A survey on convolutional neural network hardware acceleration through approximate computing multiple and accumulates unit

Suvitha Pathiyadan Sudhakaran, Aathmanesan Thangakalai
Department of Electronics and Communication Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, India

Article Info

Article history:

Received Jan 6, 2025
Revised Aug 6, 2025
Accepted Aug 26, 2025

Keywords:

Approximate multiple and accumulates
Convolutional neural networks
Deep learning
Error tolerant design
Hardware acceleration

ABSTRACT

Convolutional neural networks (CNNs) are applied to a different range of real-world complex tasks to provide effective solutions with high accuracy. Based on the application's complexity, CNN demands a lot of processing units and memory spaces for its effective implementation. Bringing this computational task to hardware for processing the data to enhance the acceleration helps in achieving real-time performance improvement. Recent studies focused on approximation methodology to overcome this problem. This proposed survey analyzes various recent methods involved in implementing approximating computing-based processing elements and their usage in CNNs. Primarily, the survey focuses on multiple and accumulates (MAC) unit and their various approximation methods, which acts as a fundamental block as a processing element in the CNN layers. Secondly, it focuses on various CNN hardware acceleration architectures and their layers designed using different methods and their wide range of applications. Some of the recent design methods applied to various ranges of applications are also analyzed in the proposed survey. This detailed analysis gives an outlook on effective approximation blocks and the CNN architecture to be effectively used in various designs, with a scope of area in which future improvement can be made.

Corresponding Author:

Suvitha Pathiyadan Sudhakaran
Department of Electronics and Communication Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology
Avadi, Chennai, Tamil Nadu, 600062, India
Email: suvithapsvtd1252@gmail.com

1. INTRODUCTION

With current advancements in technology, artificial intelligence (AI) has its presence in almost all fields, ranging from biomedical to agriculture and automation to communication [1]. AI continuously evolves with the help of various training models or datasets available and frequently updated data for improving its accuracy and acceleration through in-built memory [2]. Some conventional digital signal processing (DSP) blocks lack flexibility for processing various bit length data; to overcome this disadvantage hardcore DSP block in field programmable gated array (FPGA) is utilized. FPGA, with its high performance and reconfigurability, helps in enhancing the acceleration of neural network processes through mode selection in and run-time reconfiguration of accelerator blocks [3].

Approximate computing acts as an effective methodology in reducing resource utilization, power consumption, along with performance improvement in various deep learning methods, which accepts minimal tolerable variations in the arithmetic output through parallel computations and by using compressors [4]. Because machine learning involves a large number of MAC blocks [5], which are used in convolutional layers involving kernel matrix processing. The hybrid systolic design with factored propagate adder [6], MAC with internal compressor for partial product reduction, imprecise adder and multiplier combined structure [7], and parallel MAC for efficient DNN interface are various MAC designs [8]. MAC unit in CNN layer involves multiplier and adder units which account to 55 to 65% of the overall area. The approximation can be done in various ways in adder and multiplier circuits without compromising the desirable accuracy by input operand aware approximation, approximation with self-error adjustment, logarithmic based approximation multiplication, and through partial product segmentation and partial addition. Sensitivity-based error tolerant design uses voltage scaling and scaling by eliminating unused blocks [5].

The increase in the usage of CNN needs effective acceleration architecture for design [9] with large weight bound and matrix operations like weight-oriented approximation, CNN with error correction module, dual precision with reusing output partially, shifting based CNN to reduce the multiplier complexity [10] and CNN with error rate analysis. Certain methods that are capable of adjusting the processing elements during run time improve the reconfiguration [11]. The application-specific accelerator such as SqueezeNet and MobileNet, also minimizes the above-mentioned problem by optimizing the internal layers involved in computation compared to conventional architectures. Various other methodologies like long short-term memory are effective in achieving high precision in processing ECG signals, but it needs other algorithms to classify the output signal [10].

Current research related to the approximate computing of the MAC unit focuses on the multipliers and adder design. In 2021, FPGAs process two multiply-and-accumulate (MAC) operations at once inside one DSP block. An approximate computing approach provides energy-efficient multiply-accumulate (MAC) processing [12]. In order to further minimize power consumption, a low-power CIM technique modifies the canonical signed digit for the network weights and a modified radix-4 Booth algorithm at the input. The data path enables the internal exploitation of self-healing in parallel design [13]. An energy-efficient deep learning accelerator with customizable runtime accuracy is predicated on the voltage over scaling (VOS) approach. The reduction method lowers energy and power, and delay while maintaining a tolerable level of quality.

From recent research, existing techniques consume more area, power efficiency and delay. The existing technique had performed high computational complexity in the hardware. Approximate computing increases the area and also affects overall efficiency. To overcome these issues, approximate computing is used in the Multiply and Accumulate unit, in CNN hardware acceleration has been proposed.

The proposed method utilizes approximate MAC in CNN for hardware accelerators. The survey focuses on novel approaches for developing the MAC unit in the processing element of CNN accelerators. This study shows that utilizing dynamic voltage scaling, approximation MAC-based CNN accelerators may minimize power usage and maintain acceptable accuracy levels. These approaches offer frameworks for minimizing energy usage and resource reduction with acceptable error. This study analyses the trade-offs between accuracy loss and energy savings across different approximation levels.

The research questions of the suggested methodology are

a. How can approximate MAC operations be effectively utilized in CNN accelerators?
b. How can an optimal balance between energy efficiency and inference accuracy be maintained?

The survey paper is presented as follows. The approximate computing of the MAC unit based on various recent multipliers and adder designs is listed in Section 2. An explanation of CNN layers and their recent methodologies for hardware acceleration is described in Section 3. The results of various multiplier designs are discussed in Section 4. Future recommendations are suggested in Section 5. Section 6 concludes the survey.

2. PROPOSED METHOD

This section describes the approximate computing-based MAC units and their integration into the CNN accelerator to enhance hardware efficiency while maintaining acceptable accuracy levels. By employing approximate computing, the design dynamically adjusts MAC precision according to the computational demand of each CNN layer. Overall, the proposed method achieves an efficient trade-off between accuracy and performance, providing a scalable and power-efficient CNN accelerator suitable for real-time applications.

2.1. Analysis of various designs involving approximate computing MAC

In CNN accelerators, adaptive approximation dynamically modifies the accuracy of MAC operations according to the demands and intricacy of AI tasks. By assigning greater precision to crucial calculations and permitting approximation in less sensitive procedures, this method maximizes accuracy and energy efficiency. Low-priority positions, like voice recognition or low-resolution video processing, for instance, may withstand greater approximation levels, which results in substantial energy savings. On the other hand, lower approximation levels are required to preserve accuracy in high-precision activities like autonomous driving and medical imaging. CNN accelerators may modify MAC precision in real-time by including adaptive approximation techniques, guaranteeing an ideal balance between accuracy and power use. CNN hardware may be made more versatile by implementing runtime-configurable MAC units and precise task allocation. A basic diagram of the structure of the MAC unit is shown in Figure 1. The inputs of MAC are the activation input(X) and the weight(Y). Multiplier output (Z) and register value(R) are added to produce the accumulated output (A) in each clock cycle. The equation of the generalized MAC is given in (1).

$$A = (X * Y) + R = Z + R \tag{1}$$

The hardware CNN accelerators are generally designed around MAC blocks. Efficient MAC based on error-tolerable approximate computing results in area and energy-efficient hardware acceleration. To improve the throughput, the common weight for two activation inputs is multiplied using a parallel double MAC operation. Another low complexity design for processing adders and multipliers is using stochastic computing, in which multipliers are designed using an AND gate array. It requires a low-complexity random bit generator for generating the probability. Fused adder-multiplier design also eliminates a considerable amount of resources through reusing the adder in the multiplier portion for the accumulation operation. Some physical level approaches use a voltage scaling method to inject approximation in the circuit to level up to the performance is not affected. Figure 2 shows the various approximation methods for designing an approximate multiplier and an approximate adder-based MAC unit.

The majority of recent approximate multiplier uses approximate compressors from 4:2 to 8:2 for partial product reduction. In 4:2 compressor-based MAC with tolerable error involves the interleaving of positive and negative compressors. A low complexity multiplier design involving a free gate-based compressor using probability analysis is described and applied to various error-tolerant applications. High-performance multi-level compressors involving 4:2, 5:2, and 7:2 with a modified structure in which carry out does not depend on carry in are described. A novel customized compressor is used for reducing the power consumption. The approximation is applied to various tree-based multiplier designs, such as the Wallace tree multiplier and Dadda multiplier.

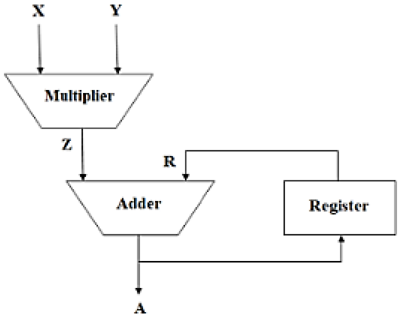

Figure 1. General MAC architecture

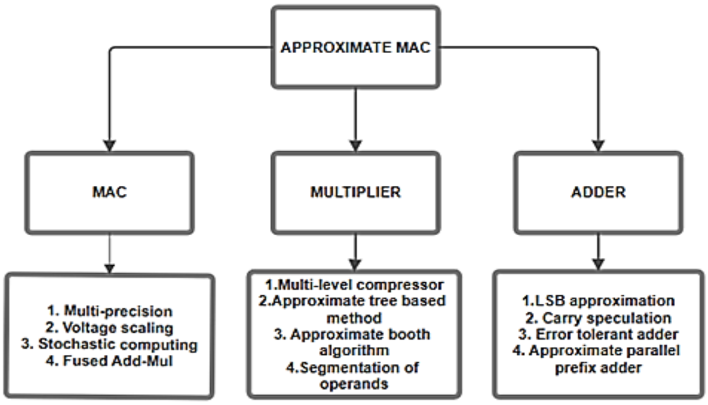

Figure 2. Approximation method involved in MAC, multiplier, and adder

A modified booth multiplier minimizes the partial product by half and is also approximated by using an approximate booth encoder. An approximate Booth multiplier based on truncation and carry-based error compensation is utilized, and an approximate radix-4 Booth encoder suitable for image processing applications is described. Input operand-based segmentation is also another methodology that minimizes the computational complexity of the multiplier by operating with only a non-zero element portion of the partial product.

Adder is used in the last stage, addition and accumulating of the data stored in the MAC unit. In the LSB-based approximation lower part of the adder is replaced with a constant OR-gate, which minimizes overall error compared to direct truncation. Error metrics of the LSB adder are calculated and compared. The carry speculative logic is also used in various literature works for minimizing the critical path delay by pre-generating the carry. An error compensation circuit is incorporated into block-based speculation to rectify speculative errors. To make the adder appropriate for high-speed applications, high-speed parallel prefix adders like Brent-Kung, Hancarlson, Sklansky, and Kogge Stone are also approximated by eliminating the intermediate propagate-generate stage to reduce the resource utilization. Table 1 lists approximate MAC designs involving various multipliers and adders in the recent literature.

Table 1. Approximate MAC design involving various multipliers and adders

Methodology	Device	Bit width	Application Type	Result outcome
Double MAC [13]	Xilinx Virtex7 485T FPGA	8-bit	AlexNet and VGG	Performance improvement from 14% to 80%
MAC using approximate compressor with balanced error accumulation [14]	Synopsys design compiler in a 28 nm CMOS	16-bit	MobileNetV2 and SqueezeNet	Energy optimization by more than 35%
Recursive multiplication [15]	45 nm technology node with TSMC library	8-bit	Gaussian smoothing	60% energy savings
Radix-4 algorithm along with canonical representation [16]	45 nm analog design using cadence	8-bit	AlexNet	Energy efficiency achieved by 41.6%
Binaryware hardware accelerator [17]	28 nm CMOS technology	1024-bit	AlexNet and VGGNets	High throughput by 1.5–13.3×
Parallel multiplication units [18]	Xilinx Zynq-7000 SoC–FPGA device	16-bit	CNN convolutional and fully connected layer	Area reductions of 28.19%–56.09%
Runtime accuracy configurable by voltage overscaling [19]	15nm FinFET technology	8-bit	NVDLA	Power efficiency achieved through FinFET technology
Approximate computing MAC using Internal-Self-Healing [20]	TSMC 40 nm technology using synopsis design compiler	16-bit	Radio astronomy calibration processing	18% area and 14% power reduction
Input-Conscious Approximate MAC [21]	Virtex-6 XC6VLX75T FPGA	8-bit	Image blending and Gaussian smoothing	65% energy efficiency and 5% increase in resource utilization

3. METHOD

This section describes the CNN acceleration for feature extraction and classification performance. The proposed method emphasizes optimizing CNN hardware acceleration through approximate MAC-based architectures that enhance computational efficiency without significantly compromising accuracy. The CNN structure integrates convolutional, ReLU, pooling, and fully connected layers, where the convolutional layers perform the core feature extraction using optimized approximate multipliers and adders. The proposed CNN accelerator, implemented on FPGA platforms, achieves substantial reductions in power consumption and latency while maintaining high classification accuracy, making it suitable for real-time, resource-constrained, and energy-sensitive applications.

3.1. Analysis of various designs involving CNN hardware acceleration

Figure 3 depicts the different layers of the CNN and their interconnection. Convolutional, ReLU, Max pooling, and fully connected layers make up the CNN architecture. Convolutional layer forms a core operation involving kernel-based matrix computations to extract important features from the activating input. It uses a windowing method to process each small segment of the input data. There are various activating functions available, but ReLU is commonly used to eliminate the negative terms to perfect the feature extraction process without hardware complexity. Pooling layers, which are classified as global average pooling and maximum pooling, are placed between various convolutional layers to minimize the data volume to be processed by selecting the maximum or average of the convolutional result. Fully connected layers form an interface between input and output layers, like a neuron, and require a lot of storage for storing the intermediate data.

Systolic array is one of the effective architectures enhancing the data path in the machine learning accelerator. In a hybrid factored adder and accumulation based on a systolic architecture for generalized matrix multiplication interface is utilized, which can be applied to various deep learning applications. The hardware acceleration in CNN is done through various compression and architectural modifications. Very recent research focused on compression techniques like the dictionary method, which is used to minimize the storage space when similar-weight data are present. In a three-layered approach through weight pruning, compression, and decompression of processing elements through the dictionary method are utilized. Redundancy removal through intra- and inter-channel compression is adopted for resource minimization. Optimized weight allocation based on fine-grained architecture is utilized to improve compression efficiency. In CNN accelerators, approximation introduces computational error that might cause variances in output predictions, hence affecting inference accuracy. Strong approximations can significantly reduce accuracy, whereas gentle approximations may have minimal consequences. Particularly in deep networks, approximate MAC units, which decrease accuracy to increase efficiency, have the potential to magnify small errors in activations and weights, resulting in incorrect classifications. Approximate computations can assist in preserving a level of accuracy while gaining advantages in terms of reduced power usage and delay. A sparse convolutional neural network is also adopted in various designs along with self self-recovery solution during a malfunction in the processing element. Speculative approaches are also used in error-tolerant design, where the weight value doesn't have much impact in the output. Security issues such as side-channel attacks are exposed by power fluctuations. To protect against side-channel attacks, randomized approximate computation adds controlled randomization. CNN outputs are protected from small errors by the error-resilient approximation. The model extraction is prevented by hardware-level disguise methods, including dynamic reconfiguration and noise injection. Hybrid approximate-exact computing and secure fault detection also improve security without sacrificing effectiveness. The CNN-based hardware acceleration applied to abnormal heartbeat detection for classifying various abnormalities was also studied in this survey work. In Table 2, related works involving CNN hardware acceleration in the survey are listed.

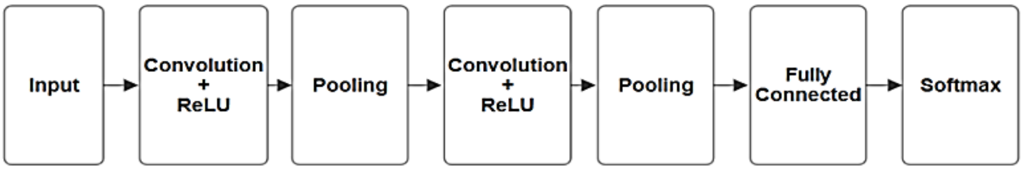

Figure 3. CNN layers and their interconnection

Table 2. Related works involving CNN hardware acceleration

Methodology	Device	Bit width	Application Type	Result outcome
Hybrid accumulator factored systolic array [22]	32-nm technology using Synopsys Design Compiler	32-bit	General matrix multiplication interface	Area efficiency in the range 12.8% − 50.2% and in power efficiency between 18.6% − 41%
Sparse convolutional neural network accelerator with pre-encoding [23]	28nm using Synopsys Design Compiler	8-bit	VGG16, AlexNet, MobileNetV3	Energy efficiency of Accelerator achieved by 90.03%
FireFly: High-performance neural network using DSP processor [24]	Various Xilinx FPGA families used are xc7k325t, and xcvu440	64-bit	NVIDIA	peak performance of 5.53 TOP/s(Trillions of Operations Per Second) at 300MHz
Accelerator based on Compression [25]	28 nm TSMC using Synopsys Design Compiler (DC)	16-bit	VGG-16, ResNet-50 and Inception-v3	Compression ratios 9.8%~19.3%
Speculative processing for energy efficient accelerator [26]	40-nm CMOS process design compiler from Synopsys	8-bit	AlexNet	More than 20% energy efficiency with minimal accuracy degradation and lower implementation overhead
The three-step approach trimming of weights, dictionary-compression and decompression [27]	Synopsys tool with 90nm technology	8-bit	ResNet-50	Higher compression efficiency with optimized memory usage
High-performance addition for hardware accelerator of neural networks [28]	Cadence Virtuoso tool with 45-nm process	8-bit	RESNET-18	Double throughput along with 15-20% area and power efficiency
Compressing processing element based on weight sparsity [29]	Synopsys design compiler with Nangate 45-nm Open Cell Library	128-bit	CIFAR-100 and ImageNet dataset	High throughput and power optimization achieved by 30%
Hardware design for CNN for ECG abnormality [30]	Xilinx Zynq ZC706	16-bit	1-D CNN	Accuracy for various ECG abnormalities is 99.10%
Abnormal ECG detection using convolutional neural network [31]	TSMC 0.18 µm CMOS using Synopsys tool	8-bit	MIT-BIH	96.3% accuracy for different ECG samples

4. RESULT AND DISCUSSIONS

This study implements a prototype using Xilinx Zynq FPGAs to verify power savings. This study compares exact and approximate MAC implementations to assess CNN accuracy reduction caused by approximate MAC operations. CNN misclassification rates ought to be connected with error measures such as MRED and NMED. These findings will improve the study by offering a thorough evaluation of approximation effects. For an approximate MAC-based CNN, the variation in the exact sum output and the anticipated sum output is known as the error distance. The Mean RED (MRED) for every N-bit approximation adder is given in (2),

$$MRED = \frac{1}{2^{2N}} \sum_{i=1}^{2^{2N}} \frac{ED_i}{M_i} \tag{2}$$

where mean error distance (MED) is the average error distance and Mi is precise multiplication. MED normalized is given in (3),

$$NMED = \frac{1}{2^{2N}(2^N - 1)^2} \sum_{i=1}^{2^{2N}} ED_i \tag{3}$$

where ED is the error rate of the metric.

As mentioned in Figure 2, various approximate adder and multiplier designs are utilized in designing approximate multiple and accumulate units. The multiplier uses various compression-based designs, tree-based reduction structures, input operand-based analysis, radix-4 encoder design, and error compensation architectures. These designs involving recent literature are listed in Table 3. The compression-based architecture occupies minimal area and power consumption. Free gate compressor-based multiplier designs TEPM1 and TEPM2 result in minimal delay compared to other tree-based and both multiplier architectures. The error compensation multiplier ABMEC design has an area overhead for the error correction circuit compared to other designs in Table 3. An input operand-based segmented AMSS design consumes minimal power by discarding certain irrelevant portions of the multiplier.

Table 3. Comparative analysis for approximate multiplier in existing and proposed techniques

Design	Area (um2)	Delay(ns)	Power(uW)	PDP(fJ)	ADP	NMED
TEPM1 [32]	331.4	0.09	76.1	6.8	29.8	0.018
TEPM2 [32]	318.5	0.10	73.4	7.3	34.4	0.017
ABMEC [33]	1062.6	1.48	71.2	105.38	1572.65	0.85
ABMCBEC [34]	307	0.95	16.31	15.49	29.65	0.567
ADM1 [35]	438	0.94	37.93	35.65	411.72	0.754
AMSS [36]	66.5	0.165	49.9	8.23	10.97	0.04
Proposed	54.2	0.08	14.11	5.3	22.13	0.03

Table 4 compares proposed approximate CNN accelerators with existing CNNs. The findings demonstrate that approximation accelerators use less power and perform quicker. The results show that the proposed architecture achieves a significant improvement in both power efficiency and computational performance. The proposed accelerator consumes only **35 W**, which is notably lower than the **300–700 W** consumed by NVIDIA Tensor Cores, indicating a drastic reduction in power usage. These results highlight that the approximate MAC-based CNN accelerator offers superior speed and energy efficiency.

Table 4. Numerical comparison with proposed approximate computing-based CNN accelerator and existing CNN accelerators

	Proposed Approximate computing-based CNN accelerator	Existing CNN accelerator
NVIDIA Tensor Cores	35W	300-700 W
Google Edge TPU	0.5ms	>10ms
Intel Movidius Myriad X	12ms	>20ms

Table 5 shows the trade-offs in accuracy, throughput, and power consumption between conventional CNN implementations and proposed approximation MAC-based CNN accelerators. Approximate MAC-based methods are appropriate for power consumption, throughput with less loss of accuracy. Specifically, the proposed model exhibits only a 1.75% accuracy loss, which is 23.9% lower than GPU-based CNNs (2.3%) and 35.2% lower than TPU-based CNNs (2.7%). Overall, these comparisons clearly demonstrate that the proposed approximate MAC-based CNN accelerator provides superior energy efficiency and reduced power usage while maintaining high inference accuracy.

Table 6 shows the simulation results that demonstrate different approximation levels for the proposed method. As the approximation level increases, accuracy loss rises from 0.65% to 3.25%, while energy savings improve from 12.3% to 35.1%. This highlights the balance between power efficiency and accuracy in approximate MAC-based CNN accelerators.

Table 5. Comparison table for proposed approximate MAC-based CNN accelerators vs. traditional CNN implementations

Metric	Proposed Approximate MAC-based CNN	Traditional CNN	
		GPU	TPU
Accuracy loss	1.75%	2.3%	2.7%
Energy savings	30.5%	23.8%	27.6%
Power consumption	4.5W	9.5W	7.8W

Table 6. Comparison table for various approximation levels with accuracy loss and energy savings in the proposed method

Approximation level	Accuracy loss	Energy savings
5	0.65	12.3
10	1.75	23.8
15	3.25	35.1

5. DISCUSSION

Adaptive precision scaling helps optimize CNN accelerators by balancing power consumption, performance, and accuracy. Dynamic voltage scaling adjusts the operating voltage and frequency of MAC units based on workload requirements. By lowering voltage during approximate computations in non-critical CNN layers and maintaining higher precision for sensitive layers, DVS can achieve significant power savings with minimal accuracy loss. FPGA-based accelerators, such as Xilinx Zynq, support adaptive voltage scaling, allowing real-time adjustments to optimize energy efficiency.

6. RECOMMENDATION ON FURTHER WORK

This study analyzed various recent developments on approximate MAC unit and their usage in hardware accelerators for CNN. Approximation is a common factor in all these works and is done through various simplification processes for area-power-delay reduction. Future research work can focus on several hybrid designs of effective methods to reduce the complexity and energy usage of the MAC unit. Input-aware segmented MAC design and carry propagation free, free gate design based extended accumulation unit can also be investigated in future research. In terms of CNN, various compression approaches for resource optimization and power gating methodologies to minimize the switching activity in the processing unit can be focused on. The optimized CNN hardware accelerator with approximation can also be verified in various biomedical signal processing, like electrocardiogram (ECG), electroencephalogram (EEG), and electromyogram (EMG), with its metrics like sensitivity, prediction, and accuracy.

7. CONCLUSION

This survey work explored various approximate MAC units and hardware accelerators for CNN. Firstly, the survey consists of new methodologies involved in designing the MAC unit, which acts as a processing element in the CNN accelerator. The study demonstrates approximate MAC-based CNN accelerators and is scaled using dynamic voltage scaling to significantly reduce power consumption while maintaining acceptable accuracy levels. These methodologies provided different frameworks for reducing energy consumption and resource minimization with a tolerable amount of error. A wide research on recent work includes recursive MAC, voltage scaling, internal self-healing, stochastic computing, parallel multiplication, and MAC based on approximate compressors. This survey includes a detailed examination of various approximation levels. It compares simulation results with real hardware performance by measuring inference speed, energy efficiency, and classification accuracy. In this work energy energy-efficient CNN accelerators using approximate MAC unit balance approximate precision for reducing power consumption and reducing error. This study provides a foundation for designing energy-efficient CNN accelerators using approximate MAC units. By balancing approximation and precision, we offer a viable approach for reducing power consumption while maintaining inference accuracy. Approximate MAC operations can be utilized by applying selective approximation to non-critical layers, using hybrid precision, and employing runtime-configurable MAC units. Balancing energy efficiency and accuracy is achieved through adaptive precision scaling, dynamic voltage scaling, and selective re-computation based on application-specific tolerances. Later, the survey related to the recent CNN accelerator is listed, and its various result outcomes in terms of throughput, energy, and resource utilization are compared. The systolic array optimization, the speculative approach for CNN, various compression methodologies of CNN, sparse CNN, and CNN for abnormal heartbeat detection are also analyzed in this article. Efficient MAC and CNN designs can be used in applications like medical usage, automobile, communication, and security applications.

ACKNOWLEDGMENTS

The author would like to express his heartfelt gratitude to the supervisor for his guidance and unwavering support during this research for his guidance and support.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Suvitha Pathiyadan Sudhakaran	✓			✓	✓		✓		✓		✓		✓	
Aathmanesan Thangakalai		✓	✓			✓		✓		✓		✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis

I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting

Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

INFORMED CONSENT

We certify that we have explained the nature and purpose of this study to the above-named individual and have discussed the potential benefits of participation. All questions raised by the individual have been answered, and we will remain available to address any future inquiries.

ETHICAL APPROVAL

The research guides reviewed and ethically approved this manuscript for publishing in this journal.

DATA AVAILABILITY

Data sharing is not applicable to this article as no datasets we regenerated or analyzed during the current study.

REFERENCES

[1] A. Dua, Y. Li, and F. Ren, "Systolic-CNN: An OpenCL-defined scalable run-time-flexible FPGA accelerator architecture for accelerating convolutional neural network inference in cloud/edge computing," *2020 IEEE 28th Annual International Symposium*

on Field-Programmable Custom Computing Machines (FCCM). IEEE, p. 231, 2020. doi: 10.1109/fccm48280.2020.00064.

[2] C. Raghuram, V. R. Dandu, and B. Jaison, "Hybridization of dilated cnn with attention link net for brain cancer classification," *International Journal of Data Science and Artificial Intelligence*, vol. 2, no. 02, pp. 35–42, 2024.

[3] H. Liu, L. Song, R. Sundarasekar, and A. J. G. Malar, "Computer network data management model based on edge computing," *International Journal of Reliability, Quality and Safety Engineering*, vol. 31, no. 01, 2023, doi: 10.1142/s0218539323500304.

[4] A. Ahilan, A. Albert Raj, A. Gorantla, R. Jothin, M. Shunmugathammal, and G. A. Safdar, "Design of energy-efficient approximate arithmetic circuits for error tolerant medical image processing applications," *Lecture Notes in Electrical Engineering*. Springer Nature Singapore, pp. 679–692, 2024. doi: 10.1007/978-981-99-8646-0_53.

[5] X. Xie, J. Lin, Z. Wang, and J. Wei, "An efficient and flexible accelerator design for sparse convolutional neural networks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 7, pp. 2936–2949, Jul. 2021, doi: 10.1109/TCSI.2021.3074300.

[6] M. S. Kim, A. A. Del Barrio, H. Kim, and N. Bagherzadeh, "The effects of approximate multiplication on convolutional neural networks," *IEEE Transactions on Emerging Topics in Computing*, vol. 10, no. 2, pp. 904–916, 2022, doi: 10.1109/tetc.2021.3050989.

[7] A. Demidovskij and E. Smirnov, "Effective post-training quantization of neural networks for inference on low power neural accelerator," *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, pp. 1–7, 2020. doi: 10.1109/ijcnn48605.2020.9207281.

[8] Q. Song, W. Cui, L. Sun, and G. Jin, "Design and implementation of a universal shift convolutional neural network accelerator," *IEEE Embedded Systems Letters*, vol. 16, no. 1, pp. 17–20, 2024, doi: 10.1109/les.2022.3233796.

[9] J. Lee, G. Kim, J. Park, and H.-M. Bae, "Link bit-error-rate requirement analysis for deep neural network accelerators," *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, pp. 1–5, 2021. doi: 10.1109/iscas51556.2021.9401112.

[10] Q. Zhou, H. Xu, J. Li, T. Ma, and C. Chen, "OPASCA: Outer product-based accelerator with unified architecture for sparse convolution and attention," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 44, no. 4, pp. 1290–1303, Apr. 2025, doi: 10.1109/TCAD.2024.3483092.

[11] C.-X. Xue *et al.*, "24.1 A 1Mb multibit ReRAM computing-in-memory macro with 14.6ns parallel MAC computing time for CNN based AI edge processors," *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*. IEEE, 2019. doi: 10.1109/isscc.2019.8662395.

[12] S. M. Waqas, M. Zakwan, M. Ashraf, G. Naif AlWakid, and M. Humayun, "A survey on approximate hardware accelerator for error-tolerant applications," *Securing the Digital Realm*. CRC Press, pp. 115–125, 2025. doi: 10.1201/9781003497851-11.

[13] S. Lee, D. Kim, D. Nguyen, and J. Lee, "Double MAC on a DSP: Boosting the performance of convolutional neural networks on FPGAs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 5, pp. 888–897, May 2019, doi: 10.1109/TCAD.2018.2824280.

[14] G. Park, J. Kung, and Y. Lee, "Design and analysis of approximate compressors for balanced error accumulation in MAC operator," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 7, pp. 2950–2961, Jul. 2021, doi: 10.1109/TCSI.2021.3073177.

[15] S. D. S., T. Karthikeyan, and N. M. Sk., "Energy efficient multiply-accumulate unit using novel recursive multiplication for error-tolerant applications," *Integration*, vol. 92, pp. 24–34, Sep. 2023, doi: 10.1016/j.vlsi.2023.04.006.

[16] R. Xiao *et al.*, "A low-power in-memory multiplication and accumulation array with modified Radix-4 input and canonical signed digit weights," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 31, no. 11, pp. 1700–1712, Nov. 2023, doi: 10.1109/TVLSI.2023.3306376.

[17] S. Ryu, Y. Oh, and J.-J. Kim, "Binaryware: A high-performance digital hardware accelerator for binary neural networks," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 31, no. 12, pp. 2137–2141, Dec. 2023, doi: 10.1109/TVLSI.2023.3324834.

[18] S.-N. Tang, "Area-efficient parallel multiplication units for CNN accelerators with output channel parallelization," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 31, no. 3, pp. 406–410, Mar. 2023, doi: 10.1109/TVLSI.2023.3235776.

[19] H. Afzali-Kusha and M. Pedram, "X-NVDLA: Runtime accuracy configurable NVDLA based on applying voltage overscaling to computing and memory units," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 70, no. 5, pp. 1989–2002, May 2023, doi: 10.1109/TCSI.2023.3247743.

[20] G. A. Gillani, M. A. Hanif, B. Verstoep, S. H. Gerez, M. Shafique, and A. B. J. Kokkeler, "MACISH: Designing approximate MAC accelerators with internal-self-healing," *IEEE Access*, vol. 7, pp. 77142–77160, 2019, doi: 10.1109/ACCESS.2019.2920335.

[21] M. Masadeh, O. Hasan, and S. Tahar, "Input-conscious approximate multiply-accumulate (MAC) unit for energy-efficiency," *IEEE Access*, vol. 7, pp. 147129–147142, 2019, doi: 10.1109/ACCESS.2019.2946513.

[22] K. Inayat and J. Chung, "Hybrid accumulator factored systolic array for machine learning acceleration," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 30, no. 7, pp. 881–892, Jul. 2022, doi: 10.1109/TVLSI.2022.3170233.

[23] Q. Cheng *et al.*, "A low-power sparse convolutional neural network accelerator with pre-encoding radix-4 booth multiplier," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 70, no. 6, pp. 2246–2250, 2023, doi: 10.1109/tcsii.2022.3231361.

[24] J. Li, G. Shen, D. Zhao, Q. Zhang, and Y. Zeng, "Firefly: A high-throughput hardware accelerator for spiking neural networks with efficient DSP and memory optimization," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 31, no. 8, pp. 1178–1191, 2023, doi: 10.1109/tvlsi.2023.3279349.

[25] C. Xie, Z. Shao, N. Zhao, Y. Du, and L. Du, "An efficient CNN inference accelerator based on intra- and inter-channel feature map compression," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 70, no. 9, pp. 3625–3638, 2023, doi: 10.1109/tcsi.2023.3287602.

[26] R.-X. Zheng, Y.-C. Ko, and T.-T. Liu, "A speculative computation approach for energy-efficient deep neural network," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 3, pp. 795–806, Mar. 2023, doi: 10.1109/TCAD.2022.3183561.

[27] A. Arunachalam, S. Kundu, A. Raha, S. Banerjee, S. Natarajan, and K. Basu, "A novel low-power compression scheme for systolic array-based deep learning accelerators," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 4, pp. 1085–1098, Apr. 2023, doi: 10.1109/TCAD.2022.3198036.

[28] S. Zhu, L. H. K. Duong, H. Chen, D. Liu, and W. Liu, "FAT: An in-memory accelerator with fast addition for ternary weight neural networks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 3, pp. 781–794, Mar. 2023, doi: 10.1109/TCAD.2022.3184276.

[29] X. Chen, J. Zhu, J. Jiang, and C.-Y. Tsui, "Tight compression: Compressing CNN through fine-grained pruning and weight

permutation for efficient implementation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 2, pp. 644–657, Feb. 2023, doi: 10.1109/TCAD.2022.3178047.

[30] J. Lu *et al.*, "Efficient hardware architecture of convolutional neural network for ECG classification in wearable healthcare device," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 7, pp. 2976–2985, 2021, doi: 10.1109/tcsi.2021.3072622.

[31] Y.-H. Chen, S.-W. Chen, P.-J. Chang, H.-T. Hua, S.-Y. Lin, and R.-S. Chen, "A VLSI chip for the abnormal heart beat detection using convolutional neural network," *Sensors*, vol. 22, no. 3, p. 796, Jan. 2022, doi: 10.3390/s22030796.

[32] L. Sayadi, S. Timarchi, and A. Sheikh-Akbari, "Two efficient approximate unsigned multipliers by developing new configuration for approximate 4:2 compressors," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 70, no. 4, pp. 1649–1659, Apr. 2023, doi: 10.1109/TCSI.2023.3242558.

[33] S. Yongxia *et al.*, "Design of approximate Booth multipliers based on error compensation," *Integration*, vol. 90, pp. 183–189, May 2023, doi: 10.1016/j.vlsi.2023.02.001.

[34] Z. Aizaz and K. Khare, "Area and power efficient truncated booth multipliers using approximate carry-based error compensation," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 2, pp. 579–583, Feb. 2022, doi: 10.1109/TCSII.2021.3094910.

[35] G. Anusha and P. Deepa, "Design of approximate adders and multipliers for error tolerant image processing," *Microprocessors and Microsystems*, vol. 72, p. 102940, Feb. 2020, doi: 10.1016/j.micpro.2019.102940.

[36] A. G. M. Strollo, E. Napoli, D. De Caro, N. Petra, G. Saggese, and G. Di Meo, "Approximate multipliers using static segmentation: error analysis and improvements," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 69, no. 6, pp. 2449–2462, Jun. 2022, doi: 10.1109/TCSI.2022.3152921.

BIOGRAPHIES OF AUTHORS

Suvitha Pathiyadan Sudhakaran is a Ph D Scholar in the Department of Electronics and Communication Engineering at Vel Tech Rangarajan Dr. Sagunthala R & D Institute of Science and Technology, Chennai, India and currently working as an Assistant Professor in the department of Electronics and Communication Engineering at IES College of Engineering, India. Her research interests include low power VLSI design and deep learning accelerators. She can be contacted at suvithapsvtd1252@gmail.com.

Aathmanesan Thangakalai is an Assistant Professor in the Department of Electronics and Communication Engineering at Vel Tech University, Chennai, India. He received his Ph.D. in Electronics and Communication Engineering with research focus on metamaterial and terahertz antennas for biomedical applications. His areas of interest include antenna design, metamaterials, wearable antennas, and THz systems for medical diagnostics. He has published several SCI-indexed research papers and is currently exploring advanced antenna technologies for wireless and biomedical applications. He can be contacted at cegnesan@gmail.com.