IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 7, 2022, accepted April 22, 2022, date of publication May 2, 2022, date of current version May 9, 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3171799*

# An Efficient In-Memory Computing Architecture for Image Enhancement in AI Applications

**MERIEM BETTAYEB[1], FAKHREDDINE ZAYER[1,2], HEBA ABUNAHLA[1], GABRIELE GIANINI[3,4], (Member, IEEE), AND BAKER MOHAMMAD[1], (Senior Member, IEEE)**

[1]System on Chip Laboratory, Electrical Engineering and Computer Science Department, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates
[2]Center for Autonomous Robotic Systems (KUCARS), Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates
[3]Department of Computer Science, Università degli Studi di Milano, 20133 Milan, Italy
[4]Emirates Innovation Center (EBTIC), Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates

Corresponding author: Baker Mohammad (baker.mohammad@ku.ac.ae)

This work was supported in part by Khalifa University Competitive Internal Research Awards (CIRA) Project 2018-026 and in part by the System on Chip Laboratory.

**ABSTRACT** Random spray retinex (RSR) is an effective image enhancement algorithm owing to its effectiveness in improving the image quality. However, the computing complexity of the algorithm, the required hardware resources, and memory access hamper its deployment in many application scenarios, for instance, in IoT systems with limited hardware resources. With the rise of artificial intelligence (AI), the use of image enhancement has become essential for improving the performance of many emerging applications. In this paper, we propose the use of RSR as a preprocessing filter before the task of semantic segmentation of low-quality urban road scenes. Using the publicly available Cityscapes dataset, we compared the performance of a pre-trained deep semantic segmentation network on dark and noisy images with that of RSR preprocessed images. Our findings confirm the effectiveness of RSR in improving segmentation accuracy. In addition, to address the computational complexity and suitability of edge devices, we propose a novel and efficient implementation of RSR using resistive random access memory (RRAM) technology. This architecture provides highly parallel analog in-memory computing (IMC) capabilities. A detailed, efficient, and low-latency implementation of RSR using RRAM-CMOS technology is described. The design was verified using SPICE simulations with measured data from the fabricated RRAM and 65 nm CMOS technologies. The approach presented here represents an important step towards a low-complexity, real-time hardware-friendly architecture and the design of retinex algorithms for edge devices.

**INDEX TERMS** Memristor crossbar, multiply and add (MAC) operations, random spray retinex, scale-to-max filtering, in-memory computing.

## I. INTRODUCTION

Digital images captured in various application areas, such as medical imaging, space exploration, and underwater environments are often characterized by low quality. This could be due to insufficient lighting conditions or the dynamic characteristics of the environment. This is also the case for images acquired for autonomous vehicle driving applications. An issue common to these images is that the difference in lightness between the bright and dark areas is large. This poses a challenge to subsequent processing tasks, such as image segmentation. Thus, a quality enhancement preprocessing step is normally added to the pipeline with the aim of increasing the overall effectiveness of the processing.

In this respect, one has to consider the tension between global and local aspects of processing. Increasing the brightness of the overall image will improve the visibility of the dark areas, but reduce the visibility of the details in the bright areas. Several global techniques have been developed to overcome the low-light enhancement problem, such as histogram equalization [1], gamma correction and tone mapping [2]. Nevertheless, these techniques result in a sort of over-exposure when the grey level in an image is concentrated at a certain intensity [3]. In addition, intensity saturation would result from inconsistent enhancement performed by these algorithms as they rely on global information from

The associate editor coordinating the review of this manuscript and approving it for publication was Donato Impedovo.

the image. This calls for the use of a differential treatment of the different image regions, based on both local and global information.

To address these challenges, more complex algorithms have been developed, such as adaptive contrast enhancements [3], [4], adaptive histogram equalization [5], [6], and Spatial Color Algorithms (SCA) [7], [8]. SCAs are driven by Retinex principles, inspired to the behavior of the Human Visual System (HVS). These algorithms – RSR [9], STRESS [10], STAR [11], and ReMark [12], to mention a few that belong to the Milano Retinex family [13] – are widely employed to enhance real-world images. They implement two important characteristics of the human color vision system: (i) the independent analysis of the color components of the visual signal; (ii) the color adjustment based on local spatial and visual information. The algorithms of the Retinex family – originally created as a model of the Human Vision System – turned out to be endowed with several desired image enhancement properties.

These algorithms use both local and global information, and this results in efficient enhancement of the dark areas.

However, these algorithms are typically computationally intensive: this makes it hard to deploy them in real-time applications (especially on resource-limited edge devices).

To address these challenges, this paper presents a novel low-complexity and real-time HW-friendly architecture and design of the Retinex algorithm. Furthermore, to the best of our knowledge, this work is the first to utilize the In-Memory Computing (IMC) feature of emerging memristor devices to reduce the power and improve the speed of the conventional Retinex algorithms.

Memristors are a type of Resistive RAM (RRAM) technology that provides low-power solutions at a low cost. A memristor contains a thin oxide film sandwiched between two metal electrodes [14] and has the ability to save information with zero leakage current, high endurance, relatively fast write time, and small cell size. Furthermore, the memristor has both storage and computing capabilities, which makes it a suitable building block for IMC [15]–[17].

The new paradigm presented in this paper supports parallel computing and provides efficiency gains in area and power. It utilizes the analog computations in the memristor crossbar and uses the same physical elements for both processing and storage [15], [16], [18]. As a result, it substantially reduces the computing complexity resulted from the data-intensive Retinex algorithm.

As a case study, among the many Retinex variants, Random Spray Retinex (RSR) has been selected as a test for the innovative proposed architecture due to its proven high effectiveness in image enhancement [19], (we quantify the image quality using several quality assessment metrics). We use RSR as a preprocessing filter before the task of semantic segmentation of low-quality urban road scenes. Using the publicly available Cityscapes dataset [20], we compare the performance of a pre-trained deep semantic segmentation network on dark and noisy images and on RSR pre-processed images. In addition, the image pixel accuracy is studied for different numbers of bits to understand the impact of this choice approach on accuracy and energy consumption. The latter study is essential for the implementation phase as it helps to identify the desired quantization level.

The main contributions of this paper are summarized as follows:

1) We implement an efficient RSR algorithm using emerging memristor technology.
2) We use the RSR algorithm as preprocessing filter for image enhancement before applying image segmentation to demonstrate the effectiveness of such an implementation in a practical setting.

The remainder of the paper is organized as follows. Section II presents a thorough background on Retinex algorithms. After that, the proposed architecture for memristor-based Retinex is described in Section III. Then, the simulation results for the proposed architecture are provided in Section IV. Finally, Section V presents the conclusions and the planned future work.

## II. RELATED WORKS

### A. THE RETINEX ALGORITHM FAMILY

Hereafter, we frame the RSR algorithm within the family of Retinex algorithms and summarize its main variants.

Several Retinex algorithms have been developed throughout the years [21], based on the original Retinex, which was formulated in the 60's and relied on random paths. This algorithm was defined by several mechanisms that specified the processing of the information collected by the random path and its progressive integration into the corrected output image. Among those mechanisms, the most characteristic was the so-called *reset*, which allowed to refer the correction to maximum values found in the vicinity of the corrected pixel. The reset mechanism was preserved by several algorithms later developed in the Retinex family, among them the algorithms of the subfamily called Milano-Retinex [13], to which RSR belongs to. For the sake of completeness we mention that this core mechanism was dropped by other simplified algorithms, among them the so-called NASA-Retinex [22]: this simplification made the processing more efficient, at the price of giving up to one of the distinctive characteristics of the Human Visual System.

The reset-based Retinex algorithms produce an enhanced image where the chromatic dominant of the light and any smooth gradients are partially suppressed [23], while scene details and edges are enhanced [24]. The Retinex algorithm performs spatial color processing, i.e. it processes colors based on their positions and combines this information with a specific equation aggregator. As for the spatial exploration, some use random walk processes [25], or their probabilistic representation [12], and some others use point sampling processes [26], [27] or their probabilistic representation [28]–[31]), while the aggregation can involve various kinds of averaging of the local intensity maxima.

RSR relies on a point sampling process, and was created by Provenzi and others [9] based on the observation that a point sampling process can explore the surroundings of a pixel in a less redundant way than a random walk.

The pixel samples were dubbed sprays, hence the name ''Random Spray Retinex (RSR)''. In RSR, for each pixel to be corrected – the target – the algorithm generates several collections of informative pixels in the neighborhood, extracts from each collection the maximum intensity and eventually uses a suitable average of those values as a white reference for rescaling the input brightness of the target to compute the output value. The authors demonstrated that the spray technique outperforms the path-based strategy for neighbor information collection.

Later, Banić *et al.* [19] proposed Light RSR, which provided an efficient algorithmic implementation reducing the computational time while keeping the same spatial sampling method of RSR. Further studies by the same authors in [32] studied the feasibility of using the RSR for global illumination estimation using the local RSR results. Moreover, a new light Random Spray Retinex-based image enhancement method was proposed by Banić and Lončarić [33]. It can be used as a color correction method, a brightness adjustment method, or both. Although it operates locally, it performs a fixed number of operations per pixel, which means that its computational speed is almost independent of the parameter values used.

Later on, RSR was combined with the Automatic Color Equalization (ACE) algorithm. The two algorithms are complementary in their spatially variant approach. As a result, the output images of the two algorithms exhibit complementary advantages and defects [34]. RSR shows good saturation properties, but has insufficient detail recovering capabilities. ACE, instead, has a propensity to put in evidence details, but it tends to wash out images.

Furthermore, Lecca *et al.* [35] modified the RSR algorithm to control the locality of the color filtering by considering the spatial image information. The spatial information is integrated into the RSR channel lightness computation at each pixel through a weighting function inversely proportional to the distance from the spray center. Finally, Tanaka *et al.* [36] produced two variants of RSR by concentrating on the region of interest (ROI): the first variant proposed a cone distribution based on anatomical data as a ROI, while the second variant focuses on the visual field information's visual resolution and considers it an ROI.

Among the most efficient and fastest implementations of RSR is FuzzyRSR [26], which exploits the same spray for the correction of several pixels.

### B. REAL TIME HARDWARE IMPLEMENTATIONS

Some hardware implementations of Retinex algorithms have been proposed in the literature. For example, a digital signal processor (DSP)-based real-time realization of the NASA-Retinex algorithms (as we mentioned in the previous subsection, algorithms very different from RSR and much less demanding in terms of performance) – a Single-Scale Retinex algorithm applied to monochrome images, and a simplified version of the Multi-Scale Retinex with color restoration [22] – were proposed in [37], [38]. In this case, however, the system performances are significantly lower than the ones of the original, not simplified, algorithm [39], [40], furthermore the DSP solution itself is not suitable for edge devices due to the high power and cost of the hardware.

Conventional architectures show huge computational costs due to the required processing layers, arithmetic operations, and the number of iterations. In order to address this issue and increase the speed of these algorithms, the implementation of a hardware accelerator based on a field-programmable gate array (FPGA) was proposed in [39]–[42].

Li *et al.* [39] presents a completely parallel architecture based on FPGAs for the implementation of multi-scale Retinex in an outdoor application. Address encoding and distributed arithmetic are used to optimize the Gaussian kernel, and concurrent multi-scale convolutions are accomplished. Furthermore, ustukov *et al.* [41] modifies the multi-scale Retinex algorithm. The algorithm performance is improved by using different methods of picture blurring, such as tabular value replacement instead of computing logarithm values. The method's ability to combine algorithms allows it to be implemented on FPGA as a threading conveyor.

Park *et al.* [43] provided a concept for the retinex video enhancing method that is low-cost and high-throughput. The hardware (HW) architecture is built using an FPGA and has a throughput of 60 frames per second for a $1920 \times 1080$ picture with little delay. By employing a tiny line buffer instead of a frame buffer, using the notion of approximation computing for the complex Gaussian filter, and creating a novel and nontrivial exponentiation operation, the suggested FPGA architecture lowers HW resources while retaining quality and speed. Moreover, Masri *et al.* [42] suggested a flexible and effective architecture for real-time video frame augmentation that may be implemented on a single FPGA. The video enhancing algorithm is based on Retinex. To regulate the dynamic range of poorly lighted photos while keeping visual details, a novel illuminance estimate methodology was used. The video enhancement settings are regulated in real time by an inbuilt microprocessor, allowing the system to adapt to the peculiarities of the incoming pictures and ambient lighting.

Nonetheless, FPGA has limited memory capacity and requires the image to be stored in external dynamic random-access memory (DRAM), which increases energy consumption and latency [39], [40], [42]. Furthermore, one has to consider the complex trade-off between performance, hardware (HW) resources, and efficiency degradation in terms of HW design. Li and Tsai [44] proposed the implementation of low-cost and high-speed HW for contrast-preserving and dynamic range compression. However, the Gaussian filter used in their algorithm can take only a small size.

Furthermore, Moore *et al.* [45] proposed a hardware implementation, which consists of resistive grids that average or smooth the pixel intensities. However, it lacks the

RESET operation, which is the essential feature of the retinex theory.

To address the challenges associated with the above-mentioned implementations, this paper presents a novel low-complexity and real-time HW-friendly architecture and design of the Retinex algorithm. Furthermore, to the best of our knowledge, this paper presents the first efficient hardware ReRAM-based implementation for the RSR algorithm. We propose using memristor-based structures that can perform highly parallel operations that reduce area and energy and accelerate the computation of the Retinex algorithm.

## III. RANDOM SPRAY RETINEX

Here, we introduce the RSR algorithm, the semantic segmentation algorithm, and the fundamental design blocks for the typical hardware implementation of the RSR algorithm.

### A. THE RSR ALGORITHM

In RSR, stochastic sampling is used for the estimation of the local white reference from the neighborhood of the target input pixel intensity $i_\tau$, where the index $\tau$ denotes that the intensity refers to the target pixel. For each chromatic channel and each target pixel, the algorithm works as follows.

1) Repeat $N$ times the following spray generation and processing cycle:
   - i Sample $n$ points from a neighborhood $\Omega_\tau$ of the target, following a particular sampling profile [31], thus obtaining an $n$-point set.
   - ii Get the corresponding sample of $n$ input intensities, $S_s^* = \{i_k\}_{k=1}^n$, where $s$ indicates the spray index: the *bare* spray).
   - iii Add the target intensity $i_\tau$, to the set and obtain an $(n+1)$ intensity set $S_s = \{\{i_k\} \cup S_s^*\}$: the augmented spray.
   - iv Compute the maximum intensity $y_s$ of the augmented spray, i.e., $y_s = \max(i_\tau, \max_k\{i_k\}_{k=1}^n)$

   After repeating $N$ times the steps (i) through (iv), a set of maxima $(y_1, y_2, \ldots, y_s, \ldots, y_N)$ is obtained.
2) Compute the harmonic average of the maxima:

$$\frac{1}{w_\tau} = \frac{1}{N}\sum_{s=1}^{N}\frac{1}{y_s}$$

   $w_\tau$ is the white reference value of $t$.
3) Set the output value $o_\tau$ for the target as, $o_\tau = i_\tau / w_\tau$.

The following three parameters affect processing performance: i) the number of sprays $N$ controls the noise: increasing $N$ lowers the chromatic noise; ii) the number of points per spray $n$ controls the sensitivity of the sampling to local intensity maxima: increasing $n$ increases the probability that a small bright patch is used as a reference white; iii) the locality of filtering (the difference between the influence from the closest points and the farther points in the neighborhood) is controlled by the sampling profile [33]. Such profile is a non-increasing function of the distance $r$ from the target and represents the probability that a pixel at that distance is picked during the sampling process [27]. Among the most used profiles are: the flat profile, and the profile that decreases as $1/(1+r)^\alpha$, as a function of the distance, with $\alpha \geq 1$ (typical values are $\alpha = 2, 3, 4$).

### B. RETINEX WITH SEMANTIC SEGMENTATION

In this section, we report the results of the study of the effects of illumination changes and contrast enhancement on the effectiveness of semantic segmentation in urban road scenes. This case can be motivated by several settings, including one of autonomous vehicle control. Autonomous vehicle applications need to operate correctly across different scenarios; however, environmental factors, such as weather and poor illumination, can deteriorate the quality of the acquired images, thus compromising safety.

Using the publicly available Cityscapes dataset, we simulate the underexposed images and compare the performance of a standard pre-trained deep semantic segmentation network on the original and dark images. The Cityscapes dataset focuses on semantic understanding of urban street scenes and has 30 classes. These semantic objects include: road, sidewalk, parking, rail track, person, rider, car, truck, bus, on rails, motorcycle, bicycle, caravan, trailer, building, wall, fence, guard rail, bridge, tunnel, pole, pole group, traffic sign, traffic light, vegetation, terrain, sky, ground, dynamic, static.

The training and evaluation of our approach are carried out using a public large-scale Audi Autonomous Driving Dataset (A2D2) [46], which contains over 40,000 labeled images, from which we take a subset of 12,497 images with dimensions of 1920 × 1208 pixels. Training images are cropped and resized to the 384 × 384 pixels size, while test images are resized so that the largest dimension is 768 pixels with the original aspect ratio preserved.

The effect on the dark image is done using the approach proposed by Christopher *et al.* [47]. They used equation (1) to model the consequences of underexposure on each pixel in an image:

$$V_2 = \begin{cases} \dfrac{V_1}{\theta_1/\theta_2} & V_1 \leq \theta_1 \\ (1-\theta_2)\dfrac{V_1-\theta_1}{1-\theta_2} + \theta_2 & V_1 > \theta_1 \end{cases} \quad (1)$$

The brightness values (i.e. the $V$ channel when the picture is converted to HSV color space) of the relevant pixels in the original and changed images are thus $V_1(0, 1)$ and $V_2(0, 1)$, respectively.

$\theta_1$ is a randomly generated threshold for each picture such that $(\mu - \sigma) \leq \theta_1 \leq \mu$ is obtained, where $\mu$ is the mean and $\sigma$ the standard deviation of all pixel values V throughout the whole image;

$\theta_2$ is a second, lower threshold that controls the amount of compression applied to dark picture sections. For the selected dataset, $\theta_2$ is set to $\theta_2 = \theta_1 \times 0.1$ such that pixels with $V_1 < \theta_1$ yield a $V_2 = V_1 \times 0.1$. Essentially, the dynamic range of pixel values below the threshold is compressed, while the dynamic range of pixel values above the threshold

**FIGURE 1.** An example image from the dataset and a modified image to simulate the darker version.

is increased. Figure 1 shows a sample picture before and after the application of equation (1).

The pre-trained DeepLab v3+ architecture [48] is used in this paper to compare the segmentation performance between the original, dark, quantized, and enhanced images. As illustrated in Figure 5, the dark and noisy images are passed through a random spray retinex algorithm to enhance the lightening of these images. Then, these enhanced images are passed to the pre-trained semantic segmentation model (DeepLab). Finally, an error is calculated as the mean pixels' cross-entropy loss between the output of the segmentation model and the ground truth segmentation.

The Accuracy (Acc), Recall (Rec), Precision (Prec), and Jaccard metric (intersection over union, IoU) of the results are calculated compared to the ground truth data for the different classes presented in the dataset. To perform the comparison, the segmentation metrics shown in Table 1 are calculated for the original image, the dark image, the enhanced image after applying RSR, and the quantized image with RSR, which is the one used for the memristor implementation as shown in the following sections. The accuracy value improved after applying RSR with n = 3 and a = 10, which is better than the dark simulated noisy pictures but not as perfect as the real image. For a broader image processing and computer vision community, we expanded the results in Table 1 with baseline techniques such as histogram equalization and some of its more advanced forms. Adjusting image intensity values, histogram equalization, and contrast-limited adaptive histogram equalization are three functions that are especially well suited for contrast enhancement. The three functions' differences are reflected in Figure 2. The first method is to adjust the image intensity values or color map, which will boost the image's contrast. By default, 1% of the data is saturated at low and high intensities of the input data. The second method is histogram equalization. It improves visual contrast by altering the values in an intensity image such that the output image's histogram closely matches the desired

**TABLE 1.** Segmentation results on cityscapes dataset.

| Dataset | Acc | Rec | Prec | IoU |
|---|---|---|---|---|
| **Original** | 0.979 | 0.918 | 0.915 | 0.854 |
| **Dark** | 0.957 | 0.853 | 0.850 | 0.753 |
| **RSR** | 0.964 | 0.868 | 0.868 | 0.776 |

histogram (uniform distribution by default). The third and last method is contrast-limited adaptive histogram equalization. It works on tiny data sections (tiles) rather than the complete picture, unlike histogram equalization. The contrast of each tile is increased such that the histogram of each output area comes close to matching the required histogram (uniform distribution by default). To prevent increasing any noise that may be present in the picture, the contrast enhancement might be minimized. Furthermore, Figure 3 shows how both $n$ and $a$ have a significant impact on the brightness adjustment.

Moreover, a test was performed on images from the ColorChecker image dataset [33], which contains 568 8-bit sRGB images, most of which have the size 874 × 583 as shown in Figure 4.

We add that the purpose of this subsection was not carrying on a comprehensive comparison of the performance of the RSR variants in terms of enhancement of the image quality: similar studies can be found in [24], [49] and in several papers devoted to variants of RSR such as [50]–[53].

### C. TYPICAL CONVENTIONAL HARDWARE IMPLEMENTATION FOR RSR

Most traditional versions of Retinex work offline, due to their computational complexity, although some versions of RSR, such as LightRSR [35] and FuzzyRSR [26] have reduced considerably the computational load, some, as SuPeR [54], at the price of some extra complexity in the code implementation. Some research has been carried on about the possibility of emulating Spatial Color Algorithms by learning the corresponding function by Artificial Neural Networks [55]: in those cases, the algorithm is fast and can be used online, however the time for training on a class of examples is non-negligible. We chose to use RSR instead of LightRSR, because the latter uses weights so that its implementation is slightly more involved than the former. The former is also more commonly adopted, so this makes it easier to compare our results with other papers.

Conventional architectures show huge computational costs due to the number of iterations and the required processing layers and arithmetic operations. To the best of the authors' knowledge, sparsity in Retinex is not yet employed in hardware. This comes from the fact that high-dimensional sparse data is usually beyond what is allowed on commodity hardware. The complexity of RSR processing is reported in terms of the number of scale to max operations, their accumulations over a given augmented spray, and the required memory resources.

Figure 6 presents the conventional architecture for RSR. The $n$ masks represent the augmented random sprays

**FIGURE 2.** Original image and enhanced images using adjusting image intensity values, histogram equalization, and contrast-limited adaptive histogram equalization example from cityscapes dataset.


**FIGURE 3.** Examples of the RSR application for different values of $n$ and $a$ for an image from the dark Cityscapes dataset. Only a brightness adjustment has been performed.

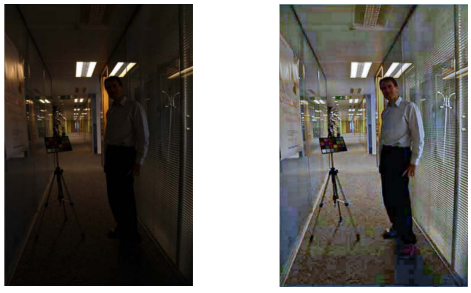


**FIGURE 4.** (a) original image from [33]. (b) application of guided image filtering for $n = 4$ and $a = 0.55$.

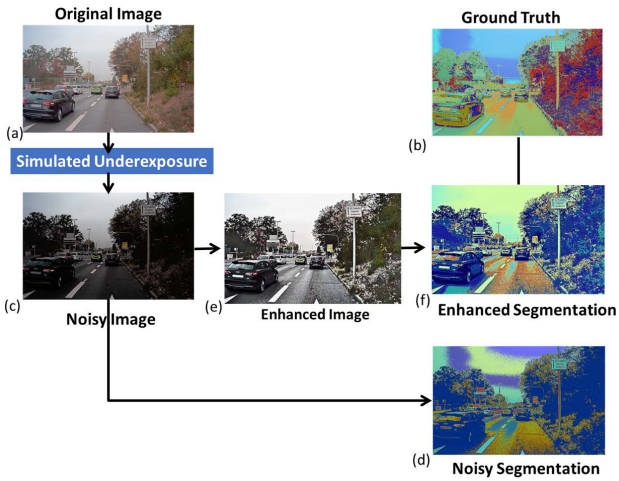


**FIGURE 5.** The training process for the RSR algorithm. First, an input image (a) is modified to simulate camera underexposure (c), then it is used as input to the RSR algorithm (e), which enhances the lightening of the image, then input that for a trained Deeplab segmentation model (f,d), which segments this enhanced image and the output is compared to the ground truth segmentation (b).

generated from the input image by shifting (e.g., block A is a unique random spray for a given targeted pixel). This is equivalent to the use of image extensions with distributed arithmetic and convolution filters [39]. The multiple random sprays are generated following a pipeline data flow. FIFOs (First-In-First-Out) are used as line buffers for most FPGA-based image processing, which means data reading is a serial operation. Pipelining can be realized with address encoding based on the random generation method.

The address-encoding concept controls the address when the sprays from the image are serially read out from the storage. As presented in Figure 6, the $x \times y \times n$ pixel values are fed to the line buffers for pipeline data flow for comparison, max scaling, accumulation, averaging, and resampling with respect to the input image. The $x$, $y$, and $n$ denote the rows, columns, and the number of pixel elements per spray, respectively. Given the methods mentioned above for HW-based retinex implementation, memristor-based in-memory computing paradigms are used to perform the scale to maximum operations and their accumulation as highlighted in Figure 6 (i.e. (b) and (c) blocks). The latter reduces memory access and computational complexity that serves the track for efficient hardware processing towards highly intensive tasks with a high order of sparsity.

### D. QUANTIZATION VS IMAGE QUALITY METRICS

Since the memristor requires a limited number of states, the intensity needs to be quantized. To study the impact of

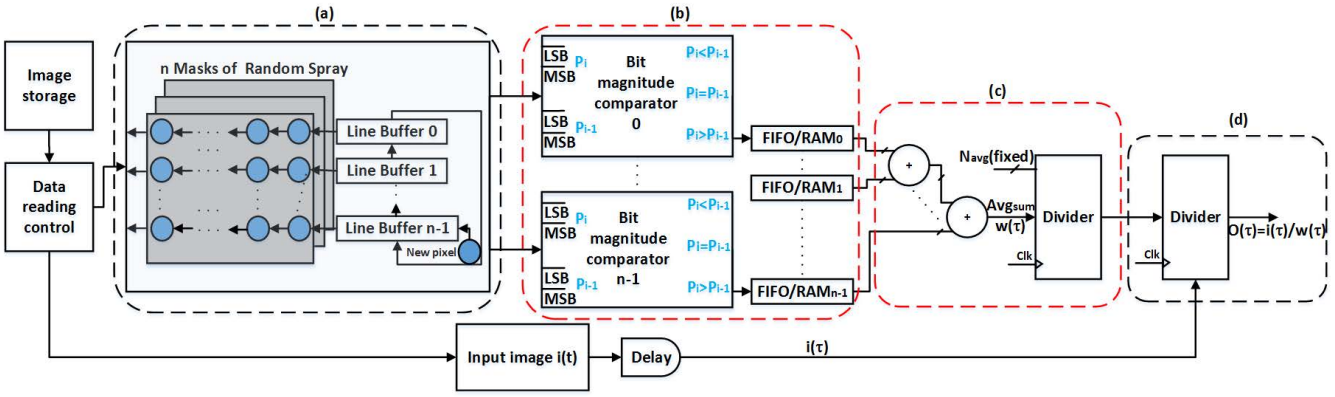


**FIGURE 6.** Conventional architecture of hardware RSR processing. (a) Random pixel generation and pipelining. (b) Comparator and scale to maximum pixel values. (c) Accumulation and averaging of maximum values. (d) Resampling with respect to the input target pixel. Both (b) and (c) functionality are implemented in memristor crossbar array.

quantization, we use seven different full-reference metrics to assess the image quality and compare it between the original image and the quantized image. The Matlab code for evaluating these metrics is provided in [56], which covers the following metrics: Structural Content (SC), Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR in dB), Normalized Cross-Correlation (NKK), Average Difference (AD), Maximum Difference (MD), and Normalized Absolute Error (NAE).

The MSE is the most common metric used in the literature for assessing image quality as it is simple and does not involve costly computations [57], [58]. MSE works satisfactorily when distortion is mainly caused by contamination of additive noise [59]. Furthermore, another famous image quality metric used by the prior studies is the PSNR, which is the ratio between the maximum power of a signal to the maximum power of a noise signal [60], [61]. PSNR is measured according to peak signal power. PSNR involves simple calculations, has a clear physical meaning, and is convenient in the context of optimization. However, PSNR is not according to the characteristics of the human visual system (HVS) [59].

As shown in Table 2 and Table 3, as the quantization level increases, the PSNR also increases, while the MSE and the NAE decrease. Moreover, the structural differences between reference and test images will generally increase as the quantization step size becomes larger. Hence, the AD is a monotonically decreasing function of the quantization step size, but the SC and MD are monotonically increasing functions of the quantization step size [62].

## IV. MEMRISTOR-BASED IN-MEMORY COMPUTING ARCHITECTURE FOR RSR

In this section, the novel proposed hybrid CMOS-Memristor architecture for the random spray retinex algorithm is detailed. In addition, the experimental and simulation results for the proposed model are also revealed.

**TABLE 2.** Image quality measures for natural image with the different quantized levels.

| Bits | MSE | PSNR | NCC | AD | SC | MD | NAE |
|---|---|---|---|---|---|---|---|
| 2 | 1.34E+04 | 6.84 | 71.78 | -74.72 | 1.6E-4 | -14 | 81.02 |
| 4 | 1.03E+04 | 7.99 | 14.67 | -63.91 | 0.003 | -8 | 17.31 |
| 5 | 8.98E+03 | 8.60 | 4.68 | -55.16 | 0.02 | 10 | 5.60 |
| 6 | 7.85E+03 | 9.18 | 1.53 | -35.79 | 0.11 | 58 | 2.08 |
| 8 | 52.35 | 30.94 | 1.08 | -4.21 | 0.86 | 10 | 0.07 |

**TABLE 3.** Image quality measures for the different quantized levels for pattern image.

| Bits | MSE | PSNR | NCC | AD | SC | MD | NAE |
|---|---|---|---|---|---|---|---|
| 2 | 1.86E+04 | 5.43 | 80.81 | -108.10 | 1.18E-4 | 0 | 93.31 |
| 4 | 1.73E+04 | 5.75 | 20.29 | -104.25 | 0.002 | 0 | 19.92 |
| 5 | 1.70E+04 | 5.83 | 12.39 | -103.07 | 0.005 | 0 | 15.29 |
| 6 | 1.25E+04 | 7.16 | 3.69 | -88.46 | 0.05 | 0 | 4.14 |
| 8 | 1.86E+04 | 5.43 | 0.03 | 107.54 | 32.36 | 255 | 0.98 |

### A. 4-BIT MEMRISTOR MODEL

The results provided in section II show that a 4-bit quantization is generally sufficient to achieve an acceptable image resolution, (see Table 2 and Table 3 for Figure 7 and Figure 8 respectively). In this work, an approximate 16-level fixed-point conductance state is adopted. This number is used to be consistent with the real memristor device fabricated by our group [18]. The writing process requires the memristive states to be separated within the switching window over the same interval.

As shown in Figure 9(a), a behavioral model is performed to fit the experimental gradual switching of the memristor. The resistance changes with a number of pulses (v(t)) are described with the following equations:

$$F = R_{max} - F(1 - e^{\alpha(v(t) - v_{max}}) \tag{2}$$

$$F = R_{max} - \frac{R_{min}}{1 - e^{-\alpha . v_{max}}} \tag{3}$$

where $R_{max} = 2800\Omega$, $R_{min} = 157\Omega$ and $v_{max} = 21$, represents, respectively, the maximum resistance, minimum resistance, and the maximum pulse number required to switch


**FIGURE 7.** Examples of the original image and the enhanced image using RSR for different quantized levels from the Audi Autonomous Driving Dataset. The first column has the original quantized image and the second column has the enhanced quantized images. (a,b) eight bits pixel representation, (c,d) two bits pixel representation, (e,f) four bits pixel representation, (g,h) five bits pixel representation, (i,j) six bits pixel representation.

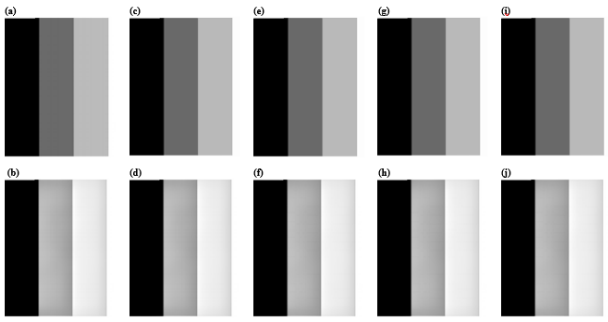


**FIGURE 8.** Examples of the original image and the enhanced image using RSR for different quantized levels for a pattern image. The first row has the original quantized image and the second row has the enhanced quantized images. (a,b) original eight bits pixel representation, (c,d) two bits pixel representation, (e,f) four bits pixel representation, (g,h) five bits pixel representation, (i,j) six bits pixel representation.

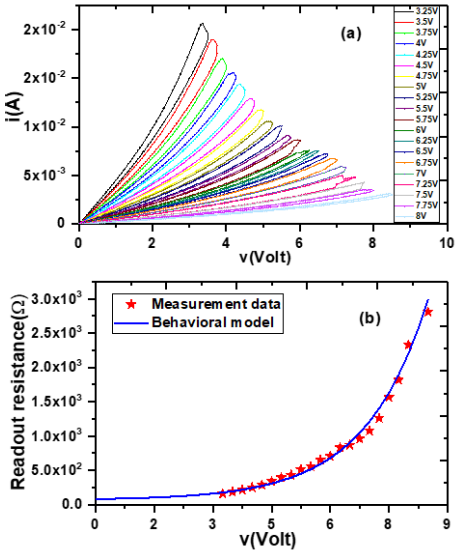


**FIGURE 9.** (a) Voltage vs. current measurement showing the gradual RESET of the memristor device. (b) Resistance modulation of the memristor device during pulse amplitude programming. Upward resistance modulation is achieved using pulses ranging from 3.25V to 8.5V by a step of 0.25V. The resistance values were readout at a non-destructive 0.001V voltage. The resistance transitions with RESET pulses are fitted using (1).

the device between the minimum and maximum resistance states. These parameters are directly extracted from the experimental data. $\alpha = 1.5V^{-1}$ is the parameter that controls the nonlinear behavior of resistance update, and $F$ is a function of $\alpha$ that fits the state transition within the range of $R_{max}$, and $R_{min}$. As shown in Figure 9(b), nonlinear change in device resistance can be obtained by tuning $\alpha$. This is equivalent to 4-bit characterization, which is used to code physical data acquired from the environment to be compatible with RRAM voltage pulse programming and efficient in-memory processing.

### B. LOGICAL MEMRISTOR-BASED IN-MEMORY COMPUTING ARCHITECTURE

Realizing a practical memristor crossbar-based in-memory computing system usually requires the integration of multiple memristor crossbar arrays [63]. In general, splitting resistive states into different arrays is beneficial for parallel computing, which is increasingly needed with increasing system scales. A single memory computing macro-core of the proposed architecture is presented in Figure 10. The latter has the fundamental one–transistor–one-memristor (1T1R) topology [16], which enables reliable and uniform analog switching behaviors. With the proposed hybrid-processing scheme, the parallel processing is implemented as in [16], which reduces the latency by a factor of $n^2$. The input pixel values are encoded by the pulse number according to its quantized bit number. This allows direct writing of resistive values to the target cross-point memristors. Using the in-memory computing paradigm, the augmented sprays are used to realize the scale-to-max operation in terms of resistance evolution over time through the selected memristor cross-point cells with the application of the equivalent modulated

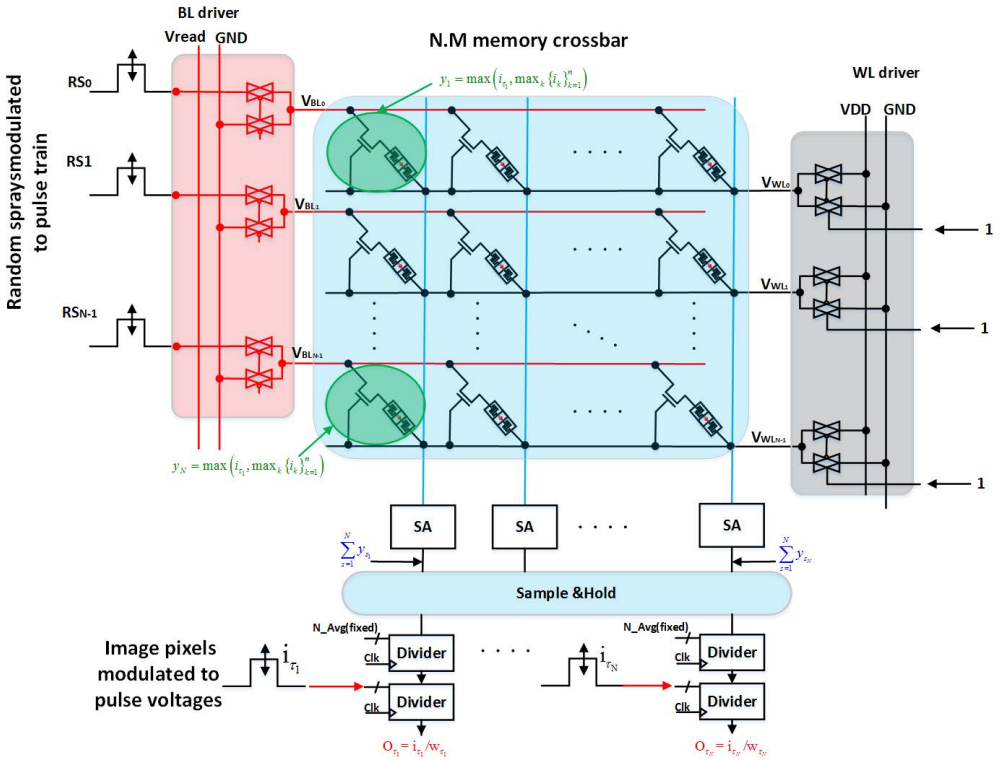


**FIGURE 10.** **In-memory computing macro core architecture of the simulated memristor-based Retinex processing unit and the relevant block modules.**

pulse signal. Furthermore, memristor arrays are highly efficient in achieving parallel Multiply and Add (MAC) operations under shared inputs for different arithmetic values.

Different fetched batches of random sprays are passed into memristor cross-point cells separately by applying the input signals as described in the previous section. The training scheme for the memristor crossbar sets the constraint for which a batch of the intermediate signal will not be supplied as input until these constraints are met. Thus, the previous batch needs to be already used to calculate the desired weight updates. Furthermore, the corresponding memristor conductance needs to be already well-tuned to the greater modulated pixel amplitude. The desired max updates of the states with respect to the pulse sequence are stored in the accessed memristor element from the crossbar. Then, the second-row memristor conductance is updated after inputting the second input batch for the same-targeted pixel. During this processing stage, another cycle of batches are inputted from another target pixel on the memristor crossbar and are fed into the unoccupied memristor-based-scale to a max operator in parallel [16]. These operations are repeatedly done through in-memory computation until the random paths are written in the memristor-based scale to max operators.

After inputting the encoded pulses that correspond to pixel intensity into the bit lines, the output currents through the source lines are sensed and accumulated. The current is the weighted sum corresponding to the input patch and the chosen augmented spray. The latest resistance values with different weights are written to different rows, and the entire memristor array operates MACs in parallel under the same inputs (read voltage). Thus, all the desired weighted-sum results are obtained concurrently. Afterthought, each output sense integrator is averaged and then resampled with the associated and predefined target pixel inputs.

Figure 11 shows a SPICE simulation [64] of the scale to max operation through different memristor cells. The different batches of such an augmented spray results in an incremental behavior of the resistance that keeps the greatest value at the end-state.

The output images have the same results as the 4-bit quantization presented in section II for both natural and pattern images. However, in the memristor-based in-memory computing system reported in [65], the resolution loss is mainly attributed to two factors: first, the presence of non-ideal device characteristics, such as device variations, array yield problems, and device reliability issues; second, as it is used in this approach, the limited precision due to weight quantization. Even though the accuracy is not fully recovered given the limited quantization precision, results suggest that the hybrid-processing method could effectively recover high-resolution accuracy by accommodating resampling with the original target pixel after averaging the accumulated memristor weights. These findings suggest that the parallel memristor-based in-memory computation is highly efficient in achieving a high resolution while greatly accelerating the RSR algorithm. In addition, the associated expenditure of chip-area is minimized by reducing the number of memory accesses and the arithmetic operations.

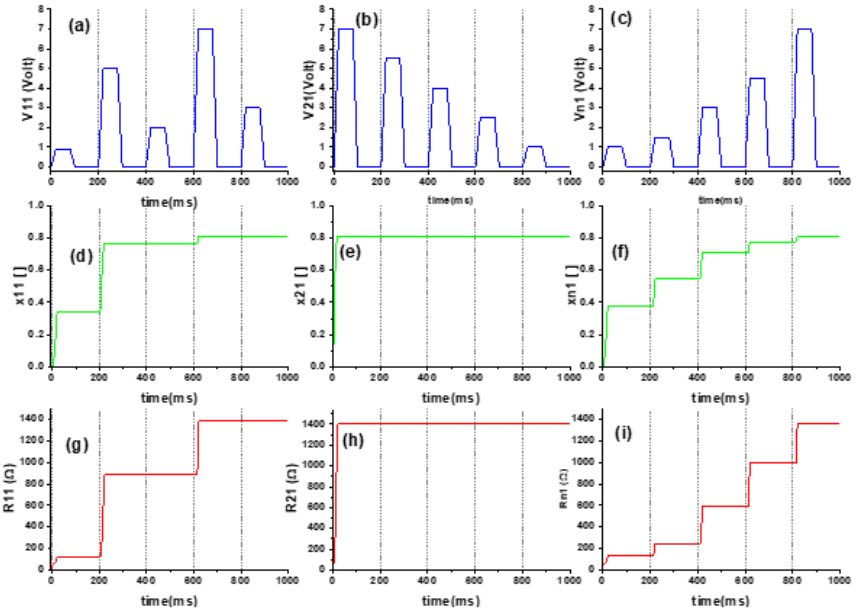


**FIGURE 11.** Example of a Spice simulation of the scale-to-max operation through the memristor model. (a), (b) and (c) Biasing voltages applied on the targeted memristor cross-points. (d), (e) and (f) the resulted evolution of the normalized internal state variable. (g), (h) and (i) the in-memory computing of the scale to max resistances following the applied voltages (quantized pixel values).

## V. PERFORMANCE ASSESSMENT AND DISCUSSIONS

In this section, a comparative analysis of the proposed memristor-based IMC architecture is presented, along with conventional FPGA-based solutions.

An image size of $I_z = 256 \times 256$ is chosen for performance evaluation (i.e, memory access ($Mem_{acc}$), number of arithmetic operations ($Ar.op$), and area cost ($A$)). Table 4 shows the metrics for evaluating the required resources for a given RSR-based target pixel computations using traditional FPGA solutions and the proposed IMC architecture.

### A. MEMORY ACCESS

The total number of memory accesses required to compute the required arithmetic operations for a given target pixel from a random path information in a traditional FPGA solution is calculated as follows;

$$Mem_{acc} = Mem_{acc}(p_{row}) \times Mem_{acc}(p_{cols}) \times n \times N, \quad (4)$$

where $p_{row}$ and $p_{cols}$ denotes the access to x,y image coordinates from line buffers of Random pixel generation and pipelining block, as shown in Figure 6. $n$ is the set-pixel-points per a single random path and $N$ presents the processing cycles. Additional $n \times N$ accesses to the scaled-to-max-values that are stored in the FIFOs in order to be accumulated, as well as $n \times N$ for the averaging and resampling processes. While for the memristor-based IMC architecture, the $n$ set-point are required to hold in-memory the given random spray and process the four-step algorithm of the targeted pixel in analog manner, as shown in Figure 10, leading to $Mem_{acc}(p_{row}) \times Mem_{acc}(p_{cols}) \times N$ memory access process. In the following, RSR's default values, $n = 250$, $N = 25$ are used for the objective assessments.

### B. ARITHMETIC OPERATIONS

Based on Figure 6, using traditional FPGA implementations, the number of arithmetic operations needed for the processing of a given pixel, are:

$$Ar.op = N.(n+1)Comp(.) + n.N(Acc(.) + Av(.) + Resamp(.)) \quad (5)$$

where $Comp(.)$, $Acc(.)$, $Av(.)$, $Resamp(.)$ are the comparators to obtain the augmented spray, accumulators, averaging and resampling operations.

While using the IMC paradigm, $N.(n+1)$ $Comp(.)$ and $Acc(.)$, are done in-memory intrinsic computations (i.e. only the access pulse train is required across its nodes). The memristor-based IMC, instead of traditional FPGA implementation, shows a vast decrease in both; the number of memory accesses and arithmetic operations, as shown in Table 4.

**TABLE 4.** Required resources and performance assessments for target pixel computations.

| Parameter | Estimation with conventional design-based FPGA solutions | IMC solution |
|---|---|---|
| $Mem_{acc}$ | Pixel generator: $164 \times 10^7$<br>Scale to max operation: 6250<br>Accumulation: 6250<br>Averaging and resampling: 6250 | Pixel voltage generator: $16.4 \times 10^5$<br>Scale to max operation: 25<br>Accumulation: 25<br>Averaging and resampling: 25 |
| $Ar.op$ | 25025 | 12550 |
| $A(F^2)$ | $750 \times 10^3$ (SRAM blocks)+<br>$8 \times 10^3$ (other blocks) | 25+260 (WL, BL switch matrices<br>+WL decoders) |

## VI. CONCLUSION

The paper presented a novel, efficient hybrid CMOS-Memristor approach for random spray retinex. The proposed solution provided high speed and energy/area-efficient architecture compared to the conventional retinex scheme. Image quality was assessed using several image quality metrics and different quantization levels. A 4-bit memristor was used as the computing element. The latter operated as a scale to maximum resistive values. MAC operations were performed in parallel for the accumulation of output currents. Generally, this design can be extended to other memristor-based in-memory computing systems that use the scale to max operators and employ sparse input data to boost their overall performance efficiently. The proposed approach is considered a great asset towards developing efficient memristor-based computer vision and deep learning applications. In addition, the usage of the RSR algorithm as preprocessing step for AI applications was investigated. The results showed improved accuracy when RSR was deployed on noisy images.

## REFERENCES

[1] T. Arici, S. Dikbas, and Y. Altunbasak, "A histogram modification framework and its application for image contrast enhancement," *IEEE Trans. Image Process.*, vol. 18, no. 9, pp. 1921–1935, Sep. 2009.

[2] X. Guan, S. Jian, P. Hongda, Z. Zhiguo, and G. Haibin, "An image enhancement method based on gamma correction," in *Proc. 2nd Int. Symp. Comput. Intell. Design*, 2009, pp. 60–63.

[3] S.-C. Huang, F.-C. Cheng, and Y.-S. Chiu, "Efficient contrast enhancement using adaptive gamma correction with weighting distribution," *IEEE Trans. Image Process.*, vol. 22, no. 3, pp. 1032–1041, Mar. 2013.

[4] D. Zhang, W.-J. Park, S.-J. Lee, K.-A. Choi, and S.-J. Ko, "Histogram partition based gamma correction for image contrast enhancement," in *Proc. IEEE 16th Int. Symp. Consum. Electron.*, Jun. 2012, pp. 1–4.

[5] J.-Y. Kim, L.-S. Kim, and S.-H. Hwang, "An advanced contrast enhancement using partially overlapped sub-block histogram equalization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 4, pp. 475–484, Apr. 2001.

[6] T. K. Kim, J. K. Paik, and B. S. Kang, "Contrast enhancement system using spatially adaptive histogram equalization with temporal filtering," *IEEE Trans. Consum. Electron.*, vol. 44, no. 1, pp. 82–87, Feb. 1998.

[7] A. Rizzi and J. J. McCann, "On the behavior of spatial models of color," *Proc. SPIE*, vol. 6493, Jan. 2007, Art. no. 649302.

[8] M. Lecca, "A gradient-based spatial color algorithm for image contrast enhancement," in *Proc. Int. Conf. Image Anal. Process.* Cham, Switzerland: Springer, 2019, pp. 93–103.

[9] E. Provenzi, M. Fierro, A. Rizzi, L. De Carli, D. Gadia, and D. Marini, "Random spray Retinex: A new Retinex implementation to investigate the local properties of the model," *IEEE Trans. Image Process.*, vol. 16, no. 1, pp. 162–171, Jan. 2007.

[10] O. Kolås, I. Farup, and A. Rizzi, "Spatio-temporal Retinex-inspired envelope with stochastic sampling: A framework for spatial color algorithms," *J. Imag. Sci. Technol.*, vol. 55, no. 4, p. 40503, 2011.

[11] M. Lecca, "STAR: A segmentation-based approximation of point-based sampling Milan Retinex for color image enhancement," *IEEE Trans. Image Process.*, vol. 27, no. 12, pp. 5802–5812, Dec. 2018.

[12] G. Gianini, A. Rizzi, and E. Damiani, "A Retinex model based on absorbing Markov chains," *Inf. Sci.*, vol. 327, pp. 149–174, Jan. 2016.

[13] A. Rizzi and C. Bonanomi, "Milano Retinex family," *J. Electron. Imag.*, vol. 26, no. 3, Mar. 2017, Art. no. 031207.

[14] Y. Lu, A. Alvarez, C.-H. Kao, J.-S. Bow, S.-Y. Chen, and I.-W. Chen, "An electronic silicon-based memristor with a high switching uniformity," *Nature Electron.*, vol. 2, no. 2, pp. 66–74, Feb. 2019.

[15] Y. Halawani, B. Mohammad, M. Al-Qutayri, and S. F. Al-Sarawi, "Memristor-based hardware accelerator for image compression," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 12, pp. 2749–2758, Dec. 2018.

[16] F. Zayer, B. Mohammad, H. Saleh, and G. Gianini, "RRAM crossbar-based in-memory computation of anisotropic filters for image preprocessingloa," *IEEE Access*, vol. 8, pp. 127569–127580, 2020.

[17] F. Zayer, W. Dghais, and H. Belgacem, "Modeling framework and comparison of memristive devices and associated STDP learning Windows for neuromorphic applications," *J. Phys. D, Appl. Phys.*, vol. 52, no. 39, Sep. 2019, Art. no. 393002.

[18] H. Abunahla, Y. Halawani, A. Alazzam, and B. Mohammad, "NeuroMem: Analog graphene-based resistive memory for artificial neural networks," *Sci. Rep.*, vol. 10, no. 1, pp. 1–11, Dec. 2020.

[19] N. Banić and S. Lončarić, "Light random sprays Retinex: Exploiting the noisy illumination estimation," *IEEE Signal Process. Lett.*, vol. 20, no. 12, pp. 1240–1243, Dec. 2013.

[20] M. Cordts, M. Omran, S. Ramos, T. Scharwächter, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset," in *Proc. CVPR Workshop Future Datasets Vis.*, vol. 2, 2015, pp. 1–4.

[21] J. J. McCann, "Retinex at 50: Color theory and spatial algorithms, a review," *J. Electron. Imag.*, vol. 26, no. 3, Feb. 2017, Art. no. 031204.

[22] D. J. Jobson, Z.-U. Rahman, and G. A. Woodell, "A multiscale Retinex for bridging the gap between color images and the human observation of scenes," *IEEE Trans. Image Process.*, vol. 6, no. 7, pp. 965–976, Jul. 1997.

[23] G. Gianini, C. Mio, L. G. Fossi, and A. Rizzi, "Gradient attenuation as an emergent property of reset-based Retinex models," in *Proc. 11th Int. Conf. Manage. Digit. EcoSystems*, Nov. 2019, pp. 324–329.

[24] M. Lecca, A. Rizzi, and G. Gianini, "Review and comparison of random spray Retinex and of its variants stress and qbrix," *Cultura e Scienza del Colore-Color Culture Sci.*, vol. 9, pp. 55–64, 2018.

[25] M. Lecca, A. Rizzi, and G. Gianini, "Energy-driven path search for termite Retinex," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 33, no. 1, p. 31, 2016.

[26] G. Gianini and A. Rizzi, "A fuzzy set approach to Retinex spray sampling," *Multimedia Tools Appl.*, vol. 76, no. 23, pp. 24723–24748, Dec. 2017.

[27] E. Provenzi, M. Fierro, A. Rizzi, L. De Carli, D. Gadia, and D. Marini, "Random spray Retinex: A new Retinex implementation to investigate the local properties of the model," *IEEE Trans. Image Process.*, vol. 16, no. 1, pp. 162–171, Jan. 2006.

[28] G. Gianini, A. Manenti, and A. Rizzi, "QBRIX: A quantile-based approach to Retinex," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 31, no. 12, pp. 2663–2673, Dec. 2014.

[29] G. Gianini, "Statistical aspects of space sampling in Retinex models," *Electron. Imag.*, vol. 28, no. 6, pp. 1–6, Feb. 2016.

[30] G. Gianini, "From samples to populations in Retinex models," *J. Electron. Imag.*, vol. 26, no. 3, Mar. 2017, Art. no. 031206.

[31] G. Gianini, M. Lecca, and A. Rizzi, "A population-based approach to point-sampling spatial color algorithms," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 33, no. 12, pp. 2396–2413, 2016.

[32] N. Banić and S. Lončarić, "Using the random sprays Retinex algorithm for global illumination estimation," 2013, *arXiv:1310.0307*.

[33] N. Banić and S. Lončarić, "Smart light random memory sprays Retinex: A fast Retinex implementation for high-quality brightness adjustment and color correction," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 32, no. 11, pp. 2136–2147, 2015.

[34] E. Provenzi, C. Gatta, M. Fierro, and A. Rizzi, "A spatially variant white-patch and gray-world method for color image enhancement driven by local contrast," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 10, pp. 1757–1770, Oct. 2008.

[35] M. Lecca and A. Rizzi, "Tuning the locality of filtering with a spatially weighted implementation of random spray Retinex," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 32, no. 10, pp. 1876–1887, 2015. [Online]. Available: http://opg.optica.org/josaa/abstract.cfm?URI=josaa-32-10-1876

[36] M. Tanaka, M. P. Lanaro, T. Horiuchi, and A. Rizzi, "Random spray Retinex extensions considering region of interest and eyemovements," *Electron. Imag.*, vol. 2020, no. 15, Jan. 2020, Art. no. 60403.

[37] G. D. Hines, Z.-U. Rahman, D. J. Jobson, and G. A. Woodell, "DSP implementation of the retinex image enhancement algorithm," *Proc. SPIE*, vol. 5438, pp. 13–24, Jul. 2004.

[38] G. D. Hines, Z.-U. Rahman, D. J. Jobson, G. A. Woodell, and S. D. Harrah, "Real-time enhanced vision system," *Proc. SPIE*, vol. 5802, pp. 127–134, May 2005.

[39] Y. Li, H. Zhang, Y. You, and M. Sun, "A multi-scale Retinex implementation on FPGA for an outdoor application," in *Proc. 4th Int. Congr. Image Signal Process.*, Oct. 2011, pp. 1788–1792.

[40] H. Tsutsui, H. Nakamura, R. Hashimoto, H. Okuhata, and T. Onoye, "An FPGA implementation of real-time Retinex video image enhancement," in *Proc. World Automat. Congr.*, Sep. 2010, pp. 1–6.

[41] D. I. Ustukov, Y. R. Muratov, and V. N. Lantsov, "Modification of Retinex algorithm and its stream implementation on FPGA," in *Proc. 6th Medit. Conf. Embedded Comput. (MECO)*, Jun. 2017, pp. 1–4.
[42] S. Marsi and G. Ramponi, "A flexible FPGA implementation for illuminance–reflectance video enhancement," *J. Real-Time Image Process.*, vol. 8, no. 1, pp. 81–93, Mar. 2013.
[43] J. W. Park, H. Lee, B. Kim, D.-G. Kang, S. O. Jin, H. Kim, and H.-J. Lee, "A low-cost and high-throughput FPGA implementation of the Retinex algorithm for real-time video enhancement," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 1, pp. 101–114, Jan. 2020.
[44] S.-A. Li and C.-Y. Tsai, "Low-cost and high-speed hardware implementation of contrast-preserving image dynamic range compression for full-HD video enhancement," *IET Image Process.*, vol. 9, no. 8, pp. 605–614, Jul. 2015.
[45] A. Moore, J. Allman, and R. M. Goodman, "A real-time neural system for color constancy," *IEEE Trans. Neural Netw.*, vol. 2, no. 2, pp. 237–247, Mar. 1991.
[46] J. Geyer, Y. Kassahun, M. Mahmudi, X. Ricou, R. Durgesh, A. S. Chung, L. Hauswald, V. Hoang Pham, M. Mühlegg, S. Dorn, T. Fernandez, M. Jänicke, S. Mirashi, C. Savani, M. Sturm, O. Vorobiov, M. Oelker, S. Garreis, and P. Schuberth, "A2D2: Audi autonomous driving dataset," 2020, *arXiv:2004.06320*.
[47] C. J. Holder, M. Khonji, and J. Dias, "Input optimisation network for semantic segmentation of underexposed images," in *Proc. IEEE Int. Symp. Saf., Secur., Rescue Robot. (SSRR)*, Nov. 2020, pp. 298–303.
[48] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2017.
[49] M. Lecca, G. Simone, C. Bonanomi, and A. Rizzi, "Point-based spatial colour sampling in Milano-Retinex: A survey," *IET Image Process.*, vol. 12, no. 6, pp. 833–849, Jun. 2018.
[50] K. Koscevic, V. Stipetic, E. Provenzi, N. Banic, M. Subasic, and S. Loncaric, "HD-RACE: Spray-based local tone mapping operator," in *Proc. 12th Int. Symp. Image Signal Process. Anal. (ISPA)*, Sep. 2021, pp. 264–269.
[51] N. Banic and S. Loncaric, "Color rabbit: Guiding the distance of local maximums in illumination estimation," in *Proc. 19th Int. Conf. Digit. Signal Process.*, Aug. 2014, pp. 345–350.
[52] N. Banic and S. Loncaric, "Towards hardware-friendly Retinex algorithms," in *Proc. 10th Int. Symp. Image Signal Process. Anal.*, Sep. 2017, pp. 104–108.
[53] N. Banic and S. Loncaric, "Puma: A high-quality Retinex-based tone mapping operator," in *Proc. 24th Eur. Signal Process. Conf. (EUSIPCO)*, Aug. 2016, pp. 943–947.
[54] M. Lecca and S. Messelodi, "SuPeR: Milan Retinex implementation exploiting a regular image grid," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 36, no. 8, pp. 1423–1432, Aug. 2019.
[55] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao, "Learning enriched features for real image restoration and enhancement," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 492–511.
[56] (2011). Athi. *Image Quality Measures*. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/25005-image-quality-measures
[57] Z. Wang and A. C. Bovik, "Mean squared error: Love it or leave it? A new look at signal fidelity measures," *IEEE Signal Process. Mag.*, vol. 26, no. 1, pp. 98–117, Jan. 2009.
[58] K. Gu, G. Zhai, X. Yang, and W. Zhang, "An improved full-reference image quality metric based on structure compensation," in *Proc. Asia Pacific Signal Inf. Process. Assoc. Annu. Summit Conf.*, Dec. 2012, pp. 1–6.
[59] T. Samajdar and M. I. Quraishi, "Analysis and evaluation of image quality metrics," in *Information Systems Design and Intelligent Applications*. New Delhi, India: Springer, 2015, pp. 369–378.
[60] P. Kaushik and Y. Sharma, "Comparison of different image enhancement techniques based upon psnr & mse," *Int. J. Appl. Eng. Res.*, vol. 7, no. 11, pp. 2010–2014, 2012.
[61] Z. Wang, M. Sabir, and A. Bovik, "A statistical evaluation of recent full reference image quality assessment algorithm," *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3441–3452, Oct. 2006.
[62] S. M. K. Silpa, "Comparison of image quality metrics," *Int. J. Eng. Res. Technol.*, vol. 1, no. 4, pp. 2278–6182, 2012.
[63] P. Yao, H. Wu, B. Gao, J. Tang, Q. Zhang, W. Zhang, J. J. Yang, and H. Qian, "Fully hardware-implemented memristor convolutional neural network," *Nature*, vol. 577, no. 7792, pp. 641–646, 2020.
[64] *Spice*. Accessed: May 1, 2022. [Online]. Available: https://www.cadence.com/
[65] S. Yu, B. Gao, Z. Fang, H. Yu, J. Kang, and H.-S. P. Wong, "A neuromorphic visual system using RRAM synaptic devices with sub-pJ energy and tolerance to variability: Experimental characterization and large-scale modeling," in *IEDM Tech. Dig.*, Dec. 2012, pp. 4–10.

**MERIEM BETTAYEB** received the bachelor's and master's degrees (Hons.) in computer engineering from the University of Sharjah (UOS), United Arab Emirates, in 2017 and 2019, respectively. She is currently pursuing the Ph.D. degree in electrical engineering and computer science with Khalifa University, United Arab Emirates. She is also working as a Teaching and Research Assistant with Khalifa University on the co-optimization of machine learning algorithms and memristor-based in-memory computing (IMC) for Internet of Things (IoT) devices. Her research projects focused on investigating the popular algorithms used for artificial intelligence (AI) applications and mapping them to efficient hardware implementations. Her research interests include exploring data reuse, data preprocessing, and the usage of the memristor crossbar architecture.

**FAKHREDDINE ZAYER** received the B.S. degree in physics and the M.S. degree in materials, nanostructures, devices, and microelectronic systems from the Faculty of Sciences of Monastir, University of Monastir, Tunisia, in 2011 and 2013, respectively, and the Ph.D. degree in electronics and computer engineering from the National Engineering School of Monastir (ENIM), Monastir University, in 2019. He has been a Postdoctoral Fellow in electrical and electronic engineering with the Center for Autonomous Robotic Systems (KUCARS), Khalifa University, United Arab Emirates, since 2021. He worked as a Research Associate with the System on Chip Center (SOCC), Khalifa University, in 2019 and 2021. Prior to joining Khalifa University, he worked as a Senior Staff Engineer in the integration of materials into the system (IMS) with the Thales Group, France, from 2014 to 2016. His research interests include the compact modeling of nanoscale devices (including reliability and testability) and computing paradigms (including neuromorphic computing and in-memory computing for logic and storage).

**HEBA ABUNAHLA** received the B.S. degree in electrical engineering from United Arab Emirates University, Al Ain, in 2007, the M.S. degree in electrical and electronics engineering from the University of Sharjah, Sharjah, in 2013, and the Ph.D. degree in electrical and computer engineering from the Khalifa University of Science and Technology, Abu Dhabi, in 2017. She is currently a Research Scientist with the Khalifa University of Science and Technology. During her research journey, she was able to build strong expertise in the field of micro- and nano-electronics design, modeling, fabrication and characterization of high-performance, and high-density memory devices. She has successfully fabricated memory devices that were uniquely applied to neuromorphic computing, hardware-security, communication, and biosensing applications. She serves as an active reviewer for several prestigious journals. She is a Guest Editor of *Frontiers in Neuroscience*. She has received several national and international awards.

**GABRIELE GIANINI** (Member, IEEE) received the M.Sc. degree, in 1992, and the Ph.D. degree, in 1996. He is currently an Associate Professor with the Department of Computer Science, Università degli Studi di Milano, Italy. He has held a visiting positions at a number of international institutions, including INSA de Lyon, France, the University of Passau, Germany, CERN, Geneva, Switzerland, Fermilab, Chicago, USA, and CBPF, Brazil. From 2005 to 2012, he has been an Adjunct Professor with the Free University of Bolzano, Italy. From 2017 to 2020, he has been a Senior Research Fellow with EBTIC/Khalifa University. His research interests include machine learning and data analytics applications. Since 1992, he has been coauthored over 200 papers published in internationally refereed journals and conferences. Since 2018, he has been an Associate Editor of the *Data Science and Engineering* (DSEJ) journal (Springer) and the *Journal of Imaging Science and Technology* (JIST) (Society for Imaging Science and Technology).

**BAKER MOHAMMAD** (Senior Member, IEEE) received the B.S. degree from the University of New Mexico, Albuquerque, the M.S. degree from Arizona State University, Tempe, and the Ph.D. degree from The University of Texas at Austin, in 2008, all in ECE. He is currently the Director of the System on Chip Center and an Associate Professor in EECS at Khalifa University. Prior to joining Khalifa University, he was a Senior Staff Engineer/the Manager at Qualcomm, Austin, USA, for six years, where he was engaged in designing high-performance and low-power DSP processors used for communication and multimedia applications. Before joining Qualcomm, he worked for ten years at Intel Corporation on a wide range of microprocessors design from high-performance, server chips more than 100Watt (IA-64), to mobile embedded processor low power sub one watt (xscale). He has over 16 years of industrial experience in microprocessor design, emphasizing memory, low power circuits, and physical design. His research interests include VLSI, power-efficient computing, high yield embedded memory, emerging technologies, such as memristor, STTRAM, in-memory-computing, and hardware accelerators for cyber-physical systems. He is also engaged in a microwatt-range computing platform for wearable electronics and WSN, focusing on energy harvesting, power management, and power conversion, including efficient DC/DC and AC/DC converters.

He authored/coauthored over 150 refereed journals and conference proceedings, more than three books, more than 18 U.S. patents, multiple invited seminars/panelists, and the presenter of more than three conference tutorials, including one tutorial on energy harvesting and power management for WSN at the 2015 (ISCAS). He is an Associate Editor of IEEE Access and IEEE Transaction on Very Large Scale Integration (VLSI) Systems. He participates in many technical committees at IEEE conferences and reviews for IEEE Transaction on Very Large Scale Integration (VLSI) Systems, *IEEE Circuits and Systems Magazine* journals. He has received several awards, including the KUSTAR Staff Excellence Award in Intellectual Property Creation, the IEEE Transaction on Very Large Scale Integration (VLSI) Systems Best Paper Award, and the 2016 IEEE MWSCAS Myrill B. Reed Best Paper Award, Qualcomm Qstar Award for Excellence on Performance and Leadership. SRC Techon's best session papers for 2016 and 2017. In 2009, the Best Paper Award for Qualcomm Qtech Conference and Intel Involve in the community award for volunteer and impact on the community.