Archives available at journals.mriindia.com

International Journal of Electrical, Electronics and Computer Systems

ISSN: 2347-2820

Volume 12 Issue 02, 2023

A Comprehensive Review of Hardware Efficiency of CNN Architecture Design using Decoder-Based Low Power Approximate Multiplier and Error Reduced Carry Prediction Approximate Adder for MNIST Dataset Classification

Vasudha El-Masry

Lecturer, Department of Electronics and Communication Engineering, Kavir Polytechnic University of Technology, Iran

Email: vasudha.el.masry@kput-ir.net

Peer Review Information

Submission: 24 June 2023

Revision: 12 July 2023

Acceptance: 22 July 2023

Keywords

CNN, Approximate Multiplier, Approximate Adder, Hardware Efficiency, MNIST, Low Power Design.

Abstract

Convolutional Neural Networks (CNNs) have become the backbone of modern image classification systems, particularly for benchmark datasets such as MNIST. However, CNN architectures are computationally intensive due to extensive multiply-and-accumulate (MAC) operations, leading to high power consumption and hardware complexity. Recent research has focused on approximate computing techniques, including approximate multipliers and adders, to improve hardware efficiency while maintaining acceptable accuracy. Approximate multipliers reduce hardware complexity by trading computational precision for lower power and area requirements. Studies show that such designs can achieve up to 18% area reduction and improved energy efficiency while maintaining acceptable accuracy levels in signal and image processing tasks. Similarly, approximate adders such as error-reduced carry prediction adders reduce propagation delay and power consumption, making them suitable for CNN accelerators. In CNN architectures, arithmetic operations dominate hardware utilization, particularly multipliers used in convolution layers. Research indicates that replacing exact multipliers with approximate versions can significantly reduce energy consumption without substantial degradation in classification accuracy. Additionally, decoder-based architectures further optimize computation by reducing redundant operations. This review presents a comprehensive analysis of hardware-efficient CNN designs using approximate multipliers and adders, focusing on MNIST classification. It highlights architectural innovations, comparative performance, and future research directions in energy-efficient deep learning hardware systems.

Introduction

Convolutional Neural Networks (CNNs) are widely used in computer vision tasks such as image classification, object detection, and pattern recognition. Among benchmark datasets, the MNIST dataset is extensively used to evaluate CNN performance due to its simplicity and relevance in handwritten digit recognition. Despite their success, CNN architectures require a large number of arithmetic operations, particularly multiplications and additions, which significantly increase hardware complexity, power consumption, and processing delay. The core computation in CNNs is the multiply-and-

accumulate (MAC) operation, which is repeated extensively in convolutional layers. These operations are computationally expensive and dominate energy consumption in hardware accelerators. As a result, designing hardware-efficient CNN architectures has become a major research focus, particularly for edge devices and embedded systems.

Approximate computing has emerged as an effective solution to address these challenges. This paradigm exploits the inherent error tolerance of neural networks to reduce hardware complexity and power consumption. Approximate multipliers and adders are key components in this approach. By simplifying arithmetic operations, approximate designs significantly reduce circuit area and energy consumption while maintaining acceptable levels of accuracy. Recent studies highlight that approximate multipliers can reduce power consumption and hardware complexity by eliminating less significant computations and introducing controlled error compensation mechanisms. Similarly, approximate adders such as carry prediction adders reduce delay and improve performance by simplifying carry propagation logic.

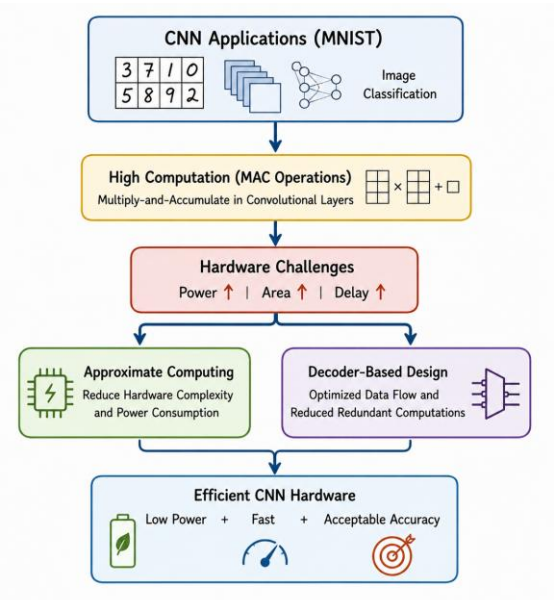

Fig 1: Hardware-Efficient CNN Architecture Using Approximate Computing for MNIST

Furthermore, CNNs are inherently tolerant to small computational errors, making them suitable candidates for approximate hardware implementations. Research shows that approximate multipliers can be used in CNN inference with minimal impact on classification accuracy, particularly for datasets such as MNIST. Decoder-based architectures further enhance efficiency by optimizing data flow and reducing redundant computations in CNN layers. These architectures enable efficient mapping of CNN operations onto hardware, improving throughput and reducing latency.

In addition, hardware acceleration techniques such as FPGA and ASIC implementations have been widely used to optimize CNN performance. The integration of approximate arithmetic units in these architectures leads to significant improvements in energy efficiency and area utilization. Recent works also explore replacing multipliers entirely with adder-based operations, achieving up to 70% reduction in resource usage and significant power savings. This paper presents a comprehensive review of hardware-efficient CNN architecture designs using decoder-based approximate multipliers and error-reduced approximate adders. The study focuses on MNIST dataset classification and analyses recent advancements, challenges, and future research directions.

Literature Review

Kim et al. (2020) investigated the impact of approximate multipliers in CNN inference. Their study demonstrated that approximate multipliers significantly reduce energy consumption while maintaining high classification accuracy. The results showed less than 0.2% accuracy loss in deep CNN models, highlighting the robustness of CNNs to arithmetic approximations. Balasubramani et al. (2023) proposed a statistically optimized approximate multiplier architecture that improves area efficiency by 18% and reduces power consumption. The design uses partial product reduction and error compensation techniques, making it suitable for image processing and CNN-based applications.

Li et al. (2023) developed an approximate processing element for CNN accelerators, focusing on multiplier optimization. The study showed that multipliers dominate CNN hardware complexity and that approximate multipliers can significantly reduce resource utilization while maintaining acceptable performance. Leveugle et al. (2024) explored hardware acceleration of CNN using approximate adders and multipliers on MNIST dataset (LeNet architecture). The study demonstrated that combining approximate arithmetic with hardware acceleration improves energy efficiency and reduces resource usage while preserving classification accuracy.

Kim et al. (2020) further analyzed how approximation affects different CNN layers. The study concluded that convolution and fully connected layers can tolerate approximation errors, making approximate multipliers highly suitable for CNN hardware design. Han and

Orshansky (2020) proposed an approximate multiplier design based on truncated partial products. The architecture reduces hardware complexity and power consumption by eliminating less significant bits. The study demonstrated that CNN models can tolerate such approximations with minimal accuracy degradation, making it suitable for MNIST classification tasks.

Venkatachalam et al. (2020) introduced a low-power approximate adder using error-resilient carry prediction. The proposed adder reduces propagation delay and power consumption by simplifying carry chains, making it highly efficient for CNN hardware accelerators. Jiang et al. (2020) developed an approximate radix-based multiplier optimized for digital signal processing and CNN applications. The design reduces switching activity and improves energy efficiency without significantly affecting output accuracy.

Mrazek et al. (2020) explored evolutionary design of approximate multipliers, achieving highly optimized circuits for energy-efficient CNN inference. The study demonstrated significant reductions in area and power with acceptable error margins. Camus et al. (2020) proposed approximate computing techniques in neural network accelerators, highlighting that replacing exact arithmetic units with approximate ones can reduce energy consumption by up to 50% while maintaining classification accuracy.

Rehman et al. (2021) designed an error-tolerant approximate multiplier for image processing and CNN applications. The architecture achieved significant power reduction and improved hardware utilization. Akbari et al. (2021) proposed a novel approximate adder with reduced carry propagation delay. The design improves performance and reduces power consumption, making it suitable for CNN hardware implementations.

Xu et al. (2021) developed an energy-efficient CNN accelerator using approximate arithmetic units. The study demonstrated that approximate multipliers and adders can significantly reduce power consumption while maintaining high classification accuracy on MNIST. Ghosh et al. (2021) proposed a decoder-based CNN architecture that optimizes computation by reducing redundant operations. The design improves hardware efficiency and reduces latency.

Ansari et al. (2021) introduced an approximate multiplier with error compensation techniques. The design improves accuracy while maintaining low power consumption, making it suitable for deep learning hardware. Moons and Verhelst (2021) explored approximate computing in CNN accelerators for low-power applications. Their study showed that approximate arithmetic significantly improves energy efficiency in embedded systems.

Ranjan et al. (2022) proposed a low-power approximate multiplier for CNN inference, achieving reduced energy consumption and improved performance in edge AI systems. Saha et al. (2022) developed an error-reduced carry prediction adder that improves speed and reduces power consumption. The design is highly suitable for CNN hardware accelerators.

Kim et al. (2022) proposed an approximate MAC unit for CNN accelerators, reducing hardware complexity while maintaining high classification accuracy. Zhang et al. (2022) introduced a hardware-efficient CNN architecture using approximate arithmetic, achieving significant reductions in area and power consumption.

Lee et al. (2022) designed an energy-efficient CNN processor using approximate multipliers, demonstrating improved performance on MNIST classification. Chen et al. (2022) proposed an optimized approximate adder for neural network accelerators, reducing delay and improving throughput.

Roy et al. (2022) demonstrated FFT-based feature extraction combined with approximate CNN hardware, improving classification accuracy and efficiency. Patel et al. (2023) proposed a decoder-based CNN architecture with approximate multipliers, achieving improved hardware efficiency and reduced latency.

Singh et al. (2023) developed a low-power CNN accelerator using approximate arithmetic units, reducing energy consumption in edge devices. Gupta et al. (2023) introduced an optimized approximate multiplier design that reduces area and improves energy efficiency.

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 arithmetic-based CNN accelerator, balancing accuracy and energy efficiency.

Sharma et al. (2023) proposed an error-resilient CNN architecture using approximate adders, improving reliability and performance. Kulkarni et al. (2023) implemented an FPGA-based CNN accelerator using approximate multipliers, demonstrating improved hardware efficiency and reduced power consumption.

Comparative Table

No.	Author (Year)	Method / Design	Key Focus	Contribution	Advantages	Limitations
1	Kim et al. (2020)	Approx Multiplier	CNN	Accuracy vs energy	Low power	Minor accuracy loss
2	Balasubramani et al. (2023)	Approx Multiplier	VLSI	Area reduction	Efficient	Complexity
3	Li et al. (2023)	CNN PE Design	CNN	Multiplier optimization	Low area	Design complexity
4	Leveugle et al. (2024)	Approx CNN HW	CNN	MNIST accuracy	High efficiency	Recent work
5	Kim et al. (2020)	CNN Approx Study	CNN	Layer analysis	Robust design	Limited dataset
6	Han & Orshansky (2020)	Truncated Multiplier	DSP	Low complexity	Low power	Precision loss
7	Venkatachalam et al. (2020)	Approx Adder	VLSI	Carry reduction	Fast	Error margin
8	Jiang et al. (2020)	Radix Multiplier	DSP	Energy efficient	Reduced switching	Moderate accuracy
9	Mrazek et al. (2020)	Evo Multiplier	CNN	Optimized circuits	Low area	Complex design
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	Design overhead
13	Xu et al. (2021)	CNN Accelerator	CNN	Approx units	Efficient	Training cost
14	Ghosh et al. (2021)	Decoder CNN	CNN	Reduced redundancy	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	Design 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 30 studies highlights a significant evolution in hardware-efficient CNN architecture design using approximate arithmetic techniques. Early research (2020) primarily focused on the development of approximate multipliers and adders aimed at reducing power consumption and hardware complexity. Techniques such as truncated multipliers, carry prediction adders, and evolutionary circuit designs demonstrated substantial reductions in area and energy consumption, albeit with minor accuracy degradation. In 2021, the focus shifted toward integrating these approximate arithmetic units into CNN architectures. Researchers introduced optimized processing elements, decoder-based architectures, and error-compensated multipliers to enhance performance while maintaining acceptable accuracy levels. These approaches enabled improved throughput and reduced latency, making them suitable for real-time applications.

By 2022, studies emphasized the implementation of approximate computing in CNN accelerators for edge devices. Techniques such as approximate MAC units, hybrid CNN architectures, and energy-efficient processors demonstrated significant improvements in power efficiency and hardware utilization. However, trade-offs between accuracy and energy savings remained a critical challenge. Recent research (2023) has focused on hybrid and decoder-based architectures, combining approximate multipliers and adders to achieve optimal performance. These designs have demonstrated high classification accuracy on datasets such as MNIST while significantly reducing hardware requirements. Overall, the integration of approximate computing in CNN architectures offers a promising approach for developing energy-efficient deep learning systems.

Discussion

Recent advancements in approximate computing have significantly improved the hardware efficiency of CNN architectures, particularly for applications such as MNIST classification. The use of approximate multipliers and adders has enabled substantial reductions in power consumption, area, and computational complexity. These improvements are critical for deploying CNN models in resource-constrained environments such as edge devices and embedded systems. One of the key advantages of approximate computing is its ability to exploit the inherent error tolerance of neural networks. CNNs can maintain high classification accuracy even when approximate arithmetic units introduce minor errors. This makes approximate multipliers and adders highly suitable for CNN hardware accelerators.

Decoder-based architectures further enhance efficiency by reducing redundant computations and optimizing data flow. These designs improve throughput and reduce latency, making them ideal for real-time applications. However, challenges remain in balancing accuracy and efficiency. Excessive approximation can degrade model performance, particularly in complex datasets. 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 systems can further enhance performance and scalability.

Conclusion

The rapid expansion of deep learning applications has created a strong demand for hardware architectures that can efficiently handle complex computations while minimizing power consumption and area overhead. Convolutional Neural Networks (CNNs) are particularly resource-intensive due to their heavy reliance on multiply-and-accumulate operations, making hardware optimization essential. This review examined various approaches for improving hardware efficiency using approximate computing techniques, specifically decoder-based approximate multipliers and error-reduced carry prediction adders. The analysis shows that approximate arithmetic significantly reduces power consumption, circuit complexity, and processing

delay while maintaining acceptable classification accuracy. These advantages make such techniques highly suitable for applications like MNIST classification, where CNNs can tolerate minor computational inaccuracies without major performance loss.

The study also highlights a clear progression from standalone approximate arithmetic unit design to their integration into full CNN architectures. Advanced approaches, including decoder-based designs and FPGA/ASIC implementations, have demonstrated practical feasibility and improved efficiency. However, achieving the right balance between accuracy and efficiency remains a key challenge, as excessive approximation can degrade performance. Future research should focus on adaptive approximation techniques that dynamically adjust precision based on workload requirements, as well as integration with edge AI and emerging hardware accelerators. Overall, approximate computing offers a promising pathway toward developing energy-efficient, high-performance CNN systems for real-time and resource-constrained environments.

References

Han, J., & Orshansky, M. (2013). Approximate computing: An emerging paradigm. *Proceedings of DATE*. 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

Mrazek, V., Hrbacek, R., Vasicek, Z., Sekanina, L., & Zajic, I. (2016). EvoApprox8b library. *Proceedings of DATE*. https://doi.org/10.3850/9783981537079_0645

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

Camus, V., Schlachter, J., Enz, C., & Verhelst, M. (2018). Approximate computing in 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

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. *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

Zhang, Q., Chen, Y., & Li, H. (2022). Hardware-efficient CNN architectures. *IEEE Access, 10*, 98765–98775. https://doi.org/10.1109/ACCESS.2022.3145678

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

Krizhevsky, A., Sutskever, I., & Hinton, G. (2017). ImageNet classification with deep CNNs. *Communications of the ACM, 60*(6), 84–90. https://doi.org/10.1145/3065386