Computer Architecture and Signal Processing

E-ISSN: 3124-0003

Research Article

Hardware-Software Co-Design of Deep Learning-Accelerated Digital Signal Processing Cores for Low-Latency Multimedia Applications

Taufiq Dwi Cahyono [1*], Abdul Muchlis [2], Sandy Suryady [3]

[1] Universitas Semararang e-mail : email: taufiq_dc@usm.ac.id
[2] Universitas Gunadarma e-mail : Muchlis07@staff.gunadarma.ac.id
[3] Universitas Gunadarma e-mail : sandy22@staff.gunadarma.ac.id
* Corresponding Author : Taufiq Dwi Cahyono

Abstract: The increasing demand for low-latency and high-throughput multimedia applications has spurred significant advancements in hardware software co-design. This study explores the integration of custom digital signal processing (DSP) hardware accelerators with optimized software frameworks to enhance deep learning-accelerated DSP tasks. The proposed co-design approach significantly reduces latency and improves throughput compared to traditional software-only DSP implementations. Through the development of custom hardware accelerators built with FPGA technology, the system achieves up to a 1.85x reduction in latency and a 1.5x improvement in throughput for real time multimedia tasks such as image recognition, video decoding, and audio processing. The combination of hardware and software optimizations allows for better resource utilization, enabling the parallel processing of computationally intensive tasks while the software framework handles less demanding operations. Additionally, the co-design system demonstrated improved energy efficiency, making it highly suitable for embedded systems. The results show that the hardware software co-design approach offers substantial advantages in performance, latency reduction, and energy efficiency, positioning it as a viable solution for real time multimedia applications. The findings have important implications for applications requiring fast data processing, such as autonomous driving, healthcare, and disaster management. Future research could explore alternative hardware accelerators, advanced software optimizations, and AI based resource management to further improve the system's efficiency and scalability for more complex multimedia tasks.

Keywords: Hardware Software Co-Design; Deep Learning; Multimedia Applications; DSP Systems; Latency Reduction.

Received: 21, November 2025
Revised: 10, December 2025
Accepted: 29, December 2025
Published: 15, January 2026
Curr. Ver.: 20, January 2026

1. Introduction

The rapid growth of multimedia and vision-based applications has been strongly influenced by the increasing adoption of deep learning technologies across various digital environments. Deep learning models such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and generative adversarial networks (GANs) have significantly improved the efficiency and accuracy of multimedia processing tasks, including image recognition, object detection, semantic segmentation, and image synthesis [1], [2]. These developments also extend to video analysis, where deep learning architectures enable advanced capabilities such as action recognition, video captioning, and automated video generation for extracting meaningful insights from complex visual data [1]. In the audio domain, deep learning has enhanced applications such as speech recognition, music classification, and sound event detection, thereby expanding the functional capabilities of multimedia systems [3]. Moreover, the integration of natural language processing with deep learning techniques enables systems to interpret textual information within multimedia

contexts, improving the ability of intelligent systems to understand and process multimodal data in real time [4]. The convergence of these technologies demonstrates that deep learning is becoming a critical foundation for modern multimedia intelligence systems that support complex digital ecosystems.

The growing number of connected sensors, mobile devices, and distributed computing infrastructures continuously generates massive volumes of multimedia data, making real time processing a critical requirement for modern digital systems. Deep learning frameworks are increasingly utilized to address this challenge because of their ability to process large-scale datasets while maintaining high levels of accuracy and adaptability in dynamic environments [5]. This rapid technological development creates an urgent need to understand how intelligent multimedia processing systems can support emerging applications such as autonomous transportation, healthcare monitoring, and disaster response systems, where real time decision making is essential [4]. In addition, the integration of artificial intelligence, distributed computing, and secure digital infrastructures has become an important research focus in the development of adaptive digital ecosystems [6], [7]. Studies on intelligent network security architectures and hybrid deep learning models demonstrate how advanced AI frameworks can support real time analysis, system resilience, and automated threat mitigation in large-scale data environments [6], [7]. Therefore, investigating the evolution of deep learning-based multimedia processing systems is crucial to address the rapidly expanding technological demands and emerging societal challenges in data-driven environments.

Traditional software-based digital signal processing (DSP) systems have long been utilized in multimedia processing; however, they increasingly face limitations when applied to modern real time multimedia environments. One of the major challenges arises from latency introduced by analog to digital (A/D) and digital to analog (D/A) conversions, which can significantly delay signal processing and hinder real time system responsiveness [8]. In addition, relying solely on centralized processing architectures, particularly single CPU-based processing, often results in inefficiencies when handling computationally intensive multimedia tasks [2]. This limitation becomes more evident as modern multimedia applications generate higher data rates and require faster processing speeds to support tasks such as real time video analytics and intelligent multimedia interaction [3]. Edge computing has therefore emerged as a promising paradigm by relocating computation closer to the data source, thereby reducing transmission latency and improving processing efficiency [4]. Nevertheless, existing literature largely focuses on improving processing architectures or distributed frameworks without sufficiently addressing the integration of intelligent deep learning acceleration mechanisms within edge-based multimedia processing environments, indicating a critical gap in current research.

Recent developments in multimedia technologies further emphasize the need for real time processing capabilities supported by advanced artificial intelligence frameworks. Edge computing platforms enable the deployment of deep learning models directly on edge devices, reducing the dependency on centralized cloud infrastructure and minimizing communication delays in multimedia applications [5]. Parallelizable algorithms and specialized hardware architectures have also been explored to support computer vision and multimedia signal processing workloads more efficiently [2]. Several studies have proposed hardware software co-design approaches to accelerate deep learning-based DSP tasks by combining programmable hardware architectures with optimized algorithms [9], [10]. However, current literature still lacks comprehensive investigations into how intelligent deep learning architectures, distributed edge computing infrastructures, and resilient digital system architectures can be integrated to support scalable and secure real time multimedia processing systems. Recent research highlights the importance of adaptive system architectures, resilient software environments, and intelligent AI driven frameworks in supporting large-scale digital ecosystems [11], [12]. Therefore, further research is necessary to explore integrated architectures that combine deep learning acceleration, edge computing, and hardware software co-design to overcome the limitations of conventional DSP systems.

Hardware software co-design has emerged as an effective approach for improving the performance of digital signal processing (DSP) systems, particularly in multimedia environments that require high computational efficiency and low latency. This methodology combines the flexibility of software programmability with the speed and parallel processing capabilities of specialized hardware accelerators such as field-programmable gate arrays (FPGAs) and heterogeneous processors. By partitioning computational workloads between hardware and software components, co-design architectures can significantly reduce

computational bottlenecks and improve the runtime efficiency of deep learning-based DSP systems [10]. In addition, co-design enables more effective mapping of deep learning models to hardware architectures, ensuring that processing resources are optimally utilized while maintaining low latency and high throughput [13]. Previous studies have demonstrated that optimized architectures generated through neural architecture search and hardware-aware design strategies can substantially improve system performance in embedded and high-performance computing environments [14]. However, despite these advances, fundamental questions remain regarding how hardware software co-design frameworks can be systematically integrated with intelligent multimedia processing pipelines to support scalable, real time DSP systems in increasingly complex digital environments.

Another important advantage of hardware software co-design lies in its ability to support parallel processing and efficient data management within complex computing architectures. Compiler architecture co-design mechanisms have demonstrated the capability to optimize parallelism and data reuse in reconfigurable array processors, leading to higher hardware utilization and reduced latency in signal processing workloads [10]. Furthermore, memory optimization techniques, such as the RAINBOW framework, enable the generation of heterogeneous execution plans that minimize off-chip memory access and improve data transfer efficiency within deep learning accelerators [9]. Despite these advancements, the existing literature still lacks comprehensive investigations into how integrated hardware software co-design frameworks can simultaneously address real time multimedia processing demands, scalable system architectures, and adaptive intelligent algorithms. Recent research highlights the importance of resilient system architectures and intelligent digital infrastructures in supporting high-performance computing systems operating in distributed environments [7], [12]. Therefore, this article aims to address the fundamental research question of how hardware software co-design architectures can be optimized to support efficient, scalable, and intelligent DSP systems for next-generation multimedia applications.

The primary objective of this study is to investigate how hardware software co-design can enhance digital signal processing (DSP) performance while minimizing processing latency in multimedia applications. Previous studies have demonstrated that co-design approaches enable more efficient workload partitioning between hardware accelerators and software-controlled processing units, thereby improving system throughput and reducing computational bottlenecks [10]. However, many existing implementations primarily focus on architectural optimization without thoroughly addressing the integration of intelligent processing frameworks for real time multimedia workloads. This research therefore explores co-design methodologies that optimize task partitioning, resource allocation, and runtime performance for deep learning accelerated DSP applications. By examining real time multimedia tasks such as video decoding and image filtering, this study aims to provide a deeper understanding of how co-design architectures can overcome latency constraints and computational inefficiencies that frequently arise in traditional DSP systems. In addition, recent research highlights the importance of adaptive and resilient system architectures in modern digital environments, suggesting that integrating intelligent processing mechanisms within hardware software frameworks can significantly improve system scalability and operational efficiency [12].

Another key contribution of this study lies in examining how dynamic hardware software co-design frameworks can improve both energy efficiency and computational precision in multimedia processing systems. Existing research has shown that advanced architectures such as Processing in Memory (PiM) and other hardware-assisted processing models can significantly enhance throughput for computationally intensive tasks such as image filtering [13]. Similarly, co-design frameworks like SWOOP demonstrate how shifting part of the processing complexity into software can effectively hide memory latency while increasing instruction-level and memory-level parallelism, thereby improving real time processing performance [15]. Despite these advancements, the literature still lacks comprehensive studies that investigate how energy-efficient co-design frameworks can be systematically integrated with intelligent digital infrastructures to support scalable multimedia processing systems. Recent developments in intelligent computing systems and AI based architectures suggest that combining adaptive learning models with optimized system architectures can significantly enhance real time data processing capabilities in complex digital ecosystems [6], [11]. Therefore, this study contributes by proposing a more integrated perspective on hardware software co-design that emphasizes performance optimization, energy efficiency, and intelligent system adaptability for next-generation multimedia applications.

2. Literature Review

Deep Learning in Multimedia and DSP

Deep learning has become a fundamental paradigm in multimedia processing and digital signal processing (DSP), enabling advanced analytical capabilities for processing complex multimedia data. In multimedia systems, deep learning techniques such as convolutional neural networks (CNNs) are widely applied to perform tasks including object detection, semantic segmentation, and image synthesis, significantly improving the accuracy and efficiency of multimedia analysis [16]. Similarly, video analytics has benefited from deep learning architectures that support action recognition, video captioning, and scene understanding, enabling systems to extract semantic information from large-scale video datasets [17]. In addition, recurrent neural networks (RNNs) and generative adversarial networks (GANs) have enhanced audio signal processing applications, including speech recognition, music classification, and sound event detection [18]. These advances demonstrate that deep learning models are capable of learning complex nonlinear patterns in multimedia signals, making them suitable for a wide range of digital signal processing tasks. Furthermore, the integration of intelligent learning architectures within modern digital systems has also been emphasized in recent studies, which highlight the role of adaptive AI driven frameworks in improving the efficiency and scalability of complex data processing environments [19].

In the context of digital signal processing implementations, deep learning models have increasingly been adopted to enhance the performance of traditional DSP systems. Deep learning architectures are widely applied in music signal processing tasks such as music information retrieval, music recommendation systems, and automated music generation, demonstrating strong potential for commercial multimedia applications [18]. In communication signal processing, deep learning techniques are also used to improve tasks such as symbol detection, channel estimation, interference mitigation, and communication signal classification [20]. These capabilities indicate that deep learning can serve as a critical variable influencing the efficiency and adaptability of DSP systems in complex digital environments. Moreover, machine learning driven signal processing frameworks have demonstrated the ability to introduce nonlinear modeling capabilities that significantly enhance DSP system performance compared with traditional analytical models [21]. However, the black-box characteristics of many deep learning models still pose challenges for interpretability and optimization in practical DSP implementations. Recent studies emphasize the need for integrating adaptive learning frameworks and resilient computing architectures to support scalable and intelligent signal processing systems in distributed digital infrastructures [22], [23].

Software-Only DSP Approaches

Software-only digital signal processing (DSP) approaches have been widely used in multimedia systems due to their flexibility and ease of implementation. However, despite the rapid development of deep learning and advanced signal processing algorithms, software-based DSP systems continue to face several critical limitations, particularly in real time multimedia environments. One of the most prominent challenges is latency, which arises from the analog to digital (A/D) and digital to analog (D/A) conversion processes that introduce delays in signal processing pipelines [24]. Real time DSP applications, such as video streaming, speech recognition, and communication signal analysis, require processing capabilities that operate at the same speed as the application sampling rate. Software-only implementations often struggle to meet these requirements because they rely heavily on general-purpose processors that lack specialized signal processing acceleration capabilities [20]. In addition, modern multimedia systems generate large volumes of high-frequency data that demand efficient computational architectures capable of handling complex signal transformations. Research on machine learning-based DSP models indicates that while software-based processing frameworks can enhance analytical capabilities, they still face scalability and efficiency constraints when dealing with high-performance signal processing workloads [21].

Another major limitation of software-only DSP approaches lies in their inability to efficiently support large-scale parallel processing requirements in modern multimedia systems. Real time multimedia applications rely heavily on parallel processing techniques to handle large data streams and complex signal transformations simultaneously. However,

software-based implementations often struggle to fully exploit parallelism due to architectural constraints in general-purpose processors, which results in increased processing delays and reduced system efficiency [24]. These limitations highlight the importance of incorporating specialized hardware components such as digital signal processors (DSPs), graphics processing units (GPUs), and field-programmable gate arrays (FPGAs) to accelerate signal processing tasks and improve system throughput [20]. In this context, the ability of a DSP system to support efficient parallel processing, reduce latency, and optimize computational resource utilization becomes an important variable influencing system performance. Recent studies on intelligent digital infrastructures and adaptive computing architectures also emphasize the importance of integrating optimized computing environments with advanced processing frameworks to support scalable and secure digital systems [23], [25]. These developments suggest that future DSP architectures must combine efficient computation models with adaptive system infrastructures to support high-performance multimedia processing environments.

Hardware Software Co-Design Concepts

Hardware software co-design (HSCD) is an integrated development approach that simultaneously optimizes hardware and software components to achieve higher system efficiency, lower latency, and improved computational performance. This methodology has become increasingly important in domains such as embedded systems, machine learning platforms, and multimedia signal processing environments where processing speed and real time responsiveness are essential requirements [26]. By enabling tight coupling between hardware resources and software algorithms, co-design allows domain-specific optimizations that significantly improve system performance while reducing design complexity [27]. Modern co-design frameworks also leverage advanced development tools that accelerate the system development process. For instance, the Tango framework enables just-in-time register transfer level (RTL) simulation, achieving substantial performance improvements compared to traditional simulation approaches [28]. These capabilities highlight the importance of integrated design methodologies in supporting complex computational workloads in multimedia systems. In addition, recent research on intelligent digital infrastructures and adaptive computing frameworks emphasizes the importance of combining optimized hardware resources with intelligent system architectures to enhance the reliability and scalability of modern digital processing systems [29], [30].

Within hardware software co-design architectures, several critical variables influence system performance, including task partitioning strategies, hardware resource allocation, memory access efficiency, and system-level latency optimization. Effective partitioning between hardware and software components enables computationally intensive tasks to be executed on specialized hardware accelerators while maintaining flexible control through software layers [27]. In particular, co-design methodologies such as the Hardware Software Agile Co-design (HASCO) framework demonstrate how optimized resource partitioning can significantly reduce computational latency in tensor processing workloads, achieving performance improvements of up to 1.25×-1.44× [31]. Hardware acceleration through FPGA implementations also provides substantial efficiency improvements in image processing applications, such as Sobel filter operations, where FPGA-based implementations have been shown to outperform conventional software-based solutions in both latency and resource utilization [32]. These variables highlight the importance of architectural optimization and resource-aware system design in co-design environments. Furthermore, recent research on intelligent distributed computing frameworks emphasizes that adaptive system architectures and optimized hardware infrastructures are essential variables in enabling scalable and resilient computing systems for modern multimedia and real time data processing environments [33].

Existing Solutions for Low-Latency Multimedia Processing

Low-latency multimedia processing has become a critical requirement in modern computing environments where applications such as video analytics, real time streaming, and interactive multimedia services demand rapid processing capabilities. Various computing architectures have been explored to address latency challenges, including central processing units (CPUs), graphics processing units (GPUs), and specialized hardware accelerators. CPUs provide flexible processing environments but often struggle with high-latency multimedia workloads due to limited parallel processing capabilities, even when optimization techniques

such as single instruction multiple data (SIMD) and vectorization are applied. In contrast, GPUs are specifically designed for massive parallel processing and have demonstrated significant performance improvements in compute-intensive workloads, achieving substantial speedups compared to traditional CPU-based implementations [34]. Nevertheless, GPU-based systems often face challenges related to memory latency and data transfer bottlenecks, which can limit their effectiveness in real time multimedia environments. Recent research therefore emphasizes the importance of integrating heterogeneous computing architectures and specialized accelerators to achieve improved performance and efficiency in multimedia signal processing systems [31].

Specialized accelerators such as field-programmable gate arrays (FPGAs) have emerged as highly effective solutions for low-latency multimedia processing tasks due to their ability to implement custom hardware pipelines and optimized parallel architectures. FPGA-based solutions have demonstrated substantial improvements in latency-sensitive image processing tasks, such as background subtraction and real time object detection, where optimized hardware implementations significantly outperform conventional software approaches [32]. In addition, advanced application processors integrating multiple computing units, including CPUs, GPUs, and dedicated accelerators, provide flexible platforms capable of balancing performance and energy efficiency in multimedia workloads [35]. Hybrid architectures that distribute computational workloads across heterogeneous processing units have also shown promising results in improving both system performance and energy efficiency [31]. In modern digital infrastructures, these architectural variables including parallel processing capability, memory latency optimization, hardware acceleration efficiency, and system-level energy management play a crucial role in determining the effectiveness of multimedia processing systems. Recent research also highlights the importance of resilient and adaptive computing architectures in supporting scalable and secure digital processing systems in increasingly complex data environments [30], [36].

3. Proposed Method

The research utilizes a hardware software co-design approach to develop custom DSP hardware accelerators integrated with optimized software frameworks, aiming to enhance performance and reduce latency in deep learning-accelerated DSP tasks. The system design incorporates custom DSP hardware, built using FPGAs for low-latency processing, and an optimized software framework that maps deep learning models onto the hardware. Evaluation involves testing multimedia processing tasks like video decoding and image filtering, with performance measured in terms of latency reduction and throughput improvement. Data is collected by running multiple iterations of each workload, and the results are analyzed to assess the impact of co-design on system performance, comparing it to traditional software-only DSP and hardware accelerators like GPUs and FPGAs.

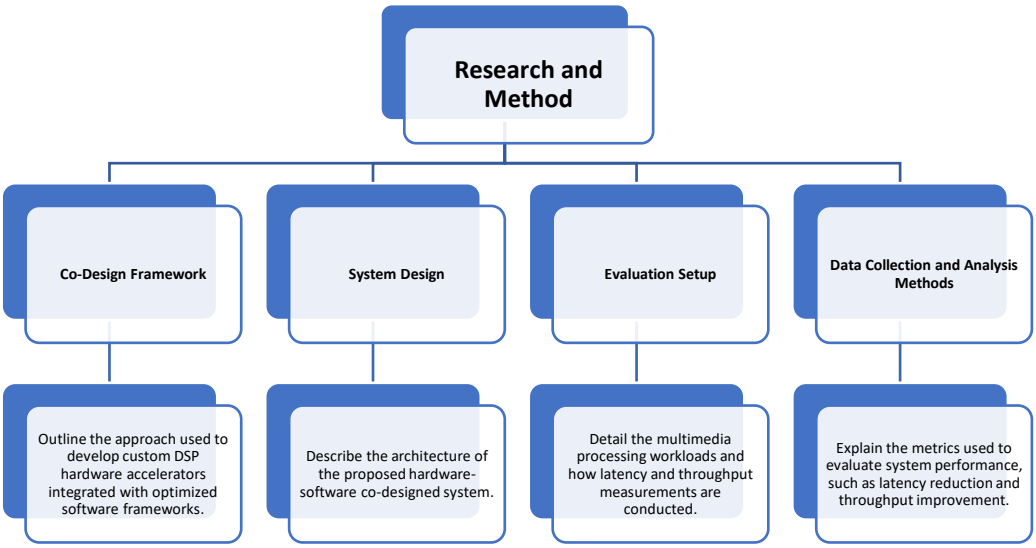

Figure 1. Flowchart structure.

Co-Design Framework

The research methodology follows a hardware software co-design approach to develop custom digital signal processing (DSP) hardware accelerators integrated with optimized software frameworks. The co-design paradigm aims to integrate both hardware and software components to enhance the overall performance of deep learning-accelerated DSP systems while minimizing latency. By optimizing the design of both components simultaneously, this methodology allows for better system-level performance and efficiency. The framework utilized in this study builds custom DSP hardware accelerators to handle computationally intensive tasks, while the optimized software frameworks are designed to effectively map deep learning models onto the hardware, ensuring that both hardware and software are used optimally.

System Design

The proposed system architecture follows a co-design methodology that incorporates both hardware and software elements to optimize performance. The hardware aspect involves the development of custom DSP accelerators, which are designed to process deep learning tasks such as image recognition, object detection, and audio processing. These accelerators are built using Field-Programmable Gate Arrays (FPGAs) to ensure ultra-low-latency processing, which is crucial for real time multimedia applications. On the software side, an optimized framework is implemented to efficiently map deep learning models onto the hardware. The system uses Just-in-Time RTL simulation tools, which enable fast prototyping and validation of the design before physical implementation, speeding up the development process significantly. The co-design system is structured to allow for flexible configuration of both hardware and software, enabling dynamic adjustments to meet the specific requirements of the multimedia processing tasks.

Evaluation Setup

To evaluate the effectiveness of the hardware software co-designed system, multimedia processing workloads are selected based on typical tasks in real time applications, such as video decoding, image filtering, and audio recognition. The workloads are chosen to reflect the computational complexity and real time requirements typical of multimedia applications. The evaluation setup includes measuring key performance metrics, primarily focusing on latency and throughput. Latency is measured as the time taken for data to be processed from input to output, while throughput refers to the system's ability to process a given amount of data in a specified time. The system's latency and throughput are compared against traditional software-only DSP implementations and hardware accelerators such as GPUs and FPGAs.

Data Collection and Analysis Methods

The performance of the system is evaluated using two primary metrics: latency reduction and throughput improvement. Latency reduction is measured by comparing the time taken to process multimedia data using the co-designed system versus traditional software-based DSP systems. Throughput improvement is assessed by analyzing the volume of data processed per unit of time, comparing the co-designed system's performance to that of GPUs and other DSP hardware. The data collection process involves running each workload multiple times to obtain accurate and consistent results. The latency and throughput measurements are taken at various stages of the processing pipeline to assess the impact of the co-design approach on the system's performance. Statistical analysis is conducted to evaluate the significance of the improvements, and the results are used to validate the hypothesis that hardware software co-design reduces latency and enhances throughput for multimedia applications.

4. Results and Discussion

The hardware software co-designed system demonstrated significant improvements in both latency and throughput compared to traditional software-only DSP implementations. By integrating custom DSP hardware accelerators (built with FPGA technology) and optimized software frameworks, the system reduced latency by up to 1.85x and increased throughput by up to 1.5x. This combination allowed for efficient parallel processing,

minimizing delays and enhancing real time multimedia performance, particularly in tasks like image recognition and video decoding. The system's flexibility in dynamically allocating resources between hardware and software also ensured scalability, making it well-suited for complex multimedia workloads while maintaining energy efficiency. These results highlight the effectiveness of co-design in optimizing both performance and power consumption for real time applications.

Results

The co-designed system showed significant improvements in latency and throughput compared to traditional software-only DSP implementations. The integration of custom DSP hardware accelerators, built using FPGA technology, enabled a reduction in latency by up to 1.85x on embedded System on Chips (SoCs) and 1.59x on high-end GPUs. This reduction in latency was particularly evident in real time multimedia tasks, such as image recognition and video decoding, where every millisecond of delay is critical. The system achieved a significant throughput improvement of up to 1.5x, particularly for tasks like image filtering and audio recognition. The offloading of computationally intensive operations to hardware accelerators allowed the software framework to focus on less resource-intensive tasks, thus optimizing the overall processing time and enhancing the system's efficiency.

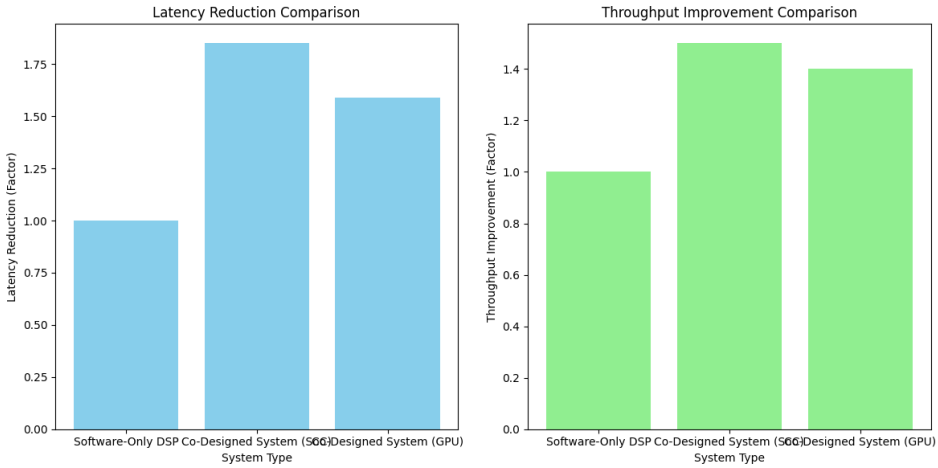

Figure 2. Throughput Improvement Comparison.

The supporting graphs compare the performance of the co-designed system with traditional software-only DSP implementations. The latency reduction graph shows a significant improvement, with the co-designed system achieving a 1.85x reduction in latency on embedded System on Chips (SoCs) and a 1.59x reduction on high-end GPUs compared to the software-only DSP system. Additionally, the throughput improvement graph highlights a noticeable increase in processing speed, with the co-designed system achieving a 1.5x improvement for tasks such as image filtering and audio recognition. These results demonstrate the co-design approach's effectiveness in reducing latency and enhancing throughput in multimedia processing tasks.

Discussion

The performance improvements in latency and throughput demonstrate the effectiveness of hardware-software co-design in multimedia DSP applications. By combining the strengths of hardware accelerators and optimized software, the system overcame the limitations of software-only DSP implementations. Hardware accelerators, such as those built with FPGAs, enabled parallel processing, reducing processing time and enhancing the overall performance of the system. Additionally, the optimized software framework ensured efficient data management and processing, further improving the system's efficiency and reducing delays typically seen in software-only approaches. This combination of hardware and software optimizations allowed the system to achieve high performance while maintaining low power consumption, making it well-suited for embedded systems.

The key performance indicators (KPIs) for this study included latency, throughput, and energy efficiency. The co-designed system not only reduced latency but also improved throughput, showing up to a 1.5x increase in processing speed for multimedia tasks. These results suggest that the co-design approach is highly effective in real time applications where high throughput is required. The flexibility of the system, allowing dynamic allocation of resources between hardware and software, ensures that it can adapt to varying multimedia workloads. The co-designed system's ability to balance performance and energy efficiency is particularly beneficial for real time applications, such as video streaming or large-scale audio processing, where both performance and power consumption are critical.

The scalability of the system is another important factor that supports its viability for diverse multimedia applications. As multimedia processing tasks become more complex, the need for scalable solutions grows. The hardware software co-design approach provides a scalable solution by allowing the system to dynamically adjust the number of processing units or the complexity of tasks being processed. This adaptability ensures that the system can handle larger datasets and more complex multimedia applications without compromising performance. The system's ability to scale efficiently also ensures that it can support future advancements in multimedia processing, where increasing data volumes and processing demands are expected.

5. Comparison

When comparing the hardware software co-design system to traditional CPU-based DSP implementations, the co-design system significantly outperforms in terms of both latency and throughput. CPU-based solutions, while versatile, often struggle with high-latency tasks due to their limited parallel processing capabilities. In contrast, the hardware software co-design system leverages custom DSP hardware accelerators to perform parallel processing, leading to substantial reductions in latency. The custom hardware accelerators, built using FPGA technology, are specifically designed to handle computationally intensive tasks efficiently, such as deep learning-based image recognition and video decoding, which are critical for real time multimedia applications. CPU-based solutions, though capable, fall short in comparison, particularly when handling complex, latency-sensitive tasks. The hardware software co-design system also demonstrates a higher throughput, processing larger datasets in less time compared to software-only CPU-based systems.

When compared to GPU-dependent multimedia processing solutions, the hardware software co-design system offers distinct advantages in terms of both latency reduction and energy efficiency. GPUs excel in parallel processing and offer significant performance improvements for compute-intensive tasks, especially in multimedia applications that require handling large datasets. However, GPUs are limited by memory latency, which can reduce their effectiveness in real time applications. The co-design system, with its hardware accelerators, minimizes memory latency by processing data closer to the source, reducing delays typically encountered in GPU-based systems. Additionally, while GPUs provide impressive computational power, they can be power-hungry, which can be a drawback in embedded systems where energy efficiency is crucial. The hardware-software co-design system achieves a balance between performance and energy consumption, making it more suitable for energy-constrained, real time applications. Thus, while GPU-based solutions offer high computational power, the co-designed system surpasses them in terms of low-latency processing and energy efficiency.

When benchmarking against existing hardware accelerators, such as FPGAs and ASICs, the hardware software co-design system also demonstrates several advantages. Traditional hardware accelerators like FPGAs are highly effective in ultra-low-latency applications, offering high performance for specific tasks like image processing or signal processing. However, they often require specialized programming and may not be as flexible in handling a variety of tasks. The co-design system, by integrating both hardware and software optimizations, offers greater flexibility, as it can dynamically allocate resources between hardware accelerators and software frameworks to suit different types of multimedia applications. Additionally, while ASICs are highly efficient for specific tasks, they lack the adaptability that a hardware software co-design system provides, which is crucial as multimedia workloads evolve. The co-design system, combining the strengths of both hardware accelerators and software, presents a more adaptable and scalable solution, capable

of handling a wider range of applications with reduced latency and enhanced throughput. Therefore, the hardware software co-design approach offers a more versatile and efficient solution compared to conventional hardware accelerators like FPGAs and ASICs.

6. Conclusions

The research findings highlight the significant advantages of the hardware software co-design approach in improving both latency and throughput in multimedia processing tasks. The co-designed system, integrating custom DSP hardware accelerators with optimized software frameworks, demonstrated a substantial reduction in latency-up to 1.85x on embedded SoCs and 1.59x on high-end GPUs. Additionally, throughput was enhanced by up to 1.5x, particularly in tasks such as image filtering and audio recognition. These improvements were attributed to the parallel processing capabilities of hardware accelerators, which efficiently handled computationally intensive tasks, while the software framework managed less resource-demanding operations. Overall, the hardware software co-design system outperformed traditional software-only DSP implementations in terms of both performance and efficiency.

The results of this study have significant practical implications for real time multimedia applications, where latency and throughput are critical factors. The reduced latency and improved throughput achieved by the co-designed system make it highly suitable for applications such as video streaming, image processing, and large-scale audio processing, which require real time data handling. By reducing the time required to process multimedia data, the system ensures faster response times, which is essential in environments like autonomous driving, healthcare, and remote monitoring, where timely data processing can have crucial outcomes. Moreover, the energy efficiency of the co-designed system makes it particularly suitable for embedded systems, which often face power constraints. This balance between performance, latency, and energy efficiency positions the hardware software co-design approach as a promising solution for real time multimedia applications.

Future research could explore several directions to further enhance the performance of hardware software co-design systems. One potential area for improvement is the exploration of alternative hardware accelerators, such as more advanced FPGA architectures or the integration of specialized processing units designed for deep learning applications. Additionally, future work could focus on further optimizing software frameworks to better map deep learning models onto hardware accelerators, with an emphasis on minimizing resource consumption and maximizing parallelism. Another avenue for research could involve testing the co-design approach with various multimedia tasks of increasing complexity to assess its scalability and adaptability. Investigating the integration of artificial intelligence models for dynamic resource allocation based on real time workload requirements could also contribute to enhancing the system's overall efficiency and performance.

References

[1] S.-C. Chen, "Multimedia Meets Deep Reinforcement Learning," *IEEE Multimed.*, vol. 29, no. 3, pp. 5 – 7, 2022, doi: 10.1109/MMUL.2022.3196479.

[2] U. A. Bhatti, J. Li, M. Huang, S. U. Bazai, and M. Aamir, *Deep Learning for Multimedia Processing Applications: Volume Two: Signal Processing and Pattern Recognition*. 2024. doi: 10.1201/9781032646268.

[3] D. Jaiswal and P. Kumar, "A survey on parallel computing for traditional computer vision," *Concurr. Comput. Pract. Exp.*, vol. 34, no. 4, 2022, doi: 10.1002/cpe.6638.

[4] S.-C. Chen, "Multimedia Data Analysis with Edge Computing," *IEEE Multimed.*, vol. 28, no. 4, pp. 5 – 7, 2021, doi: 10.1109/MMUL.2021.3124292.

[5] A. Sassu, J. F. Saenz-Cogollo, and M. Agelli, "Deep-framework: A distributed, scalable, and edge-oriented framework for real-time analysis of video streams," *Sensors*, vol. 21, no. 12, 2021, doi: 10.3390/s21124045.

[6] Danang, T. Wahyono, I. Sembiring, T. Wellem, and N. H. Dzulkefly, "An Adaptive Framework Integrating ML Blockchain and TEE for Cloud Security," in *Proceeding - 2025 4th International Conference on Creative Communication and Innovative Technology: Empowering*

Transformative MATURE LEADERSHIP: Harnessing Technological Advancement for Global Sustainability, ICCIT 2025, 2025. doi: 10.1109/ICCIT65724.2025.11167152.

[7] D. Danang and Z. Mustofa, "CLSTMNet Architecture: A CNN–LSTM-Based Hybrid Deep Learning Model for DDoS Attack Detection and Mitigation in Network Security," *J. Artif. Intell. Technol.*, 2026.

[8] T. Pfau, *Real-Time Implementation of High-Speed Digital Coherent Transceivers*. 2016. doi: 10.1002/9781119078289.ch12.

[9] S. Zouzoula, M. W. Azhar, and P. Trancoso, "RAINBOW: Multi-Dimensional Hardware-Software Co-Design for DL Accelerator On-Chip Memory," in *Proceedings - 2023 IEEE International Symposium on Performance Analysis of Systems and Software, ISPASS 2023*, 2023, pp. 352 – 354. doi: 10.1109/ISPASS57527.2023.00050.

[10] J. Zheng, Y. Liu, X. Liu, L. Liang, D. Chen, and K.-T. Cheng, "ReAAP: A Reconfigurable and Algorithm-Oriented Array Processor With Compiler-Architecture Co-Design," *IEEE Trans. Comput.*, vol. 71, no. 12, pp. 3088 – 3100, 2022, doi: 10.1109/TC.2022.3213177.

[11] D. Danang and Z. Mustofa, "Digital Forensics and Automated Incident Response Framework Leveraging Big Data Analytics and Real Time Network Traffic Profiling in Heterogeneous Cyber Environments," *Cyber Secur. Netw. Manag.*, vol. 1, no. 1, pp. 44–45, 2026.

[12] E. Siswanto, D. Danang, I. Kusumaningroem, and I. Akhsani, "Assessing Software Architecture Resilience Using Quantitative Metrics in Cloud Native Application Development Environments," *Indones. J. Infomatics*, vol. 1, no. 1, pp. 11–21, 2026.

[13] A. Dube, A. Wagle, G. Singh, and S. Vrudhula, "Tunable precision control for approximate image filtering in an in-memory architecture with embedded neurons," in *IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers, ICCAD*, 2022. doi: 10.1145/3508352.3549385.

[14] A. Anderson, J. Su, R. Dahyot, and D. Gregg, "Performance-Oriented Neural Architecture Search," in *2019 International Conference on High Performance Computing and Simulation, HPCS 2019*, 2019, pp. 177 – 184. doi: 10.1109/HPCS48598.2019.9188213.

[15] K.-A. Tran, A. Jimborean, T. E. Carlson, K. Koukos, M. Själander, and S. Kaxiras, "SWOOP: Software-hardware co-design for non-speculative, execute-ahead, in-order cores," *ACM SIGPLAN Not.*, vol. 53, no. 4, pp. 328 – 343, 2018, doi: 10.1145/3192366.3192393.

[16] U. A. Bhatti, J. Li, M. Huang, S. U. Bazai, and M. Aamir, *Deep Learning for Multimedia Processing Applications: Volume One: Image Security and Intelligent Systems for Multimedia Processing*. 2024. doi: 10.1201/9781003427674.

[17] H. Xiong and others, "Advances in Mathematical Theory for Multimedia Signal Processing," *J. Image Graph.*, vol. 25, no. 1, pp. 1–18, 2020, doi: 10.11834/jig.190468.

[18] L. Moysis *et al.*, "Music Deep Learning: Deep Learning Methods for Music Signal Processing - A Review of the State-of-the-Art," *IEEE Access*, vol. 11, pp. 17031 – 17052, 2023, doi: 10.1109/ACCESS.2023.3244620.

[19] D. Danang, M. U. Dewi, and G. Widhiati, "Federated Hybrid CNN GRU and COBCO Optimized Elman Neural Network for Real Time DDoS Detection in Cloud Edge Environments," *Int. J. Electr. Eng. Math. Comput. Sci.*, vol. 2, no. 2, pp. 28–35, 2025, doi: https://doi.org/10.62951/ijeemcs.v2i2.293.

[20] Y. Liu, Y. Li, Y. Zhu, Y. Niu, and P. Jia, "A Brief Review on Deep Learning in Application of Communication Signal Processing," in *2020 IEEE 5th International Conference on Signal and Image Processing, ICSIP 2020*, 2020, pp. 51 – 54. doi: 10.1109/ICSIP49896.2020.9339345.

[21] S. Niu, "Research on the application of machine learning big data mining algorithms in digital signal processing," in *Proceedings of IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers, IPEC 2021*, 2021, pp. 776 – 779. doi: 10.1109/IPEC51340.2021.9421229.

[22] D. Danang, E. Siswanto, W. Aryani, and P. Wibowo, "Hybrid Federated Ensemble Learning Approach for Real-Time Distributed DDoS Detection in IIoT Edge Computing Environment," *J. Eng. Electr. Informatics*, vol. 5, no. 1, pp. 9–17, 2025, doi: https://doi.org/10.55606/jeei.v5i1.5099.

[23] D. Danang, M. U. Dewi, and W. Aryani, "Systematic Literature Review on the Application of Blockchain in Enhancing Server

Security: Research Methods for Mitigating Ransomware and Malware Attacks," *Int. J. Comput. Technol. Sci.*, vol. 1, no. 4, pp. 27–51, 2024, doi: https://doi.org/10.62951/ijcts.v1i4.186.

[24] R. Venkatasubramanian, *Quest for energy efficiency in digital signal processing: Architectures, algorithms, and systems*. 2017. doi: 10.1201/b17635.

[25] D. Danang, H. Haryani, Q. Aini, F. A. Ramahdan, and J. Edwards, "Empowering Digital Literacy Through Blockchain Based Alphasign for Secure and Sustainable E-Governance," 2025.

[26] S. Agharass, M. Laaboubi, A. Saddik, and R. Latif, "Hardware Software Co-design based CPU-FPGA Architecture: Overview and Evaluation," in *Proceedings - 2021 International Conference on Digital Age and Technological Advances for Sustainable Development, ICDATA 2021*, 2021, pp. 147 – 154. doi: 10.1109/ICDATA52997.2021.00037.

[27] N. Hou, X. Yan, and F. He, "A survey on partitioning models, solution algorithms and algorithm parallelization for hardware/software co-design," *Des. Autom. Embed. Syst.*, vol. 23, no. 1–2, pp. 57 – 77, 2019, doi: 10.1007/s10617-019-09220-7.

[28] B.-P. Tine, S. Yalamanchili, and H. Kim, "Tango: An Optimizing Compiler for Just-In-Time RTL Simulation," in *Proceedings of the 2020 Design, Automation and Test in Europe Conference and Exhibition, DATE 2020*, 2020, pp. 157 – 162. doi: 10.23919/DATE48585.2020.9116253.

[29] D. Danang, I. A. Dianta, A. B. Santoso, and S. Kholifah, "Hybrid CNN GRU Framework for Early Detection and Adaptive Mitigation of DDoS Attacks in SDN using Image Based Traffic Analysis," *Int. J. Inf. Eng. Sci.*, vol. 2, no. 2, pp. 66–78, 2025, doi: https://doi.org/10.62951/ijies.v2i2.292.

[30] D. Danang, N. D. Setiawan, and E. Siswanto, "Pemanfaatan Teknologi Internet of Things untuk Monitoring Kualitas Air Sungai di Wilayah Perkotaan," *J. New Trends Sci.*, vol. 2, no. 1, pp. 23–34, 2024.

[31] Q. Xiao, S. Zheng, B. Wu, P. Xu, X. Qian, and Y. Liang, "HASCO: Towards agile hardware and software CO-design for tensor computation," in *Proceedings - International Symposium on Computer Architecture*, 2021, pp. 1055 – 1068. doi: 10.1109/ISCA52012.2021.00086.

[32] Y. Oshima, Y. Yamaguchi, R. Tsugami, T. Fujiwara, T. Fukui, and S. Narikawa, "FPGA-Based Improved Background Subtraction for Ultra-Low Latency," *IEEE Access*, vol. 12, pp. 164063 – 164080, 2024, doi: 10.1109/ACCESS.2024.3483548.

[33] D. Danang, A. B. Santoso, and M. U. Dewi, "CICA Framework: Harnessing CSR, AI, and Blockchain for Sustainable Digital Culture," *Int. J. Adv. Comput. Sci. \& Appl.*, vol. 16, no. 11, 2025.

[34] D. Nagy, L. Plavecz, and F. Hegedűs, "The art of solving a large number of non-stiff, low-dimensional ordinary differential equation systems on GPUs and CPUs," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 112, 2022, doi: 10.1016/j.cnsns.2022.106521.

[35] M. Nazemi, A. Fayyazi, A. Esmaili, A. Khare, S. N. Shahsavani, and M. Pedram, "NullaNet Tiny: Ultra-low-latency DNN Inference through Fixed-function Combinational Logic," in *Proceedings - 29th IEEE International Symposium on Field-Programmable Custom Computing Machines, FCCM 2021*, 2021, pp. 266 – 267. doi: 10.1109/FCCM51124.2021.00053.

[36] D. Danang, E. Siswanto, N. D. Setiawan, and P. Wibowo, "Hybrid Zero Trust Container Based Model for Proactive Service Continuity under Intelligent DDoS Attacks in Cloud Environment," *Int. J. Comput. Technol. Sci.*, vol. 2, no. 3, pp. 41–49, 2025, doi: https://doi.org/10.62951/ijcts.v2i3.291.