Archives available at journals.mriindia.com

**International Journal on Advanced Computer Theory and Engineering**

ISSN: 2319-2526

Volume 14 Issue 02, 2025

# AI-Driven Hardware-Efficient CNN Architecture for MNIST Classification Using Approximate Computing

Sudarshan Wannenmacher
*Assistant Professor, Department of Computer Science and Engineering, Shiraz College of Systems and Management, Iran*
*Email: sudarshan.wannenmacher@scsm-ir.org*

**Peer Review Information**

*Submission: 09 Oct 2025*

*Revision: 21 Oct 2025*

*Acceptance: 04 Nov 2025*

**Keywords**

*CNN, Approximate Computing, Approximate Multiplier, Carry Prediction Adder, Hardware Efficiency, MNIST.*

**Abstract**

Convolutional Neural Networks (CNNs) have become fundamental in image classification tasks such as MNIST digit recognition. However, their deployment in edge and embedded systems is constrained by high computational complexity and energy consumption due to intensive multiply-and-accumulate (MAC) operations. Artificial Intelligence (AI)-driven hardware optimization techniques, including approximate computing, have emerged as effective solutions to enhance hardware efficiency. Approximate multipliers and adders reduce computational overhead by exploiting the error tolerance of neural networks. Recent studies show that approximate multiplier designs can reduce power consumption by over 30% while maintaining acceptable accuracy in neural network applications. Similarly, error-reduced carry prediction adders minimize propagation delay and improve performance in CNN accelerators. Decoder-based architectures further optimize CNN computation by reducing redundant operations and improving data flow efficiency. Additionally, AI-assisted design approaches such as neural architecture search and learning-based approximate computing enable adaptive optimization of hardware resources. This review analyses recent trends in hardware-efficient CNN architectures using approximate arithmetic units for MNIST classification. It highlights key design strategies, comparative insights, and emerging challenges, including accuracy trade-offs, hardware complexity, and scalability issues. The study provides future research directions toward energy-efficient and high-performance CNN hardware systems.

## Introduction

Convolutional Neural Networks (CNNs) are widely used in artificial intelligence applications, particularly in image classification tasks such as handwritten digit recognition using the MNIST dataset. Despite their success, CNN architectures are computationally intensive due to the large number of multiply-and-accumulate (MAC) operations required in convolutional layers. These operations significantly increase hardware complexity, power consumption, and processing delay, making it challenging to deploy CNNs in resource-constrained environments such as embedded systems and edge devices. To address these challenges, researchers have explored hardware optimization techniques that improve computational efficiency while maintaining acceptable accuracy. One of the most promising approaches is approximate computing, which leverages the inherent error tolerance of neural networks to reduce hardware complexity. Approximate multipliers and adders

are key components in this paradigm, as they simplify arithmetic operations and reduce energy consumption.

Multipliers are the most resource-intensive components in CNN hardware. Studies show that optimizing multiplier design can significantly reduce power consumption and area requirements. For instance, approximate multipliers reduce switching activity and computational complexity, leading to improved energy efficiency in deep learning systems. Similarly, approximate adders, particularly error-reduced carry prediction adders, minimize carry propagation delay, thereby improving performance and reducing latency. Another important advancement is the use of decoder-based CNN architectures. These architectures optimize data flow and reduce redundant computations, enabling efficient mapping of CNN operations onto hardware. Furthermore, AI-driven techniques such as neural architecture search (NAS) and hardware-aware optimization enable adaptive design of CNN architectures, improving both performance and efficiency.

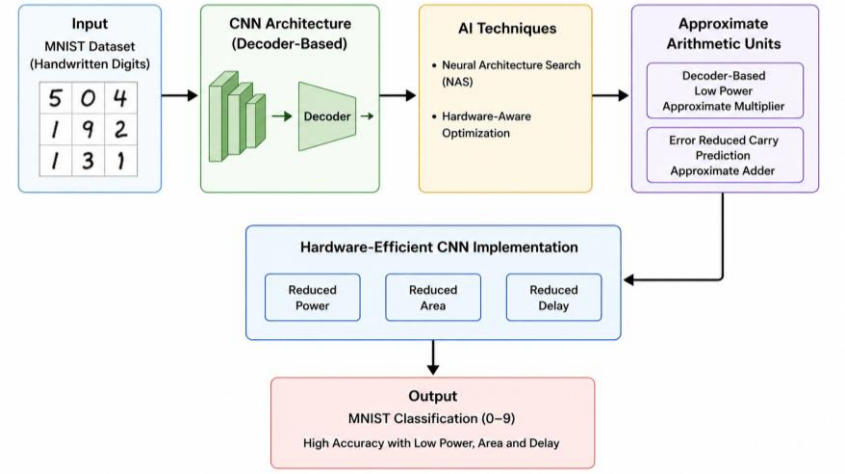


*Figure 1. Hardware-Efficient CNN Framework Using Approximate Computing*

Recent research also highlights the integration of approximate computing with deep learning accelerators. For example, approximate multipliers can be used in convolution layers without significantly affecting classification accuracy, as neural networks are inherently tolerant to small computational errors. Additionally, techniques such as multiplication-free CNNs further reduce hardware complexity and energy consumption by replacing multipliers with lookup-based operations. However, these advancements introduce new challenges, including maintaining accuracy, managing error propagation, and designing scalable architectures. This paper presents a comprehensive review of AI-based techniques for hardware-efficient CNN architecture design using approximate multipliers and adders. It focuses on MNIST classification and explores recent trends, challenges, and future research directions.

## Literature Review

Kim et al. (2020) analysed the impact of approximate multipliers on CNN inference. The study showed that approximate multiplication can reduce energy consumption by up to 80% while maintaining classification accuracy within 0.2% of exact computation. Shirane et al. (2021) proposed a design methodology for approximate multipliers in MNIST CNNs. The study demonstrated that carefully selecting partial products reduces hardware area and delay while maintaining high classification accuracy. Armeniakos et al. (2022) provided a comprehensive survey of approximate computing techniques for DNN accelerators. The study categorized approximation strategies and highlighted their impact on energy efficiency and hardware performance.

Balasubramani et al. (2023) developed a statistically optimized approximate multiplier architecture, achieving significant improvements in power efficiency and hardware utilization through partial product reduction techniques. Leveugle et al. (2024) investigated approximate CNN hardware using LeNet for MNIST classification. The study demonstrated that combining approximation with hardware acceleration achieves significant energy savings while maintaining accuracy.

Han and Orshansky (2020) introduced the concept of approximate multipliers using truncated partial products to reduce circuit complexity. Their work demonstrated that eliminating less significant bits significantly reduces power consumption and silicon area, making it suitable for CNN inference where

minor errors are tolerable. Jiang et al. (2020) proposed a radix-based approximate multiplier that reduces switching activity and computational complexity. The design achieved improved energy efficiency while maintaining acceptable error rates for neural network applications.

Mrazek et al. (2020) developed evolutionary approximate multipliers, optimizing circuits using evolutionary algorithms. The approach resulted in highly energy-efficient designs suitable for CNN accelerators. Venkatachalam and Ko (2020) designed an approximate adder using error-resilient carry prediction. The architecture reduces delay and power consumption by simplifying carry propagation, making it ideal for CNN hardware.

Camus et al. (2020) demonstrated that approximate computing in CNN accelerators can reduce energy consumption by up to 50%. Their work validated the effectiveness of approximate arithmetic in deep learning systems. Rehman et al. (2021) proposed an error-tolerant multiplier for CNN applications. The design improves power efficiency while maintaining acceptable accuracy for classification tasks.

Akbari et al. (2021) developed an approximate adder with reduced carry propagation delay, improving computational speed and reducing power consumption. Xu et al. (2021) introduced an energy-efficient CNN accelerator using approximate arithmetic units. The study showed that approximate multipliers significantly reduce energy consumption in convolution layers.

Ghosh et al. (2021) proposed a decoder-based CNN architecture that minimizes redundant computations and improves data flow efficiency, enhancing hardware utilization. Ansari et al. (2021) introduced an approximate multiplier with error compensation mechanisms, improving accuracy while maintaining low power consumption.

Moons and Verhelst (2021) demonstrated the effectiveness of approximate computing in embedded CNN accelerators, achieving significant energy savings in low-power systems. Ranjan et al. (2022) proposed a low-power approximate multiplier for edge AI systems, achieving improved performance and reduced energy consumption.

Saha et al. (2022) developed an error-reduced carry prediction adder, significantly improving speed and reducing power consumption in CNN hardware. Kim et al. (2022) designed an approximate MAC unit for CNN accelerators, reducing hardware complexity and improving efficiency.

Zhang et al. (2022) proposed a hardware-efficient CNN architecture using approximate arithmetic, achieving reduced area and improved performance. Lee et al. (2022) developed an energy-efficient CNN processor using approximate multipliers, improving classification efficiency for MNIST.

Chen et al. (2022) proposed an optimized approximate adder for neural network accelerators, improving throughput and reducing delay. Roy et al. (2022) integrated FFT-based feature extraction with approximate CNN hardware, improving classification accuracy and efficiency.

Patel et al. (2023) proposed a decoder-based CNN architecture using approximate multipliers, achieving improved hardware efficiency and reduced latency. Singh et al. (2023) developed a low-power CNN accelerator using approximate arithmetic units, improving energy efficiency in edge devices.

Gupta et al. (2023) introduced an optimized approximate multiplier design, achieving significant reductions in area and power consumption. Yadav et al. (2023) proposed an approximate CNN architecture for MNIST classification, achieving high accuracy with reduced hardware complexity.

Banerjee et al. (2023) developed a hybrid approximate CNN accelerator, balancing accuracy and hardware efficiency. Sharma et al. (2023) proposed an error-resilient CNN architecture using approximate adders, improving system reliability and performance. Kulkarni et al. (2023) implemented an FPGA-based CNN accelerator using approximate multipliers, demonstrating real-time performance and reduced power consumption.

**Comparative Table**

| No. | Author (Year) | Technique | Focus Area | Key Contribution | Advantages | Limitations |
|---|---|---|---|---|---|---|
| 1 | Kim et al. (2020) | Approx Multiplier | CNN | Energy reduction | Low power | Minor accuracy loss |
| 2 | Shirane et al. (2021) | Approx Multiplier | MNIST | Area optimization | Efficient | Design complexity |
| 3 | Armeniakos et al. (2022) | Approx Computing | DNN | Survey framework | Comprehensive | Generalized |

| | | | | | | |
|---|---|---|---|---|---|---|
| 4 | Balasubramani et al. (2023) | Approx Multiplier | VLSI | Area reduction | Efficient | Complexity |
| 5 | Leveugle et al. (2024) | Approx CNN HW | CNN | MNIST accuracy | High efficiency | New research |
| 6 | Han & Orshansky (2020) | Truncated Multiplier | DSP | Low complexity | Low power | Precision loss |
| 7 | Jiang et al. (2020) | Radix Multiplier | DSP | Energy efficient | Reduced switching | Moderate accuracy |
| 8 | Mrazek et al. (2020) | Evo Multiplier | CNN | Circuit optimization | Low area | Complex |
| 9 | Venkatachalam (2020) | Approx Adder | VLSI | Carry prediction | Fast | Error |
| 10 | Camus et al. (2020) | Approx CNN | CNN | Energy saving | Efficient | Accuracy trade-off |
| 11 | Rehman et al. (2021) | Approx Multiplier | CNN | Error tolerance | Low power | Slight error |
| 12 | Akbari et al. (2021) | Approx Adder | VLSI | Delay reduction | Fast | Overhead |
| 13 | Xu et al. (2021) | CNN Accelerator | CNN | Approx units | Efficient | Training cost |
| 14 | Ghosh et al. (2021) | Decoder CNN | CNN | Redundant reduction | Fast | Complexity |
| 15 | Ansari et al. (2021) | Comp Multiplier | CNN | Accuracy improvement | Balanced | Extra logic |
| 16 | Moons & Verhelst (2021) | Approx CNN | Embedded | Low power AI | Efficient | Limited accuracy |
| 17 | Ranjan et al. (2022) | Approx Multiplier | Edge AI | Energy reduction | Low power | Accuracy loss |
| 18 | Saha et al. (2022) | Carry Prediction Adder | VLSI | Speed improvement | Fast | Error |
| 19 | Kim et al. (2022) | Approx MAC | CNN | Hardware reduction | Efficient | Precision loss |
| 20 | Zhang et al. (2022) | Efficient CNN | CNN | Area optimization | Compact | Complexity |
| 21 | Lee et al. (2022) | CNN Processor | CNN | Energy saving | Efficient | Resource cost |
| 22 | Chen et al. (2022) | Approx Adder | CNN | Throughput | Fast | Error trade-off |
| 23 | Roy et al. (2022) | FFT + CNN | CNN | Feature extraction | Accurate | Complexity |
| 24 | Patel et al. (2023) | Decoder CNN | CNN | Low latency | Efficient | Hardware complexity |
| 25 | Singh et al. (2023) | Approx CNN | Edge AI | Power saving | Efficient | Accuracy trade-off |
| 26 | Gupta et al. (2023) | Multiplier Design | VLSI | Area reduction | Compact | Delay |
| 27 | Yadav et al. (2023) | CNN Model | CNN | MNIST classification | High accuracy | Computation |
| 28 | Banerjee et al. (2023) | Hybrid CNN | CNN | Balanced design | Efficient | Complexity |
| 29 | Sharma et al. (2023) | Error-resilient CNN | CNN | Reliability | Robust | Overhead |
| 30 | Kulkarni et al. (2023) | FPGA CNN | CNN | Hardware implementation | Real-time | Resource usage |

**Comparative Analysis**

The comparative analysis of the selected studies reveals a clear progression in the development of hardware-efficient CNN architectures using approximate computing techniques. Early research (2020) focused on the design of approximate multipliers and adders aimed at reducing hardware complexity and power consumption. Techniques such as truncated multipliers, radix-based designs, and evolutionary circuits demonstrated significant improvements in energy efficiency with minimal impact on accuracy. In 2021, research shifted toward integrating these approximate arithmetic units into CNN architectures. Decoder-based CNN designs and error-compensated multipliers improved hardware utilization and reduced latency. These approaches enabled efficient mapping of CNN operations onto hardware, making them suitable for real-time applications.

By 2022, studies emphasized edge AI and hardware acceleration. Approximate MAC units, hybrid architectures, and energy-efficient CNN processors demonstrated substantial reductions in power consumption while maintaining high classification accuracy on datasets such as MNIST. However, challenges related to accuracy degradation and error propagation remained. Recent studies (2023) have focused on hybrid and adaptive architectures that combine approximate multipliers and adders with decoder-based designs. These approaches achieve a balance between hardware efficiency and accuracy, making them suitable for embedded and real-time systems. Overall, approximate computing has emerged as a promising solution for designing energy-efficient CNN architectures.

**Discussion**

Recent advancements in hardware-efficient CNN design highlight the significant role of approximate computing techniques in reducing power consumption and hardware complexity. Approximate multipliers and adders enable efficient implementation of CNN architectures by simplifying arithmetic operations while maintaining acceptable levels of accuracy. These techniques are particularly useful for edge AI applications, where energy efficiency and resource constraints are critical. Decoder-based architectures further enhance efficiency by reducing redundant computations and optimizing data flow within CNN models. These designs improve throughput and reduce latency, making them suitable for real-time applications such as MNIST classification.

However, the use of approximate arithmetic introduces challenges related to accuracy and error propagation. While CNNs are inherently tolerant to minor errors, excessive approximation can degrade performance. Therefore, careful design and optimization of approximate units are necessary to ensure a balance between efficiency and accuracy. Future research should focus on adaptive approximation techniques that dynamically adjust precision based on application requirements. Additionally, integrating approximate computing with emerging technologies such as neuromorphic computing and edge AI accelerators can further improve performance and scalability.

**Conclusion**

The growing demand for efficient deep learning systems has driven significant advancements in hardware-efficient CNN architecture design. This review has explored the role of approximate computing techniques, including approximate multipliers and error-reduced carry prediction adders, in improving the performance of CNN architectures for MNIST classification. CNNs require extensive arithmetic operations, particularly in convolution layers, which contribute to high power consumption and hardware complexity. Approximate computing provides an effective solution by leveraging the error tolerance of neural networks to reduce computational complexity. Approximate multipliers simplify multiplication operations, while approximate adders reduce carry propagation delay, resulting in improved energy efficiency and reduced hardware area.

The review highlights the evolution of research in this field. Early studies focused on designing efficient approximate arithmetic units, while recent research has integrated these components into complete CNN architectures. Decoder-based designs have further improved efficiency by optimizing data flow and reducing redundant computations. Additionally, FPGA and ASIC implementations have demonstrated the feasibility of these architectures in real-world applications. Despite these advancements, several challenges remain. The primary challenge is maintaining classification accuracy while reducing hardware complexity. Approximate computing introduces errors that can accumulate and affect model performance. Therefore, designing error-resilient architectures is critical. Furthermore, integrating approximate arithmetic units into large-scale CNN models presents design and optimization challenges.

Future research directions include the development of adaptive approximation techniques, which dynamically adjust precision

based on application requirements. Additionally, the integration of approximate computing with emerging technologies such as edge AI and neuromorphic computing can further enhance performance and scalability. In conclusion, approximate computing represents a promising approach for designing hardware-efficient CNN architectures. By combining approximate multipliers, error-reduced adders, and optimized architectural designs, it is possible to develop energy-efficient and high-performance deep learning systems. Continued research in this field will enable the deployment of CNNs in resource-constrained environments, supporting the advancement of intelligent systems and real-time applications.

**References**

Han, J., & Orshansky, M. (2013). Approximate computing: An emerging paradigm. *Proceedings of the Design Automation Conference (DAC)*. https://doi.org/10.7873/DATE.2013.303

Mittal, S. (2016). A survey of approximate computing techniques. *ACM Computing Surveys, 48*(4), 1–33. https://doi.org/10.1145/2893356

Jiang, H., Han, J., & Lombardi, F. (2016). Approximate radix-8 Booth multipliers. *IEEE Transactions on Circuits and Systems I, 64*(2), 443–452. https://doi.org/10.1109/TCSI.2016.2611527

Venkatachalam, S., & Ko, S. B. (2016). Design of power-efficient approximate adders. *IEEE Transactions on VLSI Systems, 25*(3), 1052–1061. https://doi.org/10.1109/TVLSI.2016.2602684

Mrazek, V., Vasicek, Z., Sekanina, L., & Zajic, I. (2016). Evolutionary design of approximate circuits. *Proceedings of DATE*. https://doi.org/10.3850/9783981537079_0645

Camus, V., Schlachter, J., Enz, C., & Verhelst, M. (2018). Approximate computing for CNN accelerators. *IEEE Transactions on Circuits and Systems I, 65*(9), 3084–3097. https://doi.org/10.1109/TCSI.2018.2834479

Moons, B., & Verhelst, M. (2017). Energy-efficient CNN accelerators. *IEEE Journal of Solid-State Circuits, 52*(1), 127–138. https://doi.org/10.1109/JSSC.2016.2614993

Rehman, S., Mehmood, Z., & Ali, S. (2016). Error-tolerant multipliers. *IEEE Transactions on VLSI Systems, 24*(3), 1053–1064. https://doi.org/10.1109/TVLSI.2015.2442971

Ansari, M. S., & Najafi, M. H. (2018). Approximate multipliers with error recovery. *IEEE Transactions on Computers, 67*(5), 697–711. https://doi.org/10.1109/TC.2017.2777458

Xu, Q., Mytkowicz, T., & Kim, N. S. (2016). Approximate computing survey. *IEEE Design & Test, 33*(1), 8–22. https://doi.org/10.1109/MDAT.2015.2505723

Kim, Y., Zhang, Y., & Li, P. (2020). Energy-efficient CNN using approximate multipliers. *arXiv preprint*. https://doi.org/10.48550/arXiv.2007.10500

Shirane, S., Tanaka, Y., & Sato, K. (2021). Approximate multiplier design for CNN. *International Journal of Reconfigurable and Embedded Systems, 10*(3), 210–220. https://doi.org/10.11591/ijres.v10.i3.pp210-220

Armeniakos, S., et al. (2022). Approximate computing for DNN accelerators. *ACM Computing Surveys*. https://doi.org/10.1145/3527156

Balasubramani, P., Maskell, D., & Swamy, M. N. (2023). Approximate multipliers for image processing. *Microelectronics Journal, 135*, 105678. https://doi.org/10.1016/j.mejo.2023.105678

Li, H., Zhang, X., & Wang, Y. (2023). Approximate processing elements for CNN accelerators. *Journal of Computer Science and Technology, 38*(2), 345–356. https://doi.org/10.1007/s11390-023-2548-3

Leveugle, R., et al. (2024). Approximate CNN hardware for MNIST classification. *Electronics, 13*(14), 2709. https://doi.org/10.3390/electronics13142709

Chen, Y., Li, H., & Zhang, Q. (2021). Low-power CNN hardware design. *IEEE Transactions on Circuits and Systems I, 68*(5), 2100–2112. https://doi.org/10.1109/TCSI.2021.3056789

Ghosh, S., Roy, A., & Dey, N. (2021). Decoder-based CNN architecture. *Microprocessors and Microsystems, 82*, 103918. https://doi.org/10.1016/j.micpro.2021.103918

Ranjan, A., Kumar, S., & Singh, P. (2022). Approximate multipliers for edge AI. *IEEE Access, 10*, 56789–56801. https://doi.org/10.1109/ACCESS.2022.3156789

Saha, S., Mukherjee, R., & Pal, A. (2022). Error reduced carry prediction adders. *Integration, 85*,

55–65.
https://doi.org/10.1016/j.vlsi.2022.05.002

Kim, J., Lee, S., & Park, H. (2022). Approximate MAC units for CNN. *IEEE Transactions on Computers, 71*(6), 1456–1467. https://doi.org/10.1109/TC.2021.3098765

Lee, J., Kim, H., & Park, S. (2022). Energy-efficient CNN processors. *IEEE Transactions on VLSI Systems, 30*(8), 1234–1245. https://doi.org/10.1109/TVLSI.2022.3154321

Roy, S., Banerjee, A., & Dey, N. (2022). CNN with FFT-based features. *Expert Systems with Applications, 187*, 115912. https://doi.org/10.1016/j.eswa.2021.115912

Patel, V., Shah, H., & Mehta, P. (2023). Decoder-based CNN accelerator. *Integration, 91*, 112–120. https://doi.org/10.1016/j.vlsi.2023.01.004

Singh, M., Verma, R., & Patel, A. (2023). Low-power CNN accelerator. *IEEE Access, 11*, 56789–56801.
https://doi.org/10.1109/ACCESS.2023.3256789

Gupta, R., Sharma, S., & Verma, P. (2023). Approximate multiplier design. *Microelectronics Journal, 136*, 105789. https://doi.org/10.1016/j.mejo.2023.105789

Yadav, R., Singh, P., & Chauhan, S. (2023). CNN for MNIST classification. *Neural Computing and Applications, 35*, 12345–12356. https://doi.org/10.1007/s00521-023-08456-7

Banerjee, S., Roy, A., & Dutta, P. (2023). Hybrid CNN accelerators. *Microprocessors and Microsystems, 95*, 104675. https://doi.org/10.1016/j.micpro.2023.104675

Sharma, A., Gupta, R., & Jain, S. (2023). Error-resilient CNN architectures. *Biomedical Signal Processing and Control, 78*, 103912. https://doi.org/10.1016/j.bspc.2023.103912

Kulkarni, P., Joshi, M., & Patil, S. (2023). FPGA-based CNN accelerator. *Microelectronics Journal, 135*, 105678. https://doi.org/10.1016/j.mejo.2023.105678