



Energy Efficient Spin-Based Implementation of Neuromorphic Functions in CNNs

SANDEEP SONI  (Graduate Student Member, IEEE), **GAURAV VERMA**  (Graduate Student Member, IEEE),
HEMKANT NEHETE (Graduate Student Member, IEEE),
AND BRAJESH KUMAR KAUSHIK  (Senior Member, IEEE)

Department of Electronics and Communication Engineering, Indian Institute of Technology Roorkee, Roorkee 247667, India

CORRESPONDING AUTHOR: BRAJESH KUMAR KAUSHIK (e-mail: bkk23fec@iitr.ac.in)

ABSTRACT Convolutional neural networks (CNNs) offer potentially a better accuracy alternative for conventional deep learning tasks. The hardware implementation of CNN functionalities with conventional CMOS based devices still lags in area and energy efficiency. This has necessitated the investigations of unconventional devices, circuits, and architectures to efficiently mimic the functionality of neurons and synapses for neuromorphic applications. Spin-orbit torque magnetic tunnel junction (SOT-MTJ) device is capable of achieving energy and area efficient rectified linear unit (ReLU) activation functionality. This work utilizes the SOT-MTJ based ReLU for activation and max-pooling in a single unit to eliminate the need of dedicated hardware for pooling layer. Moreover, 2×2 multiply-accumulate-activate-pool (MAAP) is implemented by using four activation pairs each of which is fed by the crossbar output. The presented approach has been used to implement various CNN architectures and evaluated for CIFAR-10 image classification. The number of read/write operations reduce significantly by 2X in MAAP based CNN architectures. The results show that the area and energy in MAAP based CNN is improved by at least 25% and 82.9%, respectively, when compared with conventional CNN designs.

INDEX TERMS Convolutional neural network (CNN), spintronics, spin-orbit torque (SOT), multiply-accumulate (MAC), magnetic tunnel junction (MTJ).

I. INTRODUCTION

The enormous development in the domain of artificial intelligence (AI) has led to the improvement in the performance of smart computing systems for image classification and similar tasks. The essential concept behind each of such high-end applications is driven by deep convolutional neural networks (CNNs) models that are growing computationally expensive [1]. This leads to the serious challenge of efficient computations in resource-constrained smartphones, health-care systems, and other edge devices. The need for the area and power-efficient computing systems has necessitated to explore unconventional computing architectures mainly inspired by the outstanding efficiency of the biological brain [2]. CNNs are one of the most effective architectures for computer-vision tasks including image recognition and data classification. The computational complexity of CNNs makes on-chip implementations expensive for resource-constrained

hardware. The CNNs consist of convolutional, pooling, and a fully connected layer including an activation function. Convolutional layers extract the important features from input datasets and their configuration depends on the type of application. Pooling is one of the most efficient features to reduce the computation complexity of CNNs [3]. The activation function is used to add nonlinearity and remove redundant data while keeping relevant features. Rectified linear units (ReLU) are the most widely used activation function due to its quick converging speed and reduce gradient vanishing problems during training [4]. ReLU is computationally effective as compared to sigmoid and tanh because all the neurons are not activated at the same time and the output is always zero for negative inputs. The fully connected layers are organized at the end of the architecture to flatten the features and classify the output. The concept is to limit the number of distinct computations in CNN by accumulating

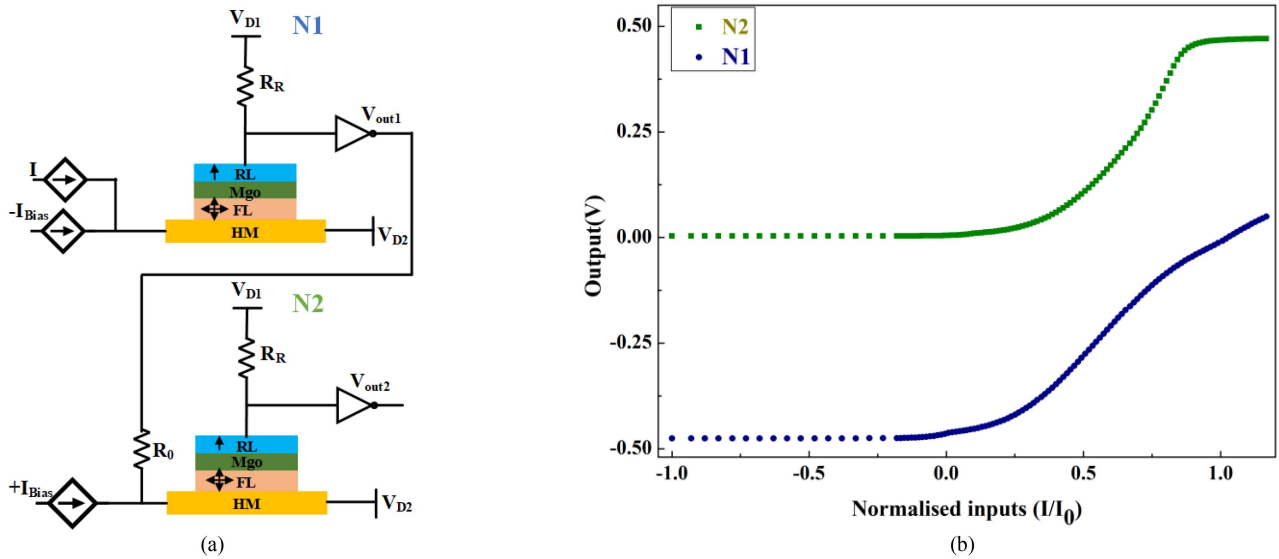


FIGURE 1. Activation pair structure (a) with two neuron structure (N1 and N2) cascaded each other and (b) the output characteristics of N1 and N2 with ± 1 bias shows the approximately mimic the ReLU functionality.

the convolution, activation, and pooling operations in a single step [5]. This also reduces the number of peripheral circuitry and related operations such as minimizing the memory and buffer requirements.

The implementation of CNN architectures requires specialized hardware with integrated caches and controllers for ReLU, sigmoid, and tanh along with maximum or average pooling. ReLU is the most preferable for CNNs and deep neural networks (DNNs) due to its unsaturated nonlinear functionality. Elbtity et al. [6] have previously investigated the spin-orbit torque magnetic random-access memory (SOT-MRAM) device as both binary synapses and a sigmoidal neuron. A heterogeneous mixed-precision and mixed-signal CPU-in-memory analog computation (IMAC) architecture to realize energy and performance improvements for the FC and convolution layers of CNN models has been presented in [6]. Similarly, Amin et al. [7] have presented an idea of using a voltage divider and inverter with spin devices to implement an area and power-efficient analog sigmoid neuron. However, the architecture-level analysis is limited only to the multilayer perceptron network. The implementation of analog based compute in-memory (CIM) is limited by area and power consumption of converters. Hence, digital counter parts are being explored for efficient neuromorphic hardware implementation. Wang et al. [8] have proposed a digital CIM-based architecture by using a toggle SOT-MRAM device to perform the computation entirely within the bit-cell array as opposed to a peripheral circuit. Furthermore, time domain CIM have been reported by Zhang et al. [9] using a highly reconfigurable array of field-free SOT-MRAM that can be applied to construct the convolutional neural network.

This work extends the implementation of spintronic-based ReLU and multiply-accumulate-activate-pool (MAAP) as reported in [10] to the complete architecture-level analysis of

CNN for image classification. The key contribution of this work are as follows:

- SOT-MTJ-based ReLU has been used to implement activation and max-pooling in a single unit (MAAP unit) to eliminate the need for dedicated hardware for the pooling layer.
- The presented MAAP approach has been used to implement various CNN architectures such as VGG-16, Lenet, and AlexNet, and evaluated for performance parameters such as area and energy consumption for CIFAR-10 image classification.
- In comparison to the conventional CNN architectures, the proposed design shows an improvement of 82.9% and 25% in energy and area consumption, respectively, for the CIFAR-10 image classification application.

II. DEVICE AND NEURON STRUCTURE

Neurons and synapses are the fundamental building elements of neuromorphic computing. The perpendicular SOT-MRAM devices are utilized to implement the neuron functionality. The perpendicular magnetic anisotropy device offers a significant advantage in terms of thermal stability, scaling, and power efficiency as compared to the in-plane device configuration [11], [12], [13], [14]. In SOT-MTJ device, MTJ is primarily composed of three layers: a reference layer (RL), a free layer (FL), and an oxide thin barrier. In a SOT-MTJ device, a SOT current is channelled through heavy metal (HM) to generate spin current [12]. As a result, a fieldlike and in-plane torque is induced that switch magnetization in the hard axis of FL. The charge current impacts both the voltage and the magnetization. The leakage power dissipation is minimal in it since the HM has a very low resistance relative to the R_{MTJ} , synaptic resistors, and load resistors. The pair of output inverters operating in a linear regime is used to read out

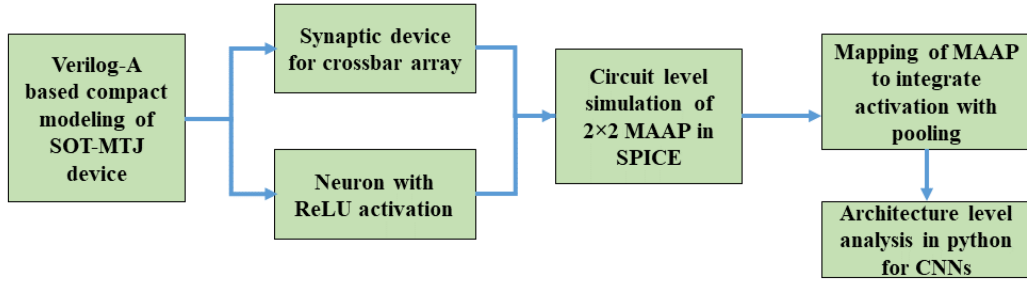

FIGURE 2. Framework of the presented work.

TABLE 1. Performance Parameters for Neuron Structure Using SOT-MTJ Device

Parameter	Description	Design value
t_{FL}	Free layer thickness (nm)	1
M_s	Magnetization (A/m)	10^6
K_u	Magnetic anisotropy (erg/cm ³)	7.06×10^6
B_h	Barrier height	0.4
D	Diameter of MTJ (nm)	70
$l \times b \times t$	HM layer (nm \times nm \times nm)	$200 \times 80 \times 3$
TMR	Tunnel magnetoresistance	150%
t_{ox}	Thickness of oxide layer (nm)	1.1
R_R	Reference resistance (k Ω)	10.5
ρ	Resistivity of HM layer (Ω m)	10^{-4}
V_{DD}	Supply voltage of inverter (V)	± 0.5
α	Damping constant	0.05
θ	Spin hall angle	0.3

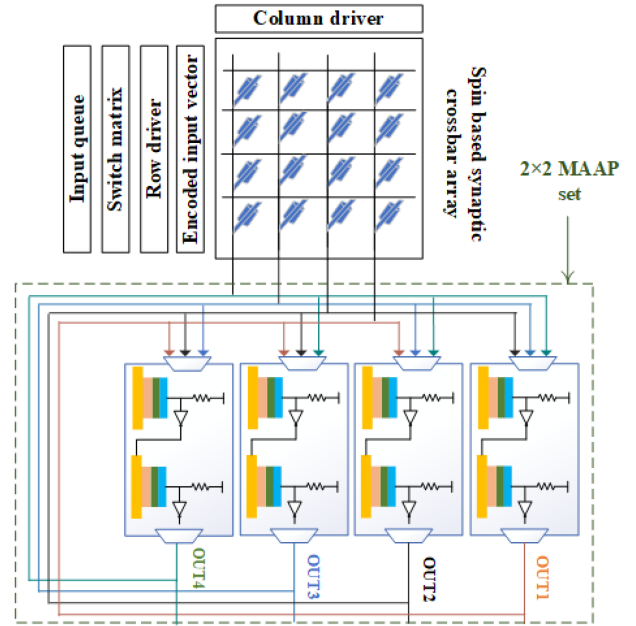
the output of the neuron. This neuron structure functionally behaves as a voltage divider circuit where reference resistance (R_R) is fixed and SOT-MTJ device is act as variable resistance (R_{MTJ}) as shown in Fig. 1(a). The offset current (I_0) is added to the input current that comes from crossbar array to compensate the difference between V_{D2} and ground, and the resulting accumulated current is then passed to the SOT-MTJ, that controls the output voltage. The two voltage divider circuits (N1 and N2) are cascaded with biasing of -1 and $+1$ to the first and second SOT-MTJ devices, respectively, to achieve ReLU neuron functionality as illustrated in Fig. 1(b). This structure has two SOT-MTJ devices with opposite polarity in the magnetization.

The magnetization dynamics of the FL can be modeled by the LLGS (1) as shown below [14]:

$$\frac{d\vec{m}}{dt} = -\gamma \left(\vec{m} \times \vec{H}_{eff} \right) + \alpha \left(\vec{m} \times \frac{d\vec{m}}{dt} \right) + \vec{\tau}_{DL} + \vec{\tau}_{FL} \quad (1)$$

$$\vec{H}_{eff} = \vec{H}_K + \vec{H}_{th} + \vec{H}_D \quad (2)$$

where, γ is a gyromagnetic ratio, α is a damping constant, H_{eff} is an effective magnetic field, τ_{DL} and τ_{FL} are damping and fieldlike torque, respectively. The effective field consists of uniaxial anisotropy field (H_K), thermal field (H_{th}), and demagnetization field (H_D). The device used in this work


FIGURE 3. Hardware implementation using MAAP integrated with spin-based crossbar array.

is optimized with respect to various parameters tabulated in Table 1. The critical charge current (I_{C0}) of SOT-MTJ device for neuron structure can be evaluated as [15]:

$$I_{C0} = \frac{2q\alpha}{\hbar} M_s t_{FM} \left(H_K + \frac{H_D}{2} \right) \cdot \frac{1}{\chi} \quad (3)$$

The notations q , \hbar , t_{FM} and M_s represent is the electron charge, reduced Planck's constants, FM layer thickness, and saturation magnetization, respectively. The ratio of spin current to charge current determines the spin injection efficiency of HM, which is denoted by χ .

The framework for the presented work is shown in Fig. 2. The device level and circuit level simulation has been performed in SPICE to extract the parameters such as area, power, and latency. Further, a python code has been implemented for mapping the device and circuit parameters for complete CNN. The architecture level analysis

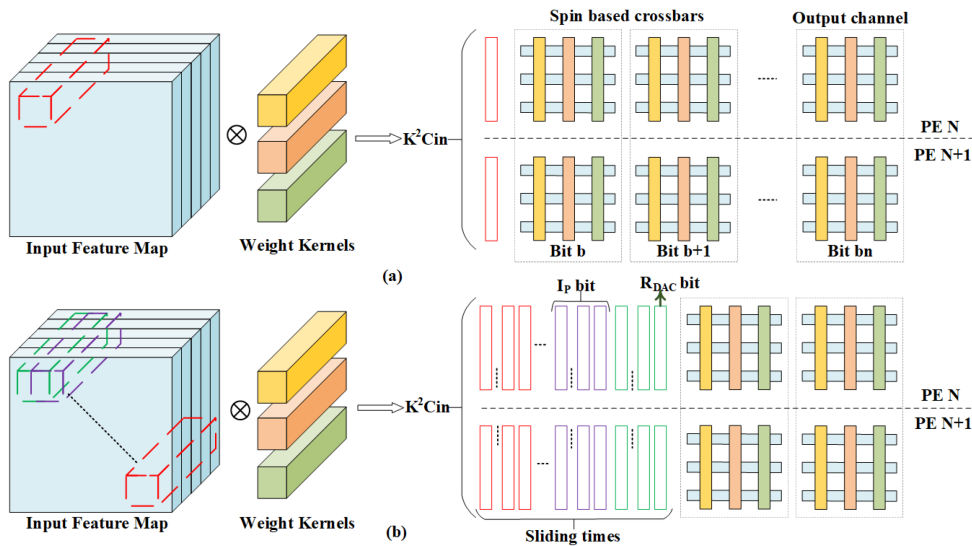


FIGURE 4. Mapping of (a) CNN kernels and (b) input feature map on spin based crossbars. K = Kernel size, Cin = Input channels, bn = Weight precision, I_p = Input precision, R_{DAC} = DAC resolution.

using spin-based MAAP set is presented in the subsequent section.

III. MAAP BASED CNN ARCHITECTURE

Pooling functionality is one necessary operation involved in neuromorphic computing that takes place usually after activation [16]. Max-pooling operation gives a single output from a set of input values depending on the highest value after non-linear activation. Hence, to eliminate separate hardware for the implementation of the pooling function, the integrated MAAP functionality is presented with the SOT-MTJ devices [10]. The activation pairs are associated with inputs from a spin-based crossbar array that formed the basic component of the MAAP set. A spin-based synaptic crossbar array for multiply-accumulate (MAC) operation that is connected to spin-based MAAP units to implement the neuron activation and pooling functionality is shown in Fig. 3. The data (weight matrix) is mapped to the synaptic device conductance states of the compute in-memory (CIM) crossbar array organized into multiple crossbars. Each crossbar array is integrated with rows and column peripherals to perform computations on encoded input data vectors. The computation in such crossbar array architectures is inherently parallelized for matrix-vector computations of machine-learning algorithms. In the crossbar, MAC operation are implemented in a single step by activating all the columns of the array resulting in parallel vector-matrix-multiplication (VMM) of encoded input vectors with weight matrix. This property achieves significant enhancements in computational volume and power efficiency of neuromorphic hardware.

The conventional memristor-based CIM DNN accelerators utilize the concept of analog MAC operation using synaptic devices. The 3D convolution kernels used in online training are flattened into 1D column vectors and mapped onto devices in one column of crossbars as shown in Fig. 4. The

feature extracting kernels of one convolution layer are placed in different columns of the same crossbar. In this work, the precision of spin based synaptic devices is limited to single bit. As a result, the convolution kernels from two dimensions are split to complete the mapping. Since, the weight precision is higher than the number of synaptic resistance levels, several crossbars in one processing element (PE) are used to store bits of convolution kernels. The synaptic device used in this work is SOT-MRAM with storage of one bit per synaptic cell. Hence, for b -bit weight precision, b crossbars in one PE are required to implement mixed precision CNN. In the crossbar, the 1D input vector is encoded as voltage pulse signal using DACs onto word lines. Similarly, input feature maps for each sliding window is implemented sequentially in multiple crossbars during computation cycles.

The implementation of 2×2 MAAP sets connected by inhibitory feedback and the spin-based crossbar integrated in the CNN architecture is shown in Fig. 5. The crossbar is controlled by row and column driver to read and write operation of synaptic devices for weight updation. The input vector is encoded in form of voltage pulses that are fed using input queue, and the row/column drivers activate the necessary bit lines/word lines for MAC operation. The resultant of weighted sum from the crossbar is fed into MAAP circuit. This includes the activation as well as maximum pooling functionality for the respective layer output. Each input line from the crossbar array includes a contribution from the twice-inverted output of every other activation pair. The winner-take-all implementation allows the largest input to push the remaining activation pairs to zero. Hence, the maximum pooling computation is also integrated along with MAC operations.

The MAAP unit at architecture level is connected with a buffer, shared storage and controller. The crossbar arrays are connected to ADC / DACs, row-column decoders to form a PE. Multiple PEs sharing a common PE buffer are controlled

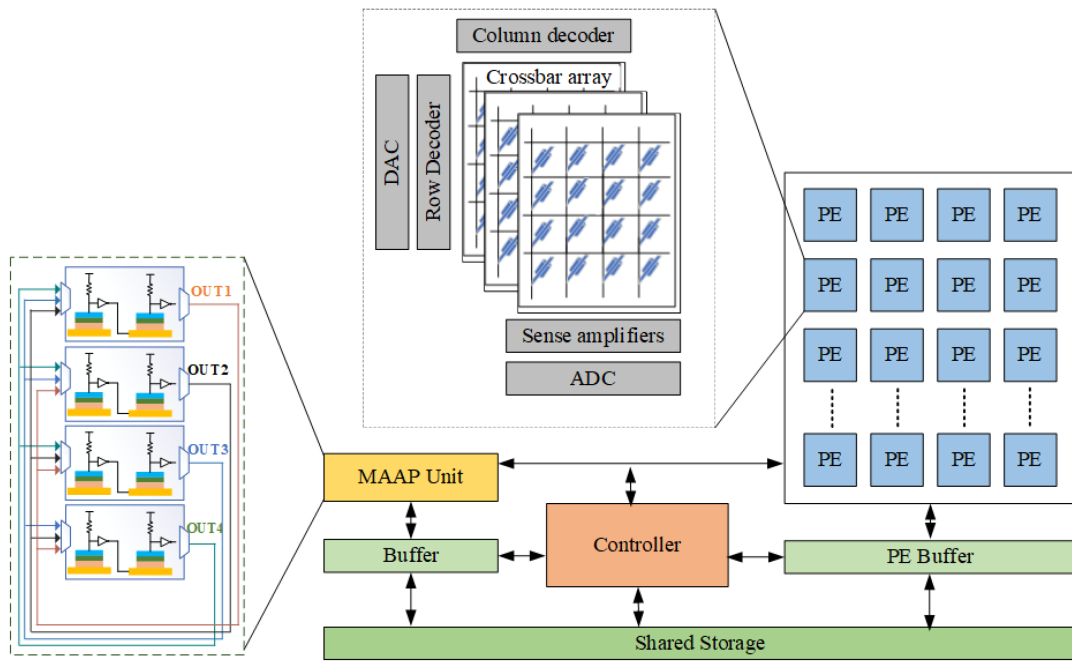


FIGURE 5. The 2×2 MAAP set is utilized for pooling in the proposed MAAP-based CNN architecture.

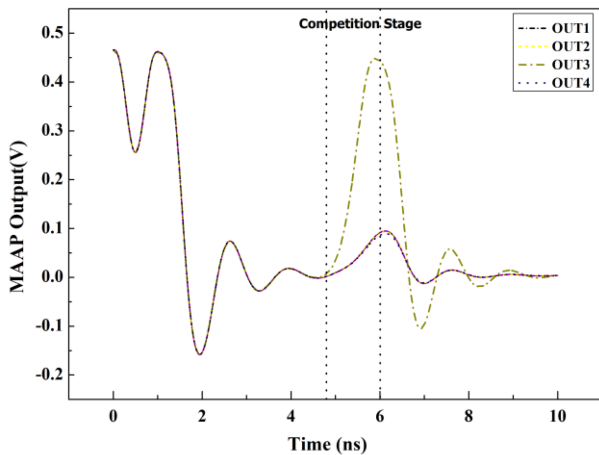


FIGURE 6. The winner-take-all competition is demonstrated by showing the time versus output voltage of each activation pair in a 2×2 MAAP set, where only one of the four activation pairs can be triggered at a time with the other three remaining at almost zero state.

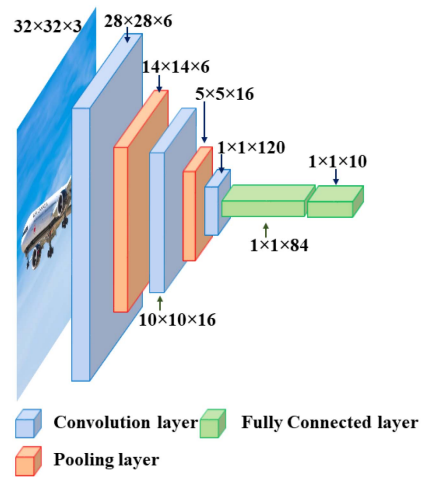


FIGURE 7. CNN architecture of Lenet for CIFAR-10 image classification.

through the controller to implement complete CNN functionality at hardware level. The parallel read-out computation is performed column wise along read word-lines. The current output flowing through the synaptic devices gives the vector product of input voltage vector and the synaptic conductance matrix. Parallel read-write lines are simultaneously activated by encoded voltage pulses from pre-neurons and the output weighted summation current accumulates at the end of bit-lines [4], [17]. The resultant weighted sum current is received by the proposed neuronal device to implement activation and pooling function. The SPICE simulation of four activation pairs is performed with an optimized input in the range of

$[-1, 1]$. The normalized result is determined by summing all four output voltages and equals the rectified value of maximum among the four external normalized inputs as shown in Fig. 6. The response of all the outputs nearly overlap each other (OUT1, OUT2, OUT4) that is dominated by the winner output (OUT3) of MAAP set. The winner-take-all competition results in the activation of largest input activation pair during competition duration and the remaining activation pairs are deactivated due to inhibitory feedback coming from the remaining activation pairs. After competition stage all the activation pairs are stabilised to zero.

The energy consumption and simulation time needed for complete MAAP computation is optimized by parameters

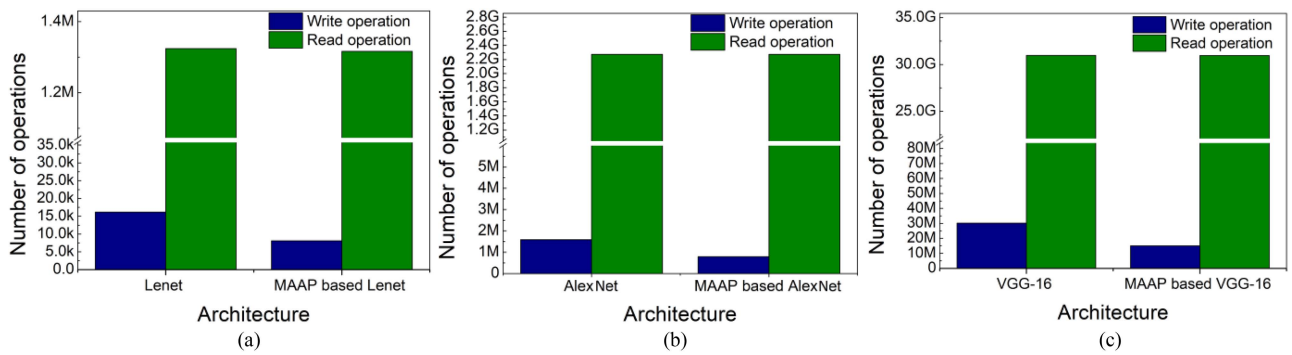


FIGURE 8. The read and write operation of CNN with (a) Lenet, (b) AlexNet and (c) VGG-16 architectures, as well as a comparison to proposed MAAP-based CNN architectures.

TABLE 2. Performance Comparison of Proposed MAAP Based CNN Architecture With Conventional CNN Architecture for CIFAR-10 Classification

Performance parameters	Lenet [18]	Proposed MAAP-based Lenet	AlexNet [18]	Proposed MAAP-based AlexNet	VGG-16 [18]	Proposed MAAP-based VGG-16
Total area (mm ²)	4.68	3.51	55.57	50.13	280.08	254.44
Pooling area (mm ²)	0.735	0.551	8.73	3.13	44.12	23.20
Pooling energy (nJ)	68.11	11.64	39.44	7.01	147.93	23.41
Average latency (ms)	0.31	0.22	0.75	0.54	4.78	4.50
Accuracy (%)	72.4	69.83	84.45	82.03	92.81	91.23

of spin devices such as size of the FM layer, MTJ resistance, critical current. With the advanced developments in MTJ based devices and circuits for neuromorphic computing, this work presents novel architecture that integrates the activation-pooling, and synaptic performance for neuromorphic functions simultaneously using spintronic devices. The spin-based synapses are arranged in a crossbar to implement the MAC computation for CNNs. Lenet architecture has been implemented using MAAP unit for CIFAR-10 dataset classification as shown in Fig. 7. The three architectures are chosen with different number of pooling layers used in order to highlight the benefit of proposed MAAP functionality [18]. The integration of activation and pooling in single circuit provides advantage of better resource efficiency at hardware level for networks with more pooling layers.

IV. RESULTS AND DISCUSSIONS

The proposed spin based neuromorphic circuits are implemented for image classification tasks using distinct architectures. The device and circuit level performance are simulated using the SPICE framework with 45nm CMOS technology node. The energy performance of MAAP set is calculated as the sum of neuron structures, inverters and crossbar array energy consumption. The average energy consumption of 2 × 2 MAAP set is nearly to 0.5–4 pJ with the variation of device parameters that specify the significant robustness in the energy performance. The modified MNSIM framework is used for implementation of CNN architectures to extract the performance parameter such as accuracy, area, energy and latency [19], [20]. Firstly, a Lenet CNN architecture shown in

Fig. 7. with convolution-activation-pooling layer arrangement is implemented with conventional CMOS and proposed device. The results for image classification on CIFAR-10 dataset are compared in Table 2 which shows that proposed approach achieves 47.41% and 84.17% area and energy efficiency respectively. The significance of proposed approach in terms of resource utilization is achieved with negligible loss in classification accuracy.

The convolution, activation, and pooling computations in the CNN architecture require a significant read and write operation, that increases the energy and latency required for the image classification task. The integrated MAAP operation in place of activation and pooling reduces a significant read and write operation, that lowers the energy and delay needed for the image classification task for deep CNNs. Fig. 8 plots the results for reduction in number of read-write operations with MAAP based approach. The number of write operations reduce by 2X for all the three network architectures. The read operations lower by 2.235×10^6 for MAAP based AlexNet and 1.2×10^6 for MAAP based VGG-16 architectures.

V. CONCLUSION

The proposed SOT-MTJ based CNN architecture integrates multiple neuromorphic functions simultaneously for AI applications. This demonstrates remarkable efficiency of the ReLU activation integrated with max pooling to form MAAP structure that significantly lowers area and energy consumption. The presented work is suitable for resource-constrained hardware implementation of more complex NN architectures.

REFERENCES

- [1] Y. Luo and S. Yu, "Accelerating deep neural network in-situ training with non-volatile and volatile memory based hybrid precision synapses," *IEEE Trans. Comput.*, vol. 69, no. 8, pp. 1113–1127, Aug. 2020, doi: [10.1109/TC.2020.3000218](https://doi.org/10.1109/TC.2020.3000218).
- [2] A. Sengupta and K. Roy, "Encoding neural and synaptic functionalities in electron spin: A pathway to efficient neuromorphic computing," *Appl. Phys. Rev.*, vol. 4, no. 4, 2017, Art. no. 041105, doi: [10.1063/1.5012763](https://doi.org/10.1063/1.5012763).
- [3] A. Sayal, S. Fathima, S. T. Nibhanupudi, and J. P. Kulkarni, "COMPAC: Compressed time-domain, pooling-aware convolution CNN engine with reduced data movement for energy-efficient AI computing," *IEEE J. Solid-State Circuits*, vol. 56, no. 7, pp. 2205–2220, Jul. 2021, doi: [10.1109/JSSC.2020.3041502](https://doi.org/10.1109/JSSC.2020.3041502).
- [4] G. Lin and W. Shen, "Research on convolutional neural network based on improved Relu piecewise activation function," *Procedia Comput. Sci.*, vol. 131, pp. 977–984, 2018, doi: [10.1016/j.procs.2018.04.239](https://doi.org/10.1016/j.procs.2018.04.239).
- [5] M. Yildirim, "Analog circuit architecture for max and min pooling methods on image," *Analog Integr. Circuits Signal Process.*, vol. 108, no. 1, pp. 119–124, 2021, doi: [10.1007/s10470-021-01842-x](https://doi.org/10.1007/s10470-021-01842-x).
- [6] M. Elbtity, A. Singh, B. Reidy, X. Guo, and R. Zand, "An in-memory analog computing co-processor for energy-efficient CNN inference on mobile devices," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI*, 2021, pp. 188–193, doi: [10.1109/ISVLSI51109.2021.00043](https://doi.org/10.1109/ISVLSI51109.2021.00043).
- [7] M. H. Amin, M. Elbtity, M. Mohammadi, and R. Zand, "MRAM-based analog sigmoid function for in-memory computing," in *Proc. ACM Great Lakes Symp. VLSI*, 2022, pp. 319–323, doi: [10.1145/3526241.3530376](https://doi.org/10.1145/3526241.3530376).
- [8] J. Wang et al., "Reconfigurable bit-serial operation using toggle SOT-MRAM for high-performance computing in memory architecture," *IEEE Trans. Circuits Syst. I: Regular Papers*, vol. 69, no. 11, pp. 4535–4545, Nov. 2022, doi: [10.1109/TCSI.2022.3192165](https://doi.org/10.1109/TCSI.2022.3192165).
- [9] Y. Zhang et al., "Time-domain computing in memory using spintronics for energy-efficient convolutional neural network," *IEEE Trans. Circuits Syst. I: Regular Papers*, vol. 68, no. 3, pp. 1193–1205, Mar. 2021, doi: [10.1109/TCSI.2021.3055830](https://doi.org/10.1109/TCSI.2021.3055830).
- [10] A. W. Stephan and S. J. Koester, "Spin Hall MTJ devices for advanced neuromorphic functions," *IEEE Trans. Electron Devices*, vol. 67, no. 2, pp. 487–492, Feb. 2020, doi: [10.1109/TED.2019.2959732](https://doi.org/10.1109/TED.2019.2959732).
- [11] I. Ahmed, Z. Zhao, M. G. Mankalale, S. S. Sapatnekar, J. P. Wang, and C. H. Kim, "A comparative study between spin-transfer-torque and spin-Hall-effect switching mechanisms in PMTJ using SPICE," *IEEE J. Explor. Solid-State Comput. Devices Circuits*, vol. 3, pp. 74–82, Dec. 2017, doi: [10.1109/JXCDC.2017.2762699](https://doi.org/10.1109/JXCDC.2017.2762699).
- [12] M. Kazemi, G. E. Rowlands, E. Ipek, R. A. Buhrman, and E. G. Friedman, "Compact model for spin-orbit magnetic tunnel junctions," *IEEE Trans. Electron Devices*, vol. 63, no. 2, pp. 848–855, Feb. 2016, doi: [10.1109/TED.2015.2510543](https://doi.org/10.1109/TED.2015.2510543).
- [13] A. Nisar, S. Dhull, S. Mittal, and B. K. Kaushik, "SOT and STT-based 4-bit MRAM cell for high-density memory applications," *IEEE Trans. Electron Devices*, vol. 68, no. 9, pp. 4384–4390, Sep. 2021, doi: [10.1109/TED.2021.3097294](https://doi.org/10.1109/TED.2021.3097294).
- [14] S. Wasef and H. Fariborzi, "Spin-Based voltage comparator using spin-hall effect driven nanomagnets," *AIP Adv.*, vol. 10, no. 3, 2020, Art. no. 035116, doi: [10.1063/1.5130491](https://doi.org/10.1063/1.5130491).
- [15] W. H. Butler et al., "Switching distributions for perpendicular spin-torque devices within the macrospin approximation," *IEEE Trans. Magn.*, vol. 48, no. 12, pp. 4684–4700, Dec. 2012, doi: [10.1109/TMAG.2012.2209122](https://doi.org/10.1109/TMAG.2012.2209122).
- [16] Q. Lou, C. Pan, J. McGuinness, A. Horvath, A. Naemi, and X. S. Hu, "A mixed signal architecture for convolutional neural networks," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 15, no. 2, pp. 1–26, 2019, doi: [10.1145/3304110](https://doi.org/10.1145/3304110).
- [17] D. Bankman, L. Yang, B. Moons, M. Verhelst, and B. Murmann, "An always-on 3.8 $\mu\text{J}/86\%$ CIFAR-10 mixed-signal binary CNN processor with all memory on chip in 28-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 54, no. 1, pp. 158–172, Jan. 2019, doi: [10.1109/JSSC.2018.2869150](https://doi.org/10.1109/JSSC.2018.2869150).
- [18] Y. Ling et al., "A RRAM based max-pooling scheme for convolutional neural network," in *Proc. IEEE 5th Electron Devices Technol. Manuf. Conf.*, 2021, pp. 1–3, doi: [10.1109/EDTM50988.2021.9421061](https://doi.org/10.1109/EDTM50988.2021.9421061).
- [19] L. Xia et al., "MNSIM: Simulation platform for memristor-based neuromorphic computing system," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 37, no. 5, pp. 1009–1022, May 2018, doi: [10.1109/TCAD.2017.2729466](https://doi.org/10.1109/TCAD.2017.2729466).
- [20] Z. Zhu et al., "MNSIM 2.0: A behavior-level modeling tool for memristor-based neuromorphic computing systems," in *Proc. ACM Great Lakes Symp. VLSI*, 2020, pp. 83–88, doi: [10.1145/3386263.3407647](https://doi.org/10.1145/3386263.3407647).