Analog Image Denoising with an Adaptive Memristive Crossbar Network

O. Krestinskaya 1, K. Salama 1 and A.P. James2
1King Abdullah University of Science and Technology, Saudi Arabia; 2Digital University Kerala

Abstract—Noise in image sensors led to the development of a whole range of denoising filters. A noisy image can become hard to recognize and often require several types of post-processing compensation circuits. This paper proposes an adaptive denoising system implemented using analog in-memory neural computing network. The proposed method can learn new noises and can be integrated into or alone with CMOS image sensors. Three denoising network configurations are implemented, namely, (1) single layer network, (2) convolution network, and (3) fusion network. The single layer network shows the processing time, energy consumption and on-chip area of 3.2µs, 21nJ per image and 0.3mm² respectively, meanwhile, convolution denoising network correspondingly shows 72µms, 236µJ and 0.48mm². Among all the implemented networks, it is observed that performance metrics SSIM, MSE and PSNR show a maximum improvement of 3.61, 21.7 and 7.7 times respectively.

Index Terms—Memristor, RRAM Denoising, Near-Sensor Processing, Neural Networks

I. INTRODUCTION

The CMOS image sensors capture the pixel information through photo-diodes and CMOS-based amplification circuits [1]. The image pixels’ noise gets injected due to non-idealities in the integrated devices and is impacted by temperature, frequency of usage, and device parasitics [2]. Traditionally, these analog signal noises are suppressed by dedicated filters [3] and converted to digital domain, which rejects certain amount of signal noise and allows using the data for further noise compensation by a digital microprocessor. An alternative school of thought is to make the analog sensor intelligent by incorporating denoising filters directly into the pixels [4], [5]. We propose incorporating denoising into pixel sensors using a continuously trainable memristor-based neural network in analog domain for near-sensor on edge processing. The neural network can be trained for different types of noises and compensates for noises originating from the sensor and related interface circuits.

When using a neural network for inference tasks with noisy images, the commonly known approaches are to apply post-processing techniques or to incorporate the denoising as part of learning. The system pre-trained with noiseless data fails to process, recognize or classify noisy images [6], [7]. Most commonly, image sensor noise is compensated by converting sensor outputs to the digital domain using ADC, and applying denoising and filtering techniques [3], [8] (shown in Fig. 1 (a)). To use this digital data in an analog memristor-based neural network, it should be converted back to analog domain for dot product computation. In this conversion, ADC and DAC contribute to power consumption, speed and on-chip area overhead [9].

For analog memristor-based neural network applications, the desirable approach is reading sensor output directly without converting it to the digital domain (Fig. 1 (b)) [10]. However, retraining of the whole network for new noisy inputs is computationally expensive, and requires additional backpropagation circuits and memory contributing to power and area overhead, especially in analog domain [11], [12]. Moreover, in a real-time system, the input noise can be unknown, making it challenging to pre-train the system for a particular noise type.

To solve these issues, we propose integrating a small trainable memristor-based denoising network close to the sensor that can be trained with varying noise and is able to denoise the data without retraining the whole system (Fig. 1 (c)).

The denoising network can be as small as a single-layer dense network without needing backpropagation circuits to retrain it. In turn, training circuits of reduced complexity are more suitable for implementation on edge devices. The other approach is to introduce small convolution/deconvolution (CNN) network, which requires fewer memristors to store the weights, but has more complex training circuits.

Previously, complex software-based deep neural networks have been used for image denoising [13], [14], [15]. Also, RRAM crossbar based networks have been useful for image storage [16] and edge detection [17]. The studies of memristor based neural networks for denoising applications are very limited. The existing state-of-the-art works focus on cellular neural networks [18], [19], only Gaussian noise.
[18]–[20] and binary images [18], [20], containing limited evaluation of denoising metrics. In this work, we demonstrate the generalized architecture suitable for various types of noises for binary and RGB images, compare the performance to the most common denoising techniques and illustrate the network fusion approach for efficient processing of different types of noises. We evaluate the performance of the proposed system considering memristor non-idealities, and estimate the processing speed, energy consumption and on-chip area of the complete system.

II. DENOISING NETWORK FOR NEAR-SENSOR PROCESSING

A. Origin of noise

When developing denoising method for image sensors, multiple noise types need to be considered. The most common four types are considered in the paper, and relevance of such noise in image sensors is listed as:

1) **Gaussian:** This noise is additive to the pixel, reflective of Johnson–Nyquist noise (thermal noise) and reset noise of capacitors (kTC noise) [21]. The amplifier noise is unavoidable and forms part of the read noise from the image sensors.

2) **Salt and pepper (S&P):** This noise originates in data conversion and transmission processes, such as from errors in ADC circuits. It changes a dark pixel to a bright or from bright to the dark. Pixel interpolations and mean filtering are popular approaches to compensate for this noise [22].

3) **Poisson (Shot) noise:** This noise is approximated with a Poisson distribution and at very high-intensity levels as Gaussian noise. It originates from statistical quantum fluctuations that occur in photodetector at high exposures and can also arise from dark leakage current in the image sensor [21].

4) **Speckle:** This noise reflects the unwanted modifications of the desired signal. As the scatterers are not identical, the detected signal becomes sensitive to small changes in object surfaces. This is granular interference to the signal, and is very common in radars and medical images [23].

B. Denoising network architecture

The denoising network for near-sensor processing and retraining for different types of noises can be implemented in two ways: (1) single-layered dense network and (2) multi-layered convolution/deconvolution network (CNN). Both approaches, as shown in Fig. 2, have advantages and drawbacks. In both approaches, positive and negative weights of the network are implemented with two memristors and opamp based readouts. Also, we assume crossbars to be separated to $256 \times 64$ crossbar tiles [24].

1) **Dense Network for denoising:** Fig. 2 (a) illustrates the training of a single-layer fully-connected denoising network. Comparing to backpropagation, retraining of such a network is less complicated. It is based on root mean square error (RMSE) calculation between the ideal and real outputs and updating RRAM weights according to this error. It involves multiplication of error with the corresponding inputs. Such a network is smaller and consumes less energy due to simplified training. But as in all shallow networks, the number of devices to train is higher than in a CNN when aiming for the same inference performance (Table III).

2) **Training of CNN denoising network:** Fig. 2 (b) illustrates more complex denoising network involving convolution and deconvolution parts. The noisy image is applied to a convolution layer that extracts useful features, and the pooling layer reduces the dimensions of the images. Then, the 3D convolution collects all the feature maps from the previous
TABLE I: Comparison of SSIM, MSE and PSNR performance metrics for noisy and denoised images, and effect of memristor non-idealities.

| Noise        | Gaussian | Salt and Pepper | Poisson | Speckle |
|--------------|----------|-----------------|---------|---------|
| **Comparison of noisy and denoised images, MNIST:** | [0.29, 20.5] | [4.82, 0.18] | [2.90, 17.0] | [0.16, 17.1] |
| **(dB)**     | [0.69, 0.51] | [-, 0.69] | [0.30, 0.16] | [-, 0.16] |
| **Simulation results and comparison** | [0.10, 0.06] | [0.07, 0.07] | [0.69, 0.74] | [-, 0.74] |
| **Effect of memristor non-idealities** | [18.9, 21.9] | [6.98, 18.7] | [13.1, 18.6] | [-, 17.0] |

Limited number of stable resistive states [256, 128, 64, 16, 2]

| Noise        | Gaussian | Salt and Pepper | Poisson | Speckle |
|--------------|----------|-----------------|---------|---------|
| **MNIST (Gaussian, 0.01)** | [0.01, 0.01, 0.01, 0.01] | [10.0, 10.0, 10.0, 10.0] | [0.01, 0.01, 0.01, 0.01] | [10.0, 10.0, 10.0, 10.0] |
| **Dropout (10, 20, 30)** | [0.01, 0.01, 0.01, 0.01] | [10.0, 10.0, 10.0, 10.0] | [0.01, 0.01, 0.01, 0.01] | [10.0, 10.0, 10.0, 10.0] |
| **Pruning (10, 20, 30)** | [0.01, 0.01, 0.01, 0.01] | [10.0, 10.0, 10.0, 10.0] | [0.01, 0.01, 0.01, 0.01] | [10.0, 10.0, 10.0, 10.0] |

III. SIMULATION RESULTS AND COMPARISON

1) Denoising performance: The proposed system has been tested for MNIST [25] and CIFAR-10 [26] datasets (Fig. 3 (a-b)) for eight different input noises [27]: Gaussian noise with $\sigma = 0.01, 0.1, 0.5$, Salt and Pepper (S&P) noise of 10%, 25%, and 50%, Poisson, and Speckle noise shown. In Table I, Structural similarity index (SSIM) [28], mean square error (MSE) and Peak Signal-to-Noise Ratio (PSNR) have been measured for all the images calculating the average score for the whole testing set. For MNIST dataset, the dense network results are slightly better than CNN, as convolution and pooling compresses some structural information. Denoised images have higher SSIM improved up to 2.28 and 3.61 times for MNIST and CIFAR images. The gap between SSIM of noisy and denoised images becomes more significant for the case where structural information is nearly destroyed by noise, e.g. Gaussian noise with $\sigma = 0.5$ or S&P noise of 50%. Mostly, MSE of the denoised images is 10 times lower than of the noisy ones, which can be up to 16.3 and 21.7 times lower for MNIST and CIFAR datasets for high Gaussian noise. On average, PSNR is increased by 5-10dB for images with medium noise and up to 20dB for images with high noise. Maximum PSNR improvement is 5.1 and 7.7 times for MNIST and CIFAR images.

2) Device non-idealities, dropout and pruning: Fig. 3 (c-d) and Table I illustrates how the denoised image’s quality is affected by memristor non-idealities [29], dropout and pruning. For limited number of stable resistive states $L < 16$, denoised image quality is significantly reduced and is worse than the noisy images. The desirable number of states is $L = 64-128$ for MNIST and $L = 256$ for CIFAR. The conductance variation with $\sigma = 0.025$ reduces SSIM and PSNR by 18% and 40%, respectively. Fig. 3 (e) demonstrates how dropout (randomly dropping out input neurons [30]) and pruning (randomly disconnecting memristor devices introducing sparsity [31]) can be used to reduce the number of active devices in the fully connected denoising layer. The dropout and pruning of 20% devices cause around 2dB reduction of PSNR (Table I).

III. Comparison to the conventional methods: Table II shows the comparison of dense network performance with conventional denoising methods [32] for MNIST dataset, where top 1 result is highlighted in yellow and top 3 are highlighted in green. For conventional methods, the highest scores from several iterations are considered. The proposed method is more generalized showing stable performance and is among top 3 best results for all types of noises.

4) Hardware performance estimation: Table III shows on-chip area, power consumption, processing time and energy per image denoising for inference and training circuits in 65nm CMOS technology. We take into account measurements data from $256 \times 64$ crossbar tiles with power consumption of $111\\mu W$ and on-chip area of $2800\mu m^2$ per tile [24]. A
single read and write operation is estimated as 50ns and 80ns (12.5-20MHz) with 50 update pulses on average [33], [34]. The sequential and parallel operations are considered for both networks. In sequential inference in dense network, the outputs in each tile share a single readout circuit, while in parallel processing each output is connected to a separate readout. In dense network, we consider opamp based summation circuits to sum up the outputs from several tiles. In CNN design, we map CNN kernels to the crossbars by unfolding and duplicating weights [35], [36]. In parallel processing in CNN, the convolution kernels are duplicated, which require 400 times more RRAM devices. In CNN, a single readout and ADC per tile is assumed. The opamp design adapted from [37] consumes 262.4µW of power and 25.8µm² of area. We use an 1.731µm²/0.48µW analog current multiplier [38], adding opamp based voltage-to-current converters, which makes the area and power of a multiplier 53.33µm² and 525.28µW. For storage of intermediate outputs, we assume 8-bit 3000µm²/40µW ADC [24] and 6T SRAM cells of 0.525µm² [39]. The pooling is CNN is performed by an opamp-based max-pooling circuit [40].

In the training circuits, we consider a single difference amplifier, multiplier, ADC per output crossbar tile in both networks. The training time is estimated considering forward and backward propagation, CMOS processing and update time. All crossbar tiles are updated in parallel sequentially inside the tile. We do not consider digital control circuits, DACs and adders in CNN design, assuming them to be a part of the control circuits overhead calculation. The energy consumption of the proposed dense denoising network is the same as of a digital pre-processor for median filtering and 22 times lower than for NL Means filtering for MNIST images [41]. Also, the dense network with parallel processing has 47 times smaller area and 22 times lower power consumption than state-of-the-art 65nm 8-bit image denoising accelerator [42].

![Network fusion approach for denoising images under unknown variable noise conditions.](image1)

![Classification accuracy improvement with denoising for different types of noises for (a) MNIST and (b) CIFAR10.](image2)

### IV. Conclusion

We proposed adaptive denoising memristive networks that can integrate into analog image sensors and trained on-chip adapting to new noisy conditions. This approach allows re-training only a small denoising network without retraining the entire system for new noisy conditions. For adapting to different types of noises, the fusion approach is introduced. The method is verified for image classification. The application scope of the proposed system can be extended to different problems that require real-time image denoising as a pre-processing step.

![Network fusion approach for denoising images under unknown variable noise conditions.](image1)

![Classification accuracy improvement with denoising for different types of noises for (a) MNIST and (b) CIFAR10.](image2)
