IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 24, 2020, accepted June 15, 2020, date of publication June 19, 2020, date of current version June 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3003874

Spatially Adaptive Image Denoising via Enhanced Noise Detection Method for Grayscale and Color Images

AMANDEEP SINGH[1], (Member, IEEE), GAURAV SETHI[1], AND G. S. KALRA[2], (Member, IEEE)
[1]School of Electronics and Electrical Engineering, Lovely Professional University, Jalandhar 144411, India
[2]Department of Electronics and Communication Engineering, CT Group of Institutions, Jalandhar 144020, India

Corresponding author: Amandeep Singh (amansandhu6788@gmail.com)

ABSTRACT Keeping in view the variety of the applications, image denoising still remains the unexplored territory for the researchers. There are many pros and cons in existing denoising algorithms. The two prime cons of image denoising algorithms are (i) Over and under detection of noisy pixels (ii) Low performance at high noise levels. So, in order to overcome these existing issues, a spatially adaptive image denoising via enhanced noise detection method (SAID-END) is proposed for grayscale and color images. The denoising is achieved using a two-stage sequential algorithm, the first stage ensures accurate noise estimation by eliminating over and under detection of noisy pixels. The second stage performs image restoration by considering non-noisy pixels in estimation of the original pixel value. To enhance the accuracy while denoising high-density impulse noise and artifacts, both noise estimation and restoration stages are using a spatially adaptive window (window expands to spatially connected area), the size of the window depends upon the noise level in the vicinity of the reference noisy pixel. The two stages of the proposed method are referred to as (i) Enhanced adaptive noise detection (ii) Non-corrupted pixel sensitive adaptive image restoration. The proposed method is evaluated by two test steps to ensure its versatility and robustness. In the first step, the proposed method is tested on a wide standard data set of color and grayscale images affected by impulse noise and artifacts. The results of proposed method are compared with well-known methods compatible for denoising impulse noise and artifacts. In the second step, the results of proposed method are compared with the recent state of the art algorithms for traditional test images. The result shows that the proposed method outperforms the existing denoising methods when applied to grayscale and color images.

INDEX TERMS Adaptive window, denoising, image enhancement factor (IEF), peak signal to noise ratio (PSNR), structural similarity index (SSIM).

I. INTRODUCTION

Various applications like recognition, edge detection, medical imaging and satellite imaging require high-quality noise free images. So, it is a necessity to denoise the images as preprocessing to such applications. It is important to retain information such as edges, texture and structure details while performing the image denoising. Specifically, edges are extremely important in the biomedical field analysis like forensic examination and hairline cracks in bones. Noise and artifacts are two major contributors in image quality deterioration. In digital images, salt & pepper noise deteriorate the image quality by introducing extreme values 0 and 255 [1]. The artifacts represent deterioration of image quality due to spots and scratches.

Median filter (MF) [2], [3] is one of the most popular standard non-liner filter to remove impulse noise. This filter performs well for low level of noise but the performance of filter reduces drastically as noise level increases. Modified forms of MF are commonly used and preferred for image denoising till date. Decision Based Median Filter (DBMF), Center Weighted Median Filter (CWMF), Progressive Switching Median Filter (PSMF), Different Applied Median Filter (DAMF), and Iterative Mean Filter (IMF) are some examples of methods modified from MF. DBMF [4]–[6] is an effective method for denoising low and mid noise density affected images. This method introduces blurring and artifacts at high noise levels. The CWMF [7], [8] algorithm

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

provides better proximity to original values by providing more weight to center values of the window. This method achieves better visual performance but as more values get corrupted in the selected window due to high noise density, its performance dilutes. PSMF [9]–[11] is a two-stage cascaded process, first noise is detected and then restored iteratively. Pixel values of the current iteration are considered for calculation of pixel value in the next iteration. This method can denoise impulse noise and low level of blotches. DAMF [11] algorithm was developed to operate on a wide range of impulse noise. It can successfully denoise all range of impulse noise but its performance declines sharply as noise density becomes very high. IMF [12] is using a fixed window base iterative mechanism for high-density impulse noise reduction. This is a very promising method and achieves the desired results. The use of a fixed window provides high speed operation of IMF algorithm but affects its accuracy when very high noise density is present in the window. Some other algorithms were also designed by using trimmed values to avoid noise effect on original value estimation. Rank Ordered Absolute Differences with Trimmed Global Mean filter (ROAD-TGM) [6] is a window base two-stage algorithm, where the first stage focus on noise detection and the second stage ensures the desired restoration. This algorithm uses TGM when all values of the selected window are noisy. Similarly, adaptive unsymmetric trimmed shock filter (AUTSF) [13] is also a two-stage process for the detection and restoration of noisy image. This algorithm performs well on both color and grayscale images. Modified cascaded filter (MCF) [14] is a hybrid approach using trimmed median values to neglect the effect of noise on the restoration stage. This algorithm can operate well on color images affected by impulse noise. Fuzzy decision-based algorithms and supervised data-driven models were also developed to enhance the image denoising for impulse noise. Adaptive Type-2 Fuzzy Filter (FDS, fuzzy denoising for Impulse noise) [15] this is also two-stage algorithms, where first stage operates to classify pixel as good or bad and second stage, uses the weighted mean value for the restoration of noisy value. Iterative scheme-inspired network (IIN) [16] denoises on the basis of training data, the accuracy of the algorithm depends on size and type of images in the dataset.

As discussed above various algorithms are available to denoise the noisy image, these algorithms work mainly on two-stage procedure (i) Detection of noisy pixel and (ii) Restoration of the noisy pixel. The success of such algorithms depends upon the individual performance of these respective stages [17]–[21]. For the detection stage, the performance of the algorithm depends upon how accurately locations of corrupted pixels are detected. Detecting corrupted pixels locations in the presence of noise is a very challenging task that causes false detection once the noise level increases to a certain level. This leads to the problem of under and overdetection of noisy pixels. Similarly, for noisy pixels, restoration stage performance depends upon how close the algorithm restores the corrupted pixel value to the original value. As noise level increases in the image more of the neighboring pixels tend to get corrupted and it is very difficult to restore the desired values of the pixel [22]–[25]. Therefore, in order to overcome the problem of over/under detection and to restore the value of corrupted pixels close to the original values, the SAID-END method is proposed. The proposed method works exceedingly well in high noise scenarios for both grayscale and color images.

The main contributions of our work are as follows

1. We propose an enhanced adaptive noise detection algorithm to overcome the problem of over/under noise detection. The proposed method confirms the noisy pixel by using systematic thresholding and similarity index formulation.
2. We propose a non-corrupted pixel sensitive adaptive image restoration to increase the accuracy of the image restoration stage. This stage excludes the noisy values from contributing to original value estimation and it ensures the maximum number of noise-free pixels in the selected window. This process uses a spatially adaptive window with maximum non-corrupted pixel ratio criteria.

In the past decade, numerous contributions were made for denoising grayscale images and the challenges were addressed from diverse and many points of view. But significantly fewer contributions were made while addressing the issue of color image denoising [26], [27]. In this article, the focus is to provide a novel approach that is highly effective for both grayscale and color images. The proposed image denoising algorithm is applied to a variety of grayscale and color image data set. Experimental results demonstrate that it achieves high denoising performance in terms of Peak Signal-to-Noise Ratio (PSNR) [11], [28]–[31], Image Enhancement Factor (IEF) [12], [14], [32] and Structural Similarity Index (SSIM) [11], [20], [33]–[36], that is superior than conventional denoising methods.

The paper is organized as follows. The required preliminaries for data set generation are presented in Section II. The proposed SAID-END algorithm is explained in Section III. In Section IV, experimental results, discussion, and comparison with existing algorithms is presented. The concluding remarks are drawn in Section V.

II. PRELIMINARIES FOR NOISY DATA SET GENERATION

Two data sets are used in the evaluation process of the proposed method first, datasets used in this paper consists of 50 grayscale images from the Brodatz texture dataset (Fig. 1) [37] and 16 color images from the University of South California miscellaneous dataset volume 3 (Fig. 2) [38]. The first data set is used to validate the performance on wide data set for both noise and artifact. Secondly, some commonly used traditional test images like Lena (grayscale and color) and Peppers (grayscale) are used for comparison of the proposed method with the recent state of the art methods. All simulations were carried out in MATLAB (MathWorks, Natick, MA, USA).

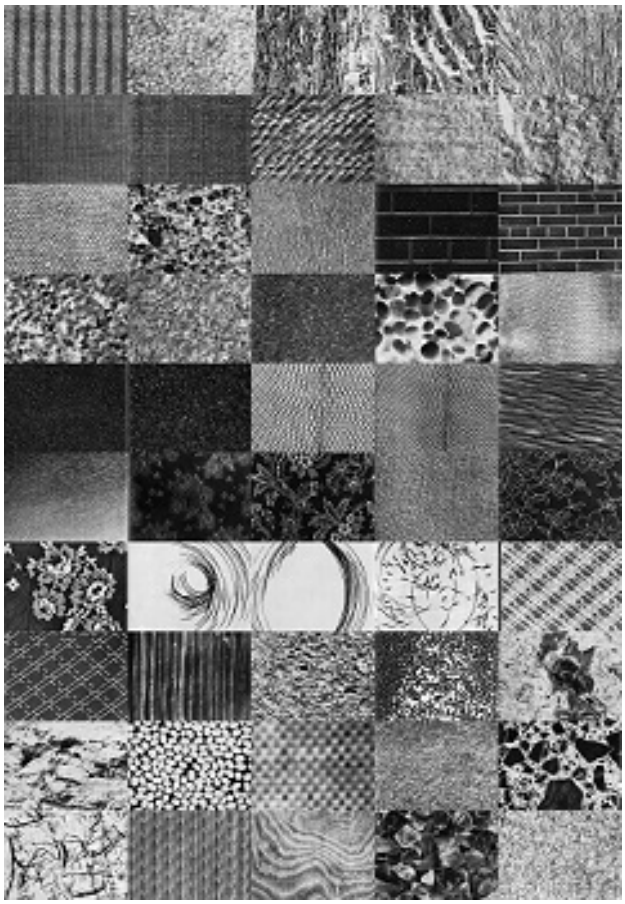

FIGURE 1. Brodatz texture dataset of grayscale images.

FIGURE 2. University of South California miscellaneous dataset of color images.

Let's discuss first data set for test one, this case is to establish the robustness of the proposed method, two types of artifacts i.e., Strip lines and Blotches [39], [40] and one types of noise i.e., Salt & Pepper [41], [42] are added in the images at varying levels (Fig. 3).

Since the work is to propose the denoising algorithm which can denoise the images affected with any noise level (low to high noise level). So, in order to check the robustness of the proposed method, the noise and artifacts are added in the image in eight steps each as shown in (Fig. 3). First Salt and Pepper noise is added to an image in eight steps from 10% to 80% (10% is considered as low level 80% is considered as high noise level). Vertical and horizontal strip lines are added in eight steps starting from 2 pixels wide strip lines to 9-pixel wide strip line with an increment of one-pixel width. Similarly, Blotches artifacts are also introduced in an image with 8 levels starting from a square of 2 * 2 to 9 * 9 with an increment of one. The noise (Salt & Pepper) and artifacts (strip lines and blotches) are added one by one and then the proposed method is applied to noisy images to achieve a noise-free image. The protocol of adding noise and artifacts in the images is as follows.

- The noise (Salt & Pepper) having 10% of corruption level is added in the image (as shown in Fig. 3).
- Then, the noise level is increased by 10% to achieve a total noise level of 20% and this increased noise level of 20% is added in the image.
- Similarly, keep on increasing the noise level by 10% until the noise level reaches up to 80%.
- So, the database of corrupted images is created by adding noise level started from 10% to 80%.
- Similarly, create the data set of corrupted images using Strip lines and Blotches.
- The process of dataset creation & addition of different noise levels from 1 to 8 in both grayscale & color images is shown in Fig. 3.

Afterward, noise affected data in both grayscale and color images were denoised using existing well-known algorithms. ROAD-TGM [6], DBMF [6], [10], (CWMF) [7], [43], (PSMF) [9], [10], (MF) [2] and proposed method. A comparison of the proposed method with the five other existing algorithms is done using box plots. Each box plot for the grayscale image dataset represents values obtained from 400 denoised images (50 images * 8 levels of noise). Similarly, the color image data set each box plot represents parameter values obtained from 128 images (16 images * 8 levels of noise). Our methodology is shown schematically in Fig. 3.

For evaluation of the proposed method with the recent state of the art algorithms some commonly used traditional images i.e., Lena (grayscale and color) and Peppers (grayscale) are corrupted with 10% to 90% impulse noise. Then the proposed method along with the recent state of the art algorithms i.e., FDS, DAMF, IIN and IMF are applied to grayscale image dataset for performance comparison. For color image denoised, a performance comparison is drawn between AUTSF, MCF and proposed method. The grayscale image and color image comparison are drawn on the basis of

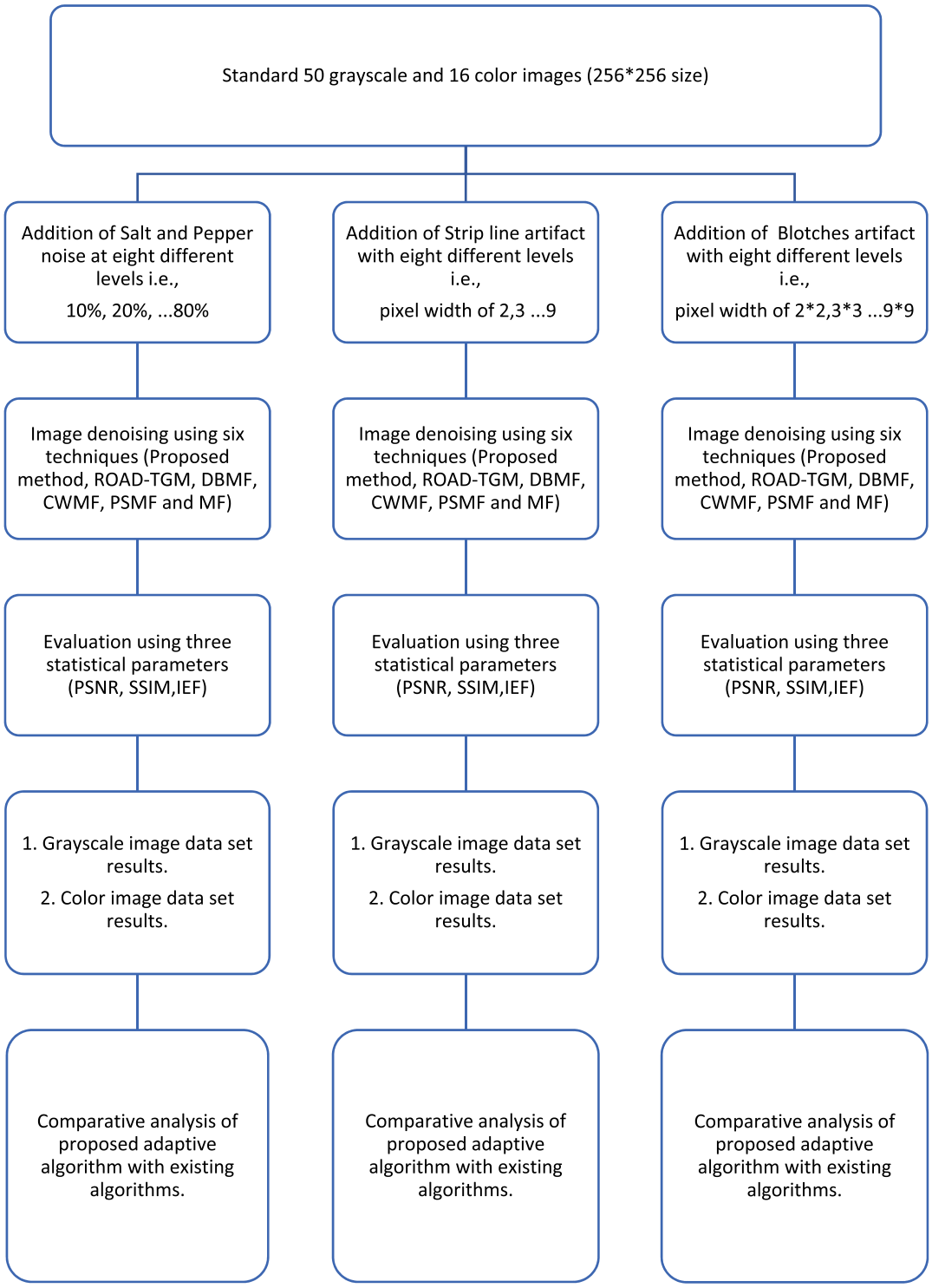

FIGURE 3. Explanation of test one Dataset, the procedure for addition of different noises, artifacts and description of protocol.

PSNR and SSIM parameters as these are commonly preferred parameters.

III. ALGORITHM OF PROPOSED DENOISING METHOD

In this paper, the SAID-END method is proposed. The proposed method works on the concept of finding noisy pixels using systematic thresholding and spatially adaptive window hence overcoming the problem of under & overdetection. Secondly, the original value of the noisy pixel is restored adaptively by adjusting the statistical parameter median. The proposed method consists of two stages.

(1) Enhanced pixel adaptive noise detection

(2) Non-corrupted pixel sensitive adaptive image restoration

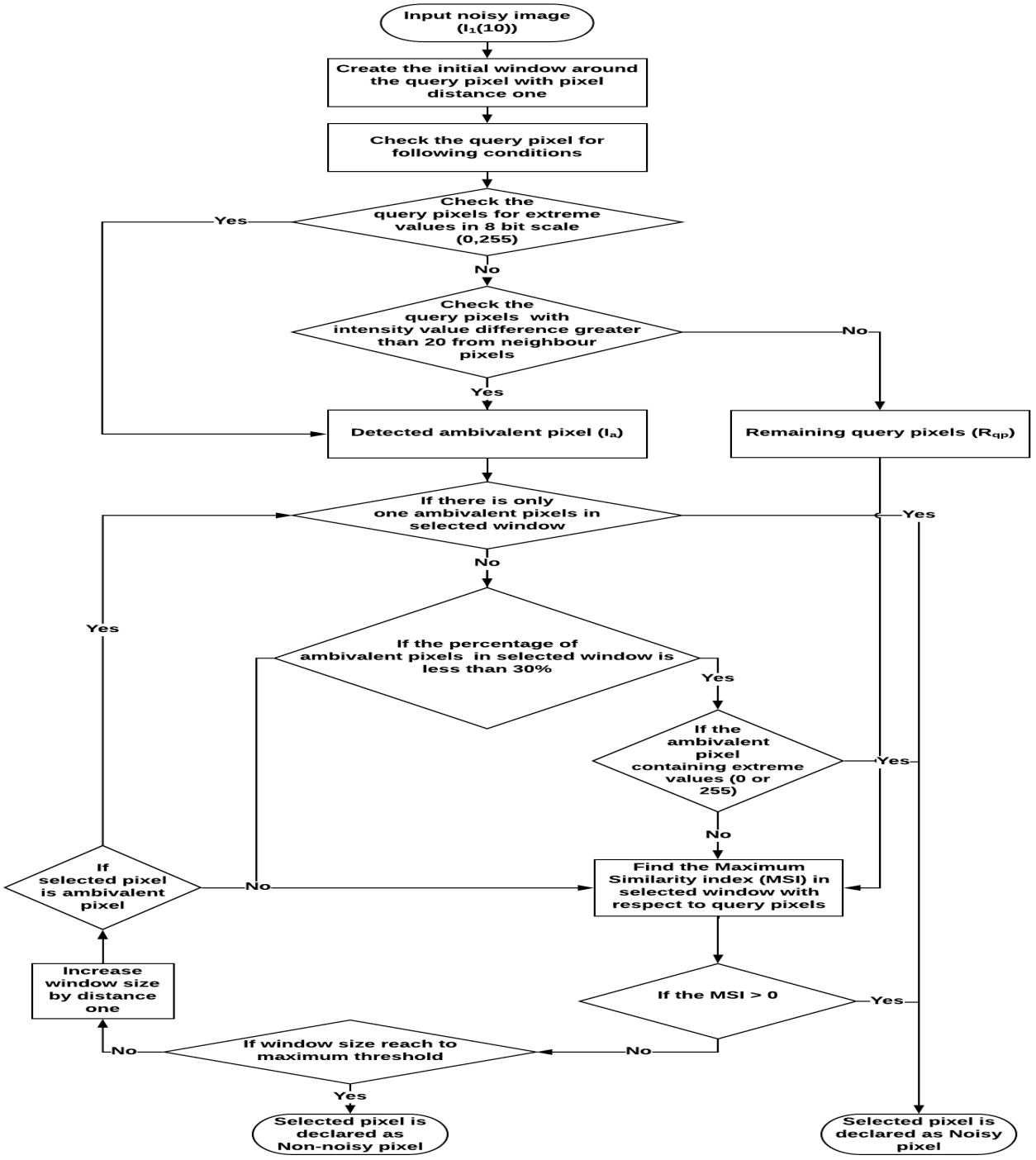

FIGURE 4. Flow chart of the detection stage of the proposed method.

Following is a detailed explanation of two stages of the proposed method.

A. ENHANCED ADAPTIVE NOISE DETECTION

The detailed flowchart of the detection stage of the proposed method is shown in Fig 4. As mentioned earlier, the noise is added in the image ranges from 1 to 8 levels. Let assume the image having Salt and Pepper noise with a noise level of 10%.

The objective of this stage is to detect noisy pixels. Let's assume $I_i\,(j)$ as a noisy dataset of grayscale images.

$$I_i(j)\ where\ \begin{matrix} i = Number\ of\ images(1\ to\ 50\ for\ grayscale\ dataset) \\ j = Noiselevel\ (10\%, 20\% \ldots 80\%) \end{matrix} \tag{1}$$

The first image ($i = 1$) is having a noise level of 10% ($j = 10$). So, the first image having a noise level of 10% is denoted as I_1 (10) (flowchart is shown in Fig. 4)

a) Consider the query pixel (first pixel) of I_1 (10) and create the window around query pixel with distance one (create a window on the immediate neighborhood).
b) Check whether the query pixel is having abrupt intensity values i.e., (i) 0 or 255 (ii) check if the intensity difference of query pixel with neighbor pixels in the window is greater than 20.
c) If any of the above condition is valid then, consider this query pixel as ambivalent pixel (doubtful to be the noisy pixel (I_a)). More confirmation is required to declare I_a as a noisy pixel.
d) Further, check if the selected window contains single or multiple ambivalent pixels.
e) If I_a is the only pixel in the window having extreme value or only pixel with intensity difference greater than 20 from surrounding pixels then it is a noisy pixel. If not then check how many pixels in the window are having extreme values and intensity difference greater than 20.
f) Then, count how many $I'_a s$ are present in the selected window. Let's assume there are 's' ambivalent pixels I_{as} ($I_{as_1}, I_{as_2}, I_{as_3} \ldots$). The formula for calculating Ambivalent Pixels Percentage (*APP*) is given below.

$$\%APP = \frac{I_{as}}{Total\ number\ of\ pixels\ in\ selected\ window} * 100 \tag{2}$$

g) The pixel I_a is considered noisy, if *APP* satisfy the following condition.

$$APP \leq D_{ref} \quad \text{Where } D_{ref} = 30\% \tag{3}$$

If *APP* does not satisfy the above criteria then check the similarity of I_a with the neighbor pixels.
h) To declare I_a as a noisy or non-noisy pixel, Calculate Maximum Similarity Index (MSI) [44] which is

$$MSI = [sum\ (MSIR)] - WP_{ref} \tag{4}$$

where MSIR is the Maximum Similarity Index Range which is calculated as.

$$MSIR = \frac{SIR_i}{S_{ref}} \tag{5}$$

where,

i = 1, 2, ... WP_{ref} (Window pixel reference)
Similarity reference (S_{ref}) = 20
WP_{ref} = 30% *of total number of pixels in the window*

Similarity Index Range (*SIR*) i.e difference of I_a with all non-extreme pixels in the window NN_{ij} ($NN_{i1}, NN_{i2}, NN_{i3}, \ldots, NN_{in}$). Where 'i' represent window number and 'j' represent the pixel number of respective window (NN_{11} indicates the first pixel of the first window). So, to create a vector *SIR* equation is given below.

$$SIR = [d_1, d_2, \ldots, d_n] \tag{6}$$

$$d_1 = |NN_{11} - I_a| \tag{7}$$

172	165	167	165	170
170	170	166	172	171
171	175	170	173	169
177	167	164	169	168
173	174	171	168	166

FIGURE 5. Original Matrix (without any noise).

Similarly, d_2 *to* d_n can be created by varying pixel represented by j in the above equation. Arrange *SIR* in ascending order.
i) Ambivalent pixel I_a can be declared as a noisy pixel. If MSI satisfies the following conditions.

$$Result = \begin{cases} noisy & if\ MSI > 0 \\ non\ noisy & otherwise \end{cases} \tag{8}$$

j) If *MSI* of pixel is lesser than zero or ratio of corrupted pixels is greater than D_{ref} then expand the window and repeat step d to step j.
k) In case window expanded to maximum window size (pixel distance thirteen) while MSI remains less than zero, then the status of I_a will be fixed as non-noisy.
l) For R_{qp} *pixels* (*remaining query pixels*) repeat steps h to j.

Let's understand the detection stage with the help of the following examples. Let's say the matrix of the original image having intensity values of pixels as shown in Fig. 5.

The original matrix (prior to addition of noise is shown in Fig. 5) is corrupted by the different types of noises and artifacts to express the cases of detection stage. The following cases are discussed.

Case 1 (a), (b): When single-pixel is corrupted in the selected window having extreme values (0 or 255).
Case 1 (c), (d): When single-pixel is corrupted in the selected window having value between 0 and 255.
Case 2: When less than 30% of pixels in a selected window are ambivalent pixels.
Case 3: When more than 30% of pixels in a selected window are ambivalent pixels.
Case 4: Edge preservation.

Firstly, let's take an example of noisy image affected by Salt and Paper noise shown in Fig. 6.

1) CASE 1 (A), (B): WHEN SINGLE-PIXEL IS CORRUPTED IN THE SELECTED WINDOW HAVING EXTREME VALUES (0 OR 255)

As shown in Fig. 6, there are two windows in each window there is only one corrupted (ambivalent) pixel i.e., pixel having value '0' in window NN_1 and another pixel having extreme value '255' in window NN_2. So as per the proposed method, this pixel will be considered as noisy (as there is only one ambivalent pixel in each of the windows) shown in Fig. 6.

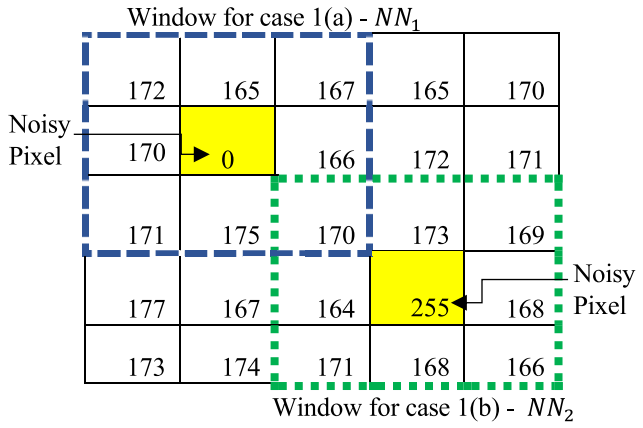

FIGURE 6. Matrix for noise detection case 1 (a) and (b).

2) CASE 1 (C), (D): WHEN SINGLE-PIXEL IS CORRUPTED IN THE SELECTED WINDOW HAVING VALUE BETWEEN 0 AND 255

Pixels of an image can take non-extreme values now let us consider the case 1 (c) (shown in Fig. 7) which consist of non-extreme value in the selected window. Pixel in window NN_3 is having value '100' which is having a difference greater than 20 (reference intensity difference threshold) from every neighbour pixel in the window. So, it will be considered as the odd man out and labeled as a noisy pixel. Case 1 (d) (shown in Fig. 7) takes care of those noises which can attain any value except from value neither extreme in nature nor having a difference greater than 20 from every neighbor pixel in the selected window. In such cases, it is difficult to decide the Remaining Query Pixels (R_{qp} as shown in Fig. 4) is a noisy pixel or original pixel. To overcome this challenge distance base similarity is calculated for the R_{qp} and similarity conditions need to be satisfied for thirty percent pixels of the total number of pixels in the considered window. Otherwise, the pixel will be considered as a noisy pixel. Maximum Similarity Index (MSI) calculations are already discussed above (refer to equation no.4 - 8) in this paper. In window NN_4 of case 1 (d) (Fig. 7) the remaining query pixel (R_{qp}) value is 145 which is not having an intensity difference of 20 from every other pixel in the window. So, it is important to check the similarity of query pixel with its non-extreme neighbors. Let us understand this example in detail by applying the equation number 4 and onwards respectively. In this case R_{qp} is having value 145 and d_1 *to* d_n are the values of its neighbor pixels. Let's start by creating the Similarity Index Range (*SIR*).

$$\begin{aligned} SIR = [&|170-145|, |173-145|, |169-145|, \\ &|164-145|, |168-145|, |171-145|, \\ &|168-145|, |166-145|] \end{aligned} \tag{9}$$

$$SIR = [25, 28, 24, 19, 23, 26, 23, 21] \tag{10}$$

Arrange the *SIR* in ascending order as follows

$$SIR = [19, 21, 23, 23, 24, 25, 26, 28] \tag{11}$$

Calculate 30% of the total number of pixels in the selected window, window NN_4 is a having dimensions 3×3 (total 9 pixels). So, 30 percent of 9 is 2.7 which is can be rounded off to 3. Now consider the first 3 distance from *SIR* vector (as 3 is the result of 30 % of 9 pixels) and then divide *SIR* values with Similarity reference (S_{ref}) which is defined as 20 as shown in the following operation.

$$MSIR = \left[19/20, 21/20, 23/20\right] \tag{12}$$

$$MSIR = [0.95, 1.05, 1.15] \tag{13}$$

$$MSI = [0.95 + 1.05 + 1.15] - 3 = 0.15 \tag{14}$$

As *MSI* is greater than zero in this example the respective R_{qp} pixel will be considered as a noisy pixel.

3) CASE 2: WHEN LESS THAN 30% OF PIXELS IN A SELECTED WINDOW ARE AMBIVALENT PIXELS

To understand this case in detail let's consider an example in Fig. 8. In this case, a window is considered with the initial distance one from ambivalent pixels and is initially having 3×3 dimensions. Consider two ambivalent pixels in both windows (NN_1& NN_2) and the rest of the pixels are non-corrupted pixels. So specifically, for this case 7 pixels in

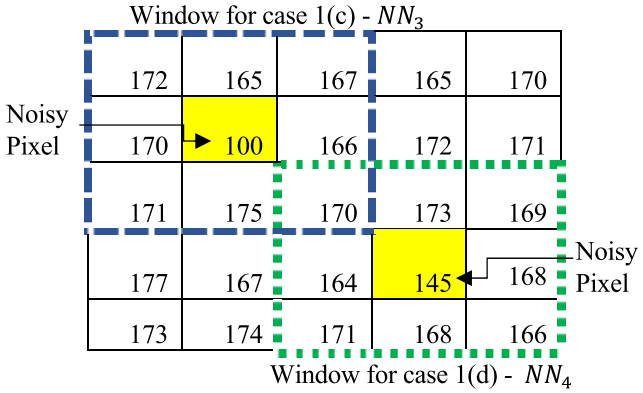

FIGURE 7. Matrix for noise detection case 1 (c) and case 1 (d).

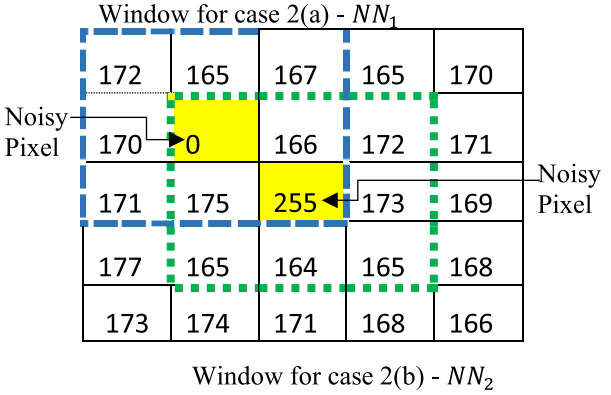

FIGURE 8. Matrix for multiple ambivalent pixels in the selected window.

each window are non-corrupted and two are query pixels out of total 9 pixels. The corrupted (ambivalent) pixels are highlighted with yellow color in both windows. The algorithm will calculate percentage of ambivalent pixels (*APP* refer to equation no.3) in the selected window. The percentage of ambivalent pixels should be less than the considered threshold value i.e., $D_{ref} = 30\%$ and $APP = 22\%((2/9)^*100 = 22\%)$. So, when the calculated percentage of ambivalent pixels is less than the considered threshold value then the ambivalent pixel will be considered as a noisy pixel.

4) CASE 3: WHEN MORE THAN 30% OF PIXELS IN A SELECTED WINDOW ARE AMBIVALENT PIXELS

In this example, three pixels are considered as ambivalent pixels out of total 9 pixels (initial window is of 3 * 3 dimensions) as shown in Fig. 9. The ambivalent pixel percentage is 33% ($(3/9)^*100 = 33\%$) which is greater than the considered threshold value ($D_{ref} = 30\%$). In such a condition when the noise level is higher than the threshold value, the algorithm will increase the window size by pixel distance one. The new window is having 5 * 5 dimensions shown in Fig. 10 and the total number of pixels are now 25 in the window. The algorithm will again calculate the Ambivalent Pixel Percentage (*APP* refer to equation no.3) for the considered case it will be 12% ($(3/25)^*100 = 12\%$) which is less than the considered threshold value ($D_{ref} = 30\%$), So query pixel will be declared as a noisy pixel.

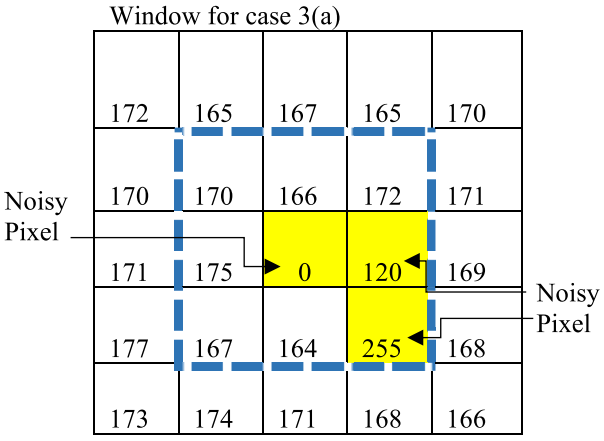

FIGURE 9. Matrix for multiple ambivalent pixels in selected window.

Due to the high amount of noise present in cases where multiple noise pixels are existing in a window. Ambivalent pixel percentage can have value higher than 30%, in such cases algorithm will keep on increasing the window size by pixel distance one and repeat the ambivalent pixel percentage calculation again until ambivalent pixel percentage becomes less than 30%. In proposed method increasing the window size is restricted to maximum pixel distance thirteen, if till this limit of pixel distance thirteen ambivalent pixel percentage remains higher than 30% then the pixel is considered as original pixel.

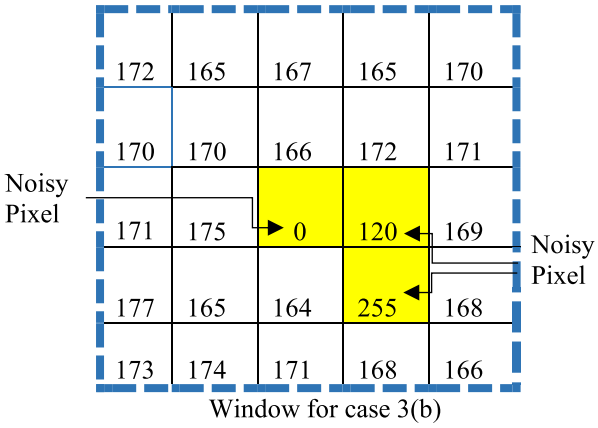

FIGURE 10. Matrix for multiple ambivalent pixels in selected window of 5 * 5 size.

5) CASE 4: EDGE PRESERVATION

To understand the edge and detail preservation process of the proposed method in natural conditions, let us consider an example of a standard Lena image. For this purpose, three edges from different locations are marked to represent different contrast situations and have been discussed case by case. These cases are as follows:

Case 4-A: Edge pixel on Hat (shown in Fig.11 (a))
Case 4-B: Edge pixel on shoulder (shown in Fig.11 (b))
Case 4-C: Edge pixel on mirror (shown in Fig.11 (c)).

The direction of the edge is marked by the yellow color in Fig.11(e-g). The proposed method preserve edge by doing classification of edges as non-corrupted pixels.

6) CASE 4-A: EDGE PIXEL ON HAT

The first example is considered from Lena hat as shown in Fig. 11(b) and its respective intensity values are presented in Fig. 11(e). Let us form a 3×3 initial window by considering the center pixel ($\boldsymbol{R_{qp}}$ *is remaining query pixel*) as a base is shown in Fig.11(e) which is under evaluation (originally an edge pixel). To verify the pixel as noisy or non-noisy for this case we need to satisfy ***MSI*** equation. Let us compute MSI in steps (as per step h and i of the proposed algorithm) as mentioned in the proposed method.

Computation of Similarity Index Range (***SIR***) from values of Fig.11(e) by using equation 6-7.

$$\text{SIR} = [|104-96|, |87-96|, |82-96|, |80-96|, |117-96|, |95-96|, |150-96|, |151-96|] \quad (15)$$

$$\text{SIR} = [8, 9, 14, 16, 21, 1, 54, 55] \quad (16)$$

Arrange the SIR in ascending order as follows

$$\text{SIR} = [1, 8, 9, 14, 16, 21, 54, 55] \quad (17)$$

To ensure the detailed preservation we are using 30% criteria of window size. The rationale of using a 30% value is that the proposed method on this data set gives the best results when 30% criteria are used. This criterion is set to 30% to make the algorithm work well in considerably high noise conditions

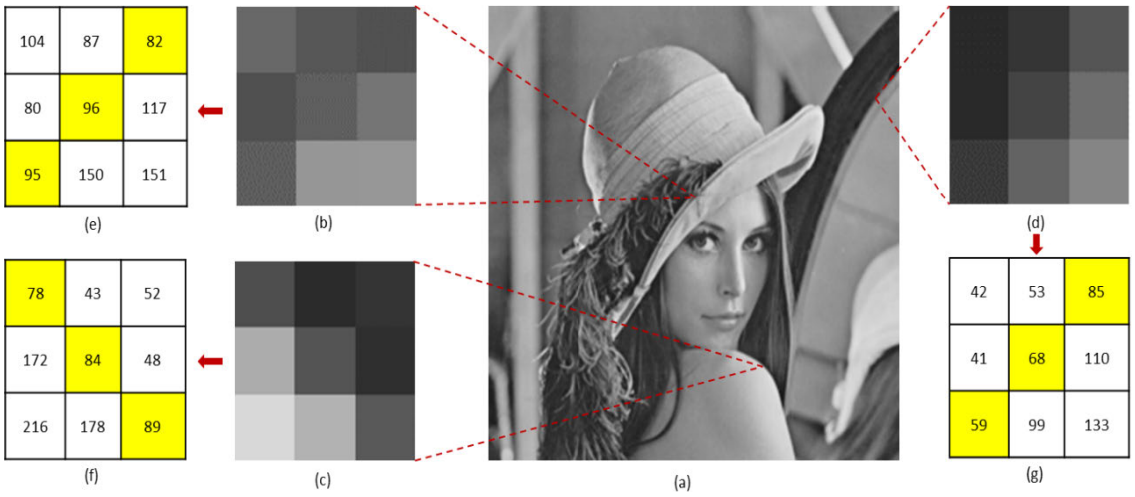

FIGURE 11. Edge preservation process (a) Original Lena image; (b) Zoomed hat edge; (c) Zoomed shoulder edge; (d) Zoomed mirror edge; (e) Intensity values of picture (b); (f) Intensity values of picture (c); (g) Intensity values of picture (d).

(as noise increases, the count of non-corrupted pixel reduces). Calculate 30% of the total number of pixels in the selected window, the window has initial dimensions of 3 × 3 (total 9 pixels). So, 30 percent of 9 is 2.7 which can be rounded off to 3.

Thirty percent criteria also ensure quality with an increase in window size (adaptive window). Now consider the first 3 distance from ***SIR*** vector (as 3 is the result of 30 % of 9 pixels) and then divide ***SIR*** values with similarity reference ($\boldsymbol{S_{ref}}$) which is defined as 20. $\boldsymbol{S_{ref}}$ is a weight that provides tolerance to the algorithm with varying noise ratio, for this paper it is set to 20 as this tolerance weight is working well on a large dataset considered in this paper. MSI is calculated as shown in the following equations by using equations 4-5.

$$\text{MSIR} = [1/20, 8/20, 9/20] \tag{18}$$
$$\text{MSIR} = [0.05, 0.4, 0.45] \tag{19}$$
$$\text{MSI} = [0.05 + 0.4 + 0.45] - 3 = -2.1 \tag{20}$$

As ***MSI*** is less than zero in this example the respective $\boldsymbol{R_{qp}}$ pixel will be considered as a non-noisy pixel. So, the edge will be preserved (original intensity value will be kept as such). In this way proposed algorithm finds the pixels on edges as non-noisy.

7) CASE 4-B: EDGE PIXEL ON SHOULDER

MSI is calculated from intensity values of Fig.11(f) on similar lines as in above-mentioned case by considering the initial window size as 3 × 3.

$$\text{SIR} = [|78 - 84|, |43 - 84|, |52 - 84|, |172 - 84|, |48 - 84|, |216 - 84|, |178-84|, |89-84|] \tag{21}$$
$$\text{SIR} = [6, 41, 32, 88, 36, 132, 94, 5] \tag{22}$$

Arrange the SIR in ascending order as follows

$$\text{SIR} = [5, 6, 32, 36, 41, 88, 94, 132] \tag{23}$$
$$\text{SIR} = [5/20, 6/20, 32/20] \tag{24}$$
$$\text{MSIR} = [0.25, 0.3, 1.6] \tag{25}$$
$$\text{MSI} = [0.25 + 0.3 + 1.6] - 3 = -0.85 \tag{26}$$

As ***MSI*** is less than zero in this example the respective $\boldsymbol{R_{qp}}$ the pixel will be considered as a non-noisy pixel. So the edge will be preserved.

8) CASE 4-C: EDGE PIXEL ON MIRROR

In this case, find out the MSI of intensity values given in Fig. 11(g) on a similar pattern as mentioned in the proposed method (step h and i).

$$SIR = [|42 - 68|, |53 - 68|, |85 - 68|, |41 - 68|, |110 - 68|, |59-68|, |99-68|, |133 - 68|] \tag{27}$$
$$SIR = [26, 15, 17, 27, 42, 9, 31, 65] \tag{28}$$

Arrange the *SIR* in ascending order as follows

$$SIR = [9, 15, 17, 26, 27, 42, 9, 31, 65] \tag{29}$$
$$MSIR = [9/20, 15/20, 17/20] \tag{30}$$
$$MSIR = [0.45, 0.75, 0.85] \tag{31}$$
$$MSI = [0.45 + 0.75 + 0.85] - 3 = -0.95 \tag{32}$$

As *MSI* is less than zero in this example the respective R_{qp} pixel will be considered as a non-noisy pixel. So the edge will be preserved (edge pixel will be kept in original form).

B. NON-CORRUPTED PIXEL SENSITIVE ADAPTIVE IMAGE RESTORATION

Once the pixel is detected as noisy pixel (as shown in Fig. 6 - 10), next stage is to restore the original values of noisy pixels. For this purpose, the restoration stage is proposed. The flow chart of the restoration process is shown in Fig. 15. In case of high noise density fixed window size is a prime reason for the loss of edge information. To overcome this issue, noise level based adaptive window is preferred to ensure high amount of non-corrupted pixels in the window [45], [46]. The noise restoration stage uses the location of noisy pixels

identified by noise detection stage. This stage ensures a minimum of 70% non-corrupted pixels (maximum non-corrupted pixel ratio criteria) are used to estimate the original value to increase accuracy. The maximum non-corrupted pixel ratio criteria is implemented by an adaptive window with the condition on noise level less than 30%. The use of local information of non-corrupted in the window can preserve edge details to a certain extent. To understand the process of noise restoration stage in detail let us consider the following cases.

1) CASE 1: WHEN A SINGLE PIXEL IS NOISY PIXEL IN SELECTED WINDOW

This case restores the original value of noise detected in case 1 (a)-(d) of the detection stage. In the considered matrix for noise restoration, noise location is already known as a result of detection stage. The algorithm will utilize the location of noisy pixel and create an initial window with distance one which results in 3 * 3 matrix. Algorithms utilizing only non-corrupted neighbor pixels to restore the value of noisy pixel (non-corrupted and corrupted pixels are already identified in the detection stage). The median of non-corrupted pixels in the selected window is taken and replaced with the value of noisy pixel (noisy pixels in the selected window are not included in the median calculation). This process of restoration stage is shown in Fig. 15. Consider case 1 (a) window NN_1 (as shown in Fig. 6) and case 1 (c) window NN_3 (as shown in Fig. 7) in these windows noisy pixel value is replaced by median value 170 (calculated by taking a median of non-corrupted pixels of the respective window). For case 1 (b) window NN_2 (as sown in Fig. 6) and case 1 (d) window NN_4 (shown in Fig. 7) noisy pixel is restored with median value 169 (as integer value is required in the image, so round off operation is applied on decimal values). Final restored matrix from noisy matrix shown in Fig. 6 and Fig. 7 is presented in Fig. 12.

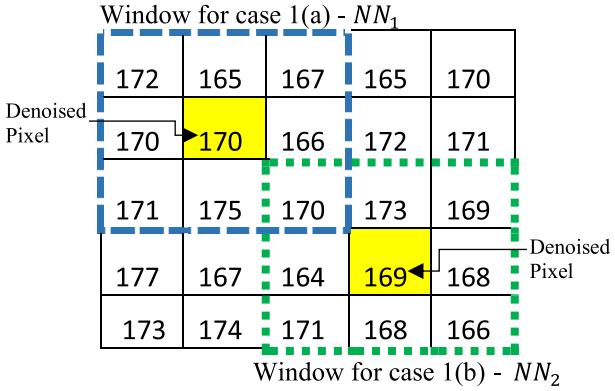

FIGURE 12. Matrix for single restored pixels in selected window.

2) CASE 2: WHEN LESS THAN 30% OF PIXELS IN A SELECTED WINDOW ARE NOISY PIXELS

Noisy pixels in a considered window NN_1 and window NN_2 of matrix shown in Fig. 8 are multiple. Let's consider case 2(a) window NN_1 (shown in Fig. 8) first, Algorithm will calculate percentage of noisy pixels in the selected window. Percentage of noisy pixels should be less than the considered threshold value of 30%, in the considered window it is 22% ((2/9)*100 = 22%). So, the median value of seven non-corrupted pixels is calculated as 170 and replaced with the noisy pixel in case 2(a) window NN_1. As now noisy pixel of case 2(a) window NN_1 is restored, now case 2(b) window NN_2 will have only one noisy value which is filtered as case 1 of the restoration stage already discussed above. So median value of eight non-corrupted pixels in the window NN_2 is replaced with noisy pixel (median value 169). Restored matrix from noisy matrix shown in Fig. 8 is presented in Fig. 13.

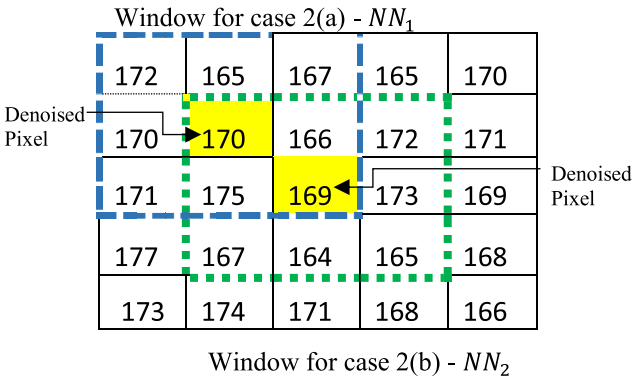

FIGURE 13. Matrix for multiple restored pixels in selected window.

3) CASE 3: WHEN MORE THAN 30% OF PIXELS IN A SELECTED WINDOW ARE NOISY PIXELS

When noisy pixels in a considered window are high (greater than 30%) shown in Fig. 9. Then window size is increased by one and noisy pixel percentage in a selected window is calculated again (Noisy Pixel Percentage (*NPP*) calculation is similar to calculation of *APP* simply replace ambivalent pixels with detected noisy pixels in the equation no.2). This process is repeated until the noisy pixel percentage in the selected window is less than D_{ref} which is 30% and then the median is calculated from non-corrupted values. For denoising of the matrix shown in Fig. 9 it is required to increase the window size by one (as noise pixel percentage is greater than 30% in case 3(a)), the new window size will be of 5 × 5 and noisy pixel percentage is 12% ((3/25)*100 = 12). As 12% is less than considered threshold value of 30%, so the median value of non-corrupted pixels from the current window is calculated as value '170' (out of 25 total values 23 are considered for median calculation) which is replaced with noisy pixel value '0'. Now for noisy pixel having value '120', again window is created with distance one (3 × 3) and now the percentage of noisy pixel is calculated as 12% ((2/9)*100 = 12%) which is less than considered threshold value 30%. Now, this pixel will be denoised as case 2 where the median is calculated by ignoring the other noisy pixels in the window. So, for this pixel median is calculated as 169

(actual value was 173 it is not possible to achieve the same value every time). For noisy pixel value 255 case 1 will be applicable as the other two pixels in the window are already restored. So the median is calculated as 169 with a similar procedure to case 1 of image restoration stage and replaced with noisy pixel (shown in Fig.14).

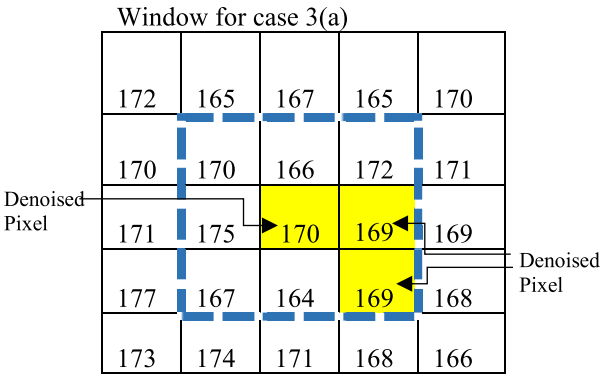

FIGURE 14. Matrix for multiple restored pixels in selected window.

IV. RESULTS AND DISCUSSION

This section evaluates the performance of the proposed method and comparison is done with existing denoising algorithms. This comparison is carried out in two test stages. Let's discuss the first stage for the comparison of the proposed method with well-known methods based on the wide dataset. As mentioned earlier the noise (salt pepper) and artifacts (Strip lines & blotches) are added in images (Grayscale and color images). For each image, different noise levels were incrementally added from noise level one to noise level eight and 2 pixels to 9 pixels range is used for strip lines and blotches artifacts, forming sequences of increasingly corrupted images.

For the visual understanding of images before and after denoising, a set of grayscale images are present in Fig 16 and color images are present in Fig 17. A comparative analysis of the proposed method with existing denoising algorithms on grayscale images corrupted by level 1 to level 8 of noise and artifacts is shown in Fig. 18–20. The PSNR, SSIM and IEF values are used to evaluate the performance of the proposed and existing algorithms on all grayscale images and color images. The PSNR, SSIM and IEF values for each image and for each noise and artifact level were calculated for the denoised image produced by both the proposed and existing algorithms. These results are shown in box plots representation for all noise levels and all images. The results prove that the proposed method is better than all other algorithms in comparison, producing better PSNR, SSIM and IEF values in all cases. Typically, the overall mean PSNR value of the proposed method is 23.95, which is higher than all other models: ROAD-TGM (PSNR = 19.82), DBMF (PSNR = 18.17), CWMF (PSNR = 16.03), PSMF (PSNR = 15.50), and MF (PSNR = 14.68). The box plots also show that the proposed method outperformed all other

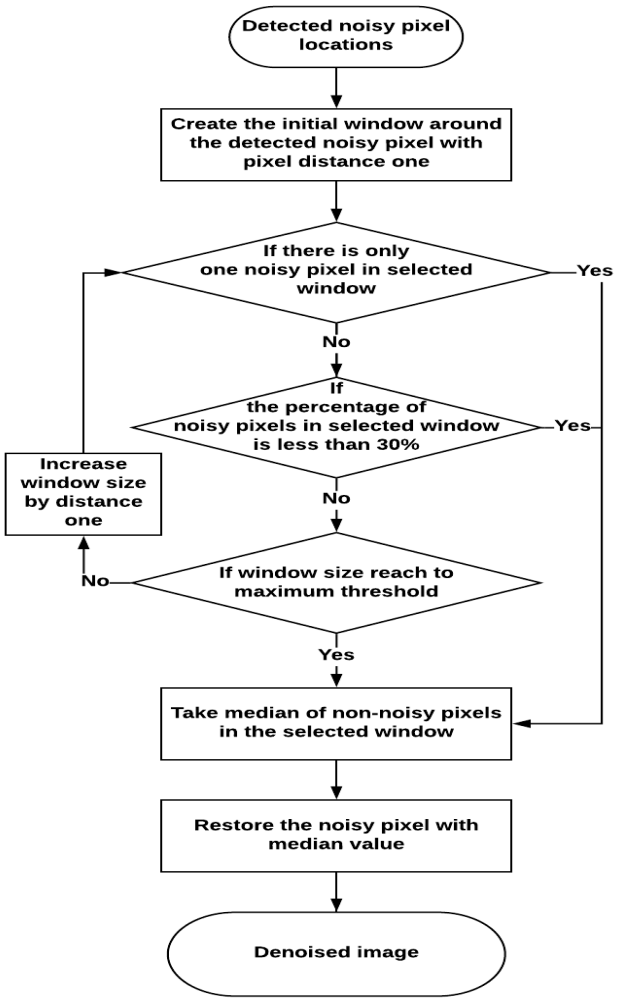

FIGURE 15. Noise restoration stage algorithm.

algorithms in terms of the SSIM parameter. The mean SSIM value of the proposed method is 0.83, which is higher than values for all other models: ROAD-TGM (SSIM = 0.63), DBMF (SSIM = 0.55), CWMF (SSIM = 0.38), PSMF (SSIM = 0.44) and MF (SSIM = 0.41). The mean IEF value of the proposed method is 47.12, which is higher than values for all other models: ROAD-TGM (IEF = 17.96), DBMF (IEF = 9.15), CWMF (IEF = 7.0), PSMF (IEF = 5.73) and MF (IEF = 4.81).

The results of various denoising techniques on images corrupted by strip lines and blotches artifacts are shown in Fig.19 and Fig 20. Again, the performance of the proposed method is better than all other algorithms, producing better PSNR, SSIM and IEF values in all cases.

The mean PSNR, mean SSIM and mean IEF values of the proposed method for strip lines artifact are: PSNR = 25.73, SSIM = 0.92, IEF = 17.71 and for blotches artifacts are: PSNR = 45.51, SSIM = 0.99, IEF = 20.80, which are higher than all other algorithms considered in the comparison. The box plots clearly demonstrate the superiority of the proposed method compared to other existing algorithms.

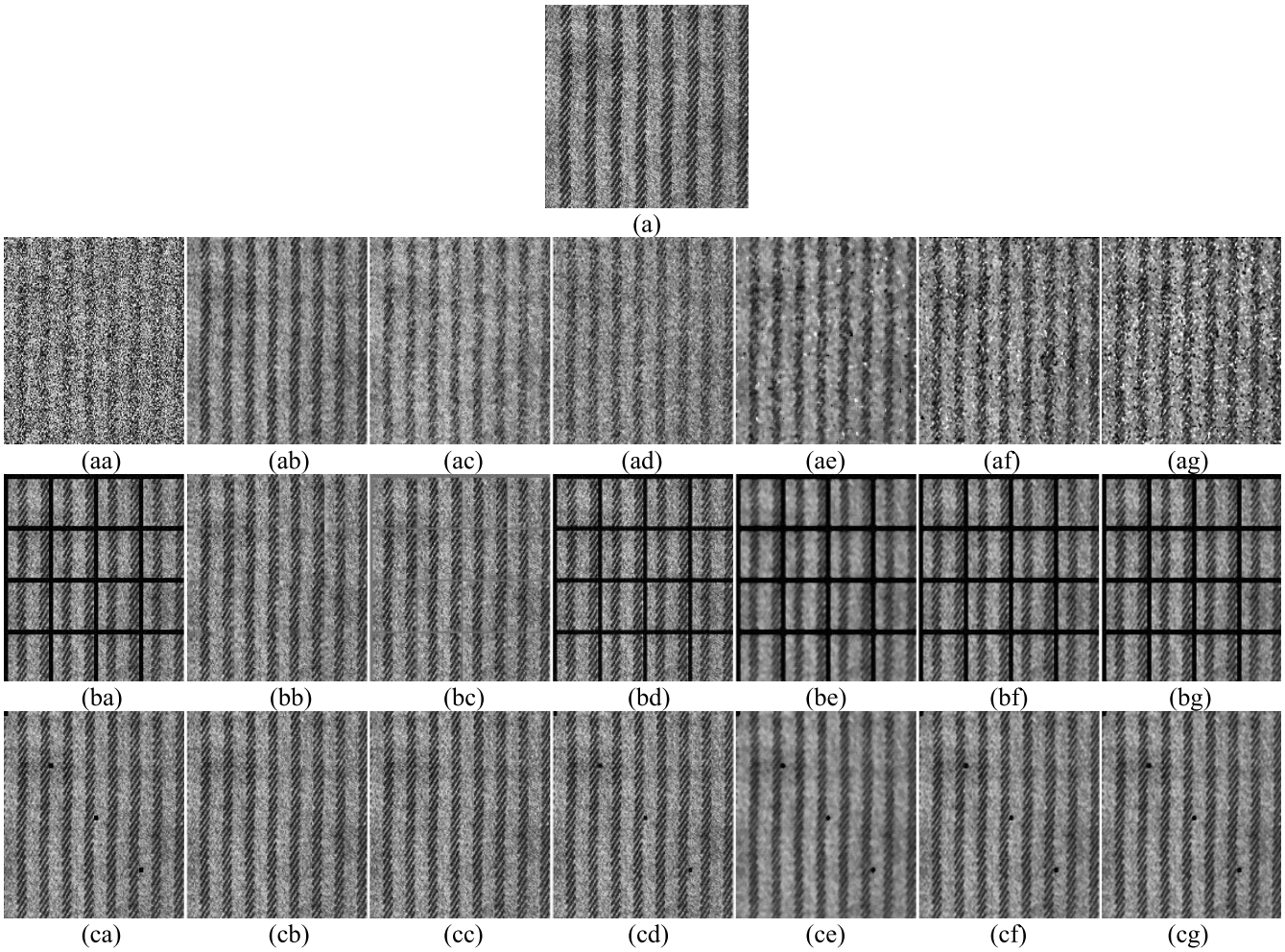

FIGURE 16. (a) Original texture image, (aa) Texture image corrupted by 50% Salt and Pepper noise, (ba) Texture image corrupted by 6 pixel wide strip line artifact, (ca) Texture image corrupted by 6 * 6 pixel blotches artifacts, (ab–cb) Image restored using proposed method from (aa–ca), (ac–cc) Image restored using the ROAD-TGM from (aa–ca), (ad–cd) Image restored using the DBMF from (aa–ca), (ae–ce) Image restored using the CWMF from (aa–ca), (af–cf) Image restored using the PSMF from (aa–ca), (ag–cg) Image restored using the MF from (aa–ca). NB: ROAD-TGM, Rank-Ordered Absolute Differences Trimmed Global Mean Filter; CWMF, Center Weighted Median Filter; DBMF, Decision-Based Median Filter; PSMF, Progressive Switching Median Filter; and MF, Median filter.

Our comparative analysis of denoising algorithms on color images affected by Salt and Pepper noise, Strip lines artifacts and Blotches artifacts are shown in Fig. 21 and Fig. 23, respectively. In the case of noise and artifacts, the proposed method produced a PSNR value of 36.37, SSIM value of 0.92 and IEF value of 72.23, which were higher than ROAD-TGM, DBMF, CWMF, PSMF, and MF algorithms. In the case of Strip lines artifacts and Blotches artifacts, the proposed method outperformed the other existing algorithms, with a PSNR value of 40.91,58.83; SSIM value of 0.96, 0.99 and IEF value of 51.45,26.12 respectively.

In the second stage of comparison, the proposed method is evaluated with the recent state of art methods. This comparison is performed for both grayscale and color image as discussed in section two. For color image denoising comparison, colored Lena image affected with (10% to 90%) impulse noise is used as a test image. The main parameter in this comparison is PSNR as this parameter is commonly used by recent methods for color image denoising, but for proposed method SSIM results are also presented along with PSNR values in Table. 1. The proposed method outperforms the recent state of the art methods by gaining superior mean PSNR value 35.35 for the noise range of 10% to 90% in comparison to denoising performance of AUTSF (32.70) and IMF (28.19). For color image denoising proposed algorithm obtained high mean SSIM value '0.93' for the image affected with low to high density of noise.

Traditional Lena and Peppers grayscale images corrupted with low to high density of impulse noise (10% to 90%) are denoised using the proposed method. This comparison with FDS, DAMF, IIN and IMF methods is presented in Table. 2, where '—' indicate unavailability of value. To achieve fair performance comparison, only

FIGURE 17. (a) Original color image, (aa) Color image corrupted by 50% Salt and Pepper noise, (ba) Color image corrupted by 6 pixel wide strip line artifact, (ca) Color image corrupted by 6 * 6 pixel size blotches artifacts,(ab–cb) Image restored using proposed method from (aa–ca), (ac–cc) Image restored using the ROAD-TGM from (aa–ca), (ad–cd) Image restored using the DBMF from (aa–ca), (ae–ce) Image restored using the CWMF from (aa–ca), (af–cf) Image restored using the PSMF from (aa–ca), (ag–cg) Image restored using the MF from (aa–ca). NB: ROAD-TGM, Rank-Ordered Absolute Differences Trimmed Global Mean Filter; CWMF, Center Weighted Median Filter; DBMF, Decision-Based Median Filter; PSMF, Progressive Switching Median Filter; and MF, Median filter.

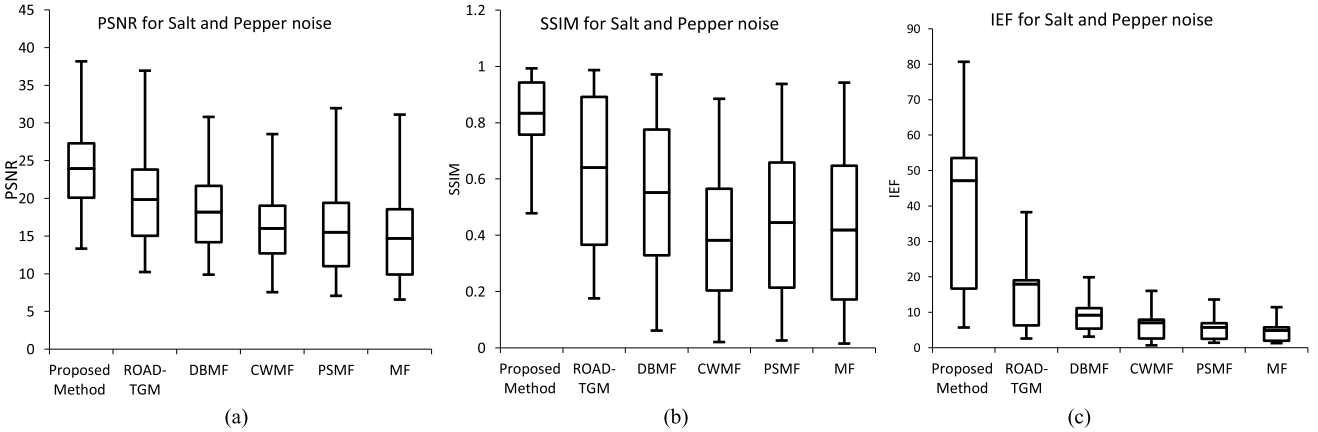

FIGURE 18. Comparative analysis of algorithms for 10% to 80% Salt and Pepper noise-affected grayscale images (a) Peak signal to noise ratio (PSNR); (b) Structural Similarity Index (SSIM); (c) Image enhancement factor (IEF). NB: ROAD-TGM, Rank-Ordered Absolute Differences Trimmed Global Mean Filter; CWMF, Center Weighted Median Filter; DBMF, Decision-Based Median Filter; PSMF, Progressive Switching Median Filter; and MF, Median filter.

common test images and statistical paraments of the recent state of the art algorithms are used. Performance evaluation of grayscale images using parameter PSNR and SSIM, shows the superiority of the proposed method among the recent state of the art algorithms (refer to Table. 2).

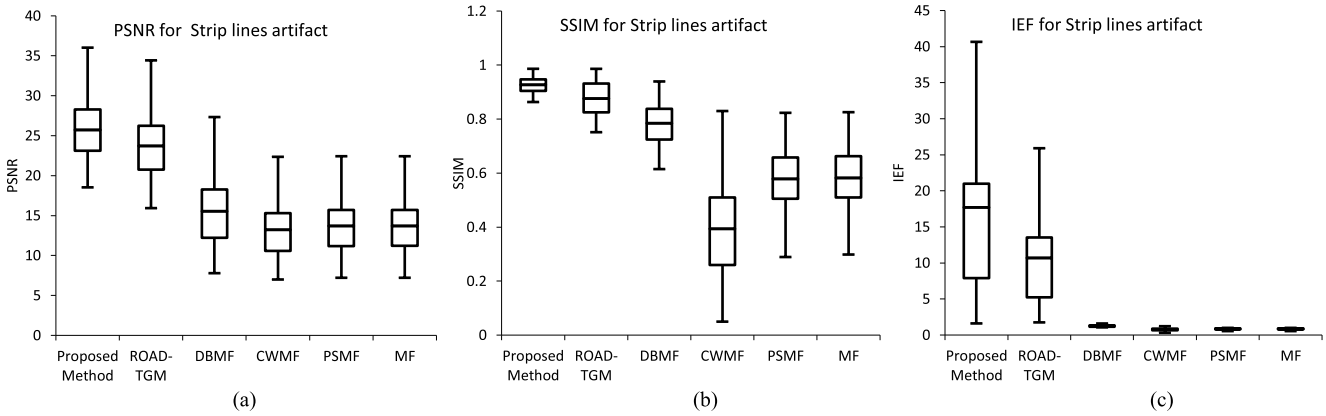

FIGURE 19. Comparative analysis of algorithms for 2 pixels to 9 pixels wide strip line-affected grayscale images (a) Peak signal to noise ratio (PSNR); (b) Structural Similarity Index (SSIM); (c) Image Enhancement Factor (IEF). NB: ROAD-TGM, Rank-Ordered Absolute Differences Trimmed Global Mean Filter; CWMF, Center Weighted Median Filter; DBMF, Decision-Based Median Filter; PSMF, Progressive Switching Median Filter; and MF, Median filter.

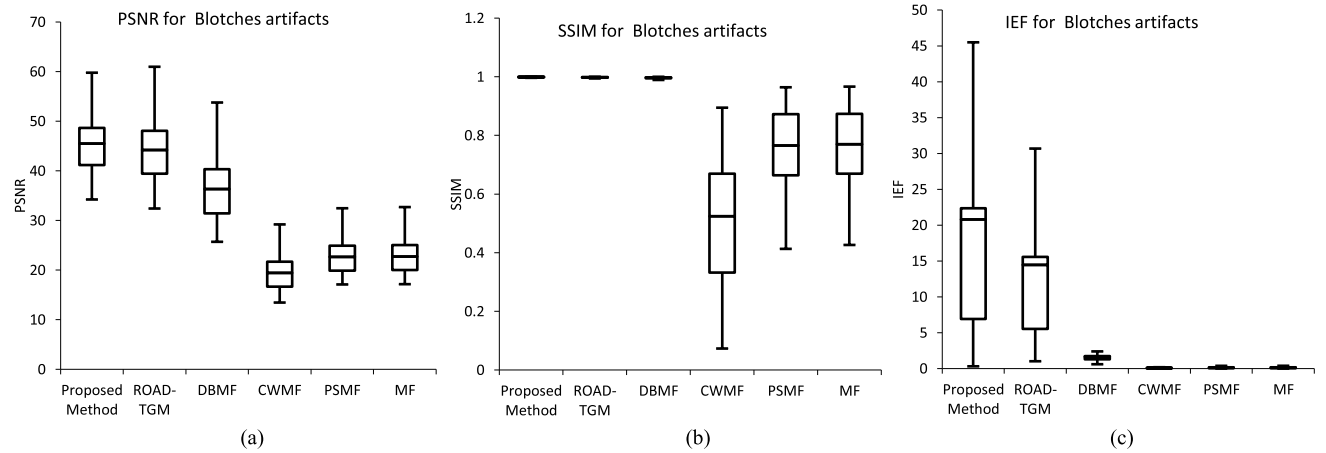

FIGURE 20. Comparative analysis of algorithms for 2 * 2 pixel to 9 * 9 pixel size blotches-affected grayscale images (a) Peak signal to noise ratio (PSNR); (b) Structural Similarity Index (SSIM); (c) Image Enhancement Factor (IEF). NB: ROAD-TGM, Rank-Ordered Absolute Differences Trimmed Global Mean Filter; CWMF, Center Weighted Median Filter; DBMF, Decision-Based Median Filter; PSMF, Progressive Switching Median Filter; and MF, Median filter.

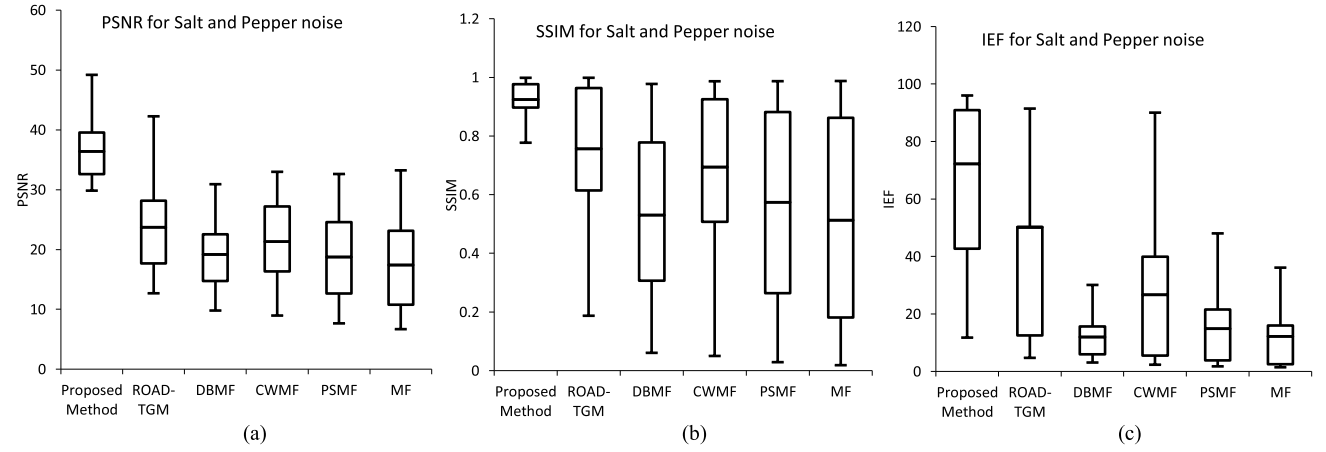

FIGURE 21. Comparative analysis of algorithms for 10% to 80% Salt and Pepper noise-affected color images (a) Peak signal to noise ratio (PSNR); (b) Structural Similarity Index (SSIM); (c) Image enhancement factor (IEF). NB: ROAD-TGM, Rank-Ordered Absolute Differences Trimmed Global Mean Filter; CWMF, Center Weighted Median Filter; DBMF, Decision-Based Median Filter; PSMF, Progressive Switching Median Filter; and MF, Median filter.

The proposed method achieves mean PSNR value of 36.36 in comparison to lower values of the recent state of the art methods (FDS = 29.33, DAMF = 33.73, IIN = 28.74, IMF = 34.45) for grayscale Lena image. Again, on Lena grayscale test image the proposed method achieves better performance for SSIM parameter by obtaining the

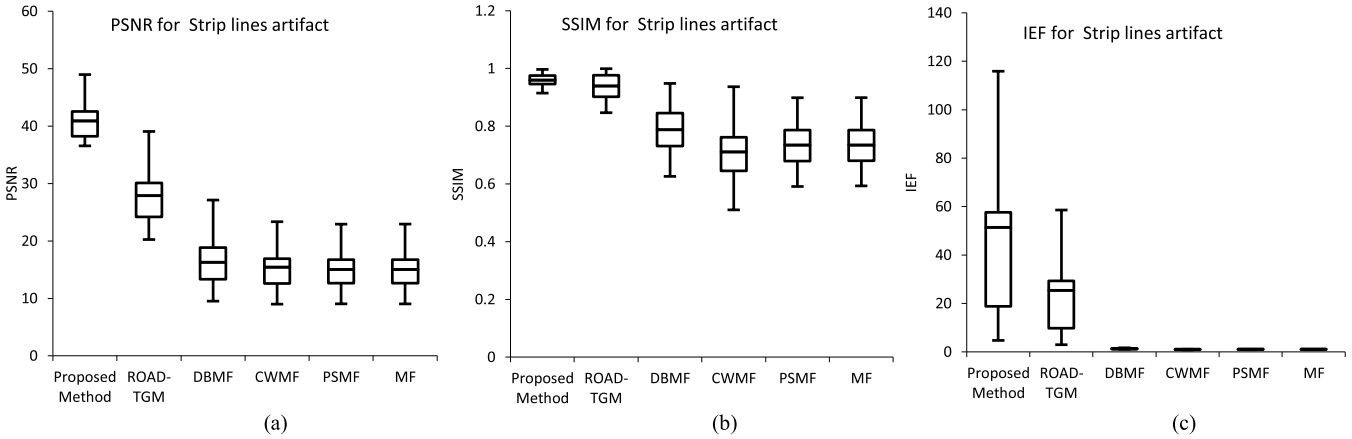

FIGURE 22. Comparative analysis of algorithms for 2 pixels to 9 pixels wide strip line-affected color images (a) Peak signal to noise ratio (PSNR); (b) Structural Similarity Index (SSIM); (c) Image Enhancement Factor (IEF). NB: ROAD-TGM, Rank-Ordered Absolute Differences Trimmed Global Mean Filter; CWMF, Center Weighted Median Filter; DBMF, Decision-Based Median Filter; PSMF, Progressive Switching Median Filter; and MF, Median filter.

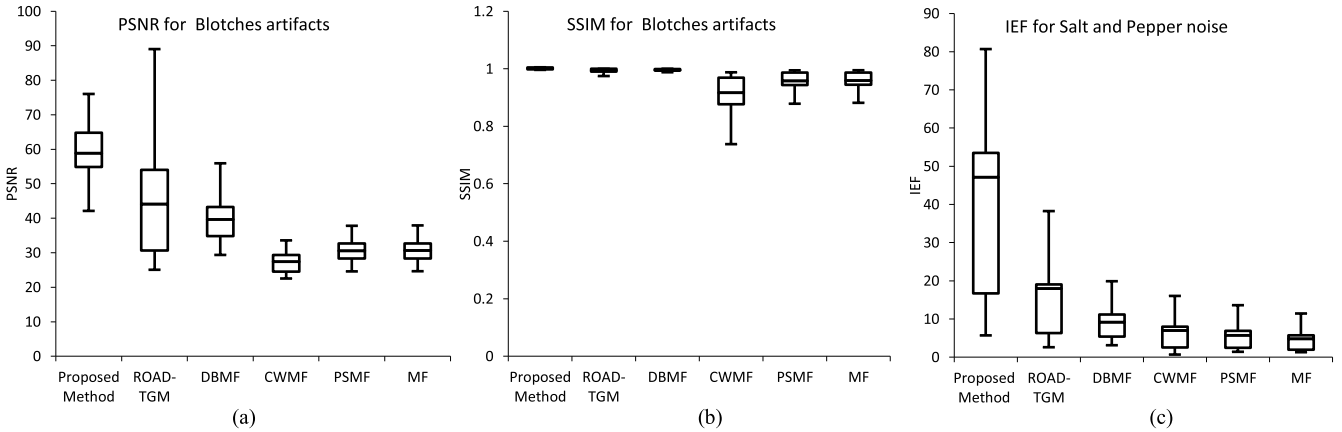

FIGURE 23. Comparative analysis of algorithms for 2 * 2 pixel to 9 * 9 pixel size blotches-affected color images (a) Peak signal to noise ratio (PSNR); (b) Structural Similarity Index (SSIM); (c) Image Enhancement Factor (IEF). NB: ROAD-TGM, Rank-Ordered Absolute Differences Trimmed Global Mean Filter; CWMF, Center Weighted Median Filter; DBMF, Decision-Based Median Filter; PSMF, Progressive Switching Median Filter; and MF, Median filter.

TABLE 1. Color image denoising comparison of the proposed method with the recent state of the art methods.

Test Image	Noise level		10%	20%	30%	40%	50%	60%	70%	80%	90%	Mean
Lena	AUTSF [13]	PSNR	41.51	38.12	35.91	34.20	32.60	31.05	29.22	27.38	24.29	32.70
		SSIM	—	—	—	—	—	—	—	—	—	—
	MCF [14]	PSNR	—	—	38.95	36.55	34.20	31.42	26.41	21.34	8.46	28.19
		SSIM	—	—	—	—	—	—	—	—	—	—
	Proposed Method	**PSNR**	**44.1988**	**41.34**	**38.99**	**36.66**	**34.32**	**32.93**	**31.96**	**30.11**	**27.82**	**35.35**
		SSIM	**0.9855**	**0.9740**	**0.9663**	**0.9544**	**0.9399**	**0.9295**	**0.9080**	**0.8721**	**0.8346**	**0.93**

*For Table 1-2 The values in bold represent better PSNR and SSIM as compared to several state-of-the-art algorithms. ** For Table 1-2 '—' indicates value not available for denoising method.

mean SSIM value 0.94 in comparison to the recent state of art algorithms (FDS = 0.83, DAMF = 0.91, IIN = N.A, IMF = 0.92). Similarly, the proposed method performs well on the second traditional test image (Peppers). The proposed method repeats its success over the recent state of art methods for both the parameters PSNR and SSIM by achieving higher values. The proposed method achieves mean parameter (PSNR/SSIM) values (35.71/0.92) followed

TABLE 2. Grayscale image denoising comparison of the proposed method with recent state of the art methods.

Test Image	Noise level		10%	20%	30%	40%	50%	60%	70%	80%	90%	Mean
Lena	FDS [15]	PSNR	41.40	37.25	34.49	31.67	28.99	26.54	23.95	21.39	18.30	29.33
		SSIM	0.9894	0.9759	0.9573	0.9293	0.8858	0.8280	0.7441	0.6379	0.5020	0.83
	DAMF [11]	PSNR	42.97	39.29	36.84	34.94	33.21	31.64	30.22	28.53	25.93	33.73
		SSIM	0.9902	0.9788	0.9655	0.9494	0.9304	0.9064	0.8770	0.8370	0.7620	0.91
	IIN [16]	PSNR	—	31.43	29.50	27.62	26.39	—	—	—	—	28.74
		SSIM	—	—	—	—	—	—	—	—	—	—
	IMF [12]	PSNR	43.48	40.18	37.05	35.40	33.98	32.49	31.23	29.70	27.42	34.55
		SSIM	0.9913	0.9796	0.9675	0.9541	0.9383	0.9183	0.8953	0.8623	0.8058	0.92
	Proposed Method	**PSNR**	**45.21**	**42.35**	**39.80**	**37.67**	**35.33**	**33.94**	**32.97**	**31.12**	**28.83**	**36.36**
		SSIM	**0.9957**	**0.9842**	**0.9765**	**0.9646**	**0.9501**	**0.9397**	**0.9182**	**0.8823**	**0.8448**	**0.94**
Peppers	FDS [15]	PSNR	40.6500	36.9000	34.3200	31.7200	29.3200	26.8300	24.1100	21.3700	18.1500	29.26
		SSIM	0.9825	0.9627	0.9396	0.9079	0.8687	0.8110	0.7355	0.6360	0.5085	0.82
	DAMF [11]	PSNR	41.5200	37.8900	35.6700	33.9500	32.5500	31.3100	29.7900	28.2800	25.8700	32.98
		SSIM	0.9815	0.9606	0.9389	0.9131	0.8866	0.8541	0.8180	0.7719	0.7049	0.87
	IIN [16]	PSNR	—	27.23	26.21	24.82	23.98	—	—	—	—	25.56
		SSIM	—	—	—	—	—	—	—	—	—	—
	IMF [12]	PSNR	41.83	38.59	36.65	35.14	33.9	32.69	31.43	30.01	27.88	34.24
		SSIM	0.9858	0.9684	0.9504	0.9299	0.9091	0.8846	0.8572	0.8219	0.7700	0.90
	Proposed Method	**PSNR**	**43.41**	**40.13**	**38.13**	**36.76**	**35.08**	**33.84**	**32.91**	**31.18**	**29.83**	**35.70**
		SSIM	**0.9922**	**0.9863**	**0.9674**	**0.9433**	**0.9256**	**0.9009**	**0.8799**	**0.8521**	**0.8187**	**0.92**

by FDS (29.26/082), DAMF (32.98/087), IIN (25.56/N.A) and IMF (34.24/0.90).

V. CONCLUSION

In order to overcome the performance issues of the existing denoising methods, a two-stage SAID-END denoising algorithm has been proposed. At first, the proposed method overcomes the issues of over/under detection of noisy pixels by using enhanced adaptive noise detection stage. This stage uses systematic thresholding with iterative similarity indexing to ensure the accurate categorization of noisy and non-noisy pixels. After the classification of pixels, the task was to reduce the impact of high-density noise on original value estimation which was achieved by using a non-corrupted pixel sensitive adaptive image restoration stage. This stage ensures the computation of restored value would be carried out only when a good amount of non-corrupted pixels are available in the window. This process has been implemented using an adaptive window mechanism with non-corrupted pixel ratio criteria. Once the non-corrupted pixel ratio criteria is satisfied, the original value of noisy pixel was restored using statistical measure i.e., median of non-corrupted pixel values. The two-stage test has been carried out on the proposed method to evaluate its performance. The first test was carried out to validate the operativity of the proposed method on a wide range of noise and artifacts affected dataset. The proposed method has shown better PSNR, SSIM and IEF performance when compared with some well-known algorithms for a wide dataset of color and grayscale images. The second test stage was performed to evaluate the proposed method in comparison to the recent state of art algorithms. The commonly referred traditional test images have been used to perform this comparison. The proposed algorithm has shown improved performance on the basis of PSNR and SSIM parameters. In the future, this work can be extended by increasing the proximity of restored value to the original value to achieve higher detail preservation.

REFERENCES

[1] T. Veerakumar, B. N. Subudhi, S. Esakkirajan, and P. K. Pradhan, "Context model based edge preservation filter for impulse noise removal," *Expert Syst. Appl.*, vol. 88, pp. 29–44, Dec. 2017.
[2] T. Nodes and N. Gallagher, "Median filters: Some modifications and their properties," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 30, no. 5, pp. 739–746, Oct. 1982.
[3] A. Noor *et al.*, "Median filters combined with denoising convolutional neural network for Gaussian and impulse noises," *Multimedia Tools Appl.*, 2020, doi: 10.1007/s11042-020-08657-4.
[4] J. Lu, K. B. Letaief, J. C.-I. Chuang, and M. L. Liou, "M-PSK and M-QAM BER computation using signal-space concepts," *IEEE Trans. Commun.*, vol. 47, no. 2, pp. 181–184, Feb. 1999.
[5] 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.
[6] G. S. Kalra and S. Singh, "Efficient digital image denoising for gray scale images," *Multimedia Tools Appl.*, vol. 75, no. 8, pp. 4467–4484, Apr. 2016.
[7] T. Chen and H. Ren Wu, "Application of partition-based median type filters for suppressing noise in images," *IEEE Trans. Image Process.*, vol. 10, no. 6, pp. 829–836, Jun. 2001.
[8] S.-J. Ko and Y. H. Lee, "Center weighted median filters and their applications to image enhancement," *IEEE Trans. Circuits Syst.*, vol. 38, no. 9, pp. 984–993, Sep. 1991.

[9] Z. Wang and D. Zhang, "Progressive switching median filter for the removal of impulse noise from highly corrupted images," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 46, no. 1, pp. 78–80, Jan. 1999.

[10] S. Esakkirajan, T. Veerakumar, A. N. Subramanyam, and C. H. PremChand, "Removal of high density salt and pepper noise through modified decision based unsymmetric trimmed median filter," *IEEE Signal Process. Lett.*, vol. 18, no. 5, pp. 287–290, May 2011.

[11] U. Erkan, L. Gökrem, and S. Engino lu, "Different applied median filter in salt and pepper noise," *Comput. Electr. Eng.*, vol. 70, pp. 789–798, Aug. 2018.

[12] U. Erkan, D. N. H. Thanh, L. M. Hieu, and S. Enginoglu, "An iterative mean filter for image denoising," *IEEE Access*, vol. 7, pp. 167847–167859, 2019.

[13] T. Veerakumar, B. N. Subudhi, S. Esakkirajan, and P. K. Pradhan, "Iterative adaptive unsymmetric trimmed shock filter for high-density Salt-and-Pepper noise removal," *Circuits, Syst., Signal Process.*, vol. 38, no. 6, pp. 2630–2652, Jun. 2019.

[14] B. Karthik *et al.*, "Removal of high density salt and pepper noise in color image through modified cascaded filter," *J. Ambient Intell. Hum. Comput.*, 2020, doi: 10.1007/s12652-020-01737-1.

[15] V. Singh, R. Dev, N. K. Dhar, P. Agrawal, and N. K. Verma, "Adaptive Type-2 fuzzy approach for filtering salt and pepper noise in grayscale images," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 5, pp. 3170–3176, Oct. 2018.

[16] M. Zhang, Y. Liu, G. Li, B. Qin, and Q. Liu, "Iterative scheme-inspired network for impulse noise removal," *Pattern Anal. Appl.*, vol. 23, no. 1, pp. 135–145, Feb. 2020.

[17] A. Buades, B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Model. Simul.*, vol. 4, no. 2, pp. 490–530, Jan. 2005.

[18] C. Kervrann and J. Boulanger, "Optimal spatial adaptation for patch-based image denoising," *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 2866–2878, Oct. 2006.

[19] P. Chatterjee and P. Milanfar, "Is denoising dead?" *IEEE Trans. Image Process.*, vol. 19, no. 4, pp. 895–911, Apr. 2010.

[20] Z. Sun, B. Han, J. Li, J. Zhang, and X. Gao, "Weighted guided image filtering with steering kernel," *IEEE Trans. Image Process.*, vol. 29, pp. 500–508, 2020.

[21] A. Rahiman V and S. N. George, "Robust single image super resolution using neighbor embedding and fusion in wavelet domain," *Comput. Electr. Eng.*, vol. 70, pp. 674–689, Aug. 2018.

[22] A. Buades, B. Coll, and J. M. Morel, "Image denoising methods. A new nonlocal principle," *SIAM Rev.*, vol. 52, no. 1, pp. 113–147, Jan. 2010.

[23] R. Dosselmann and X. D. Yang, "A comprehensive assessment of the structural similarity index," *Signal, Image Video Process.*, vol. 5, no. 1, pp. 81–91, Mar. 2011.

[24] B. Zhang and J. P. Allebach, "Adaptive bilateral filter for sharpness enhancement and noise removal," *IEEE Trans. Image Process.*, vol. 17, no. 5, pp. 664–678, May 2008.

[25] H. Talebi and P. Milanfar, "Global image denoising," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 755–768, Feb. 2014.

[26] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. Image Process.*, vol. 17, no. 1, pp. 53–69, Jan. 2008.

[27] N. Joshi, C. L. Zitnick, R. Szeliski, and D. J. Kriegman, "Image deblurring and denoising using color priors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1550–1557.

[28] H. R. Sheikh and A. C. Bovik, "Image information and visual quality," *IEEE Trans. Image Process.*, vol. 15, no. 2, pp. 430–444, Feb. 2006.

[29] Y. A. Vlasov and S. J. McNab, "Coupling into the slow light mode in slab-type photonic crystal waveguides," *Opt. Lett.*, vol. 31, no. 1, p. 50, Jan. 2006.

[30] X. Wang, S. Shen, G. Shi, Y. Xu, and P. Zhang, "Iterative non-local means filter for salt and pepper noise removal," *J. Vis. Commun. Image Represent.*, vol. 38, pp. 440–450, Jul. 2016.

[31] F. Ahmed and S. Das, "Removal of high-density Salt-and-Pepper noise in images with an iterative adaptive fuzzy filter using alpha-trimmed mean," *IEEE Trans. Fuzzy Syst.*, vol. 22, no. 5, pp. 1352–1358, Oct. 2014.

[32] V. Gupta, D. K. Gandhi, and P. Yadav, "Removal of fixed value impulse noise using improved mean filter for image enhancement," in *Proc. Nirma Univ. Int. Conf. Eng. (NUiCONE)*, Nov. 2013, pp. 1–5.

[33] A. Hore and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *Proc. 20th Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 2366–2369.

[34] D. Brunet, E. R. Vrscay, and Z. Wang, "On the mathematical properties of the structural similarity index," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1488–1495, Apr. 2012.

[35] T.-L. Lin, N.-C. Yang, R.-H. Syu, C.-C. Liao, W.-L. Tsai, C.-C. Chou, and S.-L. Chen, "NR-bitstream video quality metrics for SSIM using encoding decisions in AVC and HEVC coded videos," *J. Vis. Commun. Image Represent.*, vol. 32, pp. 257–271, Oct. 2015.

[36] A. Shah, J. I. Bangash, A. W. Khan, I. Ahmed, A. Khan, A. Khan, and A. Khan, "Comparative analysis of median filter and its variants for removal of impulse noise from gray scale images," *J. King Saud Univ.–Comput. Inf. Sci.*, to be published, doi: 10.1016/j.jksuci.2020.03.007.

[37] T. Randen. (2007). *Brodatz Textures*. Accessed: Jun. 16, 2019. [Online]. Available: http://Www.Ux.Uis.No/~Tranden/Brodatz.Html and http://www.ux.uis.no/~tranden/brodatz.html

[38] *Volume 3: Miscellaneous—USC Viterbi | Ming Hsieh Department of Electrical Engineering*. Accessed: Jun. 15, 2019. [Online]. Available: https://minghsiehee.usc.edu/volume-3-miscellaneous/

[39] S. Manikandan and D. Ebenezer, "A nonlinear decision-based algorithm for removal of strip lines, drop lines, blotches, band missing and impulses in images and videos," *EURASIP J. Image Video Process.*, vol. 2008, pp. 1–10, Dec. 2008.

[40] H. Yous, A. Serir, and S. Yous, "CNN-based method for blotches and scratches detection in archived videos," *J. Vis. Commun. Image Represent.*, vol. 59, pp. 486–500, Feb. 2019.

[41] W. Li, Y. Sun, and S. Chen, "A new algorithm for removal of high-density salt and pepper noises," in *Proc. 2nd Int. Congr. Image Signal Process.*, Oct. 2009, pp. 1–4.

[42] L. Xu, W. Jing, H. Song, and G. Chen, "High-resolution remote sensing image change detection combined with pixel-level and object-level," *IEEE Access*, vol. 7, pp. 78909–78918, 2019.

[43] R. H. Chan, C. Hu, and M. Nikolova, "An iterative procedure for removing random-valued impulse noise," *IEEE Signal Process. Lett.*, vol. 11, no. 12, pp. 921–924, Dec. 2004.

[44] C.-T. Lu, Y.-Y. Chen, L.-L. Wang, and C.-F. Chang, "Removal of salt-and-pepper noise in corrupted image using three-values-weighted approach with variable-size window," *Pattern Recognit. Lett.*, vol. 80, pp. 188–199, Sep. 2016.

[45] Z. Huang, Y. Zhang, Q. Li, T. Zhang, and N. Sang, "Spatially adaptive denoising for X-ray cardiovascular angiogram images," *Biomed. Signal Process. Control*, vol. 40, pp. 131–139, Feb. 2018.

[46] L. Fan, X. Li, H. Fan, and C. Zhang, "An adaptive boosting procedure for low-rank based image denoising," *Signal Process.*, vol. 164, pp. 110–124, Nov. 2019.

AMANDEEP SINGH (Member, IEEE) received the B.Tech. degree in electronics and communication engineering from Punjab Technical University, Punjab, India, in 2010, and the M.Tech. degree in electronics and communication engineering from Lovely Professional University, Punjab, in 2012, where he is currently pursuing the Ph.D. degree. He has authored and coauthored more than 20 research articles in reputed conferences and journals. His current research interests include image analysis, machine learning, image reconstruction, biomedical image processing, and computer vision.

GAURAV SETHI received the Ph.D. degree in electronics and communication engineering from the Dr. B. R. Ambedkar National Institute of Technology, Jalandhar, Punjab, India, in the area of biomedical image processing.

He is currently working as a Professor with the School of Electronics and Electrical Engineering, Lovely Professional University, Phagwara, Punjab. He has published many research papers and book chapters in reputed International Journals, books, and Conferences. He has been guiding five Ph.D. Scholars, M.Tech. Students, and B.Tech. Projects in the area of artificial intelligence, the IoT, medical image, and signal processing. He has 15 years of rich experience of teaching and research and technical education management. His research interests include digital signal processing, medical imaging, artificial intelligence, image processing, and affordable healthcare services. He is a member of Indian Science Congress, Kolkata. He is also a Reviewer of many SCI and Scopus indexed Journals.

G. S. KALRA (Member, IEEE) received the M.Tech. and Ph.D. degrees in electronics and communication engineering from Punjab Technical University, in 2010 and 2016, respectively. He has been with the Department of Electronics and Communication Engineering, CT Group of Institutions, Jalandhar, Punjab, India, where he is currently a Professor and the C. T. Group Director. His current research interests include image denoising, image segmentation, image enhancement, computer vision and their applications in image processing, and computer vision.