SERBIAN JOURNAL OF ELECTRICAL ENGINEERING
Vol. 23, No. 3, October 2026, 377-399
UDC: 004.932.4:621.391.822 **DOI: https://doi.org/10.2298/SJEE2603377P**
Original scientific paper

Dual-stage Enhanced Decision based Unsymmetric Trimmed Mean-Median Filter for High Density Salt and Pepper Noise Removal

Aksha Perween[1], Anshu Kumari[1], Abhishek Kumar[2], Sanjeev Kumar[2]

Abstract: This paper presents a high-performance enhanced decision based unsymmetric trimmed mean-median filter (EDBUTMMF) to detect and remove high density salt and pepper noise from an image. Unlike conventional decision-based filters that directly classify all extreme-valued pixels as noisy, the proposed method first verifies whether an extreme-valued pixel is genuinely corrupted before restoration. The process of denoising is implemented in two stages: noise detection followed by noise removal. The proposed filter initially detects noisy pixels iteratively through several phases and replaces the noisy pixels with the trimmed method. Unsymmetric trimmed filtering omits impulse noise values before performing the median calculation, ensuring that the replacement pixel is derived exclusively from the uncorrupted neighbourhood samples. This yields improved edge preservation and restoration accuracy, especially when dealing with high noise densities images. Dual-stage method separates noise detection and restoration to avoid unnecessary modification of uncorrupted pixels, especially at high density noise. Performance evaluation shows both quantitative parameters MSE, PSNR and SSIM achieves better results, while providing improved visual quality and lower computational time compared with several existing decision-based filtering methods.

Keywords: Salt and pepper noise, Trimmed median filter, Decision based algorithm, Adaptive median filter.

[1] Sarala Birla University, Department of Computer Science & Engineering, Ranchi, Jharkhand, India
aksha.perween78@gmail.com, https://orcid.org/0009-0009-0126-6998
anshu.kumari7424@gmail.com, https://orcid.org/0009-0006-4709-0893
[2] Sarala Birla University, Department of Electronics & Communication Engineering, Ranchi, Jharkhand, India
a.kumar1049@gmail.com, https://orcid.org/0009-0009-2033-452X
sanjeevsingh.ece@gmail.com, https://orcid.org/0000-0002-4662-288X

Colour versions of the one or more of the figures in this paper are available online at https://sjee.ftn.kg.ac.rs

© *Creative Common License CC BY-NC-ND*

1 Introduction

Impulse noise in images is caused by bit errors in transmission or introduced during the signal acquisition stage [1 – 2]. Impulse noise is generally classified into salt and pepper noise (SPN) and random-valued impulse noise (RVIN). Among these, SPN is more frequently encountered in digital imaging systems and remains challenging to remove at high noise densities while preserving image details. Denoising an image is a fundamental and critical operation in image processing, as various fields, including edge detection, object segmentation, feature extraction, and medical imaging, necessitate high-quality, noise-free images [3 – 4]. Numerous researchers have proposed a variety of techniques for the removal of SPN from images, which can be categorized into linear and non-linear filters. The linear filters perform better for removing impulse noise at low noise densities. However, at high noise densities, the filtered images become blurred and fail to preserve the high-frequency content of image edges. Thus, non-linear filters have been used to provide effective filtering performance for impulse noise removal and preservation of other details of images.

Among the non-linear filters, the Standard Median Filter (SMF) is one of the earliest and most widely used techniques. It performs satisfactorily when the noise density is low; however, its performance deteriorates as the noise density increases because the median value becomes unreliable when the filtering window contains a large number of corrupted pixels. To improve this limitation, the Adaptive Median Filter (AMF) [5] employs a variable window size that expands until sufficient noise-free pixels are available for restoration. Although this adaptive strategy extends the operating range of the SMF, the enlarged window often introduces excessive smoothing, resulting in blurred edges and the loss of fine image details. Moreover, when the neighbourhood itself is heavily corrupted, increasing the window size alone is insufficient to provide reliable restoration.

Recognising these limitations, the Decision-Based Algorithm (DBA) [6] introduced a different strategy by first identifying whether the processing pixel is noisy and filtering only the detected noisy pixels. Unlike the SMF and AMF, the DBA leaves uncorrupted pixels unchanged, thereby preserving image details more effectively. However, its performance also degrades at high noise densities because repeated replacement of neighbouring corrupted pixels produces noticeable streaking artifacts. To reduce this problem, the Decision-Based Unsymmetric Trimmed Median Filter (DBUTMF) [7] computes the median after removing the noisy pixels from the filtering window. This modification reduces the streaking effect and generally provides better restoration than the DBA. Nevertheless, when all the pixels within the selected window are corrupted, no valid pixels remain after trimming, making it impossible to compute the median. The Modified Decision-Based Unsymmetric Trimmed Median Filter

(MDBUTMF) [8] addresses this situation by replacing the corrupted pixel with the mean of the selected window whenever all neighbouring pixels are noisy. Although this modification improves the restoration performance under highly corrupted conditions, the mean calculated from an entirely noisy neighbourhood may still lead to smoothing and loss of image details. Therefore, despite the progressive improvements from the SMF to the MDBUTMF, existing methods continue to struggle with preserving image details while maintaining reliable restoration performance at very high noise densities. This limitation provides the motivation for developing a more robust decision-based filtering approach.

An adaptive switching median filter [9] that made use of local noise density worked well when the noise level was low, but its effectiveness dropped as the noise density increased. To address this limitation, [10] came up with a decision-based partial trimmed global mean filter, which offered better edge preservation in images heavily affected by SPN noise. Extending the decision-based approach further, [11] designed a high-performance median filter that improved impulse noise detection while keeping computational requirements reasonable.

Among the more intelligent solutions, a deep learning-based switching filter for color images [12], showing clear improvements over traditional filters, though at the cost of added complexity. In the same direction of handling high noise levels, [13] proposed an adaptive min–max weighted median filter tailored for medical images, which proved highly effective in maintaining important details even under severe noise conditions.

Although [14] achieved real-time performance, its restoration accuracy decreases under extremely high noise densities. The adaptive threshold selection in [15] improved grayscale image restoration; however, its performance depends strongly on threshold estimation. Approaching the problem from a different angle by combining smoothing and sharpening filters, [16] not only reduces noise but also improves contrast, thereby avoiding the blurring problem often associated with denoising. Among the fuzzy logic approach, [17] proposed an adaptive Type-2 fuzzy detection method with regression-based filtering (IAFDLRBF), which stood out for its ability to handle very high noise densities while still preserving fine image details. Most recently, [18] proposed a hybrid adaptive and modified decision-based filter, merging adaptive detection with decision-based replacement. Their approach improved structural similarity and visual quality, especially for natural and medical images.

Although considerable progress has been made, existing filtering methods continue to suffer at high noise densities which results in streaking effect, blurring effect, texture preservation and new artifacts. Existing decision-based filters for SPN noise removal often experience a decline in their effectiveness when the noise density becomes very high, due to the ambiguity in deciding window size and the influence of corrupted pixels during restoration. Given these limitations,

the proposed filter Dual-stage enhanced decision based unsymmetric trimmed mean-median filter (EDBUTMMF) separates noise detection from pixel restoration through two adaptive stages. First stage performs adaptive noise identification and second stage performs restoration using enhanced trimmed mean-median estimation. The Dual-stage method ensures the separation between noisy pixels and non-noisy extreme-valued image features, confines the restoration process only on flagged pixels, without disturbing uncorrupted regions, and maintains reliable performance even at high noise densities. This method is designed to reduce computational complexity.

The main contribution of the proposed method EDBUTMMF is:

– Adaptive verification significantly reduces false detection and improves preservation of genuine image details.
– Less computational time compared to existing decision-based filters, making it suitable for practical images.
– Improves denoising performance across high noise densities while preserving image details and edges.

The paper is organized as follows: brief introduction and related work are discussed in Sec.1, the proposed method EDBUTMMF is explained in Sec.2. Illustrations of the proposed method are presented in Sec.3. Simulation results on different images in Sec. 4, followed by conclusions in Sec. 5.

2 Proposed Method

SPN noise randomly replaces some pixel values in the image with either the maximum intensity (255) or the minimum intensity (0) for an 8-bit image with a given probability p. Let X be the noise free image, Y be the SPN corrupted image and Z the denoised image of dimension $M \times N$ having pixel value at position (i, j) expressed as:

$$Y(i,j) = \begin{cases} q, & \text{with probability } p \\ \tilde{q}, & \text{with probability } 1-p \end{cases} \tag{1}$$

where $q = 0$ or 255 is a random variable dependent on the noise process and $\tilde{q} \neq 0$ or 255.

The proposed EDBUTMMF method has two stages, noise detection and noise removal. As DBUTMF and MDBUTMF have no noise detection stage where every pixel with 0 or 255 is considered as noise, the proposed EDBUTMMF initially treats such pixels as noise candidates. Their actual status is subsequently verified through an adaptive neighbourhood analysis before any denoising is performed. This analysis decreases the false detection and loss of

image details, particularly in homogeneous regions and under high-density SPN conditions.

2.1 Noise detection

Noise detection is an important step in image processing, as its primary objective is to accurately detect noisy pixels in noisy image. False detection occurs when an uncorrupted image pixel is incorrectly classified as noisy, leading to degradation of image details and edge information. Therefore, an effective noise detection algorithm should minimize false detection while maintaining a high detection rate for genuinely corrupted pixels.

For the 8-bit image, if processing pixel is either maximum (255) or minimum (0) value grey level, then the processing pixel is initially marked as noise candidate. At high noise densities or uniform region there is a high probability of initially marked noise candidate pixel to be a part of image itself. Therefore, to decrease the probability of false detection of pixel, the processing window size adaptively increases.

To check whether all the pixel values in a window are equal to q (either 0 or 255):

$$U(W_k(i,j)) = \begin{cases} 1, & \text{if } W_k(i,j) \forall (i,j) = q, \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

The formula introduces a uniformity predicate, $U(W_k(i,j))$, which is a binary test of homogeneity within a window $W_k(i,j)$ defined as: $W_k(i,j) = (2k+1) \times (2k+1)$, and $k = 1,2,3,4,5$.

The proposed algorithm progressively enlarges the adaptive window size until at least one reliable noise-free pixel is identified for noise verification and restoration. This strategy improves detection robustness, particularly at high salt-and-pepper noise (SPN) densities, where smaller neighbourhoods are more likely to be completely corrupted. However, excessively increasing the window size may incorporate pixels from distant regions, reducing local spatial information and adversely affecting edge preservation and spatial accuracy [17]. It also increases computational complexity and processing time. Therefore, an appropriate maximum window size is required to achieve a balance between restoration quality and computational cost.

Based on extensive simulations, the maximum adaptive window size was limited to 11×11. As shown in **Table 1**, increasing the window size improves restoration performance; however, beyond 11×11, the improvement in PSNR becomes relatively marginal, while computational time continues to increase.

Thus, the 11×11 neighbourhood provides a suitable trade-off between PSNR and computational time and was selected as the maximum window size.

If the adaptive window reaches beyond the maximum window size and contains at least one reliable non-extreme-valued pixel, the available noise-free information is used for restoration. However, if the entire neighbourhood consists only of identical extreme-valued pixels (all 0s or all 255s), the processing pixel is considered to belong to a probable homogeneous image region rather than being classified as noisy. In this case, reliable local information for restoration is unavailable; therefore, the processing pixel is retained unchanged. This strategy reduces false detection of genuine extreme-valued pixels and prevents restoration artifacts caused by unreliable neighbourhood estimates, thereby improving the robustness of the proposed method at high SPN densities.

The binary indicator $F(i,j)$ now can be represented as for given $Y(i,j)=q\in\{0,255\}$,

$$F(i,j)=\begin{cases}1, & \text{if } U(W_k(i,j),q)=1,\\ 0, & \text{otherwise.}\end{cases} \tag{3}$$

The noise detection process is given in Algorithm 1 and flowchart as shown in Fig. 1.

Algorithm 1: Noise Detection

Step 1: Set window size, $W_k(i,j)=(2k+1)\times(2k+1)$ and $k=1,2,...,5$.

Step 2: Assume that the pixel being processed is $Y(i,j)$.

Step 3: If $0<Y(i,j)<255$, then $Y(i,j)$ is considered as an uncorrupted pixel and retained. Otherwise, go to step 4.

Step 4: If $Y(i,j)=0\,||\,255$, then $Y(i,j)$ is considered as a candidate of being a noise, go to step 5.

Step 5: For all k :

$if\ (Y(i,j)=0\wedge\forall x\in W_k(i,j),x=0)\vee(Y(i,j)=255\wedge\forall x\in W_k(i,j),x=255),$

$F(i,j)=1$

$else\,F(i,j)=0$

2.2 Noise removal

In this step, only identified noisy pixels in noise detection stage will be filtered out, preserving the local features. The noisy pixel is replaced by trimmed median value. Moreover, the noisy pixel is replaced by the mean value of the filtering window, if the algorithm fails to detect noise free pixel at 11×11.

Our objective is to determine the actual value of $Y(i,j)$ for all $F(i,j)=1$. The filtering window denoted as W_k, is determined by a parameter k. For a

given k, the dimension of the window is $(2k+1)\times(2k+1)$, where k is a set of range from 1 to 5, which corresponds to filtering window sizes varying from W_1 up to W_5. Let $V_k = \{x \in W_k(i,j) \mid x \neq 0 \wedge x \neq 255\}$ the set of trimmed pixels $(x \neq 0 \wedge x \neq 255)$ and $Z(i,j)$ be estimated denoised pixel value.

$$Z(i,j) = \begin{cases} (V_k)_{median}, & \text{if } V_k \neq 0, \\ (W_k)_{mean}, & \text{otherwise,} \end{cases} \tag{4}$$

where

$$(V_k)_{median} = \underset{m}{\arg\min} \sum_{i=1}^{n} |x_i - m| \tag{5}$$

and

$$(W_k)_{mean} = \frac{1}{n} \sum_{i=1}^{n} x_i \,. \tag{6}$$

A corrupted pixel is substituted by the median of its clean neighbours in an adaptive window, with a fallback to the mean if no clean neighbours are discovered. The mean value is employed only when no valid noise-free pixels remain within the maximum adaptive window, making median estimation infeasible. This situation occurs only in severely corrupted neighbourhoods where reliable local information is unavailable. Under such circumstances, the mean provides a stable fallback estimate while avoiding repeated propagation of extreme-valued pixels. Since the adaptive detection stage significantly reduces the frequency of this condition, the influence of the mean-based restoration on the overall image quality is limited.

The algorithm 2 for noise removal and flowchart as shown in Fig. 2 above describes the process for noise removal.

Algorithm 2: Noise Removal

Step 1: Set window size, $W_k(i,j) = (2k+1)\times(2k+1)$ and $k = 1,2,...,5$.

Step 2: Assume that the pixel being processed is $Y(i,j)$ *and* $F(i,j) = 1$

Step 3: Read the pixels from the processing sliding window and store it in a set.

Step 4: If the current window does not contain all elements as 0 and 255. Then eliminate 0 and 255 and store it in the set V_k and compute the median value $(V_k)_{median}$ of the remaining elements. Replace $Y(i,j)$ with the $(V_k)_{median}$. Otherwise, go to step 5.

Step 5: Increase the window size by varying $k = 1,2,....,5$ and go to step 3.

Step 6: If $k = 5$,then go to step 7.

Step 7: Replace $Y(i,j)$ with the mean of pixel values present in the current window $(W_k)_{mean}$.

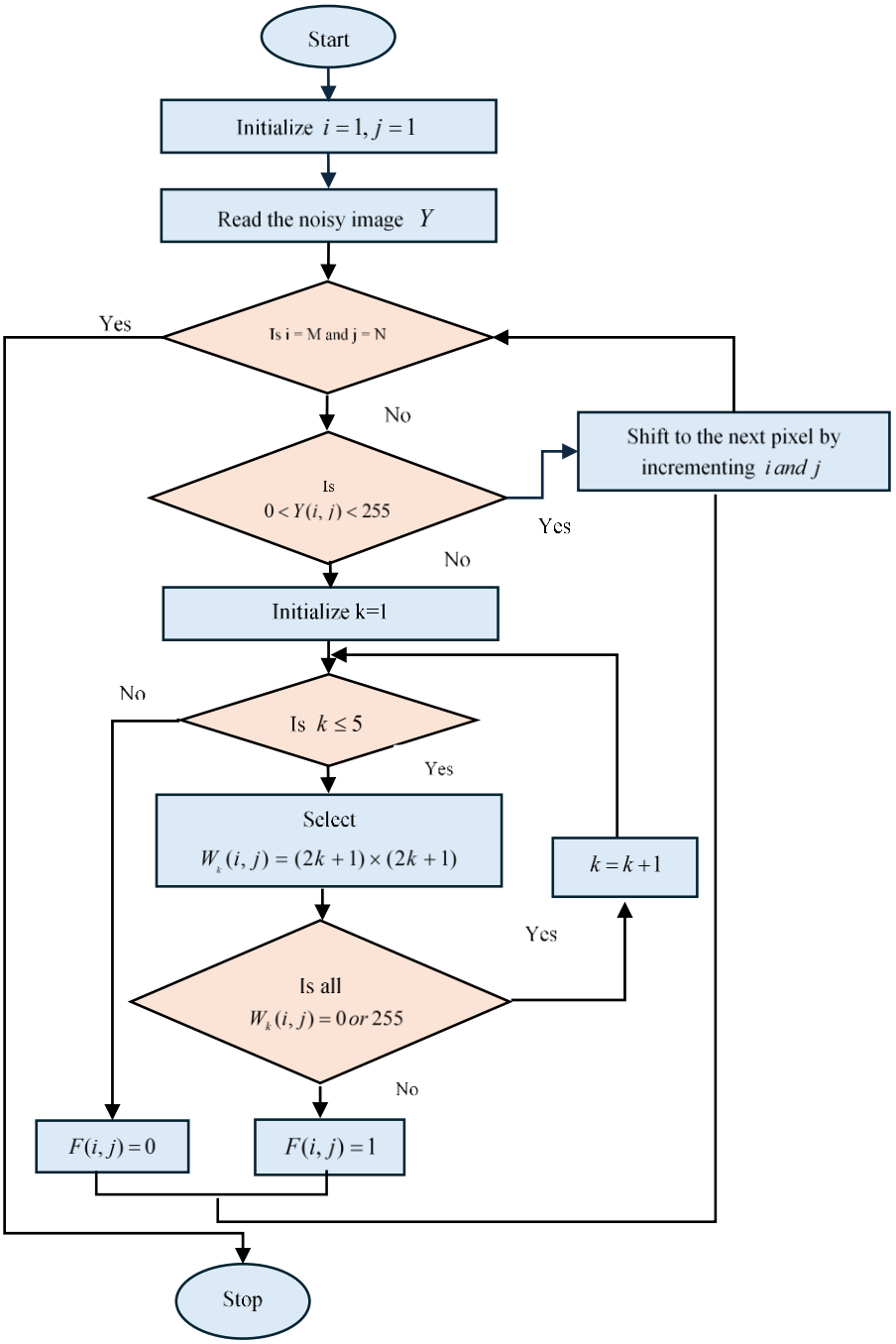


Fig. 1 – *Flow Process of Noise Detection.*

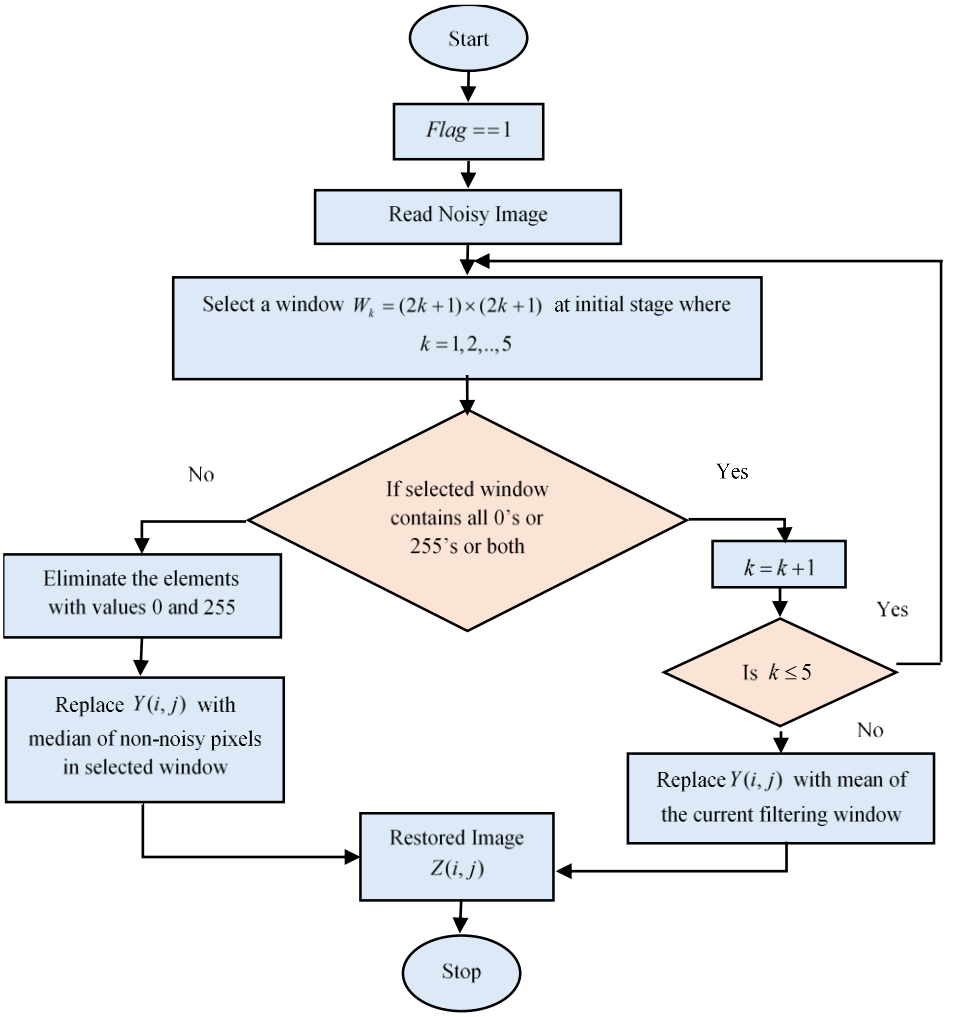


Fig. 2 – *Flow Process of Noise Removal.*

3 Illustration of Proposed Method

The illustration of the various cases of the proposed method is discussed here.

Case i): If the processing pixel (centre pixel) $Y(i,j)$ of the selected 3×3 window (initial window size) is uncorrupted, $0 < Y(i,j) < 255$, then it does not require further processing as shown in Fig. 3.

Here, the processing pixel $Y(i,j)$ is 164 which is an uncorrupted pixel i.e, $F(i,j) = 0$. This means $Y(i,j)$ remains unchanged.

255	167	160
162	**164**	158
0	159	155

Fig. 3 – 3×3 *window (Centre pixel is uncorrupted).*

Case ii): If the processing pixel $Y(i, j)$ of the selected 3×3 window (initial window size) as shown in Fig. 4 is 0 or 255 and the pixel values in the selected window are identical, the window size increases for $k = 1, 2, ...5$ (in this case, window size up to 5×5 are considered for simplicity as shown in Fig. 5). If all the pixels are identical for maximum window size, the pixels are considered as part of image and $F(i, j) = 0$ is marked.

255	255	255
255	**255**	255
255	255	255

Fig. 4 – 3×3 *window (Centre pixel is corrupted and its neighbour are identical).*

255	255	255	255	255
255	255	255	255	255
255	255	**255**	255	255
255	255	255	255	255
255	255	255	255	255

Fig. 5 – 5×5 *window (window size of Fig. 4 is increased by 2).*

Case iii): If the processing pixel $Y(i, j)$ of the selected 3×3 window (initial window size) is a corrupted pixel, $Y(i, j) = 0$ *or* 255, and the neighbouring pixel contains some uncorrupted pixels which means the elements present in the selected window are not identical. That means $F(i, j) = 1$ as shown in Fig. 6.

If $Y(i, j) = 0$ is the processing pixel, eliminate the SPN noise present in the selected window. 1-D array of pixel values of the selected window is created [255 167 255 162 0 158 0 159 157]. After the elimination of noisy pixel, the array will be [167 162 158 159 157], then the median value is calculated, which is 159. Hence, replace the processing pixel $Y(i, j)$ by 159.

255	167	255
162	**0**	158
0	159	157

Fig. 6 – 3×3 *window (Centre pixel is corrupted but its neighbour is uncorrupted).*

If $Y(i, j) = 0$ is the processing pixel, eliminate the SPN noise present in the selected window. 1-D array of pixel values of the selected window is created [255 167 255 162 0 158 0 159 157]. After the elimination of noisy pixel, the array will be [167 162 158 159 157], then the median value is calculated, which is 159. Hence, replace the processing pixel $Y(i, j)$ by 159.

Case iv): If the processing pixel $Y(i, j)$ and neighbouring pixel values are all corrupted but not identical i.e., it contains 0's and 255's in the selected 3×3 window as shown in Fig. 7. With $F(i, j) = 1$, then increase the window size.

255	255	0
0	**0**	255
255	255	0

Fig. 7 – 3×3 *window (Centre pixel is corrupted but its neighbours are not identical).*

If the processing pixel $Y(i, j)$ is 0. It is checked whether the elements present in the selected 3×3 window is identical or not, in this case it is not. Then the processing pixel $Y(i, j)$ is considered as noise i.e., $F(i, j) = 1$. Therefore, the window size is increased by 2 as shown in Fig. 8.

41	41	255	43	255
0	255	255	0	255
42	0	**0**	255	40
0	255	255	0	0
255	45	255	46	255

Fig. 8 – 5×5 *window (window size of Fig.* 7 *increased by* 2*).*

The 1-D array of the selected window will be [41 41 255 43 255 0 255 255 0 255 42 0 **0** 255 40 0 255 255 0 0 255 45 255 46 255]. After eliminating 0's and 255's the pixel values of the selected window will be [41 41 43 42 40 45 46]. Here the median value is 42. Hence, the processing pixel $I(i, j)$ is replaced by 42.

4 Simulation Results

The performance of the proposed algorithm is simulated on three standard greyscale (Lena, House and Cameraman) and two real-time images. The real-time images were captured by a smartphone camera and resized to size of 512×512. The simulation was done on a system running Intel(R) Core (TM) i7, 13th Gen CPU 64-bit and 16 GB of RAM. In the simulation, images are corrupted by salt & pepper noise with varying levels of noise ranging from 10% to 90%. Initially, to select the window size for noise detection and noise removal, several simulations are done by varying the window size for various noise densities. The simulations are performed on all the test image and its corresponding PSNR and run-time are computed in **Table 1** (all values are averaged of five trials).

Table 1

PSNR (dB) and run time (sec) for a set of images with varying window sizes.

Image	Window size	Noise density (%)					
		20		50		80	
		PSNR	Run Time (s)	PSNR	Run Time (s)	PSNR	Run Time (s)
Lena	9 × 9	39.85	1.36	33.31	3.38	27.50	6.64
	11 × 11	39.04	1.05	33.21	3.37	28.46	4.61
	13 × 13	39.07	1.83	33.16	2.39	26.40	6.63
	15 × 15	39.08	1.97	33.17	4.40	26.35	9.64
Cameraman	9 × 9	39.75	0.67	33.09	0.88	27.84	2.71
	11 × 11	39.94	0.36	33.05	0.85	27.90	1.63
	13 × 13	39.87	0.37	32.76	0.92	27.86	2.61
	15 × 15	39.68	0.94	32.93	0.90	27.82	2.66
House	9 × 9	45.81	1.16	38.27	1.37	32.92	2.64
	11 × 11	46.05	0.94	38.48	1.26	33.07	1.28
	13 × 13	46.00	0.96	38.42	1.39	32.99	3.66
	15 × 15	45.97	1.16	38.29	1.36	32.85	3.62

As shown in **Table 1**, there is not much improvement in PSNR as window size increases rather it increases the run time of algorithm, therefore based on extensive simulations, the stopping criteria for maximum window size was limited to (11×11) which provides good trade-off between PSNR and computation time.

Restoration performances are quantitatively measured by mean squared error (MSE), peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM).

$$MSE = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{n} [Z(i,j) - Y(i,j)]^2 , \tag{7}$$

$$PSNR = 10 \times \log_{10} \left(\frac{255^2}{MSE} \right) , \tag{8}$$

and

$$SSIM(X,Z) = \frac{(2\mu_X \mu_Z + C_1)}{(\mu_X^2 + \mu_Z^2 + C_1)} \times \frac{(2\sigma_{XZ} + C_2)}{(\sigma_X^2 + \sigma_Z^2 + C_2)} , \tag{9}$$

where M and N is the row and column of the image, $Y(i,j)$ represents the noisy image, and $Z(i,j)$ represents the denoised image, $\mu_X, \mu_Z, \sigma_X^2, \sigma_Z^2$ are mean and variances and σ_{XZ} is the covariance between original (noise free) image X and denoised image Z.

The PSNR and MSE of the proposed algorithm are calculated, and the performance of the proposed algorithm is compared against the existing algorithms at different noise densities for grey-scale images. The proposed filter gives better results at high noise densities as compared to AMF, DBA, DBUTMF, MDBUTMF, and IAFDLRBF as shown in Fig. 9, Fig. 10 and Fig. 11 for Lena, House and Cameraman image respectively.

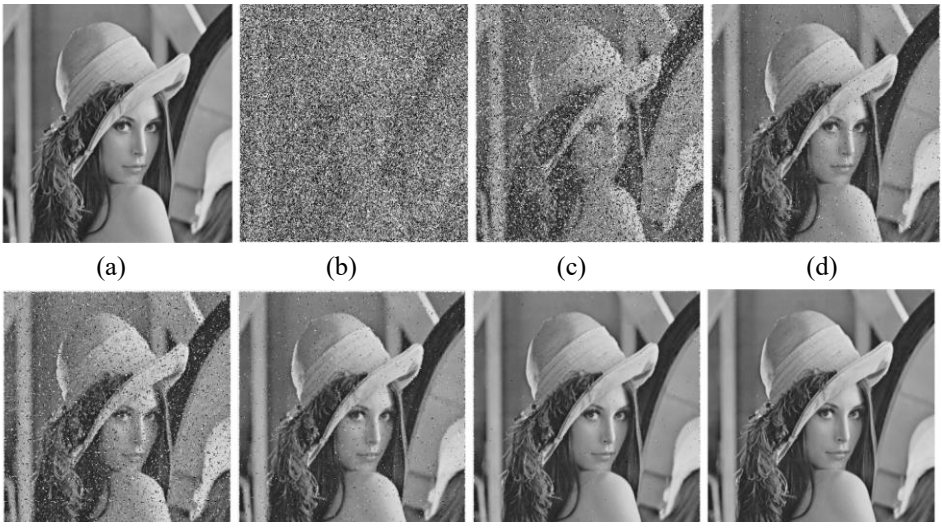


Fig. 9 – *Results for 80% noise corrupted Lena image* (a) *original Lena image;* (b) *80% noise corrupted image. Restoration results of:* (c) *AMF;* (d) *DBA;* (e) *DBUTMF;* (f) *MDBUTMF;* (g) *IAFDLRBF;* (h) *EDBUTMMF.*

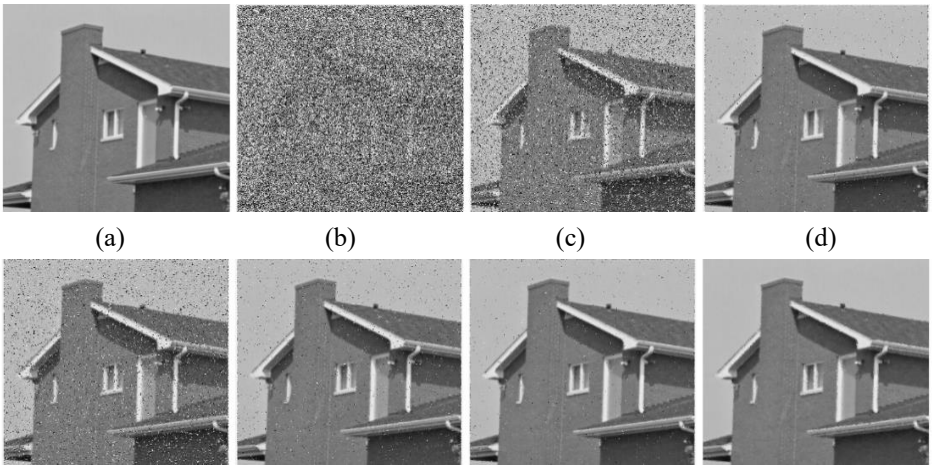


Fig. 10 – *Results for 80% noise corrupted House image:* (a) *original House image;* (b) *80% noise corrupted image. Restoration results of:* (c) *AMF;* (d) *DBA;* (e) *DBUTMF;* (f) *MDBUTMF;* (g) *IAFDLRBF;* (h) *EDBUTMMF.*

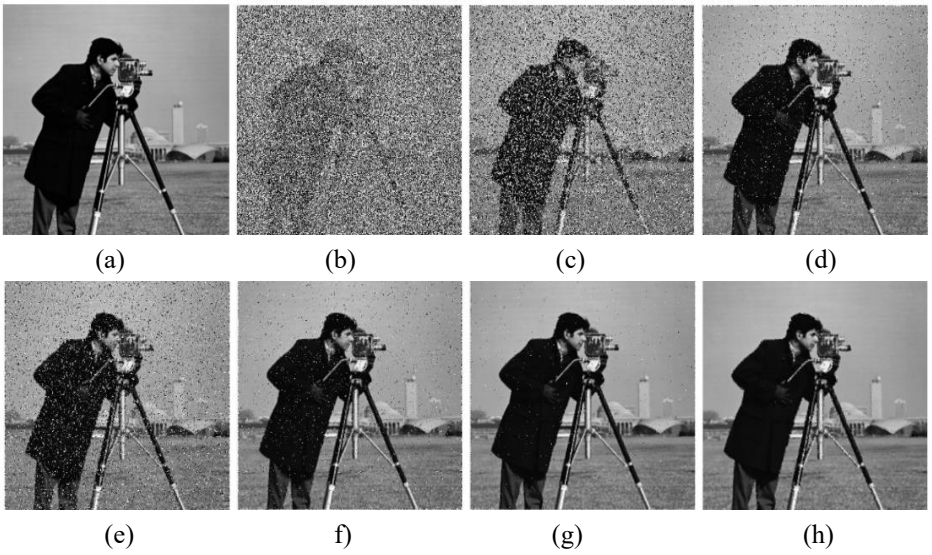


Fig. 11 – *Results for 80% noise corrupted Cameraman image:* (a) *original Cameraman image;* (b) *80% noise corrupted image. Restoration results of:* (c) *AMF;* (d) *DBA;* (e) *DBUTMF;* (f) *MDBUTMF;* (g) *IAFDLRBF;* (h) *EDBUTMMF.*

Table 2 demonstrate the experimental results of MSE and **Table 3** tabulates the PSNR and SSIM values of different algorithms for the grey-scale Lena image. The performance of the proposed algorithm is compared with various existing

filters, namely, AMF, DBA, DBUTMF, MDBUTMF and IAFDLRBF. It can be revealed that traditional filters such as AMF, DBA, DBUTMF, and MDBUTMF perform reasonably well at low noise densities but their PSNR values drop sharply as the noise level increases, indicating poor robustness under heavy noise. In contrast, the proposed EDBUTMMF consistently achieve higher PSNR across all noise densities, reaching 42.92 dB at 10% noise and still maintaining 26.15 dB even at 90% noise, whereas DBUTMF falls to just 9.51 dB. These findings highlight that while conventional methods are suitable for light to moderate noise, advanced approaches such as IAFDLRBF and especially EDBUTMMF provide superior noise removal capability and detail preservation, making them more reliable for high-density SPN noise scenarios. Figs. 12 and 13 demonstrate the graph-based comparison of MSE and PSNR performance metrics for various existing algorithms for Lena image.

Table 2
Comparison of MSE values of different filters for Lena image at different noise densities.

Noise Density (%)	AMF	DBA	DBUTMF
10%	17.50	17.62	4.20
20%	23.01	41.40	9.70
30%	37.67	63.25	16.75
40%	57.02	106.18	26.00
50%	86.11	149.65	85.91
60%	125.62	214.82	234.46
70%	183.68	369.89	312.98
80%	298.59	409.64	447.30
90%	592.44	562.0	529.14
Noise Density (%)	**MDBUTMF**	**IAFDLRBF**	**EDBUTMMF**
10%	13.30	6.36	3.31
20%	21.78	8.33	6.53
30%	38.82	15.10	11.17
40%	67.00	21.63	16.94
50%	118.05	31.26	21.98
60%	153.13	45.29	33.42
70%	243.26	74.14	55.42
80%	454.02	128.84	90.79
90%	544.23	279.30	157.79

Table 3
Comparison of PSNR (dB) and SSIM of different filters for Lena image at different noise densities.

Noise Density (%)	Metric	AMF	DBA	DBUTMF
10	PSNR	35.70	35.67	41.89
	SSIM	0.75	0.82	0.88
20	PSNR	34.51	31.96	38.26
	SSIM	0.71	0.81	0.83
30	PSNR	32.37	30.12	35.89
	SSIM	0.68	0.75	0.80
40	PSNR	30.57	27.87	33.98
	SSIM	0.63	0.71	0.76
50	PSNR	28.78	26.38	28.79
	SSIM	0.58	0.68	0.72
60	PSNR	27.14	24.81	24.43
	SSIM	0.55	0.63	0.62
70	PSNR	25.49	22.45	19.03
	SSIM	0.51	0.57	0.54
80	PSNR	23.38	20.28	14.07
	SSIM	0.46	0.48	0.51
90	PSNR	20.63	17.12	9.51
	SSIM	0.39	0.41	0.46
Noise Density (%)	**Metric**	**MDBUTMF**	**IAFDLRBF**	**EDBUTMMF**
10	PSNR	36.89	40.09	42.92
	SSIM	0.91	0.94	0.99
20	PSNR	34.75	38.92	39.98
	SSIM	0.84	0.91	0.97
30	PSNR	32.24	36.34	37.65
	SSIM	0.89	0.88	0.96
40	PSNR	29.87	34.78	35.84
	SSIM	0.74	0.85	0.94
50	PSNR	27.41	33.18	34.71
	SSIM	0.65	0.78	0.92
60	PSNR	26.28	31.57	32.89
	SSIM	0.61	0.72	0.90
70	PSNR	24.27	29.43	30.69
	SSIM	0.60	0.71	0.87
80	PSNR	21.56	27.03	28.55
	SSIM	0.56	0.67	0.83
90	PSNR	18.38	23.67	26.15
	SSIM	0.51	0.64	0.76

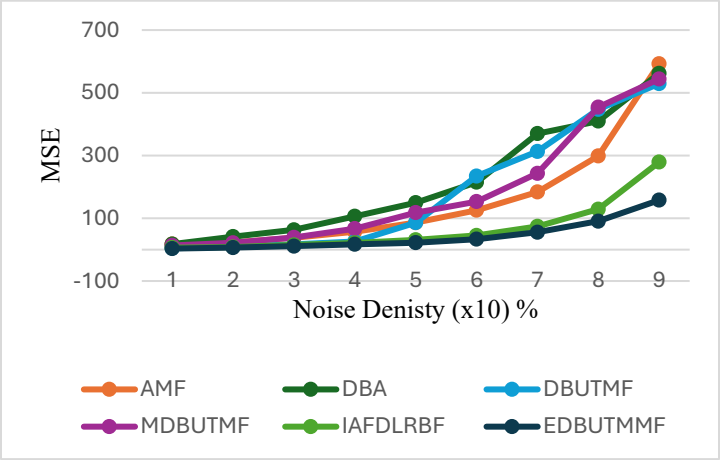


Fig. 12 – *MSE performance graph of proposed algorithm and existing algorithms at different noise densities for Lena image.*

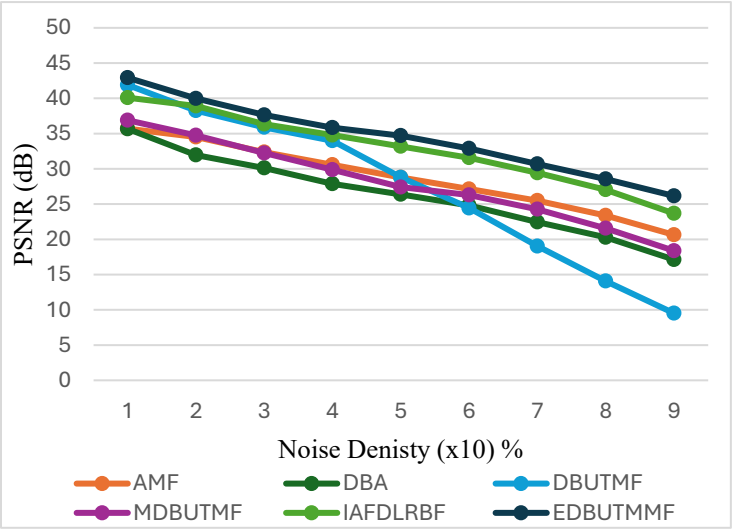
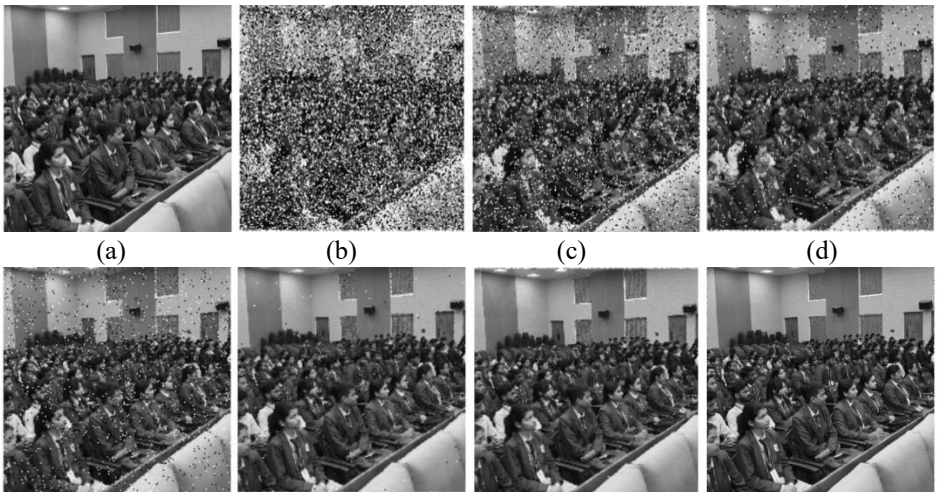


Fig. 13 – *PSNR performance graph of proposed algorithm and existing algorithms at different noise densities for Lena image.*

The simulation result of the proposed method at 80% noise density on real time image, Sample Image 1 and Sample Image 2 is shown in Fig. 14 and Fig. 15 respectively. The images obtained using the existing methods in Fig. 14c–14g successfully suppress a significant portion of the SPN, residual noisy pixels are still visible, particularly in homogeneous regions and around object boundaries. In contrast, the proposed EDBUTMMF method, shown in Fig. 14h, produces the most visually pleasing restoration. Most of the impulse noise is effectively removed while preserving important structural information. The audience

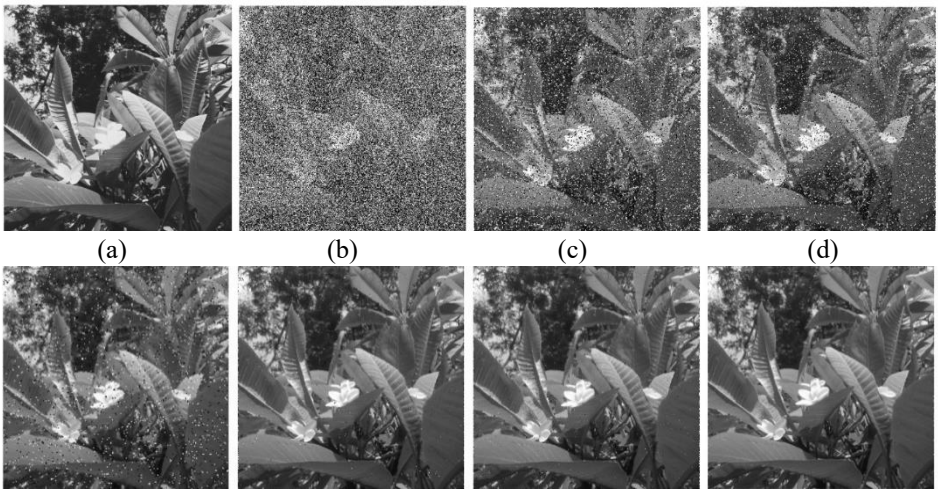
members, seating arrangement, and wall boundaries, appear significantly noise free, with well-preserved edges and improved local contrast.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 14 – *Results for 80% noise corrupted of Sample Image 1:* (a) *original Sample Image 1;* (b) *80% noise corrupted image. Restoration results of:* (c) *AMF;* (d) *DBA;* (e) *DBUTMF;* (f) *MDBUTMF;* (g) *IAFDLRBF;* (h) *EDBUTMMF.*



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 15 – *Results for 80% noise corrupted Sample Image 2:* (a) *original Sample Image 2;* (b) *80% noise corrupted image. Restoration results of:* (c) *AMF;* (d) *DBA;* (e) *DBUTMF;* (f) *MDBUTMF;* (g) *IAFDLRBF;* (h) *EDBUTMMF.*

The comparison of PSNR and SSIM of different filters for Sample Image 1 and Sample Image 2, at different noise densities are presented in **Table 4** and **5**.

Table 4
Comparison of PSNR (dB) and SSIM of different filters for Sample Image 1 at different noise densities.

Noise Density (%)	Metric	AMF	DBA	DBUTMF
10	**PSNR**	34.63	35.13	41.26
	SSIM	0.73	0.81	0.87
20	**PSNR**	33.47	31.48	37.69
	SSIM	0.69	0.80	0.82
30	**PSNR**	31.40	29.67	35.35
	SSIM	0.66	0.74	0.79
40	**PSNR**	29.65	27.45	33.47
	SSIM	0.61	0.70	0.75
50	**PSNR**	27.92	25.98	28.36
	SSIM	0.56	0.67	0.71
60	**PSNR**	26.33	24.44	24.06
	SSIM	0.53	0.62	0.61
70	**PSNR**	24.73	22.11	18.74
	SSIM	0.49	0.56	0.53
80	**PSNR**	22.68	19.98	13.86
	SSIM	0.45	0.47	0.50
90	**PSNR**	20.01	16.86	9.37
	SSIM	0.38	0.40	0.45
Noise Density (%)	**Metric**	**MDBUTMF**	**IAFDLRBF**	**EDBUTMMF**
10	**PSNR**	43.16	46.91	41.50
	SSIM	0.86	0.91	0.96
20	**PSNR**	40.66	45.54	38.66
	SSIM	0.98	0.86	0.94
30	**PSNR**	37.72	42.52	36.41
	SSIM	0.84	0.87	0.93
40	**PSNR**	34.95	40.69	34.66
	SSIM	0.87	0.94	0.91
50	**PSNR**	32.07	38.82	33.56
	SSIM	0.76	0.91	0.89
60	**PSNR**	30.75	36.94	31.80
	SSIM	0.71	0.84	0.87
70	**PSNR**	28.40	34.43	29.68
	SSIM	0.70	0.83	0.84
80	**PSNR**	25.23	31.63	27.61
	SSIM	0.66	0.78	0.80
90	**PSNR**	21.50	27.69	25.29
	SSIM	0.60	0.75	0.73

Table 5
Comparison of PSNR (dB) and SSIM of different filters for Sample Image 2 at different noise densities.

Noise Density (%)	Metric	AMF	DBA	DBUTMF
10	PSNR	42.48	42.45	40.34
	SSIM	0.89	0.98	0.85
20	PSNR	41.07	38.03	36.84
	SSIM	0.84	0.96	0.80
30	PSNR	38.52	35.84	34.56
	SSIM	0.81	0.89	0.77
40	PSNR	36.38	33.17	32.72
	SSIM	0.75	0.84	0.73
50	PSNR	34.25	31.39	27.72
	SSIM	0.69	0.81	0.69
60	PSNR	32.30	29.52	23.53
	SSIM	0.65	0.75	0.60
70	PSNR	30.33	26.72	18.33
	SSIM	0.61	0.68	0.52
80	PSNR	27.82	24.13	13.55
	SSIM	0.55	0.57	0.49
90	PSNR	24.55	20.37	9.16
	SSIM	0.46	0.49	0.44
Noise Density (%)	**Metric**	**MDBUTMF**	**IAFDLRBF**	**EDBUTMMF**
10	PSNR	35.53	48.51	51.93
	SSIM	0.88	0.84	0.92
20	PSNR	33.46	47.09	48.38
	SSIM	0.81	0.91	0.87
30	PSNR	31.05	43.97	45.56
	SSIM	0.86	0.86	0.94
40	PSNR	28.76	42.08	43.37
	SSIM	0.71	0.83	0.95
50	PSNR	26.40	40.15	42.00
	SSIM	0.63	0.94	0.91
60	PSNR	25.31	38.20	39.80
	SSIM	0.59	0.87	0.89
70	PSNR	23.37	35.61	37.13
	SSIM	0.58	0.86	0.95
80	PSNR	20.76	32.71	34.55
	SSIM	0.54	0.81	0.87
90	PSNR	17.70	28.64	31.64
	SSIM	0.49	0.77	0.92

The SSIM values decrease with increasing noise density for all methods due to the severe degradation of structural information as shown in **Table 4**. However, the proposed EDBUTMMF consistently achieves higher SSIM than AMF, DBA, DBUTMF, and MDBUTMF across all noise levels, indicating better preservation of image structures and edges. Although IAFDLRBF attains the highest SSIM values, the proposed method provides competitive structural similarity with significantly lower computational complexity.

For Sample Image 2 as shown in Fig. 15b, the image is severely corrupted by high-density SPN, resulting in significant loss of visual details. Although the existing methods suppress most of the noise, residual noisy pixels and smoothing effects remain, leading to partial loss of edge and texture information. In contrast, the proposed EDBUTMMF method (Fig. 15h) effectively removes the impulse noise while preserving leaf textures, flower petals, and edge details. The adaptive noise verification and selective restoration strategy reduce false detection and void unnecessary modification of uncorrupted pixels, producing a visually superior restored image.

Table 6 compares the execution time of the proposed EDBUTMMF method with various existing methods at different noise densities where it clearly reveals the less time is needed to detect and remove the noise. The execution time is tested on standard dataset image and real time image. Although the proposed algorithm uses adaptive window expansion, the additional computational cost is limited because the window is expanded only for pixels identified as noise candidates. In most cases, the search stops as soon as a noise-free neighbouring pixel is found, allowing the use of smaller windows. Moreover, only the confirmed noisy pixels are restored, avoiding unnecessary processing of uncorrupted pixels. The mean-based fallback is used only when no valid neighbouring pixels are available within the maximum window size. As a result, the overall computational cost remains low, leading to the reduced execution time observed in the experimental results.

5 Conclusion

In this paper, we have proposed an improved algorithm to remove SPN noise from images that gives better performance at high noise densities. The results demonstrate that the proposed EDBUTMMF filter achieves improved restoration performance in comparison to AMF, DBA, DBUTMF, MDBUTMF, and IAFDLRBF. The performance of the proposed algorithm has been examined in terms of MSE, PSNR and SSIM. Both visual and quantitative results have been demonstrated. The proposed algorithm result reveals the effectiveness in standard as well as real time images for removing SPN even at high noise density. The image detail preservation with reduced residual artifacts even at high noise density and less execution time makes EDBUTMMF potential suitable for

practical image processing applications. Future work will focus on extending the proposed approach to colour images and other impulse noise models.

Table 6
Comparison of Execution Time (seconds) at different noise densities.

Image	Methods	Noise Density (%)								
		10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**
Lena	AMF	5.89	6.12	5.93	6.01	6.41	7.31	8.35	10.84	17.71
	MDBUTMF	2.07	3.54	5.13	9.45	11.32	15.97	20.43	36.89	65.86
	IAFDLRBF	1.98	2.73	4.92	8.17	10.53	15.69	21.42	37.95	68.87
	EDBUTMMF	**0.91**	**1.01**	**1.87**	**2.32**	**3.67**	**3.89**	**4.75**	**5.90**	**8.54**
Cameraman	AMF	1.85	1.97	1.58	1.49	1.63	1.74	2.05	2.68	4.53
	MDBUTMF	1.86	2.56	4.78	7.56	9.42	16.26	22.52	32.59	40.72
	IAFDLRBF	1.28	1.43	1.32	3.56	4.72	3.67	6.54	7.61	7.86
	EDBUTMMF	**0.16**	**0.32**	**0.92**	**0.71**	**0.89**	**0.97**	**1.14**	**1.96**	**2.84**
House	AMF	1.43	1.37	1.56	1.78	1.86	1.95	2.74	2.93	3.23
	MDBUTMF	1.32	1.78	1.56	3.67	7.36	12.89	15.34	21.76	22.89
	IAFDLRBF	1.34	1.56	1.21	2.89	5.76	7.34	6.19	9.39	10.95
	EDBUTMMF	**0.13**	**0.87**	**0.53**	**0.98**	**1.21**	**1.44**	**1.98**	**1.34**	**2.51**
Sample Image1	AMF	3.98	4.84	6.87	7.23	7.91	8.06	8.57	8.94	9.34
	MDBUTMF	2.53	3.03	4.52	5.64	6.35	7.02	7.54	8.30	8.64
	IAFDLRBF	1.87	2.65	2.88	2.59	3.50	4.61	4.99	5.24	5.69
	EDBUTMMF	**0.14**	**0.46**	**0.75**	**0.69**	**0.97**	**1.23**	**1.64**	**2.89**	**3.95**
Sample Image2	AMF	3.33	4.07	5.47	6.60	6.96	7.34	7.89	8.14	8.49
	MDBUTMF	2.27	2.87	3.24	3.94	4.00	4.46	5.24	5.87	6.77
	IAFDLRBF	1.55	2.07	2.31	2.87	3.11	3.51	4.14	4.76	4.95
	EDBUTMMF	**0.31**	**0.64**	**0.31**	**0.86**	**0.91**	**0.98**	**1.53**	**2.96**	**3.41**

6 References

[1] J. Astola, P. Kuosmanen: Fundamentals of Nonlinear Digital Filtering, CRC Press, Boca Raton, New York, 1997.

[2] R. C. Gonzalez, R. E. Woods: Digital Image Processing, 2nd Edition, Prentice-Hall, Inc, Upper Saddle River, 2002.

[3] A. Kumar, S. Kumar, A. Pathak: Fuzzy Logic Denoiser: An Approach for Salt and Pepper Noise Removal in Digital Images, Proceedings of the IEEE 2nd International Conference on Device Intelligence, Computing and Communication Technologies (DICCT), Dehradun, India, March 2024, pp. 341 – 346.

[4] A. Kumar, S. Kumar, A. Kar: Salt and Pepper Denoising Filters for Digital Images: A Technical Review, Serbian Journal of Electrical Engineering, Vol. 21, No. 3, October 2024, pp. 429 – 466.

[5] H. Hwang, R. A. Haddad: Adaptive Median Filters: New Algorithms and Results, IEEE Transactions on Image Processing, Vol. 4, No. 4, April 1995, pp. 499 – 502.

[6] K. S. Srinivasan, D. Ebenezer: A New Fast and Efficient Decision-Based Algorithm for Removal of High-Density Impulse Noises, IEEE Signal Processing Letters, Vol. 14, No. 3, March 2007, pp. 189 – 192.

[7] K. Aiswarya, V. Jayaraj, D. Ebenezer: A New and Efficient Algorithm for the Removal of High Density Salt and Pepper Noise in Images and Videos, Proceedings of the Second International Conference on Computer Modeling and Simulation, Sanya, China, January 2010, pp. 409 – 413.

[8] S. Esakkirajan, T. Veerakumar, A. N. Subramanyam, C. H. PremChand: Removal of High Density Salt and Pepper Noise Through Modified Decision Based Unsymmetric Trimmed Median Filter, IEEE Signal Processing Letters, Vol. 18, No. 5, May 2011, pp. 287 – -290.

[9] X. Kang, W. Zhu, K. Li, J. Jiang: A Novel Adaptive Switching Median Filter for Laser Image Based on Local Salt and Pepper Noise Density, Proceedings of the IEEE Power Engineering and Automation Conference, Wuhan, China, September 2011, pp. 38 – 41.

[10] M. T. Raza, S. Sawant: High Density Salt and Pepper Noise Removal Through Decision Based Partial Trimmed Global Mean Filter, Proceedings of the Nirma University International Conference on Engineering (NUiCONE), Ahmedabad, India, December 2012, pp. 1 – 5.

[11] G. Moktader Daiyan, M. A. Mottalib, M. Mizanur Rahman: High Performance Decision Based Median Filter for Salt and Pepper Noise Removal in Images, Proceedings of the 15th International Conference on Computer and Information Technology (ICCIT), Chittagong, Bangladesh, December 2012, pp. 107 – 112.

[12] K. Radlak, L. Malinski, B. Smolka: Deep Learning Based Switching Filter for Impulsive Noise Removal in Color Images, Sensors, Vol. 20, No. 10, May 2020, p. 2782.

[13] B. Garg: An Adaptive Minimum-Maximum Value-Based Weighted Median Filter for Removing High Density Salt and Pepper Noise in Medical Images, International Journal of Ad Hoc and Ubiquitous Computing, Vol. 35, No. 2, 2020, pp. 84 – 95.

[14] T. M. Alanazi, K. Berriri, M. Albekairi, A. Ben Atitallah, A. Sahbani, K. Kaaniche: New Real-Time High-Density Impulsive Noise Removal Method Applied to Medical Images, Diagnostics, Vol. 13, No. 10, May 2023, p. 1709.

[15] N. Cao, Y. Liu: High-Noise Grayscale Image Denoising Using an Improved Median Filter for the Adaptive Selection of a Threshold, Applied Sciences, Vol. 14, No. 2, January 2024, p. 635.

[16] M. M. Weli, O. M. Abdullah: Digital Image Noise Reduction Based on Proposed Smoothing and Sharpening Filters, The Indonesian Journal of Computer Science, Vol. 13, No. 4, August 2024, pp. 5001 – 5021.

[17] A. Kumar, S. Kumar, A. Kar: Improved Adaptive Type-2 Fuzzy Detection and Simple Linear Regression-Based Filter for Removing Salt & Pepper Noise, Circuits, Systems, and Signal Processing, Vol. 43, No. 12, December 2024, pp. 7781 – 7819.

[18] F. Ullah, K. Kumar, T. Rahim, J. Khan, Y. Jung: A New Hybrid Image Denoising Algorithm Using Adaptive and Modified Decision-Based Filters for Enhanced Image Quality, Scientific Reports, Vol. 15, March 2025, p. 8971.