Energy Harvesting-assisted Ultra-Low-Power Processing-in-Memory Accelerator for ML Applications

Sanket Shukla
George Mason University
FairFax, Virginia, USA
sshukla4@gmu.edu

Sathwika Bavikadi
George Mason University
FairFax, Virginia, USA
sbavikad@gmu.edu

Sai Manoj Pudukotai Dinakarrao
George Mason University
FairFax, Virginia, USA
spudukot@gmu.edu

ABSTRACT

The proliferation of Internet of Things (IoT) and edge computing devices has become an essential aspect of our daily routines. Particularly, the rise of wearable technology like smartwatches, health trackers, and smart glasses has contributed significantly to their popularity. These gadgets are equipped with diverse sensors that enable researchers and manufacturers to collect user data. Subsequently, this data undergoes processing through on-device Machine Learning (ML) algorithms, enhancing user interactions. However, implementing ML algorithms on these compact IoTs and edge devices consumes substantial power and energy. It's crucial to recognize that these devices operate within strict energy and power constraints. Thus, optimizing battery usage is paramount for prolonging a device's lifespan. Therefore, we propose a Processing-In-Memory (PIM) architecture utilizing Look-up-Table (LUT) based processing for improved performance and energy efficiency. To further enhance energy efficiency in this work we introduce a framework that efficiently utilizes kinetic energy harvesting to intermittently support ML computations/tasks, thereby alleviating the load on the device's built-in battery. By offloading ML computations to the PIM architecture, the framework reduces the reliance on the device's internal battery power, optimizing the use of harvested kinetic energy and extending battery life. Furthermore, PIM architecture facilitates seamless integration of harvested kinetic energy, ensuring efficient ML computations with minimal energy consumption. This integrated approach presents a compelling solution for energy management in IoT and edge-based applications, as evidenced by experiments and analysis showing significant reductions in overall energy usage. We evaluated the proposed Energy Harvesting-assisted PIM architecture on various CNN architectures, such as LeNet, AlexNet, ResNet -18, -34, -50.

KEYWORDS

Energy Harvesting, IoT, Processing in Memory, Neural Networks, Low-Power

ACM Reference Format:
Sanket Shukla, Sathwika Bavikadi, and Sai Manoj Pudukotai Dinakarrao. 2024. Energy Harvesting-assisted Ultra-Low-Power Processing-in-Memory Accelerator for ML Applications. In *Great Lakes Symposium on VLSI 2024 (GLSVLSI '24), June 12–14, 2024, Clearwater, FL, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3649476.3660392

GLSVLSI '24, June 12–14, 2024, Clearwater, FL, USA

ACM ISBN 979-8-4007-0605-9/24/06.
https://doi.org/10.1145/3649476.3660392

1 INTRODUCTION

The rapid expansion of Internet of Things (IoT) and edge devices has led to a surge in interconnected devices generating vast amounts of data [13]. This data can either be processed on the device itself or sent to the cloud or a data center for analysis [22], but data transmission incurs significant power and communication overheads. On-device processing is beneficial but requires substantial power and resources, posing challenges due to battery limitations in conventional processing architectures [16]. These state-of-the-art computational devices are limited by on-chip memory, which makes them inefficient when processing large amounts of data. The inefficiency comes from the data movement of the large amounts of data between the main memory and the processing cores.

Recent research has increasingly focused on 'non-von Neumann' computing architectures [11] aiming to bridge the gap between the processor and the memory performance. Processing in memory (PIM) is a branch of non-von Neumann architectures where processing elements are integrated onto the memory chip itself. This architecture is gaining traction in real-time application domains, particularly those utilizing Convolutional and the Deep Neural Network (CNNs & DNNs) due to their ability to execute massively parallel processing with minimal latency and energy consumption.

Several recent studies have demonstrated the superior performance of PIM architectures over GPU and CPU designs for training deep neural networks and combinational optimization problems, especially in terms of throughput and energy efficiency [4]. While traditional PIM architectures like bitline-wise architecture [10] and analog crossbar array architecture [7, 9] have been considered as better alternatives to conventional computing hardware for handling the heavy computational load of DNNs, they often suffer from complexity and overhead associated with digital-to-analog (DAC) and analog-to-digital (ADC) conversions. In contrast, the recently developed Look-up-table (LUT) based PIMs have emerged as more flexible, offering superior energy efficiency for comparable performance levels [4], as seen in architectures like LAcc [8] and pPIM [1, 25]. This characteristic makes the LUT-based PIM architecture particularly suitable for implementing different operations required by DL applications such as linear algebraic operations, activation, and pooling.

To further enhance energy efficiency, in this work, we introduce energy harvesting, particularly kinetic energy harvesting (KEH) [17, 19] as a suitable solution for these devices. KEH leverages

vibrations and mechanical disturbances experienced by the computational devices to generate electrical energy [19]. Optimizing ML architectures for energy efficiency without sacrificing performance is thus crucial [21].

In the context of energy harvesting, PIM architecture efficiently uses harvested energy for computational tasks, minimizing data movement and reducing energy consumption associated with memory access. PIM architecture's parallel processing capability accelerates computations while consuming less energy compared to traditional architectures. It enables fine-grained control over memory operations, reducing computational complexity and improving power efficiency. PIM architecture's adaptability to varying energy levels allows for dynamic task scheduling, ensuring efficient computation during periods of low energy availability. By offloading tasks to PIM architecture, reliance on the device's primary power source can be reduced, maximizing the use of harvested energy.

Our work integrates an Energy Harvesting-assisted intermittent machine learning on-device, enhanced by Processing-In-Memory (PIM) architecture. This approach efficiently utilizes harvested kinetic energy, minimizing data movement between processor and memory. PIM architecture's parallel processing capability accelerates computation tasks, reduces energy consumption, and optimizes power footprint. By adapting computation to varying energy levels, PIM architecture extends the lifespan of the device and enables sustainable operation in resource-constrained environments.

Key aspects of the paper include:

- Energy Harvesting-assisted intermittent computation for efficient machine learning.
- Leveraging PIM architecture's in-memory computation techniques enhances energy efficiency and facilitates energy harvesting in IoT and edge devices.
- We evaluate the proposed Energy Harvesting-assisted PIM architecture on various CNN architectures, including LeNet, AlexNet, ResNet-18, -34, and -50, for inference applications.

By integrating these techniques, our framework offers a sustainable solution that utilizes harvested kinetic energy and enhances performance for on-device ML tasks on resource-constrained IoT and edge devices. By executing computations within memory, it reduces energy consumption associated with data movement. This capability also enables efficient utilization of harvested energy, prolonging device lifespan and supporting sustainable operation in resource-constrained environments.

2 RELATED WORK

Among multiple energy harvesting sources, kinetic energy harvesting (KEH) [17, 19] is considered relatively apt and efficient for edge and IoT devices, as these devices undergo vibrations, motion, or mechanical disturbances due to different factors such as device movement and other physical activities, which offers an intriguing opportunity to harvest the energy [19]. By capturing and converting ambient kinetic energy into usable electrical energy, it becomes possible to achieve self-sustainability and overcome the limitations of On-device battery sources.

On-device machine learning in embedded systems is a rapidly growing research area [6, 23]. IoT and edge devices face challenges due to limited memory and compute resources, making it difficult to implement Deep Neural Network (DNN) algorithms efficiently [22]. Even small DNN models like LeNet can take several minutes to execute due to millions of operations [12], highlighting the need for optimized algorithms without compromising accuracy or runtime performance [27]. These devices often rely on harvested energy, but power failures are frequent due to quickly depleting energy levels. Existing intermittent computing approaches, such as CapBand [26] and [15], focus on on-device inference using offline-trained classifiers. However, they lack efficient energy utilization strategies before powering IoT/Edge devices. To address these challenges, we propose an Energy Harvesting-assisted ultra-low-power processing-in-memory accelerator framework. This framework considers the dynamics of machine learning tasks, enhancing energy and learning efficiency systematically, and ensuring seamless operation and resource management on IoT and edge devices.

Processing-in-Memory (PIM)

For IoT and edge applications, hardware-based ML accelerators are favored for implementing ML/AI tasks due to their ability to optimize latency and throughput more effectively than software-based solutions. ML algorithms such as CNNs & DNNs typically require high-end processing support from hardware platforms. While Field-Programmable Gate Arrays (FPGAs) are commonly used for low-end real-time image processing tasks in parallel, their limited resources in terms of area, memory, and processing capabilities restrict their performance in advanced ML applications, particularly in complex and large-scale AI algorithms required for IoT applications. These applications demand ultra-low latency and high accuracy, which pose challenges for low-cost FPGA hardware accelerators.

The PIM architecture is emerging as a promising platform for implementing advanced ML algorithms, offering an alternative to traditional von Neumann computing systems. By performing computational tasks within a computational memory unit, PIMs address the 'memory wall' bottleneck inherent in traditional computing hardware, enabling massively parallel and ultra-low latency operations. While PIM devices typically excel in performing simpler operations[10] with lower data precision, such as binarized or ternarized weights, they demonstrate superior performance and efficiency compared to traditional computing devices.

Recent advancements in PIM architecture explore innovative approaches like Look-Up-Table (LUT) based processing for CNN acceleration, exemplified by architectures like LAcc[8]. These architectures either repurpose memory cells to store pre-calculated outputs of high-precision matrix multiplications or support complete CNN inference within the DRAM chip through hierarchical LUT-based processing architecture. Notably, architectures like pPIM [1, 25] achieve even higher throughput, latency, and energy efficiency compared to conventional PIM architectures.

The popularity of PIM architecture is growing, particularly in real-time application domains. These PIM architectures are capable of efficiently executing matrix-vector multiplication (MVM) operations, essential across various fields [11] including signal processing, machine learning [3], deep learning [2], stochastic computing, image recognition, object recognition [1], and cryptography [24]. However, to date, the application of PIM in energy harvesting contexts remains unexplored.

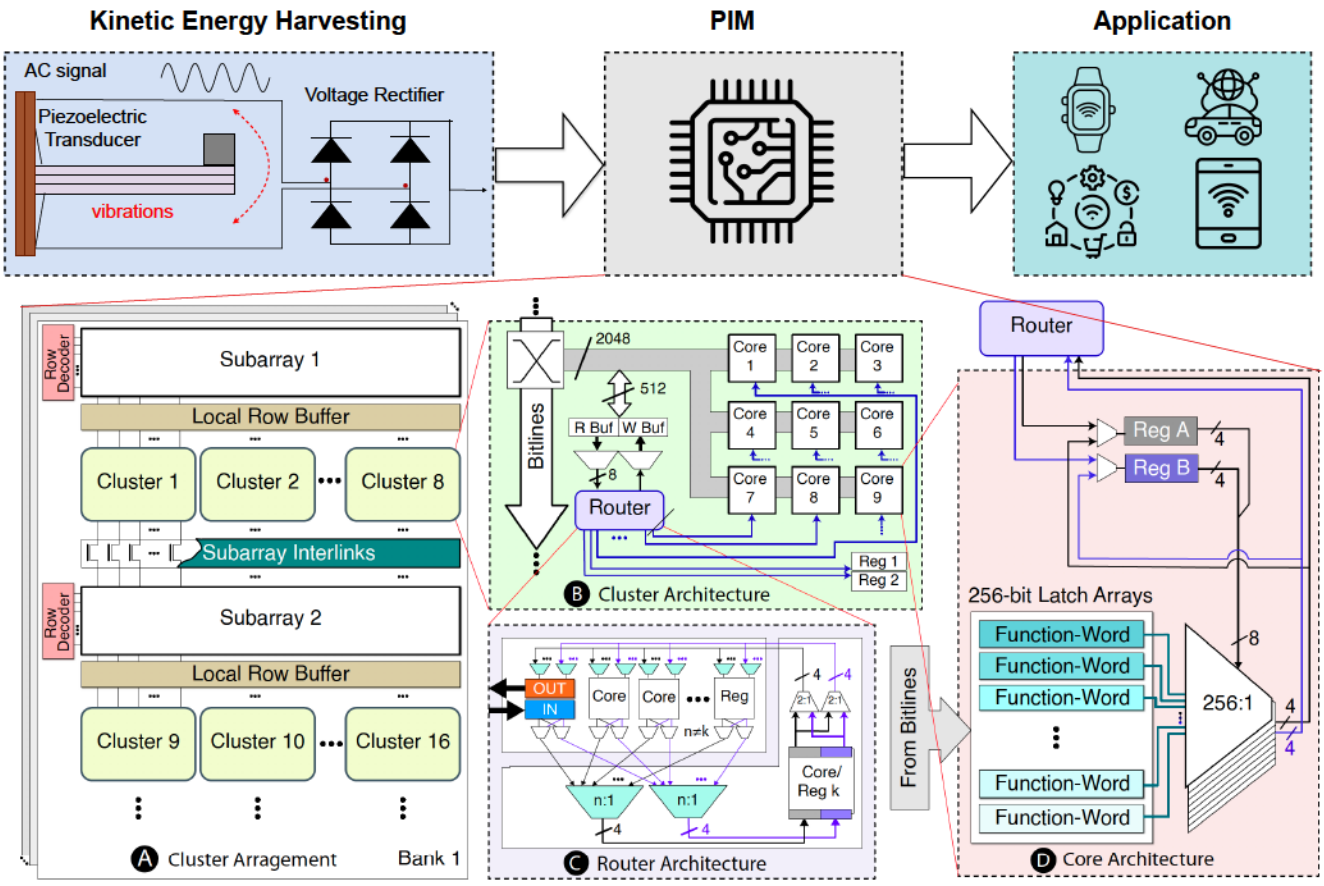

Figure 1: Hierarchical representation of the PIM Architecture showing (a) the distribution of clusters in a DRAM bank, (b) the cluster architecture, (c) the router micro-architecture, and (d) the LUT-core design

3 PROPOSED FRAMEWORK

3.1 Harvesting Piezoelectric Kinetic Energy

Piezoelectric kinetic energy harvesting (KEH) involves creating a device that captures mechanical motion or vibrations and converts them into high-frequency AC electrical signals using piezoelectric material. These signals are then rectified into DC using a diode bridge circuit and stored in a supercapacitor. This stored energy can be used to power electronic devices, reducing the need for external power sources. Efficient power management techniques optimize energy distribution and usage, making this technology valuable for self-powered IoT/Edge applications, enhancing energy efficiency, and reducing the built-in battery load.

Our design and implementation of Kinetic Energy Harvesting (KEH) are depicted in Figure 1. The prototype comprises two main parts: the Energy Harvesting section and the Load. For the Energy Harvesting component, we chose two PEH bending transducers from MIDE Systems as our PEH transducers. These transducers are lightweight at 10.4 grams each, with dimensions of $76.2 \times 31.75 \times 2.28\ mm^3$. As illustrated in Figure 1, the two PEH transducers are affixed to capture energy from vibrational motions.

The output pins of the PEH transducers connect to an energy harvesting circuit, specifically the LTC3588-1 from Linear Technology. The LTC3588-1 incorporates a low power-loss bridge rectifier to rectify the AC voltage output from the PEH transducer. Additionally, it features a high-efficiency buck converter that transfers energy stored in the capacitor into stable DC power for the load. We opted for two electrolytic capacitors with a capacitance of 470 μF and a maximum voltage rating of 25V to store the generated energy from the transducers.

When the capacitor voltage surpasses the under-voltage lockout rising threshold of the buck converter (set at 4V), the buck converter activates to discharge the stored energy. Conversely, if the capacitor voltage falls below the lockout falling threshold (set at 3.08V), the buck converter deactivates, allowing the capacitor to accumulate harvested energy.

3.2 The PIM Architecture

This work employs a PIM framework tailored to facilitate the computationally intensive tasks essential for neural network models. Specifically designed to for integration into Dual In-line Memory Module (DIMM) chips, which can be seamlessly integrated into connected on-edge devices such as a vehicle's on-board unit, this PIM architecture is ideal for IoT/edge devices with limited processing

capabilities. The hierarchical representation of the PIM architecture is outlined in Figure 1, encompasing (a) the arrangement of clusters within a DRAM bank, (b) the cluster architecture, (c) the router microarchitecture, and (d) the LUT-core design, respectively.

3.2.1 Core Architecture. To ensure functional flexibility, the PIM core employs an LUT-based design instead of a predefined logic circuit. This LUT-based approach enables the execution of in-memory arithmetic operations such as addition, multiplication, substitution, and comparison operations with significantly reduced delay compared to bitwise computations. This capability allows for the implementation of various ML algorithms using a combination of these operations.

Figure 1(d) provides a detailed overview of the architecture of a single LUT core. Within the cluster, LUT cores consist of an 8-bit 256:1 multiplexer and eight 256-bit latch arrays. The outputs of specific 8-bit operations are precomputed and stored in the latches as eight 256-bit function words. These latches can access new function words from the bit lines of the DRAM subarrays. Each LUT can generate a 4-bit data output for two input data operands with a width of 4 bits, as depicted by the A and B registers in Figure 1(d). These registers collectively control the select pins of the multiplexers, allowing them to retrieve specific 8-bit data from the eight latches representing the operation's output.

3.2.2 Cluster Architecture. The PIM cluster serves as a processing element (PE) that consolidates the operations executed by the LUT core to carry out specific tasks, like convolution operations in a neural network. Comprising nine PIM cores arranged in a 3×3 configuration, the PIM cluster is embedded within memory data to execute in-memory operations. The PIM LUT cores inside the cluster handle various logic and arithmetic operations. Connectivity among all cores within the cluster is facilitated by a router. This router facilitates data access from any core at any given time during execution, enabling the execution of tasks such as multiplication and accumulation, pooling, and convolution operations performed within a neural network.

3.2.3 Router Architecture. A routing mechanism links all nine LUT cores within a cluster, enabling direct and simultaneous communication among them. The router connects all components of the cluster and read/write ports, facilitating parallel communication. This capability allows the router to access any data at any stage of implementation.

3.3 Operations supported by the proposed PIM architecture

Each PE is capable of performing complex operations such as 8-bit MAC, 8-bit pooling, 16-bit addition. Each LUT core inside a PE can operate in parallel and perform any logic/arithmetic operation such as addition, multiplication, shifting, incrementing, decrementing, bitwise logic, inversion, counting, comparison, substitution on a pair of 4-bit operands or a single 8-bit operand. The cores are fully programmable which makes it possible to program them individually to perform a wide range of desired operations. The outputs of the LUT cores can be recirculated/redistributed among themselves in parallel via the router. Therefore, by orchestrating a scheme of multi-step operations across the nine heterogeneously programmed LUT cores inside a PE, it is possible to support complex operations such as 8-bit MAC. To summarize, each Cluster is capable of:

(1) Supporting complex operations which are amalgamations of a number of simpler logic/arithmetic operations,
(2) Scaling up the precision of an operation by adopting a multi-step operation scheme, aided by the router.

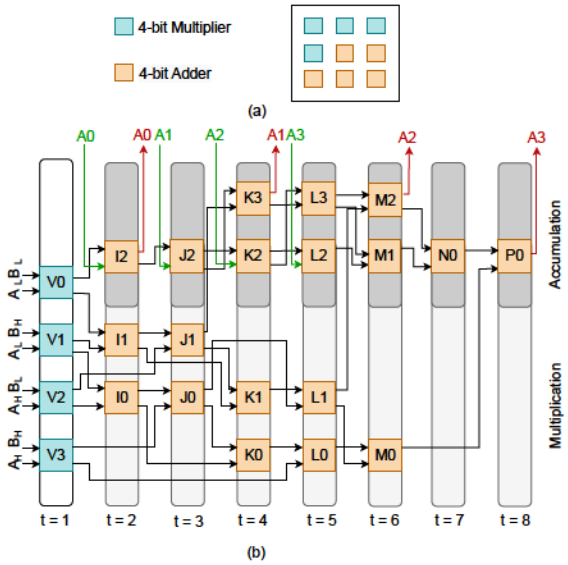

Figure 2: (a) Core Operation Mapping Scheme and (b) Step-wise execution model for 8-bit Fixed point MAC inside a PIM PE. The left and right arrows coming out of each core represent the most and least significant 4-bit of the outputs of the cores respectively. Green texts represent data from the prior round of operation. The clock-steps of the operation are designated by the values of 't' while the letters I, J, K, L, M, N & P respectively represent the corresponding stages of the partial addition operation. During each clock step, the numeric tags accompanying the letters designate multiple parallel operations across different cores.

These features make the proposed architecture a very flexible and high-performance in-memory processing architecture. Conventionally, A MAC is performed via consecutive operations of Multiplication and then Addition. However, the proposed PIM architecture partially parallelizes these two operations and thereby combines these two into one continuous operation that takes fewer clock cycles to execute. Alongside minimizing the latency, this also maximizes resource utilization in the Cluster. Figure 2 shows the programming scheme of the cores in the PE as well as the step-wise operation scheme for the 8-bit fixed-point MAC operation. The LUT cores require performing only two different types of operations: 4-bit multiplications and 4-bit additions. Four out of nine LUT cores inside a PE are programmed as multipliers and five are programmed as adders.

Alongside the 8-bit MAC, a PIM PE can also support a 4-bit MAC operation which requires fewer steps of operations, with the aid of only one multiplier and one adder core. Alongside, it can also support other operations such as max-pooling and ReLU Activation with either 8-bit or 4-bit fixed-point precision. For example, the 8-bit ReLU activation operation can be performed by implementing an 8-bit substitution table on a single LUT core that produces zero output values in response to negative inputs.

4 EXPERIMENTAL RESULTS

4.1 Experimental Setup

The proposed framework utilizes an S230-J1FR-1808XB two-layered piezoelectric bending transducer from MIDE technology with a block tip mass of 24.62g ± 0.5% which has a resonance frequency of 25 Hz. Table 1 represnts the kinetic energy captured with respect to time. All signals are sampled at a frequency of 100 Hz.

Table 1: Kinetic Energy Harvesting w.r.t. Time

Time (ms)	Power (mW)
10	0.022
20	0.027
50	0.034
100	0.063
200	0.105
250	0.117
400	0.145
500	0.162
1000	0.342
2000	0.72
3500	1.42
5000	1.9

To evaluate the performance of the proposed framework, we conducted a case study focusing on the gesture recognition dataset. The gesture recognition dataset was created by collecting the readings from an IMU sensor built on the Arduino Nano board. We captured IMU sensor reading for the following gestures (up, down, right, left, twist, punch, and flex) and the dataset comprised of nearly 70K samples. The gesture recognition data was trained and fine-tuned on Tensorflow Lite model (MobileNet [14], LeNet-5, AlexNet and ResNet [18]). These models were deployed on the Arduino board for evaluation of the classification tasks.

4.2 PIM Core and Cluster Characteristics

In this section, we evaluate the PIM in terms of performance, energy consumption, and area for DL applications. The delay & power for the PIM core and cluster are obtained from Synopsys Design Compiler using 28nm standard cell libraries from TSMC and are presented in Table 2. The delay of a single 8-bit MAC performed within a cluster involves computations inside the PIM cores as well as communication among the cores. Power consumption of the cluster is that of all the cores and the core-to-core communication. The power and delay for intra and inter subarray data transfers are obtained from [5] and [20]. These metrics are used in the system-level performance evaluation of the PIM in the next subsections.

Table 2: Characteristics of Proposed PIM components in 28 nm node

Component	Delay (ns)	Power (mW)	Active Area (μm^2)
PIM Core	0.8	2.7	4196.64
PIM Cluster (MAC Operation)	6.4	8.2-11	37769.81
Intra-Subarray Communication [20]*	63.0	0.028 μJ/comm	N/A
Inter-Subarray Communication [5] for subarrays 1/7/15 hops away*	148.5/ 196.5/ 260.5	0.09/ 0.12/ 0.17 μJ/comm	N/A

**Represented in 28nm technology node*

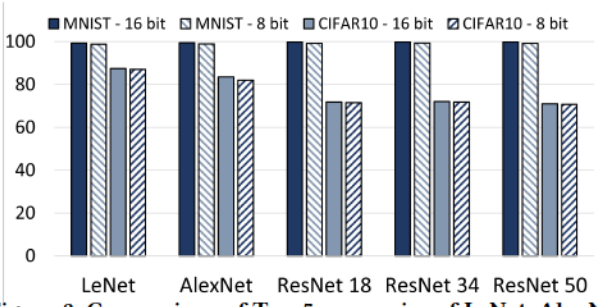

Figure 3: Comparison of Top-5 accuracies of LeNet, AlexNet, ResNet-18, -34 and -50 on MNIST, CIFAR-10 datasets for 16-bit, 8-bit data precision.

4.3 Inference Accuracy Comparison

We assess our proposed architecture across a range of cutting-edge deep neural networks, including LeNet, AlexNet, and ResNet-18, -34, and -50. These sophisticated deep learning models are deployed on our proposed hardware accelerator using both the MNIST dataset (with dimensions of 28x28x1) and the CIFAR-10 dataset (with dimensions of 32x32x3). Each dataset comprises approximately 60,000 training and 10,000 testing images distributed across 10 classes, with the objective being the accurate classification of input images.

In Figure 3, we present comparison plots of Top-5 accuracy for both 16-bit floating-point (FP) and 8-bit fixed-point data precision across both datasets. Notably, we observe that the accuracies achieved across the evaluated networks are highly similar for both 16-bit and 8-bit precision data (for both inputs and weights). For instance, the Top-1 accuracy achieved on the MNIST dataset with AlexNet implementation is 98.89% and 99.43% for 16-bit and 8-bit precision, respectively. Conversely, on the CIFAR-10 dataset, the Top-1 accuracy with AlexNet implementation is 83.5% and 82% for 16-bit and 8-bit precision, respectively. Additionally, we note that the accuracies of LeNet, AlexNet, ResNet-18, -34, and -50 on the CIFAR-10 dataset are noticeably lower compared to the MNIST dataset, as illustrated in Figure 3.

4.4 Comparative Performance Evaluation on PIM

In this section, we conduct a comparative analysis focusing on the throughput and energy efficiency of PIM for two precision modes across LeNet, AlexNet, ResNet-18, -34, and -50 algorithms. Energy efficiency is quantified as the number of frames processed within the processor per unit of energy (measured in Joules). Figures 4(a) and 4(b) depict comparisons of throughput (measured in Frames per Second) and energy efficiency (measured in Frames per Joule), for 4 bit, 8 bit (weights and input) precision modes respectively.

Considering the observed low power consumption and area efficiency of PIM, we opt to utilize a PIM bank comprising 256 PIM clusters per DRAM chip within a complete rank of DRAM chips for a DIMM configuration. Therefore, for evaluation purposes in this section, we focus on a single PIM implementation for a DIMM. In figure 4 (a),(b) it is evident that the lowest performance and energy efficiency across all CNN algorithms are achieved with 8-bit fixed-point precision mode. This precision mode necessitates operand decomposition into 4-bit segments before distribution across multiple cores (i.e., all nine cores in a cluster for MAC operation), thereby offering the least scope for parallelization.

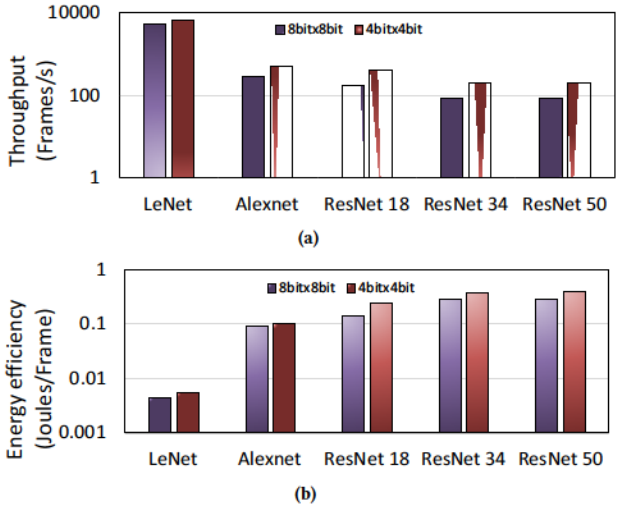

Figure 4: Comparison of (a) Throughput in Frames/s and (b) Energy efficiency in Frames/Joule for AlexNet, ResNet -18, -34, -50 inference on PIM

Conversely, the 4-bit inference mode exhibits higher throughput and superior energy efficiency due to its smaller aggregated bit-width of operand pairs, leading to reduced latency and increased parallelization. This mode inherently offers a higher degree of parallelization, performing all operations in fewer steps. Additionally, it's noteworthy that the performance and energy efficiency of PIM are closely related to the computational load imposed by different CNN algorithms.

5 CONCLUSION

In this paper, we proposed and evaluated a novel framework, "Energy Harvesting-assisted PIM Architecture" tailored for IoT/Edge applications. By leveraging kinetic energy from piezoelectric harvesters, we achieved Energy-aware intermittent ML computation, enhancing the sustainability of ML applications in resource constrained environments. Furthermore, our framework integrates energy harvesting, voltage monitoring, intermittent computing, and ML seamlessly, ensuring efficient utilization of harvested energy and enhancing on-device inference efficiency through PIM architecture without compromising performance. Additionally, we observed accelerated testing latency without compromising classification performance, demonstrating the effectiveness of our approach in optimizing energy usage and enhancing overall system efficiency. We also evaluated the Energy Harvesting-assisted PIM architecture on various CNN architectures, including LeNet, AlexNet, ResNet-18, -34, and -50, for inference applications.

REFERENCES

[1] Sathwika Bavikadi, Purab Ranjan Sutradhar, Amlan Ganguly, and Sai Manoj Pudukotai Dinakarrao. 2021. uPIM: Performance-aware Online Learning Capable Processing-in-Memory. In *IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*.

[2] Sathwika Bavikadi, Purab Ranjan Sutradhar, Amlan Ganguly, and Sai Manoj Pudukotai Dinakarrao. 2023. Heterogeneous Multi-Functional Look-Up-Table-based Processing-in-Memory Architecture for Deep Learning Acceleration. In *International Symposium on Quality Electronic Design (ISQED)*.

[3] Sathwika Bavikadi, Purab Ranjan Sutradhar, Amlan Ganguly, and Sai Manoj Pudukotai Dinakarrao. 2024. Reconfigurable Processing-in-Memory Architecture for Data Intensive Applications. In *Int. Conf. on VLSI Design (VLSID)*.

[4] Sathwika Bavikadi, Purab Ranjan Sutradhar, Khaled N. Khasawneh, Amlan Ganguly, and Sai Manoj Pudukotai Dinakarrao. 2020. A Review of In-Memory Computing Architectures for Machine Learning Applications.

[5] K. K. Chang, P. J. Nair, D. Lee, S. Ghose, M. K. Qureshi, and O. Mutlu. 2016. Low-Cost Inter-Linked Subarrays (LISA): Enabling fast inter-subarray data movement in DRAM. In *2016 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. 568–580. https://doi.org/10.1109/HPCA.2016.7446095

[6] Sauptik Dhar, Junyao Guo, Jiayi Liu, Samarth Tripathi, Unmesh Kurup, and Mohak Shah. 2021. A survey of on-device machine learning: An algorithms and learning theory perspective. *ACM Transactions on Internet of Things* (2021).

[7] A. D. Patil et al. 2019. An MRAM-Based Deep In-Memory Architecture for Deep Neural Networks. In *IEEE International Symposium on Circuits and Systems*.

[8] Q. Deng et al. 2019. LAcc: Exploiting Lookup Table-based Fast and Accurate Vector Multiplication in DRAM-based CNN Accelerator. In *ACM/IEEE Design Automation Conference (DAC)*.

[9] S. Yin et al. 2020. XNOR-SRAM: In-Memory Computing SRAM Macro for Binary/Ternary Deep Neural Networks. *IEEE Journal of Solid-State Circuits* (2020).

[10] V Seshadri et al. 2017. Ambit: In-Memory Accelerator for Bulk Bitwise Operations Using Commodity DRAM Technology. *IEEE/ACM International Symposium on Microarchitecture (MICRO)* (2017).

[11] A. Ganguly, R. Muralidhar, and V. Singh. 2019. Towards Energy Efficient non-von Neumann Architectures for Deep Learning. In *International Symposium on Quality Electronic Design (ISQED)*.

[12] Graham Gobieski, Brandon Lucia, and Nathan Beckmann. [n. d.]. Intelligence Beyond the Edge: Inference on Intermittent Embedded Systems. In *Architectural Support for Programming Languages and Operating Systems (ASPLOS '19)*.

[13] Rosilah Hassan, Faizan Qamar, Mohammad Kamrul Hasan, Azana Hafizah Mohd Aman, and Amjed Sid Ahmed. 2020. Internet of Things and Its Applications: A Comprehensive Survey. *Symmetry* (2020).

[14] Andrew G. Howard, Menglong Zhu, Bo Chen, D. Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *ArXiv* (2017).

[15] Sahidul Islam, Jieren Deng, Shanglin Zhou, Chen Pan, Caiwen Ding, and Mimi Xie. 2022. Enabling fast deep learning on tiny energy-harvesting IoT devices. In *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE.

[16] Jie Lin, Wei Yu, Nan Zhang, Xinyu Yang, Hanlin Zhang, and Wei Zhao. 2017. A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications. *IEEE Internet of Things Journal* (2017).

[17] Dong Ma, Guohao Lan, Mahbub Hassan, Wen Hu, and Sajal K. Das. 2020. Sensing, Computing, and Communications for Energy Harvesting IoTs: A Survey. *IEEE Communications Surveys Tutorials* (2020).

[18] Yoshitomo Matsubara, Marco Levorato, and Francesco Restuccia. 2022. Split computing and early exiting for deep learning applications: Survey and research challenges. *Comput. Surveys* (2022).

[19] Muhammad Moid Sandhu, Kai Geissdoerfer, Sara Khalifa, Raja Jurdak, Marius Portmann, and Brano Kusy. 2020. Towards optimal kinetic energy harvesting for the batteryless IoT. In *IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*.

[20] V. Seshadri, Y. Kim, C. Fallin, D. Lee, R. Ausavarungnirun, G. Pekhimenko, Y. Luo, O. Mutlu, P. B. Gibbons, M. A. Kozuch, and T. C. Mowry. 2013. RowClone: Fast and energy-efficient in-DRAM bulk data copy and initialization. In *IEEE/ACM International Symposium on Microarchitecture (MICRO)*.

[21] Muhammad Shafique, Rehan Hafiz, Muhammad Usama Javed, Sarmad Abbas, Lukas Sekanina, Zdenek Vasicek, and Vojtech Mrazek. 2017. Adaptive and energy-efficient architectures for machine learning: Challenges, opportunities, and research roadmap. In *IEEE symposium on VLSI*.

[22] Sanket Shukla, Setareh Rafatirad, Houman Homayoun, and Sai Manoj Pudukotai Dinakarrao. 2023. Federated Learning with Heterogeneous Models for On-device Malware Detection in IoT Networks. In *2023 Design, Automation Test in Europe Conference Exhibition (DATE)*.

[23] Sanket Shukla, P D Sai Manoj, Gaurav Kolhe, and Setareh Rafatirad. 2021. On-device Malware Detection using Performance-Aware and Robust Collaborative Learning. In *58th ACM/IEEE Design Automation Conference (DAC)*.

[24] Purab Ranjan Sutradhar, Kanad Basu, Sai Manoj Pudukotai Dinakarrao, and Amlan Ganguly. 2021. An Ultra-efficient Look-up Table based Programmable Processing in Memory Architecture for Data Encryption. In *IEEE International Conference on Computer Design (ICCD)*.

[25] P. R. Sutradhar, M. Connolly, S. Bavikadi, S. M. Pudukotai Dinakarrao, M. A. Indovina, and A. Ganguly. 2020. pPIM: A Programmable Processor-in-Memory Architecture With Precision-Scaling for Deep Learning. *IEEE Computer Architecture Letters* 19, 2 (2020), 118–121.

[26] Hoang Truong, Shuo Zhang, Ufuk Muncuk, Phuc Nguyen, Nam Bui, Anh Nguyen, Qin Lv, Kaushik Chowdhury, Thang Dinh, and Tam Vu. 2018. Capband: Battery-free successive capacitance sensing wristband for hand gesture recognition. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*.

[27] Shanglin Zhou, Mimi Xie, Yufang Jin, Fei Miao, and Caiwen Ding. 2021. An End-to-end Multi-task Object Detection using Embedded GPU in Autonomous Driving. In *2021 22nd International Symposium on Quality Electronic Design (ISQED)*.