally intensive operations are executed in parallel using the GPU [10]. In such architectures, the CPU typically manages control logic, task scheduling, and sequential operations, whereas the GPU performs repetitive mathematical calculations and large-scale parallel processing tasks. This collaboration between CPU and GPU creates an efficient heterogeneous computing environment capable of handling massive data-processing workloads.

CUDA Technology

To effectively utilize GPU-based parallel computing, it

becomes necessary to employ technologies such as CUDA (Compute Unified Device Architecture). CUDA is a parallel computing platform and programming model developed by NVIDIA that enables developers to use GPUs for general-purpose computing tasks beyond traditional graphics rendering [11, 12]. Through CUDA technology, large amounts of data can be processed simultaneously on GPU cores, greatly accelerating machine learning algorithms, scientific simulations, image processing, and other high-performance computing applications (Figure 3).

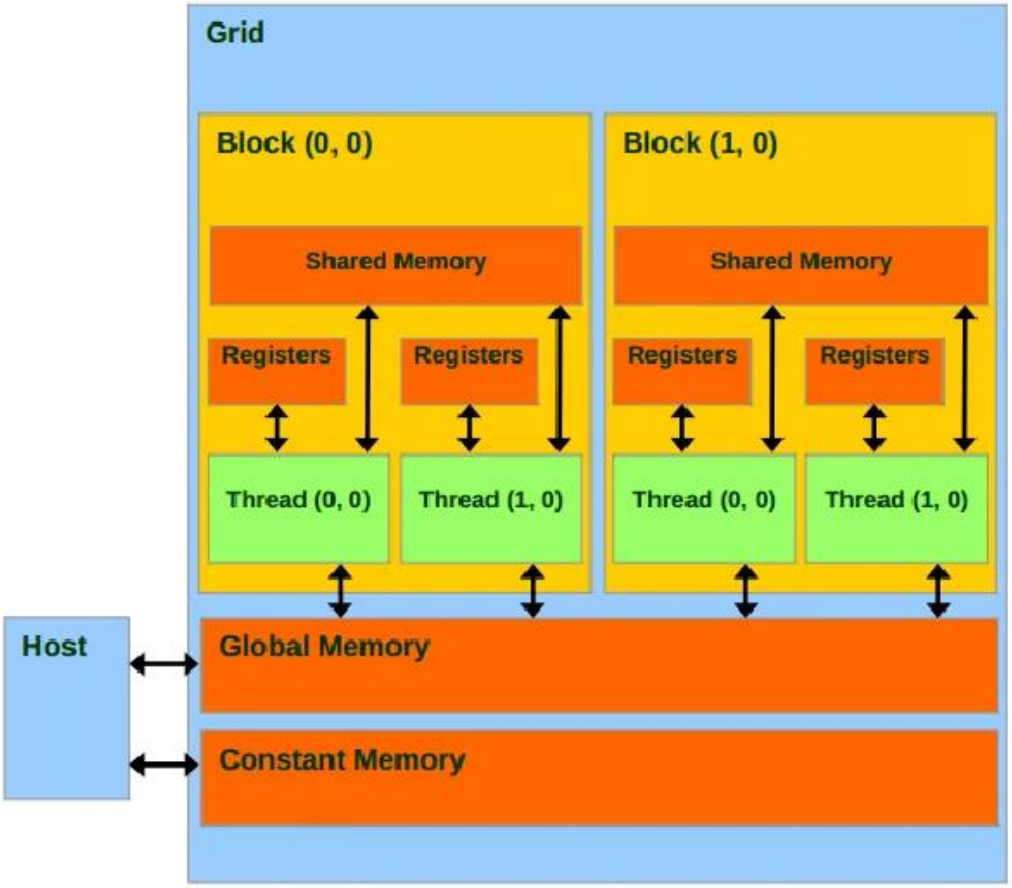

Figure 3. CUDA technology architecture

In Figure 3, the component labeled A-host represents the Central Processing Unit (CPU). The CPU operates based on the principle of Single Program, Single Data (SPSD), where a single sequence of instructions processes one set of data at a time. CPUs are designed for general-purpose computing and are highly efficient in handling sequential operations, logical decision-making, and task management within a computer system [13, 14].

On the other hand, the component labeled B-host represents the Graphics Processing Unit (GPU). Unlike the CPU, the GPU follows the concept of Single Program, Multiple Data (SPMD) or similar parallel processing models, where a single program can process multiple data elements simultaneously. This architecture enables GPUs to execute thousands of parallel operations concurrently, making them highly suitable for large-scale computations and data-intensive applications.

2. Results

The key difference between CPUs and GPUs lies in their processing structure and operational focus. While CPUs prioritize flexibility and sequential task execution, GPUs are optimized for high-throughput parallel processing. As a result, GPUs are particularly effective in machine learning, deep learning, image processing, scientific simulations, and other applications that require simultaneous processing of massive datasets. This capability allows GPU-based systems to achieve significantly higher computational performance and efficiency compared to traditional CPU-only systems (Figure 4).

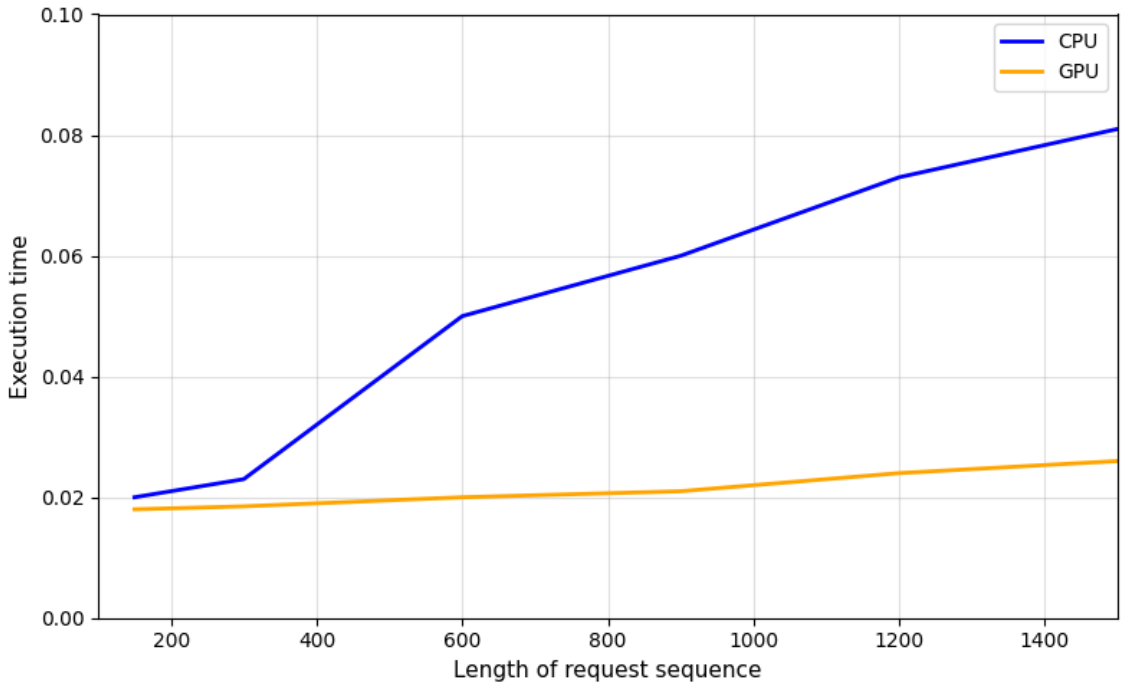

Figure 4. Performance Evaluation of CPU and GPU Architectures

Therefore, in the parallelization of machine learning processes, which represent one of the fundamental branches of artificial intelligence, performing computations primarily on Graphics Processing Units (GPUs) provides significant improvements in both processing speed and overall performance. Since machine learning algorithms involve large-scale mathematical computations and extensive data processing, GPU-based parallel computing enables these operations to be executed much faster and more efficiently compared to traditional CPU-based approaches. The use of GPUs in machine learning not only reduces computation time but also increases the scalability and effectiveness of AI systems. By leveraging parallel processing capabilities, GPUs can simultaneously handle thousands of computational tasks, making them especially suitable for deep learning, neural network training, image recognition, and other data-intensive applications. As a result, researchers and developers are able to achieve more accurate results, accelerate model training processes, and optimize resource utilization.

The analytical experiments and comparative evaluations presented in this study were conducted using a device with the technical characteristics described in the following table (Table 1).

Table 1. Processor Specifications

Processor	Model	RAM	Cores	CUDA
CPU	Intel Core i5-4210U 2.40 GHz	12 GB	2/2	—
GPU	Nvidia GeForce 820M	12 GB	96+40	8.0

Based on the research results presented above (Figure 4), it can be concluded that not only machine learning processes, but also many other complex computational tasks achieve significantly higher performance when parallelized on a computer's Graphics Processing Unit (GPU) rather than executed solely on the Central Processing Unit (CPU). The experimental results demonstrate that GPU-based parallel processing can improve time efficiency not just slightly, but by several times or even dozens of times compared to traditional

CPU-based computation.

This considerable performance improvement is mainly due to the GPU's ability to execute thousands of parallel operations simultaneously. While CPUs are optimized for sequential processing and task control, GPUs are specifically designed to handle large-scale repetitive computations in parallel, making them highly effective for data-intensive and computationally complex applications.

As a result, when tasks traditionally performed on CPUs are transferred to GPUs using heterogeneous computing systems, the overall computational efficiency increases dramatically. Such an approach enables faster execution of machine learning algorithms, reduced processing time, improved resource utilization, and enhanced system productivity. Therefore, the integration of GPU-based parallel computing technologies has become one of the key solutions for accelerating modern artificial intelligence and high-performance computing applications.

3. Conclusion

The research results demonstrate that applying parallel processing technologies in machine learning and artificial intelligence systems significantly increases computational efficiency and time performance. Comparative analyses showed that GPU-based processing using CUDA technology performs considerably faster than traditional CPU-based computation due to the large number of parallel processing cores available in GPUs. The obtained results confirm that heterogeneous computing systems effectively optimize resource utilization, reduce execution time, and improve the performance of complex computational tasks. Therefore, the use of GPUs and CUDA technology can be considered one of the most efficient approaches for accelerating modern machine learning and high-performance computing applications.

References

1. Badman, A., & Kosinski, M. (2025, December 24). Big data. What is big data? https://www.ibm.com/think/topics/big-data
2. Rahul, K., Banyal, R.K. & Arora, N. A systematic review on big data applications and scope for industrial processing and healthcare sectors. J Big Data 10, 133 (2023). https://doi.org/10.1186/s40537-023-00808-2
3. Perez-Meana, H., & Nakano-Miyatake, M. (2025). Digital Image Processing: Technologies and Applications. Applied Sciences, 15(23), 12709. https://doi.org/10.3390/app152312709
4. Y. Cheng and B. Li, "Image Segmentation Technology and Its Application in Digital Image Processing," 2021 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC), Dalian, China, 2021, pp. 1174-1177, doi: 10.1109/IPEC51340.2021.9421206.
5. Schneider, J., & Smalley, I. (2025, November 17). CPU vs. GPU Machine Learning. CPU vs. GPU for machine learning. https://www.ibm.com/think/topics/cpu-vs-gpu-machine-learning
6. SoftwareG, "How to use GPU to help CPU," [Online]. Available: https://softwareg.com.au/blogs/computer-hardware/how-to-use-gpu-to-help-cpu.
7. Nan Zhang, Yun-shan Chen and Jian-li Wang, "Image parallel processing based on GPU," 2010 2nd International Conference on Advanced Computer Control, Shenyang, China, 2010, pp. 367-370, doi: 10.1109/ICACC.2010.5486836.
8. Vasile, C.-E., Ulmămei, A.-A., & Bîră, C. (2024). Image Processing Hardware Acceleration—A Review of Operations Involved and Current Hardware Approaches. Journal of Imaging, 10(12), 298. https://doi.org/10.3390/jimaging10120298
9. CUDA Tutorial. Learn CUDA simply easy learning: https://www.tutorialspoint.com/cuda/index.htm (2016)
10. NVIDIA. Parallel programming and computing plaform | nvidia cuda. http://www.nvidia.com/object/cuda, June (2013).
11. Flinders, M., Susnjara, S., & Smalley, I. (2025, November 17). GPU. What is a graphics processing unit (GPU)? https://www.ibm.com/think/topics/gpu
12. NVIDIA, Preface - CUDA C++ Best Practices Guide 12.9 documentation". NVIDIA Corporation, May 31, 2025. [Online]. Available: https://docs.nvidia.com/cuda/cuda-c-best-practices-guide/
13. M. Harris and M. Harris, "How to access global memory efficiently in CUDA C/C++ kernels," NVIDIA Technical Blog, Oct. 16, 2025. [Online]. Available: https://developer.nvidia.com/blog/how-access-global-memory-efficiently-cuda-c-kernels/
14. M. Harris and M. Harris, "Using shared memory in CUDA C/C++," NVIDIA Technical Blog, Aug. 05,

2025. [Online]. Available: https://developer.nvidia.com/blog/using-shared-memory-cuda-cc/