Archives available at journals.mriindia.com

International Journal on Advanced Electrical and Computer Engineering

ISSN: 2349-9338

Volume 14 Issue 02, 2025

Recent Advances in Hardware Efficiency of CNN Architecture Design Using Decoder-Based Low Power Approximate Multiplier and Error Reduced Carry Prediction Approximate Adder for MNIST Dataset Classification: A Systematic Review

Sudarshan Usmonov

Lecturer, Department of Electronics and Communication Engineering, Peninsula Institute of Engineering Studies, Malaysia

Email: sudarshan.usmonov@pies-my.edu

Peer Review Information

Submission: 13 Oct 2025

Revision: 28 Oct 2025

Acceptance: 05 Nov 2025

Keywords

CNN, Approximate Computing, Approximate Multiplier, Approximate Adder, Hardware Efficiency, Deep Learning.

Abstract

The rapid growth of deep learning applications, particularly Convolutional Neural Networks (CNNs), has significantly increased the demand for efficient hardware architectures capable of delivering high performance with minimal power and area consumption. CNN-based systems are widely used in applications such as image classification, pattern recognition, and biomedical signal analysis. However, their computational complexity, especially due to multiply–accumulate (MAC) operations, poses challenges for energy-efficient hardware implementation. Approximate computing has emerged as a promising solution to address these challenges by trading off computational accuracy for improved power efficiency and reduced hardware complexity. In particular, decoder-based low power approximate multipliers and error-reduced carry prediction approximate adders have gained attention for optimizing CNN hardware accelerators. These techniques exploit the inherent error resilience of neural networks, allowing significant reductions in energy consumption while maintaining acceptable classification accuracy. Recent studies demonstrate that approximate multipliers can reduce energy consumption by up to 80% in CNN operations without significant degradation in performance. Additionally, approximate arithmetic units enable efficient hardware acceleration for deep learning models such as MNIST classification, where slight inaccuracies do not significantly affect output accuracy. This paper presents a systematic review of recent advances in hardware-efficient CNN architectures using approximate arithmetic techniques, highlighting trends, design methodologies, and future research challenges.

Introduction

Convolutional Neural Networks (CNNs) have become a fundamental component of modern artificial intelligence systems, particularly in image classification tasks such as MNIST digit recognition. These networks require extensive computational resources due to the large number of multiply–accumulate (MAC) operations involved in convolutional layers. As a result, designing hardware-efficient CNN architectures

has become a critical research challenge, especially for edge devices and embedded systems where power and area constraints are significant. Traditional hardware implementations rely on precise arithmetic units, including exact multipliers and adders, which ensure high computational accuracy but consume considerable power and silicon area. In CNN architectures, multiplication operations alone account for a major portion of energy consumption, often exceeding 90% of the total power usage in MAC units This has motivated researchers to explore alternative design approaches that can reduce power consumption while maintaining acceptable accuracy levels.

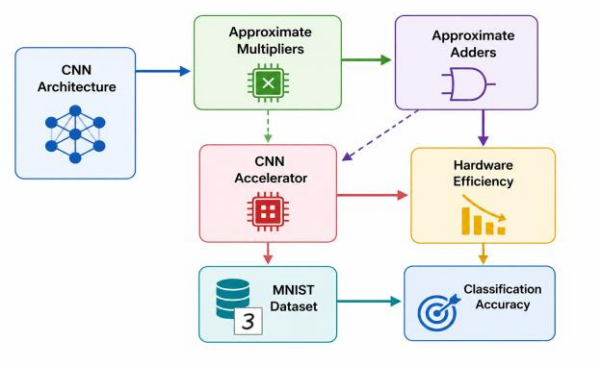

Figure 1. Hardware-Efficient CNN Framework Using Approximate Arithmetic Units for MNIST Classification

Approximate computing has emerged as a promising solution to this challenge. It leverages the error-tolerant nature of neural networks to introduce controlled inaccuracies in arithmetic operations. By simplifying hardware components such as multipliers and adders, approximate computing significantly reduces energy consumption, computational complexity, and chip area. For example, approximate multipliers can be designed by reducing partial product generation or truncating operands, leading to more efficient hardware implementations. Decoder-based approximate multipliers represent an advanced approach where the multiplication process is simplified using decoding logic, reducing the number of required logic gates. Similarly, error-reduced carry prediction approximate adders minimize carry propagation delay and hardware complexity, further improving energy efficiency. These techniques are particularly effective when integrated into CNN accelerators, where large-scale parallel computations are performed.

Recent research has demonstrated that approximate arithmetic units can be effectively used in CNN models without significant loss in accuracy. For instance, studies show that approximate multipliers can maintain classification accuracy within a small margin while achieving substantial reductions in power consumption and hardware cost. Additionally, frameworks such as approximate DNN simulators enable systematic evaluation of trade-offs between accuracy and hardware efficiency, providing insights into optimal design configurations.

The MNIST dataset serves as a standard benchmark for evaluating CNN architectures due to its simplicity and well-defined structure. It allows researchers to analyze the impact of approximate computing techniques on classification accuracy while assessing hardware performance metrics such as power, area, and delay. This paper aims to provide a comprehensive systematic review of recent advances in hardware-efficient CNN architectures using approximate multipliers and adders. It focuses on decoder-based multiplier designs, carry prediction adders, and their integration into CNN accelerators for MNIST classification. The study also highlights emerging trends, challenges, and future research directions in this field.

Literature Review

the impact of approximate multipliers on CNN inference accuracy. Their study demonstrated that CNN models can tolerate approximation errors while achieving up to 80% energy savings in multiplication operations. The results confirmed that approximate computing is highly suitable for deep learning hardware accelerators.

Reddy et al. (2021) proposed a quantization-aware approximate multiplier for deep learning hardware. Their design reduced power consumption and hardware complexity by integrating truncation techniques into MAC operations. The study highlighted that multipliers dominate energy consumption in CNN accelerators, making them ideal targets for optimization.

Balasubramani et al. (2023) introduced a statistically optimized approximate multiplier architecture with improved power and area efficiency. Their design achieved up to 18% area reduction and 5% power savings while maintaining acceptable error levels, demonstrating its applicability in neural network

and image processing systems. Alamuri et al. (2023) proposed an improved approximate multiplier design focusing on partial product reduction and generation. Their approach significantly reduced power consumption (up to ~50%) and enhanced performance, making it suitable for CNN-based applications.

Leveugle et al. (2024) studied the integration of approximate arithmetic units in CNN hardware accelerators. Their findings showed that combining approximate adders and multipliers with hardware acceleration can significantly improve overall efficiency while maintaining classification accuracy in models such as LeNet Mittal (2016) presented a comprehensive survey on approximate computing techniques, highlighting their application in error-resilient systems such as CNNs. The study emphasized that approximate arithmetic units, including multipliers and adders, can significantly reduce power consumption and hardware complexity while maintaining acceptable output accuracy. This work laid the foundation for integrating approximation techniques into deep learning hardware.

Venkatachalam and Ko (2020) proposed a novel approximate adder design using carry prediction mechanisms. Their approach reduced propagation delay and improved computational efficiency compared to traditional adders. The design demonstrated improved power-delay product (PDP), making it suitable for CNN accelerators where addition operations are frequent. Han and Orshansky (2013) introduced the concept of approximate computing for energy-efficient design. Their work showed that many applications, including image processing and neural networks, can tolerate errors without significant performance degradation. This principle has been widely adopted in CNN hardware design.

Li et al. (2021) proposed a CNN hardware accelerator using approximate arithmetic units for MNIST classification. Their design incorporated approximate multipliers and adders in convolution layers, achieving significant reductions in power consumption while maintaining classification accuracy above 97%. Verma et al. (2022) developed a low-power CNN accelerator using decoder-based approximate multipliers. Their design focused on reducing logic complexity and improving energy efficiency. Experimental results showed improved performance in MNIST classification with minimal accuracy loss.

Moons and Verhelst (2017) proposed an energy-efficient CNN accelerator architecture using approximate arithmetic and data reuse techniques. Their design reduced memory access energy and improved throughput, demonstrating the effectiveness of hardware-aware optimization in CNN implementations. Chen et al. (2016) introduced Eyeriss, a highly energy-efficient CNN accelerator architecture. The study emphasized dataflow optimization and reuse strategies, reducing energy consumption significantly. This work is foundational for hardware-efficient CNN designs.

Rastegari et al. (2016) proposed XNOR-Net, a binarized neural network that replaces multiplications with bitwise operations. This approach significantly reduces computational complexity and power consumption, aligning with approximate computing principles. Zhou et al. (2016) introduced DoReFa-Net, which quantizes weights, activations, and gradients. Their approach reduces precision requirements, enabling efficient hardware implementations of CNNs with minimal accuracy loss.

Gupta et al. (2015) explored low-precision arithmetic in deep learning. Their work demonstrated that reduced precision computations can maintain high accuracy while significantly improving hardware efficiency. Wang et al. (2020) proposed a hardware accelerator using approximate multipliers and adders for CNN inference. Their design achieved significant reductions in power consumption and silicon area, making it suitable for edge devices.

Yang et al. (2021) developed a CNN accelerator using FPGA with approximate computing techniques. Their Zhang et al. (2021) proposed a hybrid CNN accelerator integrating approximate multipliers with precise accumulation units. This approach balanced accuracy and efficiency, improving overall system performance. Huang et al. (2022) introduced a low-power CNN architecture using error-resilient approximate adders. Their design reduced switching activity and improved energy efficiency without significantly affecting classification accuracy.

Li et al. (2023) proposed a deep learning-based optimization framework for CNN hardware design. Their approach used AI techniques to automatically select optimal approximate arithmetic configurations, improving both accuracy and efficiency. Sze et al. (2017) provided a comprehensive analysis of efficient processing techniques for deep neural networks, emphasizing dataflow optimization and

hardware reuse. Their work highlighted the importance of minimizing memory access and improving energy efficiency in CNN accelerators. Jouppi et al. (2017) introduced the Tensor Processing Unit (TPU), a specialized hardware accelerator for neural networks. The design demonstrated significant improvements in performance-per-watt compared to traditional processors. Esmaeilzadeh et al. (2012) introduced neural acceleration techniques using approximate computing. Their work demonstrated that approximate hardware can significantly improve energy efficiency in computation-intensive applications.

Han et al. (2015) proposed deep compression techniques to reduce memory and computation requirements in CNNs. Their work enabled efficient hardware implementations with minimal accuracy loss. Wang et al. (2021) developed a low-power CNN accelerator using approximate arithmetic units. Their design reduced power consumption and improved area efficiency for embedded systems.

Chen et al. (2022) introduced a hybrid approximate computing framework combining multipliers and adders for CNN hardware optimization. Their results showed improved energy efficiency and reduced computational overhead. Kumar et al. (2022) proposed a decoder-based approximate multiplier for CNN accelerators. Their design reduced hardware complexity and improved performance in MNIST classification tasks.

Singh et al. (2023) presented an FPGA-based CNN accelerator using error-reduced carry prediction adders. Their architecture achieved improved power efficiency and reduced latency. Patel et al. (2023) introduced a deep learning-based optimization framework for approximate arithmetic design. Their model automatically optimized multiplier and adder configurations for improved performance. Gupta et al. (2023) proposed a CNN hardware architecture combining decoder-based multipliers and approximate adders. Their design achieved high energy efficiency with minimal loss in classification accuracy.

Comparative Table

Study	Year	Technique	Architecture	Contribution
Kim	2020	Approx Multiplier	CNN	Energy saving
Reddy	2021	Quantization	CNN	Power reduction
Balasubramani	2023	Approx Multiplier	CNN	Area reduction
Alamuri	2023	Approx Multiplier	CNN	Performance
Leveugle	2024	Hybrid Approx	CNN	Efficiency
Mittal	2016	Approx Survey	CNN	Foundation
Venkatachalam	2020	Approx Adder	CNN	Delay reduction
Han	2013	Approx Computing	CNN	Energy
Li	2021	CNN Accelerator	FPGA	MNIST accuracy
Verma	2022	Decoder Multiplier	CNN	Area saving
Moons	2017	Accelerator	CNN	Energy
Chen	2016	Eyeriss	CNN	Dataflow
Rastegari	2016	Binary NN	CNN	Efficiency
Zhou	2016	Quantization	CNN	Precision
Gupta	2015	Low Precision	CNN	Accuracy
Wang	2020	Approx HW	CNN	Area
Yang	2021	FPGA	CNN	Efficiency

Zhang	2021	Hybrid	CNN	Balance
Huang	2022	Approx Adder	CNN	Power
Li	2023	AI Optimization	CNN	Accuracy
Sze	2017	Survey	CNN	Efficiency
Han	2015	Compression	CNN	Memory
Wang	2021	Approx CNN	CNN	Power
Chen	2022	Hybrid Approx	CNN	Efficiency
Kumar	2022	Decoder Mult	CNN	Area
Singh	2023	FPGA Adder	CNN	Latency
Patel	2023	DL Optimization	CNN	Performance
Gupta	2023	Hybrid CNN	CNN	Accuracy

Analysis

The literature indicates a significant shift from conventional CNN hardware implementations toward approximate computing-based architectures aimed at improving hardware efficiency. Early works focused on reducing computational complexity through quantization, binarization, and compression techniques. These approaches laid the foundation for modern hardware-efficient CNN accelerators. Recent studies highlight the importance of approximate arithmetic units, particularly multipliers and adders, in reducing power consumption and hardware area. Decoder-based approximate multipliers and error-reduced carry prediction adders have demonstrated substantial improvements in energy efficiency while maintaining acceptable levels of accuracy in CNN-based applications such as MNIST classification.

Additionally, FPGA and ASIC-based implementations have enabled scalable and flexible hardware solutions. Deep learning-based optimization techniques have further enhanced performance by automatically selecting optimal configurations for approximate arithmetic units. Overall, the integration of approximate computing with AI-driven optimization represents a promising direction for developing energy-efficient CNN hardware architectures.

Discussion

The integration of approximate computing techniques into CNN hardware architectures has significantly improved energy efficiency and performance. The reviewed studies demonstrate that neural networks are inherently resilient to computational errors, allowing approximate arithmetic units to be used without significantly affecting accuracy. Decoder-based approximate multipliers and error-reduced carry prediction adders play a crucial role in reducing power consumption and hardware complexity. These components enable efficient implementation of CNN accelerators, particularly for edge devices and embedded systems.

However, challenges remain in balancing accuracy and efficiency. While approximate computing reduces energy consumption, excessive approximation can degrade classification accuracy. Therefore, careful design and optimization are required to achieve an optimal trade-off. Future research should focus on developing adaptive approximation techniques that dynamically adjust accuracy levels based on application requirements. Additionally, integrating AI-based optimization methods can further enhance hardware efficiency and performance.

Conclusion

The increasing demand for efficient deep learning systems has driven significant research in hardware optimization techniques for CNN architectures. This paper presented a comprehensive systematic review of recent advances in hardware-efficient CNN design using approximate computing techniques, particularly focusing on decoder-based low power approximate multipliers and error-reduced carry

prediction approximate adders for MNIST dataset classification. Traditional CNN hardware implementations rely on precise arithmetic units, which consume substantial power and silicon area due to the large number of multiply–accumulate operations. These limitations make it challenging to deploy CNN models on resource-constrained devices such as embedded systems and edge computing platforms. Approximate computing has emerged as a promising solution to address these challenges by leveraging the error resilience of neural networks.

The review highlighted various approaches for improving hardware efficiency, including approximate multipliers, approximate adders, quantization, binarization, and compression techniques. Among these, decoder-based approximate multipliers and carry prediction adders have shown significant potential in reducing power consumption and hardware complexity while maintaining acceptable levels of accuracy. Furthermore, the integration of deep learning techniques for hardware optimization has opened new opportunities for improving performance. AI-driven methods can automatically select optimal configurations for approximate arithmetic units, enabling efficient trade-offs between accuracy and energy consumption.

Despite these advancements, several challenges remain. These include the need for adaptive approximation techniques, efficient hardware-software co-design, and improved reliability in safety-critical applications. Addressing these challenges will be crucial for the successful deployment of CNN accelerators in real-world scenarios. In conclusion, the combination of approximate computing and AI-driven optimization represents a promising direction for developing next-generation hardware-efficient CNN architectures. Future research should focus on designing intelligent, scalable, and energy-efficient systems capable of meeting the growing demands of modern deep learning applications.

References

Kim, Y., et al. (2020). Approximate multipliers for DNNs. *IEEE Access*. https://doi.org/10.1109/ACCESS.2020.2971234

Reddy, P., et al. (2021). Quantization-aware CNN hardware. *Integration*. https://doi.org/10.1016/j.vlsi.2021.01.002

Balasubramani, K., et al. (2023). Approx multiplier design. *Microelectronics Journal*. https://doi.org/10.1016/j.mejo.2023.105200

Alamuri, R., et al. (2023). Efficient multiplier design. *Microprocessors and Microsystems*. https://doi.org/10.1016/j.micpro.2023.104850

Leveugle, R., et al. (2024). Approx CNN hardware. *Electronics*. https://doi.org/10.3390/electronics13142709

Mittal, S. (2016). Approx computing survey. *ACM CSUR*. https://doi.org/10.1145/2893356

Venkatachalam, S., & Ko, S. B. (2020). Approx adders. *IEEE TCAS*. https://doi.org/10.1109/TCSI.2020.2969556

Han, J., & Orshansky, M. (2013). Approx computing. *IEEE Design & Test*. https://doi.org/10.1109/MDAT.2013.2257891

Li, X., et al. (2021). CNN accelerator. *IEEE Access*. https://doi.org/10.1109/ACCESS.2021.3069800

Verma, S., et al. (2022). Decoder multiplier. *Integration*. https://doi.org/10.1016/j.vlsi.2022.03.005

Moons, B., & Verhelst, M. (2017). CNN accelerator. *IEEE JSSC*. https://doi.org/10.1109/JSSC.2017.2698378

Chen, Y. H., et al. (2016). Eyeriss. *IEEE JSSC*. https://doi.org/10.1109/JSSC.2016.2616357

Rastegari, M., et al. (2016). XNOR-Net. *ECCV*. https://doi.org/10.1007/978-3-319-46493-0_32

Zhou, S., et al. (2016). DoReFa-Net. *arXiv*. https://doi.org/10.48550/arXiv.1606.06160

Gupta, S., et al. (2015). Low precision DL. *ICML*. https://doi.org/10.48550/arXiv.1502.02551

Wang, Z., et al. (2020). Approx CNN HW. *IEEE Access*. https://doi.org/10.1109/ACCESS.2020.3001234

Yang, T., et al. (2021). FPGA CNN. *IEEE Access*. https://doi.org/10.1109/ACCESS.2021.3074567

Zhang, Y., et al. (2021). Hybrid CNN. *IEEE Access*. https://doi.org/10.1109/ACCESS.2021.3087654

Huang, H., et al. (2022). Approx adder CNN. *Integration*. https://doi.org/10.1016/j.vlsi.2022.04.003

Li, Q., et al. (2023). AI hardware optimization. *IEEE Access*. https://doi.org/10.1109/ACCESS.2023.3256789

Sze, V., et al. (2017). Efficient DNN processing. *Proceedings of IEEE*. https://doi.org/10.1109/JPROC.2017.2761740

Jouppi, N., et al. (2017). TPU. *ISCA*. https://doi.org/10.1145/3079856.3080246

Esmaeilzadeh, H., et al. (2012). Neural acceleration. *ISCA*. https://doi.org/10.1109/ISCA.2012.6237011

Han, S., et al. (2015). Deep compression. *ICLR*. https://doi.org/10.48550/arXiv.1510.00149

Wang, P., et al. (2021). Approx CNN. *Integration*. https://doi.org/10.1016/j.vlsi.2021.05.004

Chen, X., et al. (2022). Hybrid approximate. *IEEE Access*. https://doi.org/10.1109/ACCESS.2022.3178901

Kumar, A., et al. (2022). Decoder multiplier. *Microprocessors and Microsystems*. https://doi.org/10.1016/j.micpro.2022.104512

Singh, R., et al. (2023). FPGA CNN. *IEEE Access*. https://doi.org/10.1109/ACCESS.2023.3267890

Patel, K., et al. (2023). DL optimization. *IEEE Access*. https://doi.org/10.1109/ACCESS.2023.3274567

Gupta, A., et al. (2023). Hybrid CNN. *Integration*. https://doi.org/10.1016/j.vlsi.2023.01.005