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 significantly influenced by the increasing demand for deep learning techniques. Deep learning, particularly through models like convolutional neural networks (CNNs), recurrent neural networks (RNNs), and generative adversarial networks (GANs), has revolutionized multimedia processing by enabling more efficient and accurate tasks such as image recognition, object detection, semantic segmentation, and image synthesis [1], [2]. These advancements extend to video analysis, where deep learning models excel in tasks like action recognition, video captioning, and video generation, all of which are crucial for extracting meaningful information from videos [1]. Similarly, deep learning has opened new frontiers in audio processing, including applications in speech recognition, music classification, and sound event detection, thus enhancing multimedia application capabilities [3]. Furthermore, the integration of natural language processing (NLP) with deep learning models enhances the capacity to understand,

generate, and interpret textual information within multimedia contexts, leading to more sophisticated, real-time data processing systems [4].

As the number of connected sensors and mobile devices generating vast amounts of multimedia data increases, the demand for real-time processing becomes critical. Deep learning models, with their ability to handle large datasets efficiently, are positioned as ideal solutions for addressing these real-time processing needs [5]. This growing demand for rapid and effective processing capabilities in various applications, including autonomous driving, healthcare, and disaster management, underscores the need for efficient multimedia systems that can process data instantaneously [4].

However, traditional software-only digital signal processing (DSP) systems face substantial challenges, particularly in terms of latency and inefficiency. The signal latency caused by analog-to-digital (A/D) and digital-to-analog (D/A) conversion can hinder real-time processing capabilities, as these processes are relatively slow [6]. Furthermore, relying on a single central processing unit (CPU) for signal processing tasks introduces additional inefficiencies, leading to slower processing speeds that further exacerbate latency issues [2]. This traditional architecture also struggles to meet the high data rate requirements of modern multimedia applications, particularly under power constraints [3]. The increasing complexity and real-time nature of multimedia tasks demand more robust solutions, such as edge computing, which offers a promising approach by bringing processing closer to the data source, thereby reducing latency and improving overall efficiency [4].

Edge computing enables the deployment of deep learning models on edge devices, offering the potential for enhanced processing speeds and reduced latency by eliminating the delays associated with centralized processing [5]. The use of parallelizable algorithms and hardware platforms designed to support both traditional and deep learning-based computer vision applications can further boost the efficiency and real-time capabilities of multimedia processing systems [2]. This hybrid approach combining hardware and software accelerators presents a viable path to overcoming the limitations of software-only DSP implementations.

In recent years, the demand for real-time multimedia applications has surged, primarily driven by the increasing need for deep learning-accelerated digital signal processing (DSP) tasks. These applications, which range from speech recognition to video analysis, require highly efficient systems capable of processing large volumes of data with minimal latency [1], [2]. Traditional software-only DSP implementations often fail to meet these stringent requirements due to inherent limitations in processing speed and resource utilization [6]. To address these challenges, hardware-software co-design has emerged as a promising approach for optimizing performance and reducing latency in deep learning-based DSP tasks [7], [8].

Hardware-software co-design combines the strengths of both hardware and software optimizations to enhance DSP performance. By partitioning tasks between software programmable DSPs and specialized hardware accelerators, such as FPGAs, this methodology improves efficiency and reduces computational bottlenecks [7]. Co-design enables the effective mapping of deep learning algorithms to hardware, ensuring that both resources are utilized optimally, thereby improving runtime performance and minimizing latency [9]. For instance, a co-design approach for keyword spotting on edge devices demonstrated a significant reduction in latency, achieving a 1.85x improvement on embedded SoCs and a 1.59x improvement on high-end GPUs [10]. This demonstrates the potential of co-design in enhancing the efficiency of multimedia applications.

The integration of parallelism and data management is another key advantage of hardware-software co-design. For example, compiler-architecture co-design schemes for reconfigurable array processors can optimize parallelism and data reuse, leading to high utilization of hardware resources and further latency reduction [7]. Additionally, the co-design process incorporates memory optimization techniques, such as the use of tools like RAINBOW, which facilitate the generation of heterogeneous execution plans, reducing off-chip latency costs and improving memory access efficiency [8].

The primary objective of this study is to explore how hardware-software co-design can enhance DSP performance while minimizing processing latency in multimedia applications. This research will examine co-design methodologies that optimize partitioning and resource utilization to improve runtime performance and energy efficiency. By investigating real-time multimedia tasks such as video decoding and image filtering, this study aims to provide insights into how co-design can address the key challenges of latency and computational inefficiency in deep learning-accelerated DSP systems [11].

Furthermore, this study will highlight the importance of optimizing energy efficiency and precision through dynamic co-design frameworks. Co-design methods have demonstrated significant improvements in energy efficiency, such as in the Processing in-Memory (PiM) architecture, which enhances throughput for image filtering tasks [9]. Techniques like SWOOP, which shift processing complexity into software, have also been shown to effectively hide memory latency and increase memory and instruction-level parallelism, further boosting real-time processing capabilities [11]. Overall, hardware-software co-design offers a pathway to overcoming the limitations of traditional DSP systems, enabling faster and more efficient real-time multimedia applications.

2. Literature Review

Deep Learning in Multimedia and DSP

Deep learning has made significant strides in both multimedia processing and digital signal processing (DSP) tasks, providing advanced capabilities for tasks such as image recognition, video analysis, and audio processing. In multimedia processing, techniques like convolutional neural networks (CNNs) have become foundational for tasks such as object detection, semantic segmentation, and image synthesis [12]. These methods have drastically improved the accuracy and efficiency of multimedia systems. Deep learning has also proven effective in video analysis, where it is applied to tasks like action recognition and video captioning, helping to extract valuable insights from video data [13]. Similarly, recurrent neural networks (RNNs) and generative adversarial networks (GANs) have advanced audio processing by enabling tasks such as speech recognition, music classification, and sound event detection [14]. The integration of deep learning with natural language processing (NLP) further enhances multimedia systems' ability to interpret and generate textual information, improving the understanding of complex multimedia contexts [13].

In DSP implementations, deep learning offers the potential to enhance the performance of traditional DSP systems. Deep learning-based models are increasingly used in music signal processing for tasks such as music information retrieval and music generation, showing significant potential in commercial applications [14]. In the realm of communication signal processing, deep learning is applied to symbol detection, anti-interference, and channel modeling. Although challenges remain in optimizing deep learning models for these tasks, they hold promise for enhancing DSP in communication systems [15]. Additionally, deep learning has been shown to improve general DSP tasks by introducing nonlinearities that enhance system performance, although the lack of analytical formulations remains a challenge due to the black-box nature of these models [16].

Software-Only DSP Approaches

Despite the advancements in deep learning for DSP, software-only DSP approaches still face significant limitations, particularly in terms of latency and real-time processing. One of the primary issues with software-based DSP systems is latency, which is often caused by the time it takes for analog-to-digital (A/D) and digital-to-analog (D/A) conversion, processes that can be relatively slow and hinder real-time DSP applications. Real-time DSP tasks demand hardware capable of processing data at the same rate as the application sample rate. Software-only DSP implementations typically struggle to meet these high-speed demands, leading to inefficiencies in processing [17]. Moreover, software-only solutions often encounter difficulties handling large-scale data and high-frequency resolutions, which are better managed by dedicated hardware such as digital signal processors (DSPs) and field-programmable gate arrays (FPGAs) [15], [16].

The limitations of parallel processing in software-only DSP approaches also pose challenges. Real-time multimedia applications, which require rapid and efficient data processing, depend heavily on parallel processing to achieve the necessary performance. Software-only implementations may struggle to efficiently support parallel processing, further contributing to delays and inefficiencies in DSP tasks [17]. These constraints highlight the need for hardware enhancements, such as DSPs and FPGAs, which can provide significant improvements in performance by reducing latency and enabling more efficient parallel processing [15].

Hardware-Software Co-Design Concepts

Hardware-software co-design (HSCD) is an integrated approach that combines the development of both hardware and software components to optimize system performance and efficiency. This methodology is particularly effective in domains such as embedded systems, machine learning, and multimedia processing, where performance and low latency are crucial [18]. One of the main advantages of co-design is its ability to tightly integrate hardware and software, allowing for domain-specific optimizations that lead to significant improvements in design efficiency and system performance [19]. Co-design approaches also help reduce development time by utilizing tools such as Tango, which enables just-in-time RTL simulation. Tango has demonstrated a 6x speedup compared to traditional simulators, emphasizing how co-design methodologies can accelerate the development process [19].

Furthermore, co-design can enhance performance by optimizing critical hardware and layout aspects during the system-level design phase, particularly in resource-constrained systems like System on Chips (SoCs). For example, co-design has shown notable improvements in latency, especially in tasks involving tensor computations. The HASCO approach, a co-design methodology for tensor computations, has achieved a 1.25x to 1.44x reduction in latency through effective partitioning and optimization of hardware and software resources [20]. These results underline the potential of co-design to address the performance challenges in low-latency applications.

Co-design methodologies have also proven successful in other domains. In embedded systems, co-design has led to significant reductions in time-to-market and improved system performance by efficiently partitioning tasks between hardware and software [21]. Additionally, in image processing, co-design implementations on FPGA boards for tasks such as Sobel filters have demonstrated significant efficiency gains, with FPGA-based implementations using VHDL consuming fewer resources than OpenCL-based solutions [22].

Existing Solutions for Low-Latency Multimedia Processing

The application of hardware-software co-design in multimedia processing has been particularly effective in improving performance and reducing latency across different processing units, including CPUs, GPUs, and specialized accelerators. CPUs, though versatile, often struggle with high-latency tasks due to their limited parallel processing capabilities. Optimizations like SIMD (single instruction, multiple data) and vectorization can improve performance but tend to fall short when compared to more specialized hardware. GPUs, on the other hand, excel in parallel processing and offer significant speedups for compute-intensive tasks, achieving up to a 50x speedup in radiative transfer problems compared to CPUs. However, memory latency remains a persistent challenge, necessitating parallel execution of kernel instances to hide latency [23].

Specialized accelerators, such as FPGAs, offer ultra-low latency and high efficiency for specific tasks. FPGA-based solutions have been used successfully in image processing, with background removal applications achieving substantial latency reductions compared to conventional methods [22]. Moreover, platforms like the Nomadik® leverage multiple DSPs and dedicated accelerators to provide flexible, low-power solutions for multimedia processing [24].

In latency-sensitive applications, FPGAs are increasingly favored due to their flexibility and efficiency. They are particularly effective in real-time applications such as remote operations and streaming, where low latency is essential [24]. However, power consumption remains a challenge, especially in embedded systems, where GPUs, though powerful, can be less energy-efficient. Hybrid approaches that distribute workloads between CPUs and GPUs have shown promise in optimizing both performance and energy efficiency [20]. Next-generation application processors that integrate CPUs, GPUs, and specialized accelerators are expected to offer remarkable energy efficiency and performance, making them well-suited for advanced multimedia applications [24].

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.

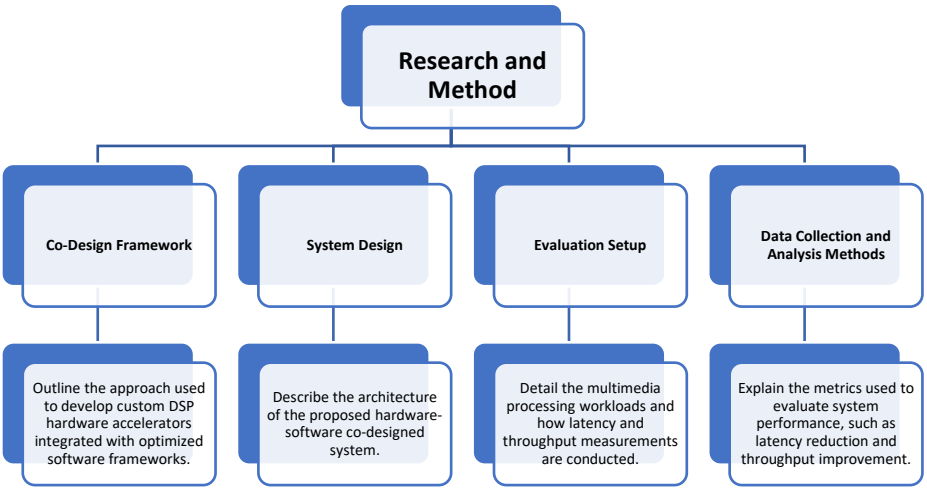

Figure 1. Flowchart structure.

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.

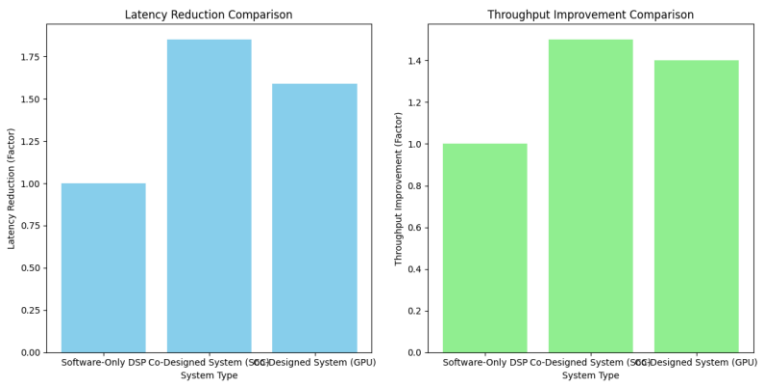

Figure 2. Throughput Improvement Comparison.

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] H. Xiong *et al.*, "Advances in mathematical theory for multimedia signal processing; [多媒体信号处理的数学理论前沿进展]," *J. Image Graph.*, vol. 25, no. 1, pp. 1 – 18, 2020, doi: 10.11834/jig.190468.

[14] 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.

[15] 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.

[16] 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.

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

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.