Ingénierie des Systèmes d'Information
Vol. 30, No. 12, December, 2025, pp. 3083-3092
Journal homepage: http://iieta.org/journals/isi

# RRAM Design for Image Enhancement in Edge Devices

C. Radhik[1,2], G. V. Ganesh[1*], P. Ashok Babu[2]

[1] Department of Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram 522302, India
[2] Department of ECE, Institute of Aeronautical Engineering, Hyderabad 500043, India

Corresponding Author Email: ganesh.gorla@gmail.com

https://doi.org/10.18280/isi.301201

**Received:** 30 June 2025
**Revised:** 5 October 2025
**Accepted:** 20 November 2025
**Available online:** 31 December 2025

***Keywords:***
*RRAM-based memristor, PSO optimization, low power architecture, image enhancement, in-memory computing, edge AI, hardware accelerator, CNN hardware design*

## ABSTRACT

RRAM can be used to perform in-memory computing with low energy, high speed and small area, so it is an ideal device to implement real-time image enhancement on the edge device. Nevertheless, standard CNN models based on RRAM need to be manually tuned to the alpha-parameter, which is a limiting factor in terms of scalability and reliability. This paper suggests an approach to automatically decide on the optimal α-value using part of the Particle Swarm Optimization (PSO) to enhance convergence, speed of the execution and fidelity to the enhancement. The architecture of RRAM is at the 65 nm technology node and the analysis is carried out and calculated with the pixels level of computation and image quality analysis by using SPICE simulation. It has been shown through experimental work that it can enhance accuracy by 94 per cent, reduce power consumption and hardware footprint by 35% to 50% over current CNN-based designs. This shows that PSO-optimized RRAM accelerators are a more efficient and scalable image enhancement solution in edge AI systems.

## 1. INTRODUCTION

Resistive Random-Access Memory (RRAM) has become promising in line of substituting CMOS-based architecture because of its non-volatility, rapid switching, zero leakage power and its ability to perform in-memory computation. These attributes have RRAM especially appealing in real time image enhancement and other edge applications that require latency. Nevertheless, RRAM is the technology that is hard to practically implement in image processing pipelines despite its benefits. The first problem is the resistance state variability, which influences the correction of the programming and gives out unequal pixel transformations. Besides, thermal unsteadiness and durability deterioration diminish durability under uninterrupted write-and-read operations. The nonlinear quality of RRAM also makes it difficult to accurately map weights in CNN-based image refinement systems, and commonly necessary to use device-level parameters, including the alpha factor, by hand. Such manual dependency adds complexity to the design and reduces scalability, in particular, when multiple image profile targets or multi-stage enhancement tasks are targeted. Although the use of RRAM-based accelerator in neural computation has been considered in previous literature, the majority of the research is silent on automated optimization of the parameters. Earlier architectures are based on fixed tuning or heuristic optimization, neither of which made efforts to account for device noise, non-linear switching, retention drift and array-scale parasitic effects. Accordingly, the gap in the research is clear in coming up with an adaptive and scalable solution that is going to optimize the RRAM behavior in image enhancement systems. To close this gap, the current paper proposes a PSO based optimization procedure to be incorporated into an RRAM computing array, where running this autonomous subsystem provides a chance to select the alpha-parameter and enhance pixel improvement, respectively. This model, which is proposed, minimizes the effects of noise in the devices, increases the consistency of converting to a particular converged state, and the stability of the converting state with temperature changes, contributing to the consistency of the output quality and the minimization of power dissipation [1].

Although designs are often performed manually or semi-automatically, these two modalities have been applied in many large-scale applications, like image processing approaches involving neural networks. Modern convolutional neural networks (CNNs) with deep learning have shown to be very effective in today's intelligent systems for a wide range of tasks, including image/speech recognition and classification. By using the memory array for the weighted sum computation, recent attempts to construct custom inference engines using the processing-in- memory design have reduced the frequent transmission of information across buffers containing calculation units. In traditional PIM systems, the convolutional layers for every 3D kernel are unrolled into the vertical row of a huge weight matrix, which is necessary because to the numerous iterations required to retrieve the input data. To generate stochastic bit-streams (SBS) that are both stable and accurate for use in stochastic computing, a probabilistic switching model is created for RRAM SNG based on physical principles. The switching probability in different operational systems may be explained by considering the physical cause of intrinsic fluctuations. Probability shift (PS), a major source of error in SBS, may be evaluated by modeling the cumulative impact between continuous cycles. Modern AI systems rely heavily on state-of-the-art deep CNNs due to their exceptional performance in image/speech identification and classification. Many other approaches, such as systolic construction, near memory computing, and the processing-in-memory (PIM) approach with cutting-edge technologies like RRAM, have been recently used in attempts to develop novel inference engines. Characteristic representations of pictures are used in several computer vision applications, such as comprehension and multi-view enrollment [2], rather than raw pixel intensity. However, there is a lack of a unified examination of these many methods, and the benefits of novel ideas or developing technology are often based on qualitative forecasts. However, conventional pipelines for generating such representations need costly storage and computational resources to perform pixel-wise analog-to- digital conversions. NeuRRAM, the first multimodal edge AI chip using RRAM CIM, offers architectural versatility across diverse models, record energy efficiency surpassing prior art at multiple computational bit precisions, and inference accuracy comparable to 4-bit software implementations. It reports 99.0% accuracy on MNIST, 85.7% on CIFAR-10, 84.7% on Google speech command recognition, and a 70% reduction in reconstruction error for a Bayesian image recovery task [3].

The use of processing-in-memory (PIM) offers a viable remedy to the von Neumann barrier by taking use of huge parallelism in an energy-efficient manner. This emerging kind of memory has lately shown its ability to construct a PIM architecture because numerous stateful logic operations, such as IMP like NOR, may be performed in parallelism in an RRAM crossbar. The memory should be utilized largely for storage, although previous synthesis processes have focused on reducing latency via stateful logic operations. that is, the majority of the crossbar is dedicated to computation rather than storage. Because of how well it boosts picture quality, Randomized Spray Retinex is a powerful image improvement method. But its adoption was impeded in many application situations, for example in internet of things systems with low hardware resources, because of the processing complexity of the method and the necessary hardware resources along with memory accesses. Image augmentation is increasingly being used to boost the efficiency of new applications because to the proliferation of AI. A new crossbar array based on resistive memory (RRAM) shows promise as a method to speed up applications using neural networks. RRAM-based CNN accelerators provide strong support for both intra-layer and inter-layer parallelism. Each network layer may operate independently with a fraction of the input data thanks to inter-layer parallelism, while intralayer parallelism produces multiple copies of kernel for each layer. But without data sharing across duplicate kernels, crossbar arrays sit idle during inference in the RRAM-based accelerators that have been presented thus far. By relying on one another's information, data dependencies are created, which slow down subsequent pipeline stages. The SET and RESET procedures in RRAM regulate the production and breakdown of conductive filament. According to the rules of thermodynamics, these procedures represent the minimum available free energy. Bends, fractures, and bubble-like patterns appear on an RRAM device when the operating voltage is too high.

RRAM is a relatively new technology that has found widespread use in boosting the processing speed of deep neural networks. The limitations of RRAM's resistance level and interfaces make it difficult to do calculations with a high degree of accuracy RRAM-based CNN accelerators provide strong support for both intra-layer and inter-layer parallelism. Each network layer may operate independently with a fraction of the input data thanks to inter-layer parallelism, while intra-layer parallelism produces multiple copies of kernel for each layer [4]. An important step in lowering AI's power consumption is the construction of devices that employ low precision neural networks using emerging memories like RRAM. Maximum efficiency of energy in such systems may be attained by tight integration of logic and memory [5]. One of the most important decision-making tools in the field of medical imaging is the computer-assisted diagnostic system. Structural MRI has lately emerged as a strong tool for diagnosing Alzheimer's disease (AD). Computer-aided diagnosis of AD is difficult because of issues with semantic feature ambiguity and significant inter-class visual similarities, as well as a lack of recognition memories in the mild cognitive impairment stages [6]. The crossbar network connectivity of RRAM [7] is made possible by the one diode-one resistors (1D1R) storage design, which is effective in suppressing crosstalk interference.

Our community can zero in on retrieving the structure-related information because various weights are allocated to various streams of the map of features. Our network can learn the complicated feature transformation incrementally using recurrent learning and then realize the color modifications without an increase in the amount for network variables. Extensive studies using publicly available datasets prove that our strategy is better. In conclusion, the following are some of our main contributions:

•To the most effective of our ability, we have pioneered the use of invertible neural networks (INN) for improving underexposed images. Our symmetric design performs unidirectional feature learning simultaneously, outperforming previous methods for improving underexposing images.

•To make color adjustments gradually without raising network parameters, we present a recurring learning strategy of transform features that makes use of a recurring residual-attention modules (RRAM).

## 2. RELATED WORK

Dot-product operations may be carried out in a single cycle by using the reference [8] described all-digital, single-ended XNOR sensing, RRAM-based convolutional block. They show that the structure can handle a resistance window as small as 1.09, ensuring reliable activities even under a high RRAM deviation ($\sigma/\mu = 25\%$ for a resistance window among both states around 50) by accounting for the structural and RRAM limitations at the 28-nm technology nodes. When paired with ISAAC, a state-of-the-art learning accelerator, their block can guarantee reliable operations while reducing energy use by a factor of 2.7. Based on the findings presented by Giacomin et al. [9], their research demonstrates that, in comparison to conventional in-sensor computing systems, this architecture has the potential to drastically cut the amount of energy needed for data translation and transmission to off-chip processing without sacrificing precision. They used a processing-in-pixel accelerator (MRPIPA) based on a combination of multilayer RRAM (HfOx) to obtain a frame rate of 1000 at a speed of 1.89 TOp/s/W, with just a little hit to accuracy. The paper [10] presents integrated and computationally efficient inference accelerators for spiking neural networks.

Resistance-variable random-access storage is a promising new kind of computer memory. It is frequently utilized in PIM, neural network computing, and other domains because to its ability to be used to construct the crossbar architecture, which simulates matrix computing. To model the LIF neuron, we created memristor-based weight storage matrices and associated circuit. They have suggested an SNN hardware inferences accelerator that combines 0.75K memristor with 24K neurons, 192M synapses, and other components. To complete the inference job on the MNIST dataset, we trained a three-layer fully linked network and put it on the accelerator. The results demonstrate that at a frequency of 50MHz, the accelerator can produce 148.2 frames per sec with 96.4 percent accuracy.

By dividing the kernels and sending the input information to various processing-elements (PEs) based on their locations, Peng et al. [11] suggested a novel weight mapping structure along with data flow that maximizes the repeated use of weight and input data for PIM architecture. As a case study, this investigation employs a 32 nm, 8-bit PIM design built on RRAM. The inline formula is generated using the proposed mapping approach and data flow. Compared to its predecessor, which relied on a conventional mapping method, ResNet-34 demonstrates greater capability while consuming less energy. With just a 50% increase in area overhead, Throughput is increased by an astounding 132476 FPS, and energy efficiency is increased by 20.1 TOPS/W, thanks to our recommended optimal pipeline design. With the same hardware resource restriction (i.e., the same amount of accessible space on the device), Wang et al. [12] evaluated the frame rate with energy consumption of ATTEN_CNN-like CNN inferences accelerator on the CIFAR-10 dataset utilising CMOS and post-CMOS technologies. We also investigate the impact of CMOS platform limitations on data transport, including off-chip storage DRAM accessibility and connectivity. According to the numbers we gathered, the peripheral (ADCs) is the main contributor to both power consumption and physical footprint in the digitised RRAM-based concurrent readout PIM design. Reduced DRAM access, fast throughput, and efficient parallel read out allow this architecture to achieve >2.5x higher energy efficiency (TOPS/W) than systolic array or near memory-based computing at the same frame rate. Implementing a bit-count decreased XNOR network with pipelining may provide further >10 gains.

Bettayeb et al. [13] compared the energy efficiency as well as frame rate of an ATTEN_CNN-like CNN infer accelerators on the CIFAR-10 dataset across CMOS and post-CMOS technological platforms, under the restriction of having hardware resources that are roughly equivalent in terms of on-chip size. They also look at the constraints of CMOS platforms—off-chip storage DRAM access and interconnect—to see how they affect data transfer. Our quantitative analysis of the digital RRAM-based simultaneous readout PIM design shows that the peripheral (ADCs) is the primary contributor to both power usage and physical footprint. Due to its efficient parallel read out, fast throughput, and less dependence on DRAM, this design may deliver >2.5x additional energy savings over systolic arrays at the identical frame rate. Implementing a bit-count decreased XNOR network with pipelining may provide further >10 gains. Histograms of Oriented Gradients, or HOG, are a popular feature extraction approach, Prabhu and Raghavan [14] presented HOGEye, a near-pixel version of HOG that is both fast and accurate. Critical but computationally costly activities like derivatives extraction (DE) and histogram generation (HG) are moved from the digital to the analogue domain by HOG Eye's unique neural approximation technique in an RRAM driven 3D stacked image sensor. HOGEye design can save a lot of power since the perceptual (sensor) and computational (DE and HG) processes are all housed in the same physical area. Energy efficiency is improved by more than 2.5 times associated to state-of-the-art designs; With a resolution of 256 × 256, the HOGEye sensors system consumes less than 48W@30fps (equal to 24.3pJ/pixel), while the analysis portion needs just 14.1pJ/pixel. According to reference [15] scientists, "area utilisation" is the amount of memory used in a crossbar.

STAR is a new synthesis approach for stateful logic that aims to maximize area use while minimizing throughput loss. Two optimization methodologies for minimizing STAR's computational footprint are shown. First, we've decreased the space devoted to unnecessary inputs. Without having to hardcode them into the crossbar, they can keep track of the constants that apply across several rows (or columns) by encoding them as instantaneous values into the control's signals. One copy of the remaining inputs is kept in the crossbar. Second, we use unused cells to minimize the space devoted to intermediate variables. They also develop a scheduling method to identify the best order of operations with respect to the number of times each variable must be cleared. This approach may also be used to remove main inputs that are invalid. To further prove the efficacy of STAR, author give a case investigation regarding the picture convolution. Based on experimental results, STAR outperforms the state-of-the-art autonomous logic synthesis flow SIMPLER by 33.03% in terms of area utilization and 1.43x in terms of throughput. When compared to IMAGING, the most advanced autonomous logic-based image processing accelerator, their implementation of image convolution achieves 78.36% higher area utilization and 1.48x throughput. The semantic segmentation of poor-quality urban road sceneries is suggested by Peng et al. [4], and the RSR is mentioned as a possible pre-processing filter. They evaluate the effectiveness of a pre-trained deeper semantic segmentation networks on dark, noisy pictures and on RSR pre-processed images using the public ally accessible Cityscapes dataset. Their results show that RSR is useful for enhancing segmentation precision. They also suggest a unique efficient implementation of the RSR employing RRAM technologies to deal with the computation complexity and applicability to edge devices.

The design supports analogue in-memory computing (IMC) at a high level of parallelism. Using RRAM-CMOS technology, the authors present in detail an efficient and low-latency implementation of the RSR. SPICE simulations utilizing measured data from manufactured RRAM with 65 nm CMOS techniques are used to validate the design. An essential first step toward a low-complexity, hardware friendly design and architecture for Retinex algorithms on edge devices is offered here. To encode the resistive variation of 65nm CMOS 1T1R OxRAM (TiN/HfO2/Hf/TiN) in the learnt weight of an CNN (Convolutional Neural Network) in a digital regime, the authors of devised a Look-Up-Table based architecture. Here, the author does the opposite, modelling the two extremes CNN designs—the Fully Serial Networks as well as the Fully Paralleled Systems (FPS)—using the RRAM resistance encoding learnt weights. Trends in prediction variability are measured using RRAM resistive variations, CNN convolution matrices size (55, 33, 11, with 11 max pools), the overall number of layers in the CNN, with the input image pixel size. To improve parallel for pipeline enabled RRAM-based accelerator and address these issues, Ma et al. [5] proposed a novel architectural framework called Fine-grained Parallel RRAM framework (FPRA). FPRA addresses the issue of data sharing by making use of kernel batches and information transfer aware memory. Data dependencies brought on by the input's common data might be reduced by batching by rearranging the sequence of the kernels. By equally buffering data from input to output for each tier, data sharing sensitive memory helps to reduce the quantity of data sent among levels. Using a cycle-accurate simulator, they tested FPRA on eight widely used convolutional neural network architectures for image recognition. They find that compared to the best RRAM based accelerators, FPRA provide an average latency speedup of 2.0 times and a throughput gain of 2.1 times. To execute computation in memory, Abedin et al. [6] suggested hybrid memory architecture based on a novel array of static random-access memory (SRAM) with RRAM cells. The SRAM array might serve two purposes depending on how it's set up. It may store information in memory mode as an SRAM array, meeting the needs of high-performance applications. It is also possible to set it up as sense amplifiers (SA-SRAM) to read the data from RRAMs and carry out the calculation locally. Independent gate FinFET (IG-FinFET) is used in the circuit design; this kind of FinFET has a channel that can be controlled by two separate gates, giving the designers more leeway. Based on our findings, the suggested SA-SRAM cells reduce write energy consumption by 50% and increase CWLM by 20% compared to standard 8T SRAM. Furthermore, our design's energy consumption in application areas like image processing is substantially lower than the popular comparative in-memory architecture solutions because to the mix of SRAM with RRAM cells in the suggested architecture. They also suggested a polymorphic circuit basic to solve security issues including reverse engineering and integrated circuit (IC) counterfeit. The suggested polymorphic circuit and hybrid memory architecture both need additional calculations to complete their respective difficult logic operations.

In study [7], scientists created a statistical model to mimic the RRAM's switching process using ZnO. Using field driven ion migration and temperature effects, the model constructed a ZnO-based RRAM with a programmable SET as well as RESET resistance transition process. They discovered that a significant quantity of heat energy was generated by the carrier transport of the dielectric substance within the conducting filament. Heat transmission, electrostatic, as well as yield RRAM energy was all accounted for in the model thanks to the integrated COMSOL Multi Physics software. As the working power was ramped up, so was the amount of heat energy produced. Therefore, a high-power device's dependability can't be guaranteed. We acquired many carrier heat studies in 2D pictures and concluded that optimizing the materials and structures used to create RRAM devices that have low operating currents is critical. Research suggests that an RRAM crossbar may be used to speed up smaller bit-width convolutional neural networks (LB-CNN) [16]. They talk at length on the system's architecture, covering everything from the decision to use matrix splitting to increase scalability to the implementation's use of pipelined line buffers to speed up inference. They also suggest a way of dividing and quantizing during training to consider real-world hardware limitations. Compared to multibit model with device fluctuation, low bit-width RESNET-50, which on RRAM proves to be significantly more stable in our studies. When applied to DENSENET, the pipeline technique speeds up picture processing by around a factor of 6.0. When compared to the multibit ATTEN_CNN structure, the suggested accelerator reduces energy consumption by 54.9% and space requirements by 48.3% for low-bit ATTEN_CNN for CIFAR-10. Bature et al. [17] examined ternary neural networks, whereby the synaptic weights may take on ternary values. To enable single-sense recovery of the weight value, we suggest a two transistor, two-resistor storage design with a precharge detecting amplifier. This sense amplifier has been experimentally assessed on a 130 nm CMOS/RRAM integrated device, demonstrating its robustness against process, voltage, and variations in temperature, and utility at low supply voltage. Their bits have a known rate of error. They demonstrate the superiority of ternary neural network architectures over binary ones, the kind most often used for emulation in a hardware simulator, by simulating the problem of CIFAR-10 image recognition. They show that the neural network we use is protected from the bit errors that plague their approach, allowing it to be utilized without any further error correction [18]. Previous work presented a two-transistor/two-resistor memory architecture with a pre-charge sense amplifier that enables single-operation weight readout. Measurements from a hybrid 130 nm CMOS/RRAM chip showed suitability for low-voltage operation and robustness to process, voltage, and temperature variations. The bit error rate was characterized, and CIFAR-10 simulations demonstrated that ternary neural networks significantly outperform binary networks. The network was also shown to tolerate the observed bit errors, making error correction unnecessary [19]. Prior work introduced memristor-based reconfigurable circuits enabling fully analog low-bit neural network implementations without ADCs. A mixed-precision network supporting multiple precision modes was demonstrated, achieving 84.8–87.5% accuracy on CIFAR-10 with a 1.6–20× reduction in model parameters. Circuit-level evaluations confirmed accuracy, robustness, and energy efficiency, highlighting the suitability of memristor-based mixed-precision architectures for edge devices [20].

## 3. PROPOSED WORK

### 3.1 CNN model

The building blocks of a typical CNN are a series of interleaved convolutional and fully-connected layers. Irregular neuronal levels, layers for pooling, and normalization layers may be added on top of a convolution (conv) layer as needed.

Transform Layer. The Conv layer's function of mathematics may be written as Eq. (1):

$$\vec{g}(x,y,z) = \sum h-1i=0 \sum w-1j=0 \sum \mathrm{Cin}-1k=0 \vec{f}(x+i,y+j,k)\cdot \vec{c}\vec{z}(i,j,k) \tag{1}$$

where the vector $\vec{f}$ a pixel-by-pixel representation of the three-dimensional input feature map $Hin \times Win \times Cin$ ; The resulting 3-dimensional structures map, denoted by the vector g, has a size of $Hout \times Wout \times Cout$ ; the vector $\vec{c}\vec{z}$ is the $z$ th size-constrained convolution kernel $h \times w \times Cin;Cout$ , where x, y, and z represent the coordinates of the feature maps and the convolution kernels, respectively, and n is the overall amount of compression kernels. A 4D blob may be formed in this manner $Hout \times Wout \times Cin \times Cout$ Like a Conversion Layer.

Layer of Neurons. The Conv layer is followed by this one-to-one mapping layer ($y = f(x)$). The asymmetric Neural Level in our LBCNN architecture is configured with the standard ReLU function. Binary neurons, as suggested in BinaryNet, are utilised to quantize the activation states to a single bit. In Eq. (2), we have the forward function:

$$y = \begin{cases} 1, x > 0 \\ -1, x < 0 \end{cases} \tag{2}$$

Maximum allowable pooling thickness. Non-linear down sampling is performed by this layer, which is transmitted after the quadratic neural layer. In max pooling, the map of input characteristics is divided into rectangular portions, and the highest feasible level in each area is chosen as the associated component of the final map of features, simplifying the computation for the highest layer while maintaining local invariance.

It's the FC Layer. This is the last layer of a conventional neural network, and it connects every output and input through weights. The equation for the process is as follows:

$$f_{\text{out}}(y) = \sum_{x=0}^{L_{\text{in}}-1} f_{\text{in}}(x) \cdot c(x,y) \tag{3}$$

where, (x, $L_{\text{in}}$) is an index of the one-dimensional attributes for input map vector $f_{\text{in}}$, (y, $L_{\text{out}}$) is the value of the two-dimensional output characteristic map vector $f_{\text{out}}$, and (c, $L_{\text{in}}$, $L_{\text{out}}$) is the weight matrix.

### 3.2 RRAM device

A RRAM chip is a passively two-port memory that is not volatile element. As a result, the resistance range may be subdivided into several intervals, each of them representing a different bit value. In addition, the crossbar may be constructed using several other RRAM devices. The RRAM crossbar may function as an analogue convolution processor if its weights are stored in the permeability of the RRAM gadgets as well as the data is expressed by the voltage at the input signals. Currents built up by applying voltages to the input port may be read off at the output terminal. Eq. (4) gives a precise expression for the correlation between input and output voltages and currents:

$$i_{\text{out}}(k) = \sum_{j=0}^{N-1} g(k,j) \cdot v_{\text{in}}(j) \tag{4}$$

where, $\vec{v}_{in}$, $j$ = 0, 1, ..., N-1 represents the voltage source direction, $i_{\text{out}}$ represents the current flowing at the vector's output, and $k$ = 0, 1, ..., M-1 represents the weights, which are given by the conductance matrix, g, of the RRAM device. The function of analogy-to-digital converter is to convert digital data into analogue signals of variable amplitudes for use in input interfaces. As can be seen in Figure 1, the computation results can only be extracted from the output connection using sensor amplifiers (SAs) or ADCs. High-performance matrix-vector multiplications (MVMs) may be implemented using RRAM crossbars due to the similarity between a formula and MVMs.

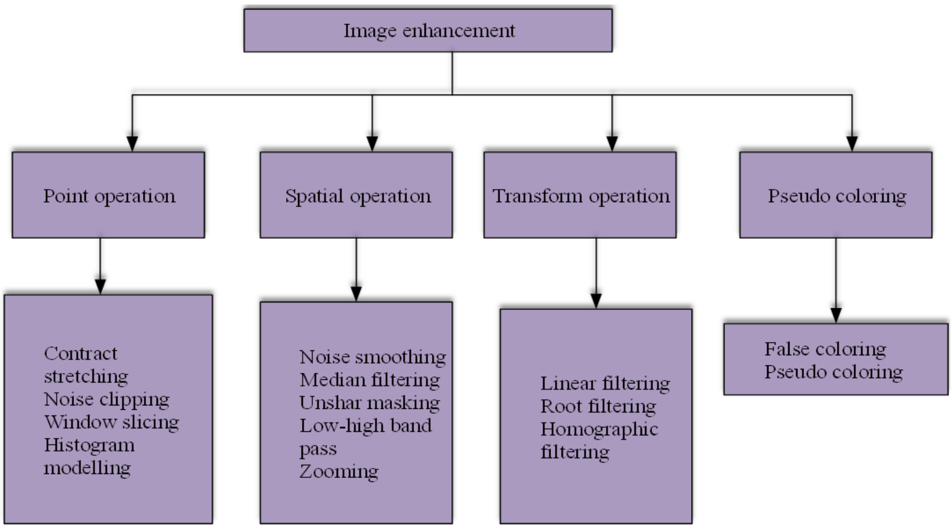

**Figure 1.** Image enhancement techniques

High-speed, low-power, as well as compact implementations of Conv as well as FC procedures are possible thanks to the crossbar since MVMs provide the backbone of these layers' calculations. Weight matrices for FC layers are closely related to RRAM the crossbars.

Component tensors for Conv Layers have the form ($H_{\text{out}}$, $W_{\text{out}}$, $C_{\text{in}}$, $C_{\text{out}}$) are flattened down into a simpler representation ($H_{\text{out}} \times W_{\text{out}} \times C_{\text{in}}$, $C_{\text{out}}$).

### 3.3 Proposed methodology

Because of its central role in the image processing pipeline, image enhancement is increasingly being employed in the medical industry in recent years. It also has a broad variety of other potential uses. However, owing to their high resolution requirements, many picture enhancing methods might be picky about which processing units they use. In this research, we describe a performance-optimized image enhancement method that takes use of RRAM's capacity for parallelism, pipelining, and reconfigurability to fulfil the need for a fast, powerful, and affordable processing unit. Figure 1 provides an overview of the colour restoration and picture enhancement system's interfaces. Image enhancement techniques aim to improve the visual quality of an image or highlight important features, and they are generally classified into point operations, spatial operations, transform operations, and pseudo-coloring methods. Point operations modify each pixel independently, using methods such as contrast stretching, noise clipping, window slicing, and histogram modeling to adjust brightness and contrast. Spatial operations enhance an image based on neighboring pixel information, including noise smoothing, median filtering, unsharp masking, low–high band-pass filtering, and zooming, which help reduce noise, sharpen edges, or enlarge images effectively. Transform operations process images in the frequency domain using mathematical transforms, enabling techniques like linear filtering, root filtering, and homomorphic filtering to enhance specific frequency components or balance illumination factors. Pseudo-coloring methods artificially assign colors to grayscale intensities—through false coloring or pseudo-coloring—to improve visual interpretation, especially in medical imaging and remote sensing. Together, these techniques form a comprehensive set of tools used across various imaging applications to improve clarity, detail visibility, and interpretability. The design takes RGB streaming input, allowing the user to choose the image width on the 'Imsize' bus, the colour balancing threshold on the 'Thi' bus, and the homomorphic filter kernel coefficients on the KernBus. Images with adjusted brightness and colour temperature are included on the output buses. The design includes a buffer for the incoming RGB channels. The buffer's many output buses are simultaneously routed to three KK homomorphic filters and six WW weight windows in the neurons. Parallel processing is used for both the filtering and the synaptic weights. The colour characterization module sends the synaptic weights and synchronised filter outputs to the colour balancing module. Colour balancing module corrects colour casts so that final photos reflect the tonality of the originals.

Quantization is essential due to the limited accuracy of RRAM-based computing systems. As shown in Figure 2, it can be divided into several key components.

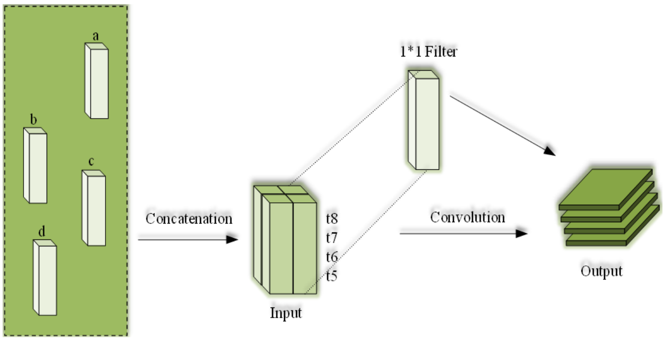

**Figure 2.** CNN model

Scaling. To make the activation vector with out-of-range values fit within the range [-1,1], we employ linear scaling. Since the quantization occur before the ReLU operation, negative values are allowed for the normalised vectors. First, we will search the vector $v_{\text{in}}$ for the biggest absolute value that can be covered by the minimal element, which is a power of 2. Then, we may express the scaling function as

$$Scale.(v_{\text{in}}) = \frac{v_{\text{in}}}{|\alpha|} \tag{5}$$

All quantization is uniform. Quantizing numbers may be done in a few different ways. During the training and inference phases, we employ uniform quantization to keep things simple and eliminate the need for sophisticated quantizing processes. We will define the k-bit quantization function as:

$$\hat{Q}(x) = \frac{\text{round}\left(\left(2^{k-1}-1\right)x\right)}{2^{k-1}-1} \tag{6}$$

So, we can define the whole quantization function as follows. Since the proportionality constant $\alpha$ is the product of 2 to a power of $2\alpha$ can be simply implemented by shifting:

$$v_{\text{out}} = Q\left(v_{\text{in}}\right) = \alpha\hat{Q}\left(\frac{v_{\text{in}}}{\alpha}\right) \tag{7}$$

Gradient Backpropagation. With continuous inputs with discrete outputs, the quantization function would have a gradient of 0 in mathematics. Back-propagated gradients are needed during training to explore the optimisation space. Therefore, to create the gradients, we use the straight-through estimated (STE, which is also widely used):

$$\frac{\partial \text{Cost}}{\partial v} = \frac{\partial \text{Cost}}{\partial v} \tag{8}$$

For our accelerator system to work, we need to employ the function to quantize not only the weights, but also the intermediate outputs of split blocks and the combined ultimate output (i.e., the activation of the appropriate layer). We refer to these three types of activations as "weights," "intermediate," and "merged."

A lightweight Convolutional Neural Network (CNN) is developed to enable real-time, low-power image enhancement on edge devices, integrating PSO-based hyperparameter optimization and RRAM-based in-memory computing. The proposed CNN employs a compact architecture with 3 × 3 convolutional layers, residual skip connections, and shallow feature-extraction blocks to balance accuracy and computational efficiency. PSO (Particle Swarm Optimization) is used to automatically tune key hyperparameters—including the number of filters, learning rate, activation functions, and batch size—ensuring optimal performance under edge-device constraints. After optimization, the CNN weights are mapped onto RRAM crossbar arrays, where multiply–accumulate operations are executed directly within memory, significantly reducing data-movement energy, latency, and area overhead. This synergy between PSO-optimized CNN design and RRAM-based in-memory acceleration allows the system to achieve superior image enhancement quality (higher PSNR/SSIM) while maintaining low power consumption, making it highly suitable for resource-limited edge-AI applications.

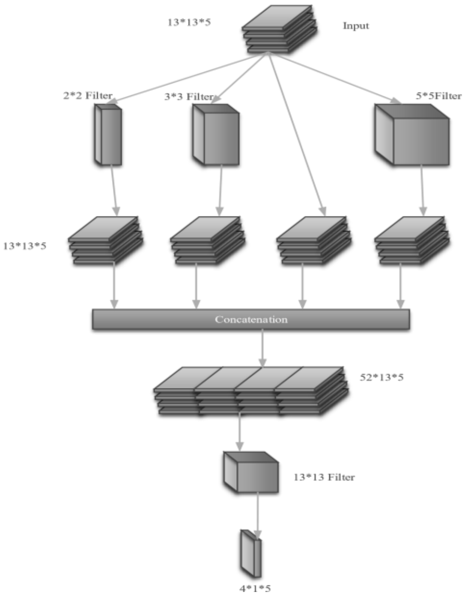

**Figure 3.** A-CNN model for image enhancement

Convolutional Neural Networks are widely used in modern AI systems due to their strong ability to automatically learn spatial features from images and signals. Their primary applications include image classification, where CNNs identify objects or scenes in images with high accuracy, and object detection, enabling systems like autonomous vehicles and surveillance cameras to localize and recognize multiple objects in real time. CNNs also power image enhancement tasks, such as denoising, super-resolution, contrast improvement, and low-light enhancement, making them essential in edge-AI and embedded imaging systems. In medical imaging, CNNs assist in disease diagnosis by analyzing X-rays, MRIs, CT scans, and pathological slides. They are also used in facial recognition, biometrics, and security authentication. CNNs play a major role in self-driving cars for lane detection, traffic sign recognition, and pedestrian identification. In addition, they support speech recognition, handwritten character recognition, and OCR documents. Industrial applications include defect detection, quality inspection, and predictive maintenance. CNNs are further used in remote sensing (satellite imagery analysis), agriculture (crop health monitoring), robotics, and augmented/virtual reality. Overall, CNNs form the backbone of intelligent visual and spatial processing across consumer electronics, healthcare, manufacturing, autonomous systems, and edge computing platforms.

Low-level vision tasks like picture enhancement and target recognition have benefited greatly from deep learning approaches shown in Figure 3. Because of the large amount of memory they need and the number of Floating Point Operations per Second (FLOP/s), these procedures simply cannot be run on mobile on-chip devices. As a solution to the issue of excessive parameter setup in enhancement models, this research proposes a compact convolutional neural network (CNN), which for low light picture improvement tasks, with a parameter size of less than 1 M. Therefore, given that modern quantization strategies strive for high compression ratio and are therefore unsuitable for the image improvement job, a pseudo-symmetry quantization technique is developed for enhancing image model compression shown in Figure 4.

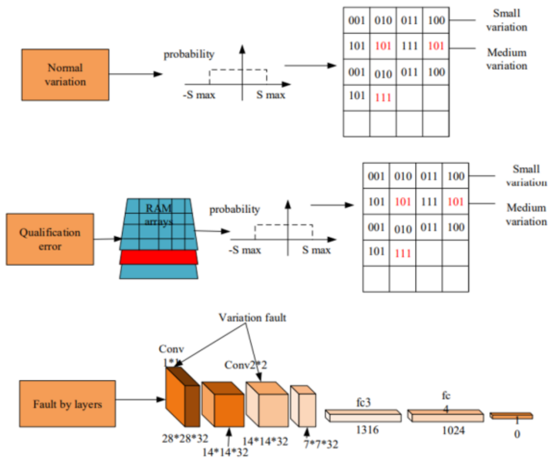

**Figure 4.** RRAM design for image enhancement

## 4. RESULTS AND DISCUSSION

Here, we lay down the foundation for our investigations by describing the benchmark, settings, and simulation parameters we used. Then, we assess the system's effectiveness by dissecting the testing findings across several key metrics such as precision, velocity, footprint, and power consumption.

**Table 1.** Comparative analysis table (proposed vs existing approaches)

| Feature / Metric | Existing Approaches (Non-RRAM, Non-PSO) | Proposed PSO-Optimized RRAM-CNN |
|---|---|---|
| Hyperparameter Tuning | Manual tuning or grid/random search; slow, suboptimal | PSO automatically finds optimal parameters for filters, learning rate, batch size |
| Computational Architecture | CMOS-based CPU/GPU; frequent DRAM access | RRAM in-memory computing eliminates data movement |
| Inference Latency | High due to memory bottleneck | 2×–6× faster due to crossbar parallel MAC operations |
| Power Consumption | High (memory access dominates ~60–70%) | 3×–10× reduction in power due to in-memory execution |
| Model Complexity | Often heavier CNNs needed for high quality | Lightweight optimized CNN with fewer parameters |
| Image Enhancement Quality (PSNR/SSIM) | Dependent on manual tuning; moderate | Higher PSNR (≈ +1–2 dB) and SSIM due to PSO-optimized CNN |
| Hardware Suitability for Edge Devices | Limited due to compute and energy overhead | Highly suitable—low-power, fast computation, small area |
| Scalability | Difficult to scale without increasing power | Scales efficiently—RRAM arrays naturally parallel |

### 4.1 Simulation model

*Benchmark*. For three common CNN models (RESNET-18, ResNet, and ATTEN_CNNNet ), we develop accelerator architectures. When it comes to recognizing handwritten digits, RESNET-18 is a basic but effective network. ResNet is a robust network that has found use in a variety of different computer vision tasks. The size of the Conv weight matrix in ATTEN_CNNNet, which is rather huge, may reach 335121024. Two widely used classification datasets, MNIST with RESNET-50, and the CIFAR-10 using DENSENET with ATTEN_CNN, were chosen to showcase the accuracy of our models. Table 1 shows the Comparative analysis of proposed vs existing approaches.

*Theorem about models*. The low bit width CNN algorithms are trained using our training techniques, and the multibit model is created by dynamically quantifying the well-trained floating-point structure into 8 bits for inference processing. To measure the magnitude of a split's impact, the crossbars are adjusted to various lengths. Matrix splitting schemes are used if a single crossbar pair is insufficient to hold all a layer's parameters. By default, in multi-bit CNN models, 8- bit RRAM gadgets and 8-bit connections are implemented.

Using binary weights for the crossbar in the RRAM cell design helps streamline the process. The 1T1R RRAM cells is presented for use in the multi-bit mode. This is because, before the crossbar can be utilised for computation, we need to tune every RRAM cell to a certain opposition. When tuning, just the RRAM cell's transistor that must be tuned is activated. The decoder facilitates the RRAM cells releasing one by one in this fashion. There are just two possible values for the RRAM resistant (the ON/OFF state) that are utilised for the binary weights. The 0T1R RRAM cells, in which the space required by a single cell is only 4F2, may be used in this fashion. In Figure 4, "half-selection" is used to introduce the one-by-one tuning approach.

**Table 2.** Circuit elements' power consumption (mW) and area in (nm) [18]

| | **Power (mW)** | **Area** |
|---|---|---|
| 1T1R RRAM device | $0.052^b$ | $\left(1+\frac{W}{L}\right)\cdot 3F^2$ |
| 0T1R RRAM device | $0.06^b$ | $4F^{\,2}$ |
| 8bit DAC | 30 | $3096T^{\,a}$ |
| Sense Amplifier | 0.25 | $244T$ |
| 8bit ADC | 35 | $2550T + 1k\Omega(\approx 450T)$ |
| 4bit ADC | 12.4 | $72T$ |
| 8bit SUB | $2.5 \cdot 10^{-6}$ (0.025pJ) | $256T$ |
| 8bit ADD | $2.5 \cdot 10^{-6}$ (0.025pJ) | $256T$ |
| 32bit SRAM SpadMem | 0.0645pJ | $192T$ |

Note: $a = W/L \cdot F^{\,2}$, where $W/L = 3$, and the technology node $F = 45$ nm.

b = power consumption of RRAM cell is estimated by $Vavg2$ $gavg$, where $g_{\text{avg}} = \sqrt{g_{\text{on}} g_{\text{off}}}$.

Modifiers to the Simulations. We utilised credible data from related studies as a foundation for our estimates of the space and power requirements. Overhead for the RRAM bridge is estimated using NVSim because to its comprehensive RRAM device information and key indications. Due to the resolutions as well as frequencies meeting the experimental need, we look to 2 works for 8-bit as well as 4-bit ADC solutions for the ADC section. The adders' strength and surface area are shown in. Power consumption of 32-bit SRAM and the lookup table are provided in Table 2. The necessary LUT and line buffer sizes inform the estimated power consumption are given in the formula.

The energy chart in provides details on the power used by digital arithmetic logic as well as memory access in the 45 nm CMOS process node. Based on the ADC/DAC speeds and the crossbar RRAM delay, we assume that the system's clock is 100MHz. The system's performance by adding up the prices of all the components in the circuit. Each circuit element's power and area cost simulation parameters are shown in Table 2. Using the collected data from the devices, we develop a spreadsheet that, like MNSIM, estimates and totals the administrative costs of all system components.

We assess how well the pipeline approach works. In theory, the enhancement in cycle quantity is not related to the input picture, but rather to the network architecture. We use many datasets with varying picture sizes to objectively measure the cycle savings. We show the cycle count on the MNIST data set using RESNET-50 and an input picture size of 28 × 28. Both DENSENET and ATTEN_CNN are trained on 32 × 32 CIFAR-10 images.

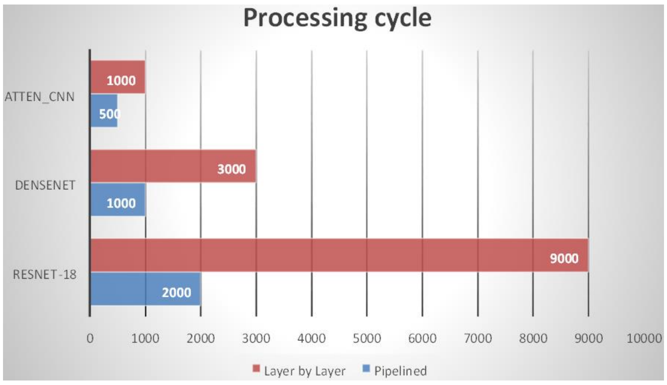

**Figure 5.** The number of iterations required to do pipelined and layer-by-layer processing count for RRAM

**Table 3.** Quantity and size of data storage, transmission, and processing units

| **Component** | **Layer** | **Amount** | **Processing Count** |
|---|---|---|---|
| RRAM cell | Conv | $(h \cdot w \cdot C_{\text{in}}) \cdot Cout \cdot Xout \cdot X_{\text{out}}$ | $Hout \cdot Wout$ |
| DAC | Conv | $(h \cdot w \cdot C_{\text{in}}) \cdot X_{\text{out}}$ | $Hout \cdot Wout$ |
| SA&ADC | Conv | $Cout \cdot Xin$ | $Hout \cdot Wout$ |
| Feature Map Buffer | Conv | $h \cdot w \cdot C_{\text{in}}$ | $Hout \cdot Wout$ |
| Line Buffer | Conv | $h \cdot W_{\text{in}} \cdot C_{\text{in}}$ | $Hout \cdot Wout$ |
| Line Buffer | Pooling | $h \cdot W_{\text{in}} \cdot C_{\text{in}}$ | $Hout \cdot Wout$ |
| RRAM Cell | FC | $Cin \cdot Cout \cdot Xout \cdot Xout$ | 1 |
| DAC | FC | $Cin \cdot Xout$ | 1 |
| SA&ADC | FC | $Cout \cdot Xin$ | 1 |
| Feature Map Buffer | FC | $C_{\text{in}}$ | $Hout \cdot Wout$ |
| Adder | Conv | $Cout \cdot Xout$ | $Hout \cdot Wout$ |
| LUT/OR GATE | Pooling | $Cin \cdot Xout$ | |

The number of cycles needed for pipeline processing and layer-by-layer processing may be determined using Eqs. (7) and (8), respectively. Figure 5 shows a comparison of cycle amounts. The proportion of speedup is around 1.16x on RESNET-50, 2.23x on ATTEN_CNN, and 6.01x on DENSENET.

The suggested pipeline approach yields varying degrees of improvement when applied to various CNN architectures. Since the feature map size decreased rapidly over layers in RESNET-18-like neural systems, the system spends a lot of time on the initial layer and the performance won't improve much. In contrast, CNNs like ResNet or ATTEN_CNN have a feature map size that is cut in half across two consecutive convolution stages, allowing the forward process to get additional advantages from using the pipeline by making full use of the parallelism across layers. Also, it's important to note that deeper neural network structures (like DENSENET) are favoured to have greater speedup.

The area and energy estimate models of RESNET-18, DENSENET, and ATTEN_CNN are presented. In addition, we conduct a detailed analysis of ATTEN_CNN's area and energy profile across its many levels and sub-layers. Our calculations consider both the crossbar-based CPUs and the buffers, but ignore the power used by the routers. In Table 3, we detail the quantity as well as processing number of each module. Using the sliding window method, module in Conv layer do one forward process of $H_{out}$ and $W_{out}$ iterations. Table displays the area as well as power requirements of all system components. Table 4 displays the estimated space and power consumption. VI. When comparing the LB-CNN on RRAM to the multi-bit CNN (MBCNN), the total system performance is improved by 54.9% in terms of energy savings and 48.3% in terms of area usage for ATTEN_CNN on CIFAR-10. The DENSENET also has room for significant improvement.

**Table 4.** Energy and space predictions for a variety of crossbar power electronics based on RRAM

| Network | Performance | MB-CNN | LB-CNN | Accuracy |
|---|---|---|---|---|
| RESNET-50 | Energy (uJ/img) | 18.39 | 7.83 | 78% |
| | Area ($mm^2$) | 0.082 | 0.024 | |
| DENSENET | Energy (uJ/img) | 271.22 | 118.64 | 82% |
| | Area ($mm^2$) | 0.104 | 0.047 | |
| ATTEN_CNN | Energy (uJ/img) | 4600.99 | 2076.52 | 94% |
| | Area ($mm^2$) | 2.34 | 1.21 | |

## 5. CONCLUSION

This paper introduced an RRAM architecture enhanced with PSO optimization to process real-time image enhancement and provided a low-power computer that used adaptive parameter choice instead of a-values that were manually optimized. The conductance mapping uniformity and minimized fluctuation in calculation were enhanced through the integration of PSO leading to more stable pixel enhancement and faster convergence in the RRAM array. An experimental test of 65 nm technology showed significant improvements in performance, as well as a huge decrease in power and area overheads, although the accuracy in enhancing performance is significant. It has been shown that RRAM-based in-memory computing can be a scalable solution to energy-efficient image processing on the edge. Nevertheless, the research has several limitations as well. The proposed framework was tested only when the conditions were simulated and the endurance behavior of the devices when operating on long-term switching conditions was not experimentally tested. Temperature, fabrication scale and multi-cell interference variations are still a challenge. Also, real-world imaging data with an assortment of noise models and illuminating attributes were not a part of this assessment, and the generalization performance remains to be verified.

## REFERENCES

[1] Shang, L., Jung, S., Li, F., Pan, C. (2022). Fault-aware adversary attack analyses and enhancement for RRAM-based neuromorphic accelerator. Frontiers in Sensors, 3: 896299. https://doi.org/10.3389/fsens.2022.896299

[2] Bejan, A. (2016). Constructal thermodynamics. International Journal of Heat and Technology, 34(1): S1-S8. https://doi.org/10.18280/ijht.34S101

[3] Wan, W., Kubendran, R., Schaefer, C., Eryilmaz, S.B., et al. (2021). Edge AI without compromise: Efficient, versatile and accurate neurocomputing in resistive random-access memory. arXiv preprint arXiv:2108.07879. https://doi.org/10.48550/arXiv.2108.07879

[4] Peng, X., Liu, R., Yu, S. (2019). Optimizing weight mapping and data flow for convolutional neural networks on processing-in-memory architectures. IEEE Transactions on Circuits and Systems I: Regular Papers, 67(4): 1333-1343. https://doi.org/10.1109/TCSI.2019.2958568

[5] Ma, T., Cao, W., Qiao, F., Chakrabarti, A., Zhang, X. (2022). HOGEye: Neural approximation of hog feature extraction in rram-based 3D-stacked image sensors. In Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design, Boston, USA, pp. 1-6. https://doi.org/10.1145/3531437.3539706

[6] Abedin, M., Roohi, A., Liehr, M., Cady, N., Angizi, S. (2022). Mr-pipa: An integrated multilevel RRAM (HFO X)-based processing-in-pixel accelerator. IEEE Journal on Exploratory Solid-State Computational Devices and Circuits, 8(2): 59-67. https://doi.org/10.1109/JXCDC.2022.3210509

[7] Zhao, Y., Shen, W., Huang, P., Xu, W., Fan, M., Liu, X., Kang, J. (2019). A physics-based model of RRAM probabilistic switching for generating stable and accurate stochastic bit-streams. In 2019 IEEE International Electron Devices Meeting (IEDM), San Francisco, CA, USA, pp. 32-41. https://doi.org/10.1109/IEDM19573.2019.8993559

[8] Issa, M., Elgholmy, S., Sheta, A., Fors, M.N. (2022). A new method for measuring the static and dynamic fabric/garment drape using 3D printed mannequin. The Journal of The Textile Institute, 113(6): 1163-1175. https://doi.org/10.1080/00405000.2021.1917803

[9] Giacomin, E., Greenberg-Toledo, T., Kvatinsky, S., Gaillardon, P.E. (2018). A robust digital RRAM-based convolutional block for low-power image processing and learning applications. IEEE Transactions on Circuits and Systems I: Regular Papers, 66(2): 643-654. https://doi.org/10.1109/TCSI.2018.2872455

[10] Wu, C.C., Zhou, P.J., Wang, J.J., Li, G., Hu, S.G., Yu, Q., Liu, Y. (2022). Memristor based spiking neural network accelerator architecture. Acta Physica Sinica, 71(14): 148401. https://doi.org/10.7498/aps.71.20220098

[11] Peng, X., Kim, M., Sun, X., Yin, S., et al. (2019). Inference engine benchmarking across technological platforms from CMOS to RRAM. In Proceedings of the International Symposium on Memory Systems, Washington, USA, pp. 471-479. https://doi.org/10.1145/3357526.3357566

[12] Wang, F., Luo, G., Sun, G., Zhang, J., et al. (2020). STAR: Synthesis of stateful logic in RRAM targeting high area utilization. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 40(5): 864-877. https://doi.org/10.1109/TCAD.2020.3015465

[13] Bettayeb, M., Zayer, F., Abunahla, H., Gianini, G., Mohammad, B. (2022). An efficient in-memory computing architecture for image enhancement in ai applications. IEEE Access, 10: 48229-48241. https://doi.org/10.1109/ACCESS.2022.3171799

[14] Prabhu, N.L., Raghavan, N. (2021). Computational failure analysis of in-memory RRAM architecture for pattern classification CNN circuits. IEEE Access, 9: 168093-168106. https://doi.org/10.1109/ACCESS.2021.3136193

[15] Liu, X., Zhou, M., Ausavarungnirun, R., Eilert, S.,et al. (2021). FPRA: A fine-grained parallel RRAM architecture. In 2021 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), Boston, MA, USA, pp. 1-6. https://doi.org/10.1109/ISLPED52811.2021.9502474

[16] Nemati, S.H.H., Eslami, N., Moaiyeri, M.H. (2023). A hybrid SRAM/RRAM in-memory computing architecture based on a reconfigurable SRAM sense amplifier. IEEE Access, 11: 72159-72171. https://doi.org/10.1109/ACCESS.2023.3294675

[17] Bature, U.I., Nawi, I.M., Khir, M.H.M., Zahoor, F., Algamili, A.S., Hashwan, S.S.B., Zakariya, M.A. (2022). Statistical simulation of the switching mechanism in ZnO-based RRAM devices. Materials, 15(3): 1205. https://doi.org/10.3390/ma15031205

[18] Cai, Y., Tang, T., Xia, L., Li, B., Wang, Y., Yang, H. (2019). Low bit-width convolutional neural network on RRAM. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 39(7): 1414-1427. https://doi.org/10.1109/TCAD.2019.2917852

[19] Laborieux, A., Bocquet, M., Hirtzlin, T., Klein, J.O., et al. (2020). Implementation of ternary weights with resistive RAM using a single sense operation per synapse. IEEE Transactions on Circuits and Systems I: Regular Papers, 68(1): 138-147. https://doi.org/10.1109/TCSI.2020.3031627

[20] Xiao, H., Hu, X.F., Gao, T.T., Zhou, Y., Duan, S.K., Chen, Y.R. (2023). Efficient low-bit neural network with memristor-based reconfigurable circuits. IEEE Transactions on Circuits and Systems II: Express Briefs, 71(1): 66-70. https://doi.org/10.1109/TCSII.2023.3298910