

# Random-Valued Impulse Noise Detection and Removal based on Local Statistics of Images

Mickael Aghajarian, John E. McInroy

Department of Electrical and Computer Engineering, College of Engineering and Applied Science  
University of Wyoming, Laramie, Wyoming, United States

**Abstract**—Random-valued impulse noise removal from images is a challenging task in the field of image processing and computer vision. In this paper, an effective three-step noise removal method was proposed using local statistics of grayscale images. Unlike most existing denoising algorithms that assume the noise density is known, our method estimated the noise density in the first step. Based on the estimated noise density, a noise detector was implemented to detect corrupted pixels in the second step. Finally, a modified weighted mean filter was utilized to restore the detected noisy pixels while leaving the noise-free pixels unchanged. The noise removal performance of our method was compared with 10 well-known denoising algorithms. Experimental results demonstrated that our proposed method outperformed other denoising algorithms in terms of noise detection and image restoration in the vast majority of the cases.

**Keywords**—Random-valued impulse noise; noise detection; image restoration; modified weighted mean filter

## I. INTRODUCTION

Image noise is an inevitable consequence of some intrinsic (e.g., sensor) and/or extrinsic (e.g., environment) factors such as imperfections in capturing instruments, bit errors in analog-to-digital conversions, malfunctions in camera sensors, and interference in transmission channels. The existence of noise not only degrades the visual quality of images but also adversely affects the performance of image processing and computer vision tasks, like classification, detection, and segmentation. Thus, image denoising is often an essential preprocessing task in the field of image processing and computer vision. The goal of an ideal image denoising method is to remove the noise while maintaining fine structures of images such as edges or corners.

Depending on the sources of noise, image noise can be classified into different categories such as impulse noise, Poison noise, and Gaussian noise. Two common types of impulse noise are the salt-and-pepper (SAP) and random-valued impulse noise (RVIN). In an 8-bit/pixel image, noisy pixels in images corrupted by SAP can take on either the minimum or maximum intensity (i.e., 0 or 255), while for contaminated images by RVIN, corrupted pixels can take any values between 0 and 255. Therefore, detecting noisy pixels contaminated by RVIN is a challenging task. Another challenging issue in detecting the noisy pixels is distinguishing between image edge pixels and corrupted pixels. The big difference between the intensity of image edge and their neighboring pixels might cause noise detectors to falsely detect the image edge pixels as noisy pixels.

Although many algorithms have been proposed for the noise removal problem, there is still room for improvement [1]–[4]. Particularly, for the RVIN removal problem, various methods have been proposed in the literature. The standard median filter (MF) [5], [6] is a widely used nonlinear filter due to its simplicity and high computational efficiency; however, it does not work well for high levels of noise and eliminates fine structures of images, and this leads to blur. In order to improve its performance, some modifications to MF have been proposed. The weighted median filter (WMF) [7] gave more weight to some pixels within the sliding window. It allowed a degree of control of the smoothing by which more image details could be preserved; however, finding suitable weights for different images was not an easy task. Center weighted median filter (CWMF) [8] was a special case of WMF which gave more weights only to the central pixel of the sliding window. The adaptive weighted median filter (AWMF) [8] was another modification to MF in which the filter weights were adapted accordingly based on the local statistics. AWMF could suppress multiplicative noise as well as additive white and impulse noise. Adaptive median filter (AMF) [9] was another method with variable sliding window size.

One characteristic of RVIN is that depending on the noise density, only some parts of image pixels are corrupted while the rest are noise-free. The main drawback of the aforementioned filters is that they restore the entire image by processing all pixels without considering whether the pixel is noisy or not. As a result, they eliminate fine details of images like edges or corners, and this leads to blur. In order to overcome this drawback, several two-step methods have been proposed that are integrated with noise detectors. In the first step, the noise detector determines whether the pixel is corrupted or not. In the second step, only the noisy pixels are restored while other pixels remain unchanged. By doing so, more image details can be preserved and in turn the quality of restored images can be improved. It should be noted that the performance of these methods heavily depends on the proper detection of noisy pixels that is a challenging task for RVIN.

The switching median filter (SMF) [10] calculated the absolute value of difference between the center pixel of the sliding window and the median. If the difference was greater than a predefined threshold, it detected the pixel as noisy and restored it by using the median filter; otherwise, it left the pixel unchanged. The first drawback of SMF is that it uses a fixed threshold to detect noisy pixels. The second drawback is that it restores corrupted pixels using the median value of the current sliding window that might include other noisy pixels, so its

performance for high levels of noise can be deteriorated. Several modifications have been developed to improve the performance of SMF. Noise adaptive soft-switching median filter (NASM) [11] used fuzzy logic to categorize image pixels into four categories named uncorrupted pixel, isolated impulse noise, non-isolated impulse noise and edge pixel. Then, depending on the pixel's characteristic, an appropriate filter (i.e., MF or proposed fuzzy WMF) was utilized to restore corrupted pixels. Adaptive impulse noise detector using CWM (ACWM) [12] utilized the differences between the center pixel of the sliding window and the output of CWM with varied center weights to detect corrupted pixels. In [13], an impulse noise detection technique for SMF was proposed (SWM) that was based on the minimum absolute value of four convolutions obtained using one-dimensional Laplacian operators. SMF with boundary discriminative noise detection for extremely corrupted images [14] used two boundaries by which image pixels were classified into three groups named lower intensity impulse noise, uncorrupted pixels, and higher intensity impulse noise. Then, a modified NASM was utilized to restore corrupted pixels. Directional weighted median filter (DWM) [15] detected the noisy pixels based on the difference between the center pixel and its neighbors aligned with four main directions. Then, WMF was applied iteratively to restore the noisy pixels. In each iteration, the threshold decreased until the maximum number of iterations was reached. In [16], SMF was modified by adding one more noise detector based on the rank order arrangement of pixels in the sliding window. Adaptive switching median filter (ASMF) [17] was another modification to SMF in which the threshold was computed locally from pixels inside the sliding window.

In recent years, some effective noise removal algorithms with local statistics-based impulse noise detectors have been developed. A new statistic based on the Rank-Ordered Absolute Difference (ROAD) [18] was introduced that represented how impulse-like a pixel was in the sense that the larger the impulse, the greater the ROAD value. Then, by incorporating this statistic into a bilateral filtering, a new nonlinear filter was proposed (trilateral filter) which could remove both Gaussian and impulse noise. The Rank-Ordered Logarithmic Difference (ROLD) [19] was developed to improve the performance of the ROAD statistic by identifying more noisy pixels with less false hits. By combining it with an edge-preserving regularization (EPR), ROLD-EPR method was implemented to remove RVIN. A partial differential equation-based image denoising method for random-valued impulse noise (NSDD) [20] was proposed in which two controlling functions were used to distinguish between edge pixels, noisy pixels, and interior pixels. In [21], a detection algorithm for RVIN (ODM) was developed that calculated the standard deviation in different directions in the filtering window. Once the optimal direction was found, a pixel was detected as noise-free if it was similar to pixels in the optimum direction. A fuzzy weighted NLM filter (FWNLM) [22] was implemented that was able to remove RVIN and mixed Gaussian-RVIN. Based on the fuzzy weighting function, the more a pixel was contaminated, the less it was used to restore images. In [23], a new WMF with a two-phase noise detector was proposed. In the first phase, the Rank-Ordered Difference of ROAD (ROD-ROAD) was introduced in which a fuzzy rule was used to

detect noisy pixels. In the second phase, another image statistic (minimum edge pixels difference) was proposed to distinguish between edge pixels and noisy candidates. To restore the corrupted images, an iterative denoising algorithm was utilized by combining the proposed two-phase noise detector and the new WMF. A new image denoising method ( $\ell_0$ TV-PADMM) [24] was implemented that was based on the total variation (TV) with  $\ell_0$ -norm data fidelity. Since the resulting optimization problem was non-convex and non-smooth, it was first reformulated as an equivalent mathematical program with equilibrium constraints and then it was solved using a proximal Alternating Direction Method of Multipliers (PADMM). In [25], a new two-phase denoising algorithm (DPC-INR) was implemented using dissimilar pixel counting. In the detection phase, the average difference scheme was used to distinguish whether two neighboring pixels were similar or not, and then the number of dissimilar pixels was compared with a threshold to determine whether the current pixel was noisy. In the filtering phase, an extended trilateral filter was utilized to restore noisy images. An adaptive rank-ordered impulse detector based on local statistics (AROPD-EPR) [26] was introduced in which a piecewise power function was applied to the rank-ordered statistic to enlarge the difference between noisy pixels and noise-free pixels. By combining the noise detector with an improved EPR filter, an effective two-stage iterative denoising algorithm was implemented to remove RVIN.

In this paper, an efficient RVIN removal method was proposed that consisted of three steps (i.e., noise density estimation, noise detection, and image restoration). We made two main contributions one of which was the noise density estimation. As opposed to most existing denoising methods that assume the noise density is known, our method estimated the noise density with high accuracy. The second contribution was proposing an effective RVIN detector using local statistics. Based on the estimated noise density, a noise detector was implemented to detect corrupted pixels. Finally, a modified weighted mean filter was utilized to restore the detected noisy pixels while leaving the noise free pixels unchanged.

The rest of the paper is organized as follows: Section II briefly reviews the ROAD statistic for detecting RVIN. The proposed method is described in Section III. Section IV presents the experimental results and draws a comparison with other state-of-the-art image denoising methods. Finally, Section V provides the conclusion.

## II. REVIEW ON THE RANK-ORDERED ABSOLUTE DIFFERENCE STATISTIC

The RVIN model with noise probability  $p$  can be described as follows where  $u_{ij}$  and  $o_{ij}$  denote the pixel intensity at location  $(i, j)$  of the noisy image and original image, respectively, and  $n_{ij}$  denotes the value of the noisy pixel at location  $(i, j)$ .

$$u_{ij} = \begin{cases} o_{ij}, & \text{with probability } 1 - p \\ n_{ij}, & \text{with probability } p \end{cases} \quad (1)$$

Unlike SAP noise that takes on either the minimum or maximum intensity, RVIN can take any values between the

minimum and maximum intensity with equal probability. Thus, detecting noisy pixels corrupted by RVIN is much more difficult than SAP.

The ROAD statistic [18] is a widely used local image statistic for detecting RVIN. Let

$$\Omega_{i,j}(N) = \{(i+s, j+t) | -N \leq s, t \leq N\} \quad (2)$$

denote the set of image coordinates in a  $(2N+1) \times (2N+1)$  window centered at  $(i, j)$  for some positive integer  $N$ . Let  $\Omega_{i,j}^0 = \Omega_{i,j} \setminus (i, j)$  be the same as  $\Omega_{i,j}(N)$  without its center coordinate. For each coordinate in  $\Omega_{i,j}^0$ , define  $d_{s,t}(u_{i,j})$  as the absolute difference in intensity of the pixels  $u_{i+s, j+t}$  and  $u_{i,j}$ , i.e.,

$$d_{s,t}(u_{i,j}) = |u_{i+s, j+t} - u_{i,j}|, \text{ for } -N \leq s, t \leq N. \quad (3)$$

After sorting the values of  $d_{s,t}(u_{i,j})$  in ascending order, the ROAD statistic can be defined as follows:

$$ROAD_m(u_{i,j}) = \sum_{k=1}^m r_k(u_{i,j}) \quad (4)$$

where  $2 \leq m \leq (2N+1)^2 - 2$  and  $r_k(u_{i,j})$  is the  $k^{\text{th}}$  smallest value of  $d_{s,t}(u_{i,j})$ .

There should be a big difference between the intensity of most corrupted pixels by RVIN and their neighbors while the intensity of most uncorrupted pixels (even edge pixels) should be close to at least half of their neighboring pixels. Therefore, the ROAD value of noisy pixels should be larger than that of noise-free pixels. As a result, the Road value can be used to detect corrupted pixels by RVIN. For a  $3 \times 3$  window (i.e.,  $N = 1$ ), it is suggested to set the value of  $m$  to 4 while for a  $5 \times 5$  window (i.e.,  $N = 2$ ) it is recommended to set the value of  $m$  to 12 [18].

### III. PROPOSED METHOD

The proposed method consists of three steps. The first step is the noise density estimation in which RVIN density is estimated with high accuracy. In the second step, a noise detector is utilized to detect corrupted pixels based on the estimated noise density. In other words, the parameters of the noise detector are determined specifically for each noise level. Finally, in the last step, a modified weighted mean filter (MWMF) is used to restore the detected noisy pixels. Each step is explained in detail in the following sections. It is worth mentioning that there are some parameters in each step whose values are determined by using an evaluation dataset that contains 20 images. The evaluation dataset is a subset of BSDS68 dataset [27]. In order to compare the performance of the proposed method with other denoising algorithms, a different dataset, containing 49 images [28], is used.

#### A. Noise Density Estimator

In the proposed method, a  $3 \times 3$  sliding window with  $m = 4$  is used to calculate the ROAD value of image pixels. If the ROAD value of a pixel is larger than a pre-defined threshold, the pixel will be considered as noisy; otherwise, the pixel will be considered as noise-free. The evaluation dataset is used to determine the value of the threshold to distinguish between noisy and noise-free pixels. The value of the threshold ( $T =$

78) is found by trial and error. By sliding the window over the entire image and counting the number of corrupted pixels, the estimated noise density can be easily computed. For instance, if the number of detected noisy pixels in a  $512 \times 512$  grayscale image is 52,430, the estimated noise density is about 20%.

#### B. RVIN Detection

In order to decide whether an image pixel  $u_{i,j}$  is noisy, all 24 neighboring pixels within a  $5 \times 5$  window, centered at  $u_{i,j}$ , are considered in two stages. In the first stage, we decide whether the center pixel  $u_{i,j}$  is a noise candidate or noise-free pixel. All detected noise candidates are considered in the second stage to make sure that they are not edge pixels, falsely detected as noise.

The underlying logic of distinguishing between noisy and edge pixels is that in a  $5 \times 5$  window that does not contain edge pixels, clean pixels should have close intensities while the intensities of contaminated pixels by RVIN vary considerably. In other words, in a smooth area of an image that does not contain edge pixels, there is only one group of clean pixels whose intensities are close together; however, if  $5 \times 5$  window contains some edge pixels, there could be two groups of pixels whose intensities might differ greatly and yet both groups could be clean pixels.

In the first stage of the proposed method, we first sort the values of all 24 neighboring pixels in ascending order, and then compute the difference between each two successive elements to generate a vector called `diff_sort_vec`. We then find two biggest groups in `diff_sort_vec` whose elements are smaller than a threshold ( $T_1$ ). We consider the first biggest group as clean pixels, called `first_clean_pxls`, because all the corresponding pixels to this group have close intensities. If there are at least six pixels in the second biggest group, we consider it as the second clean pixels group, called `second_clean_pxls`, which means the  $5 \times 5$  sliding window might contain some edge pixels. Now, if the center pixel,  $u_{i,j}$ , satisfies any of four following conditions, we detect it as a noise candidate in the first stage. The first condition is that the center pixel,  $u_{i,j}$ , is not within the range of the first clean pixels group (i.e., `first_clean_pxls`). The second condition is that the ROAD value of the center pixel ( $N = 2$  and  $m = 10$ ) be larger than or equal to a threshold ( $T_2$ ). For the third condition, let  $n_{5by5}$  denotes the number of neighboring pixels, within the  $5 \times 5$  window, whose absolute difference from the center pixel is smaller than a threshold ( $T_3$ ). The third condition is met if  $n_{5by5}$  is smaller than another threshold ( $T'_3$ ). The only difference between the third and fourth conditions is that we consider the neighboring pixels within a  $3 \times 3$  window ( $n_{3by3}$ ), instead of a  $5 \times 5$  window, and use different thresholds ( $T_4$  and  $T'_4$ ) in the fourth condition.

In the second stage, if detected noise candidates satisfy all of four following conditions, their status will be changed to noise-free pixels. In other words, the detected noise candidates in the first stage could be edge pixels (i.e., noise-free pixels). The first condition is that the center pixel,  $u_{i,j}$ , is within the range of the second clean pixels group (i.e., `second_clean_pxls`). The second condition is that the ROAD value of the center pixel be smaller than a threshold ( $T_5$ ). To meet the

third and fourth conditions,  $n_{5by5}$  and  $n_{3by3}$  must be greater than two thresholds ( $T_6$  and  $T_7$ ). The RVIN detection algorithm can be summarized as follows:

**Proposed RVIN detection algorithm:**

**Input:**  $5 \times 5$  window ( $w$ )

**Output:**  $d \in \{0,1\}$

**First stage:**

1. Remove the center pixel ( $w(3,3)$ ) of the  $5 \times 5$  window.
2. Vectorize and sort the remaining 24 pixels in ascending order to generate a vector that we call *sorted\_w*.
3. Calculate the difference between each two successive elements in *sorted\_w* to generate a vector called *diff\_sort\_vec*
4. Find two biggest groups of elements in *diff\_sort\_vec* whose elements are smaller than a threshold ( $T_1$ ). The first and second biggest groups are called *first\_clean\_pxls* and *second\_clean\_pxls*, respectively.
5. Calculate the ROAD value of the center pixel with  $N = 2$  and  $m = 10$
6. Find the number of neighboring pixels ( $n_{5by5}$ ), within the  $5 \times 5$  window, whose absolute difference from the center pixel is smaller than a threshold ( $T_3$ ).
7. Find the number of neighboring pixels ( $n_{3by3}$ ), within the  $3 \times 3$  window, whose absolute difference from the center pixel is smaller than a threshold ( $T_4$ ).
8. If  $[w(3,3) \notin \text{first\_clean\_pxls}]$  OR  $[\text{ROAD} \geq T_2]$  OR  $[n_{5by5} \leq T'_3]$  OR  $[n_{3by3} \leq T'_4]$ , then  $d = 1$ , i.e., the center pixel is a noise candidate.

**Second stage:**

9. If  $[d = 1]$  AND  $[\text{length}(\text{second\_clean\_pxls}) \geq 6]$  AND  $[w(3,3) \in \text{second\_clean\_pxls}]$  AND  $[\text{ROAD} \leq T_6]$  AND  $[n_{5by5} \geq T_6']$  AND  $[n_{3by3} \geq T_7']$ , then  $d = 0$ , i.e., the center pixel is a noise-free pixel.

**C. Image Restoration Method**

In this paper, we slightly adapted the modified mean filter (MMF) proposed in [29] to restore contaminated pixels by RVIN. If the estimated noise density is less than 35%, contaminated images were restored once; otherwise, they were restored twice to improve their visual quality. In the original MMF method, the center value of a  $3 \times 3$  sliding window is replaced by the mean of its four horizontal and vertical neighbors aligned with the four main directions, if only each of which is a noise-free pixel. If all neighbors of a center pixel are noisy, it is necessary to move toward the defined directions shown in Fig. 1 to reach closest four noise-free pixels that we call  $a, b, c,$  and  $d$ . If a pixel is noisy, it will be replaced with another noise-free pixel according to the flowchart that is shown in Fig. 2. Thereafter, the value of center pixel can be simply calculated as the mean of these noise-free pixels, i.e.

$$P0 = \frac{a+b+c+d}{4} \tag{5}$$

We make two modifications to the original MMF. The first modification is that instead of taking the mean of four noise-free pixels, we take the weighted average of four pixels according to the weights proposed in Fig. 3. In the original MMF, the neighboring pixels can be picked more than once. The second modification is that each neighboring pixel cannot be picked more than once.

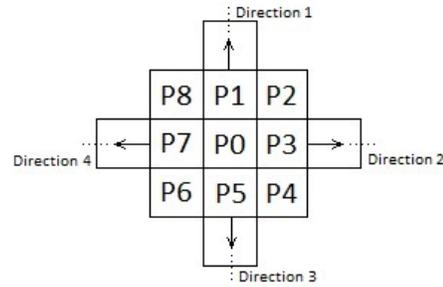


Fig. 1. Four Directions for Selecting Noise-Free Pixels [29].

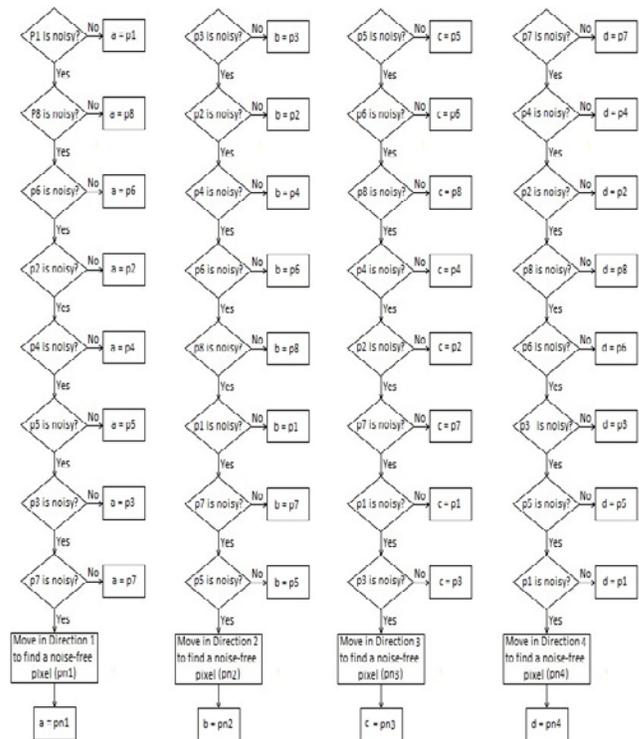


Fig. 2. Flow Chart for Selecting Noise-free Pixels [29].

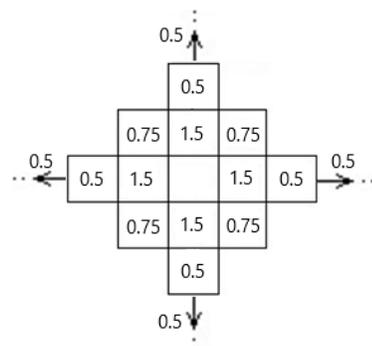


Fig. 3. Assigned Weights to Neighboring Pixels.

**IV. RESULTS AND DISCUSSION**

To compare the performance of our RVIN removal algorithm with other methods, an image dataset [28] consisting of 49 8-bit/pixel grayscale images of size  $512 \times 512$  are used. Our method is compared with 10 well-known RVIN removal methods all of which are discussed in the introduction. These

methods are ACWM [12], SWM [13], DWM [15], ROLD-EPR [19], NSDD [20], ODM [21], FWNLM [22], t0TV-PADMM [24], DPC-INR [25], and AROPD-EPR [26].

Three well-known metrics (i.e., recall, precision, and F1-score) are used to evaluate the performance of the detection algorithm. Recall (R) is the ratio of correctly detected noisy pixels to the actual noisy pixels while precision (P) is the ratio of correctly detected noisy pixels to all pixels detected as noisy. F1-score is the harmonic mean of precision and recall that is widely used in the field of information retrieval. Note that higher values of F1-score are better that requires both recall and precision be high. The best value for F1-score is one that can be reached only when both recall and precision are equal to one. These metrics can be calculated using the following formulas.

$$R = \frac{TN}{TN + FF} \quad (6)$$

$$P = \frac{TN}{TN + FN} \quad (7)$$

$$F1 - score = \frac{2}{\frac{1}{R} + \frac{1}{P}} \quad (8)$$

where TN is the number of pixels that are correctly detected as noisy, FF is the number of noisy pixels that are falsely detected as noise-free, and FN is the number of noise-free pixels that are falsely detected as noisy.

Restoration results are quantitatively measured by the peak signal-to-noise ratio (PSNR), mean absolute error (MAE), and two-dimensional correlation coefficient (COR) calculated using the following equations.

$$PSNR = 10 \log_{10} \left( \frac{255^2}{MSE} \right) \quad (9)$$

$$MSE = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (U_n(i, j) - U_d(i, j))^2 \quad (10)$$

$$MAE = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N |U_n(i, j) - U_d(i, j)| \quad (11)$$

$$COR = \frac{\sum_{i=1}^M \sum_{j=1}^N (U_n(i, j) - \bar{U}_n)(U_d(i, j) - \bar{U}_d)}{\sqrt{(\sum_{i=1}^M \sum_{j=1}^N (U_n(i, j) - \bar{U}_n)^2)(\sum_{i=1}^M \sum_{j=1}^N (U_d(i, j) - \bar{U}_d)^2)}} \quad (12)$$

where  $U_n$  and  $U_d$  are the noisy and de-noised images of size  $M \times N$ , respectively.

### A. Noise Estimation Results

Table I shows the average noise density estimation over 20 images of the evaluation dataset for each noise level. Evaluation dataset is a subset of BSDS68 dataset [27] that is used to determine the values of the parameters (e.g., thresholds). Table II demonstrates the average noise density estimation over 49 images of the test dataset [28]. As can be seen, the proposed method is able to estimate the noise density with high accuracy.

TABLE I. THE AVERAGE NOISE DENSITY ESTIMATION OVER 20 IMAGES OF THE EVALUATION DATASET

Actual noise density	20%	30%	40%	50%	60%
Estimated noise density	19.3%	28.3%	38%	48.6%	59.7%

TABLE II. THE AVERAGE NOISE DENSITY ESTIMATION OVER 49 IMAGES OF THE TEST DATASET

Actual noise density	20%	30%	40%	50%	60%
Estimated noise density	21.3%	30.2%	39.7%	50%	60.7%

### B. Noise Detection Results

The performance of RVIN removal methods with noise detector heavily depends on the proper detection of noisy pixels which is not an easy task for RVIN. We compare the performance of our proposed noise detection algorithm with ACWM, SWM, ROAD, ROLD, ECROAD, and AROPD. The average recall, precision, and F1-score for the images of the test dataset for five noise levels are shown in Fig. 4(a), 4(b), and 4(c), respectively. The results of other methods for noise detection, restoration, and run time comparison are taken from [26] in which the parameters are selected as they are suggested in the original papers.

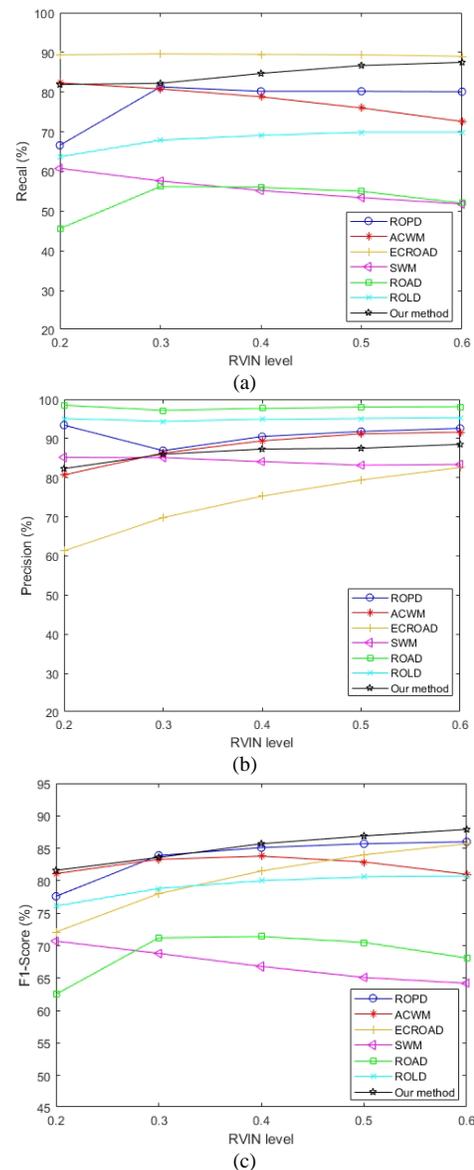


Fig. 4. Comparison of different RVIN Detectors: (a) Average Recall, (b) Average Precision, (c) Average F1-score.

As can be seen from Fig. 4(a), the average recall of our method is lower than that of ECROAD method, but ECROAD method achieved the lowest average precision (Fig. 4b) which means it falsely detects a large number of noise-free pixels as noisy. Similarly, ROAD method achieves the highest average precision and very low average recall which means it wrongly detects a large number of noisy pixels as noise-free. Thus, it is important to consider the F1-score to seek a balance between the recall and precision. As can be seen from Fig. 4(c), except for one noise level (30%), our method achieves the highest average F1-score for all other noise levels indicating the superiority of our noise detection method over other algorithms.

C. Restoration Results

Three well-known metrics (i.e., PSNR, COR, and MAE) are used to quantitatively compare the performance of image restoration. Note that higher values of PSNR and COR are better, whereas lower values of MAE are better. Table III demonstrates the average PSNR of different methods on 49 test images corrupted by five noise densities varying from 20% to 60% with increments of 10. The best average PSNR for each noise density is highlighted in bold. For four noise densities, our method achieved the highest average PSNR while for other noise density (i.e., 60%), it achieved the second best average PSNR. The average COR and MAE are shown in Tables IV and V, respectively. In terms of average COR, our method obtained the best average COR for three noise densities (20%, 30%, and 40%) while for the rest of noise densities, its result is comparable with other methods that achieved higher values for average COR. In terms of average MAE, our method outperformed all other methods for two noise densities (20% and 30%) while for other noise densities, it achieved the second best results. Fig. 5 and 6 demonstrate visual comparison between noise removal methods for the “butterfly” and “bridge” test images corrupted by 55% and 20% RVIN, respectively. It can be seen that our method can remove noise from corrupted images and preserve sharp edges and fine details of the images that yield visually pleasant restoration results.

TABLE III. IMAGE RESTORATION COMPARISON FOR DIFFERENT NOISE DENSITIES OVER 49 TEST IMAGES IN TERMS OF AVERAGE PSNR (dB)

Table with 6 columns: Method, Noise density (20%, 30%, 40%, 50%, 60%). Rows include ACWM, SWM, DWM, ROLD-EPR, NSDD, ODM, FWNLM, t0TV-PADMM, DPC-INR, AROPD-EPR, and Our Method.

TABLE IV. IMAGE RESTORATION COMPARISON FOR DIFFERENT NOISE DENSITIES OVER 49 TEST IMAGES IN TERMS OF AVERAGE COR

Table with 6 columns: Method, Noise density (20%, 30%, 40%, 50%, 60%). Rows include ACWM, SWM, DWM, ROLD-EPR, NSDD, ODM, FWNLM, t0TV-PADMM, DPC-INR, AROPD-EPR, and Our Method.

TABLE V. IMAGE RESTORATION COMPARISON FOR DIFFERENT NOISE DENSITIES OVER 49 TEST IMAGES IN TERMS OF AVERAGE MAE

Table with 6 columns: Method, Noise density (20%, 30%, 40%, 50%, 60%). Rows include ACWM, SWM, DWM, ROLD-EPR, NSDD, ODM, FWNLM, t0TV-PADMM, DPC-INR, AROPD-EPR, and Our Method.

D. Run Time

The average run time of the denoising methods on 49 test images is shown in Table VI. All experiments were performed on computers equipped with 3.40 GHz CPU. Although the run time of our method is longer than some other methods, it should be noted that it achieved better noise detection and image restoration results in the vast majority of the cases. The run time for heavy noise corruption (i.e., 40%, 50%, and 60%) is about twice longer than that of the low levels of noise corruption because in our method, highly contaminated images were restored twice to improve the quality of restored images. It is worth noting that the noise detection stage of our method took the significant portion of the run time, so future work would be to implement the noise detection algorithm in an optimized and faster way to reduce the run time. Another way to reduce the run time would be running our algorithm in parallel (i.e., parallel computing).

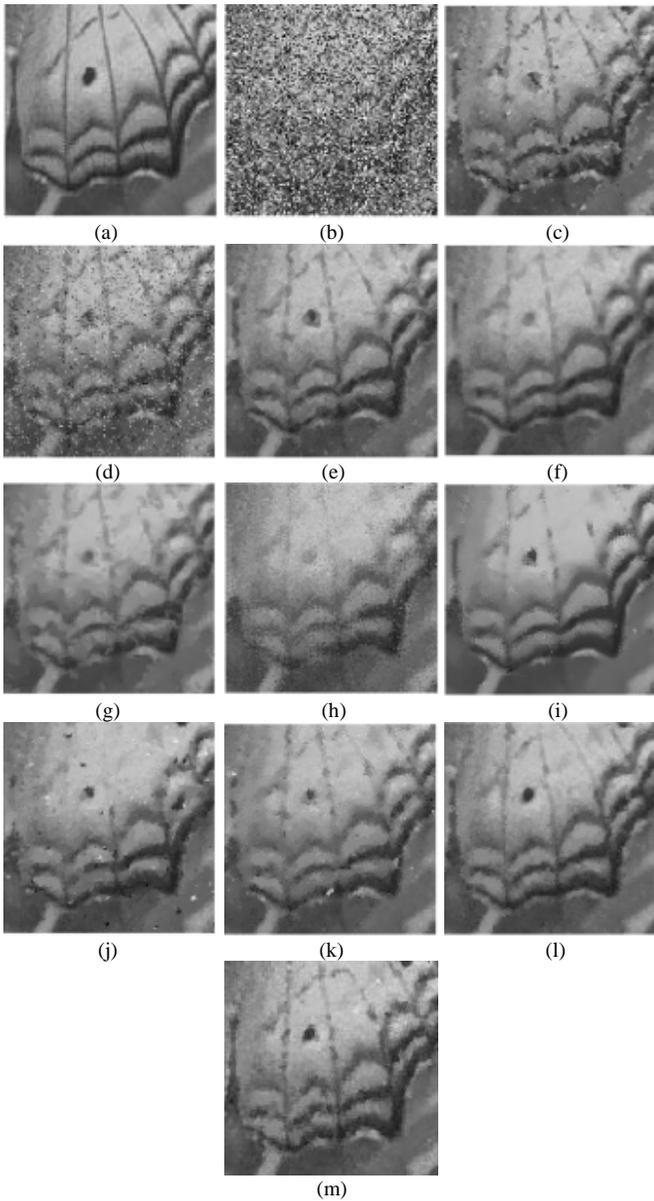


Fig. 5. Comparison of Image Restoration Results of different Methods for Image "Butterfly" Corrupted by 55% RVIN: (a) Clean Image, (b) Noisy Image (55% RVIN), (c) ACWM, (d) SWM, (e) DWM, (f) ROLD-EPR, (g) NSDD, (h) ODM, (i) FWNLM, (j) t0TV-PADMM, (k) DPC-INR, (l) AROPD-EPR, (m) our Method.

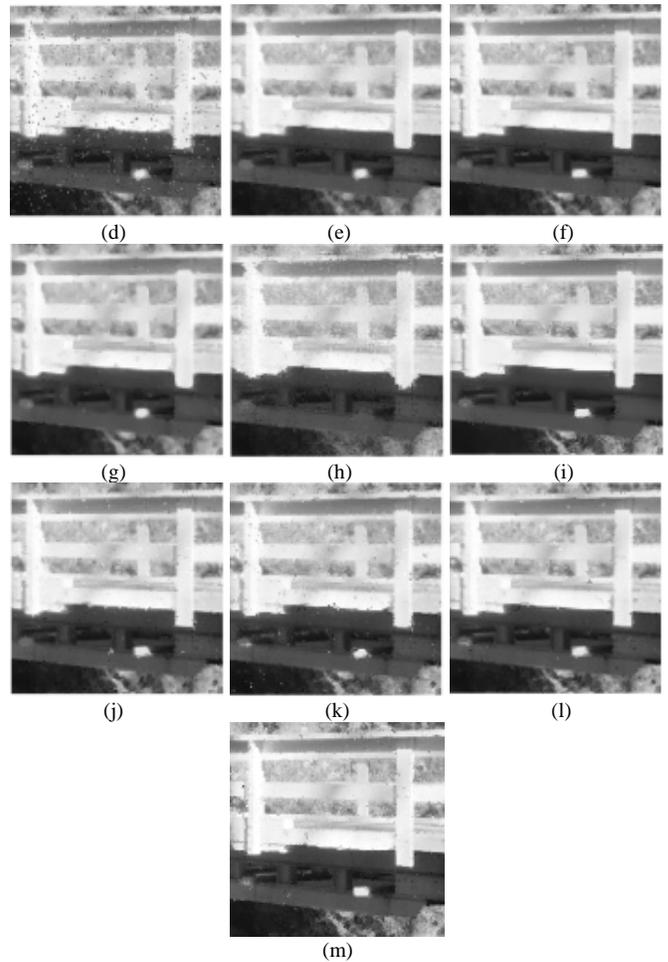
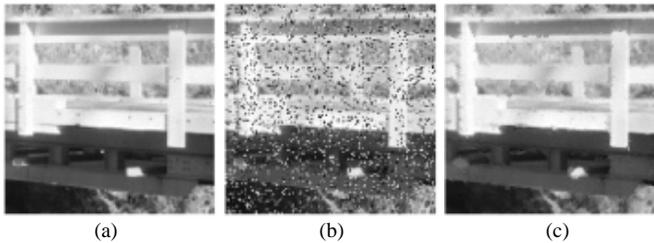


Fig. 6. Comparison of Image Restoration Results of different Methods for Image "Bridge" Corrupted by 20% RVIN: (a) Clean Image, (b) Noisy Image (20% RVIN), (c) ACWM, (d) SWM, (e) DWM, (f) ROLD-EPR, (g) NSDD, (h) ODM, (i) FWNLM, (j) t0TV-PADMM, (k) DPC-INR, (l) AROPD-EPR, (m) our Method.

TABLE VI. COMPARISON OF RUN TIME IN SECONDS

Method	Noise density				
	20%	30%	40%	50%	60%
ACWM	5.53	5.66	5.60	5.60	5.60
SWM	4.56	4.56	4.48	4.69	4.57
DWM	46.98	45.55	44.82	44.83	46.67
ROLD-EPR	1.63	4.16	5.07	5.52	7.75
NSDD	0.62	0.63	1.31	3.76	5.71
ODM	15.58	14.35	14.46	14.88	15.50
FWNLM	276.39	268.56	277.34	274.38	271.71
t0TV-PADMM	1.38	1.39	1.43	1.49	1.55
DPC-INR	9.15	13.54	14.32	14.71	14.78
AROPD-EPR	1.76	2.82	4.74	7.12	11.25
Our Method	7.16	7.33	13.89	14.19	14.27

## V. CONCLUSION

In this paper, we presented an efficient three-step noise removal method for grayscale images corrupted by RVIN. In the first step, we estimated the noise density of corrupted images with high accuracy. Based on the estimated noise density, a noise detector found corrupted pixels that were restored by using a modified weighted mean filter. In order to evaluate the performance of the proposed method, we drew a comparison with 10 denoising algorithms. In the vast majority of the cases, our method outperformed other algorithms which indicated the effectiveness of the proposed method. Future work would be to implement the noise detection algorithm in a faster way to reduce the run time.

### REFERENCES

- [1] B. Goyal, A. Dogra, S. Agrawal, B. S. Sohi, and A. Sharma, "Image denoising review: From classical to state-of-the-art approaches," *Inf. Fusion*, vol. 55, pp. 220–244, 2020.
- [2] L. Fan, F. Zhang, H. Fan, and C. Zhang, "Brief review of image denoising techniques," *Vis. Comput. Ind. Biomed. Art*, vol. 7, 2019.
- [3] K. Zhang, W. Zuo, and L. Zhang, "FFDNet: Toward a fast and flexible solution for CNN based image denoising," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4608–4622, 2017.
- [4] M. Aghajarian, J. E. McInroy, and S. Muknahallipatna, "Deep learning algorithm for Gaussian noise removal from images," *J. Electron. Imaging*, vol. 29, no. 04, p. 1, 2020.
- [5] W. K. Pratt, "Median filtering," Los Angeles, 1975.
- [6] R. E. Gonzalez, Rafael C., Woods, *Digital Image Processing*. 2002.
- [7] D. R. K. Brownrigg, "The Weighted Median Filter," *Commun. ACM*, vol. 27, no. 8, pp. 807–818, 1984.
- [8] S. J. Ko and Y. H. Lee, "Center Weighted Median Filters and Their Applications to Image Enhancement," *IEEE Transactions on Circuits and Systems*, vol. 38, no. 9, pp. 984–993, 1991.
- [9] H. Hwang and R. A. Haddad, "Adaptive Median Filters: New Algorithms and Results," *IEEE Trans. IMAGE Process.*, vol. 4, no. 4, pp. 499–502, 1995.
- [10] T. Sun and Y. Neuvo, "Detail-preserving median based filters in image processing," *Pattern Recognit. Lett.*, vol. 15, no. 4, pp. 341–347, 1994.
- [11] H. L. Eng and K. K. Ma, "Noise adaptive soft-switching median filter," *IEEE Trans. Image Process.*, vol. 10, no. 2, pp. 242–251, 2001.
- [12] T. Chen and H. R. Wu, "Adaptive impulse detection using center-weighted median filters," *IEEE Signal Process. Lett.*, vol. 8, no. 1, pp. 1–3, 2001.
- [13] S. Zhang and M. A. Karim, "A New Impulse Detector for Switching Median Filters," *IEEE Signal Process. Lett.*, vol. 9, no. 11, pp. 360–363, Nov. 2002.
- [14] P. Ng and K.-K. Ma, "A Switching Median Filter With Boundary Discriminative Noise Detection for Extremely Corrupted Images," *IEEE Trans. Image Process.*, vol. 15, no. 6, pp. 1506–1516, 2006.
- [15] Y. Dong and S. Xu, "A new directional weighted median filter for removal of random-valued impulse noise," *IEEE Signal Process. Lett.*, vol. 14, no. 3, pp. 193–196, 2007.
- [16] C. C. Kang and W. J. Wang, "Modified switching median filter with one more noise detector for impulse noise removal," *AEU - Int. J. Electron. Commun.*, vol. 63, no. 11, pp. 998–1004, 2009.
- [17] S. Akkoul, R. Lédée, R. Leconge, and R. Harba, "A new adaptive switching median filter," *IEEE Signal Process. Lett.*, vol. 17, no. 6, pp. 587–590, 2010.
- [18] R. Garnett, T. Huegerich, C. Chui, and W. He, "A universal noise removal algorithm with an impulse detector," *IEEE Trans. IMAGE Process.*, vol. 14, no. 11, pp. 1747–1754, 2005.
- [19] Y. Dong, R. H. Chan, and S. Xu, "A detection statistic for random-valued impulse noise," *IEEE Trans. Image Process.*, vol. 16, no. 4, pp. 1112–1120, 2007.
- [20] J. Wu and C. Tang, "PDE-Based Random-Valued Impulse Noise Removal Based on New Class of Controlling Functions," *IEEE Trans. Image Process.*, vol. 20, no. 9, pp. 2428–2438, 2011.
- [21] A. S. Awad, "Standard deviation for obtaining the optimal direction in the removal of impulse noise," *IEEE Signal Process. Lett.*, vol. 18, no. 7, pp. 407–410, 2011.
- [22] J. Wu and C. Tang, "Random-valued impulse noise removal using fuzzy weighted non-local means," *Signal, Image Video Process.*, vol. 8, no. 2, pp. 349–355, 2014.
- [23] L. Liu, C. L. P. Chen, Y. Zhou, and X. You, "A new weighted mean filter with a two-phase detector for removing impulse noise," *Inf. Sci. (Ny.)*, vol. 315, pp. 1–16, 2015.
- [24] G. Yuan and B. Ghanem, "LOTV-MCP: A New Method for Image Restoration in the Presence of Impulse Noise," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5369–5377.
- [25] Z. Shi, Z. Xu, K. Pang, Q. Cao, and T. Luo, "Dissimilar pixel counting based impulse detector for two-phase mixed noise removal," *Multimed. Tools Appl.*, vol. 77, no. 6, pp. 6933–6953, 2018.
- [26] Q. Xu, Y. Li, Y. Guo, S. Wu, and M. Sbert, "Random-valued impulse noise removal using adaptive ranked-ordered impulse detector," *J. Electron. Imaging*, vol. 27, no. 01, p. 013001, 2018.
- [27] S. Roth and M. J. Black, "Fields of experts," *Int. J. Comput. Vis.*, vol. 82, no. 2, pp. 205–229, 2009.
- [28] J. A. García et al., "COMPOUND GAIN: a visual distinctness metric for coder performance evaluation," <https://ccia.ugr.es/cvg/CG/base.htm>.
- [29] M. Aghajarian, J. E. McInroy, and C. H. G. Wright, "Salt-and-pepper noise removal using modified mean filter and total variation minimization," *J. Electron. Imaging*, vol. 27, no. 01, p. 013002, 2018.