International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 12 | December - 2022 Impact Factor: 7.185 ISSN: 2582-3930

High Density Salt and Pepper Noise Removal Using Fusion Model of Nonlinear Filtering Techniques

Asha Sharma[1], Dr. Kulwinder Singh[2]
[1,2]Bhai Maha Singh College of Engineering, Sri Muktsar Sahib

Abstract— In this paper, an effective technique for removal of high density salt and pepper noise through decision based unsymmetric partial trimmed adaptive neighborhood mean filter is proposed. The proposed algorithm comprises of two-phase scheme. The first phase involves Decision Based Partial Trimmed Global Mean Filter (DBPTGMF) and Decision Based Unsymmetric Trimmed Winsorized Mean Filter (DBUTWMF) to work in all possible cases and the second phase is employed after the first phase to replace the left over noisy picture elements (pixel). The proposed algorithm is the fusion of the benefits of DBPTGMF, DBUTWMF and Decision Based Adaptive Neighborhood Mean Filter (DBANMF). The proposed algorithm (PA) is examined against several grayscale pictures in terms of Peak Signal to Noise Ratio (PSNR), Image Enhancement Factor (IEF) and Structural Similarity Index Matrix (SSIM) values at high noise density and it has been observed that the P A gives better results than the Adaptive Median Filter (AMF), Decision Based Algorithm (DBA), Modified Decision Based Unsymmetric Trimmed Median Filter (MDBUTMF), Enhanced Modified Decision Based Unsymmetric Trimmed Median Filter (EMDBUTMF), Modified Decision Based Unsymmetric Trimmed Global Mean Filter (MDBUMF_GM).

Keywords—Impulse Noise; Partial Trimmed Global; Trimmed Global Mean; Unsymmetric Trimmed Median Filter; Decision Based Algorithm; Winsorized Mean.

I - INTRODUCTION

Often pictures are spoiled with impulse noise or Salt and Pepper Noise (SPN). It belongs to class of acoustical noise that bears unwanted signal that corrupts the image. It is attributable to malfunctioning of picture elements, faulty transmission or due to defective memory locations [1]. The pixel spoiled by SPN takes either minimum (0) or maximum (255) value in the dynamic ambit [2]. Several non-linear filtering techniques were used to remove impulse noise as linear filter fails to do so [3]-[5]. Median filter are the most common types of non-linear filter. These are widely used for removal of noise due to their admirable denoising ability [1] and computational efficiency [6], but as the noise density of noise the increases above 50%, the boundaries and fine details of the picture are not recovered [7].

AMF [8] and Switching Median Filter (SMF) [9] were proposed to recover picture at high noise level, but on replacing noisy picture elements (0 or 255) with median value, it affects the local features and fine edges of the image. To get over these limitations DBA [10] was advised. The algorithm works for both high and low noise density and on replacing the noisy picture element with its adjacent pixel, it results in streaking [11]. To get over this drawback, Decision Based Unsymmetric Trimmed Median Filter (DBUTMF) was advised [3]. In this algorithm, if current processing window (W_p) bears noisy picture element, then it is e exchanged with trimmed median value of W_p, but at high noise level trimmed median value cannot be attained. To get over this problem MDBUTMF was advised [12]. In this algorithm, if W_p the bears entire picture elements as either 0 or 255, then the processing pixel (W_{ij}) is e exchanged with the mean value of the W_p and if W_p bears 0 or 255 along with other picture element value the W_{ij} is exchanged with trimmed median value of the W_p. In order to get better quantitative results as compared to MDBUTMF, MDBUMF_GM [4] and EMDBUTMF [13] were advised and in this algorithm if W_p bears entire picture elements corrupted with SPN, then W_{ij} is exchanged with the trimmed global mean of W_p and if the W_p bears 0 or 255 along with non-noisy picture elements, then W_{ij} is exchanged with trimmed median value of the W_p. In EMDBUTMF, if W_p bears entire picture elements as 0's then W_{ij} is exchanged with mean of the image and if the Wp bears entire picture elements as 255's then W_{ij} is replaced with 'b win × max ( )' and if W_p bears entire picture elements as either 0 or 255, then W_{ij} is replaced with mean value of the W_p and if the W_p bears 0 or 255 along with non-noisy picture elements, then W_{ij} is exchanged with trimmed median value of the W_p. Both of these algorithms fail to preserve fine details, edges of the picture and exhibits fading. To get over this drawback, DBUTWMF is proposed [5]. In this algorithm if W_p bears entire picture element as 0 or 255, then W_{ij} is exchanged with mean of W_p and if noisy picture element is present along with non-noisy picture elements in the W_p then W_{ij} is exchanged with the trimmed modified Winsorized mean, but it did not address any case, when all the picture elements of W_p are either exclusively 0 or exclusively 255.

To get over the above short coming a novel two-phase scheme is presented in this paper for removal of high noise density and to retain the denoise image as much as possible.

The residue of the paper is ordered as follows: Section II depicts the algorithms used in the fusion of the proposed algorithm. Section III represents the proposed method. Section IV depicts the illustration of proposed algorithm in detail. Simulation results are explained in section V and finally conclusion and future scope is given in section VI.

II. BRIEF REVIEW OF FUSION FILTERINGTECHNIQUES

A. DBPTGMF [14]

This algorithm gives solution in four steps:

Step 1: If the entire picture elements of W_p are noisy then the mean of the window will also be noisy. In that case, W_{ij} is subbed by trimmed global mean.

Step 2: If all the picture elements of W_p are 0, then W_{ij} is subbed with salt noise trimmed global mean value of the picture.

Step 3: If the entire picture elements of W_p are 255, then W_{ij} of W_p is subbed with pepper noise trimmed global mean value of the picture.

Step 4: If the W_p contains 0 or 255 along with non-noisy picture elements than W_{ij} is subbed with median value.

B. DBUTWMF [12]

The Unsymmetric Trimmed Modified Winsorized Mean can be used to calculate as follows:

Step 1: Let us considered a 1D trimmed sorted array {10 100 110 199 200}.

Step 2: Now applying the Winsorized mean procedure, the lowest and highest values of the array are repeated.

Step 3: The resultant array becomes {10 10 100 110 199 200}. Then determine the Winsorized mean of resultant array.

C. DBANMF [15]

In this technique, the Winsorized mean of the second ordered neighborhood pixels is calculated as:

Step 1: First consider the second ordered neighborhood pixels.

Step 2: Sorting is applied on it.

Step 3: Then calculate the Winsorized mean of the first ordered neighborhood pixels and subbed the noisy pixel with that value.

III. PROPOSED ALGORITHM

The PA processes the spoiled image in two-phase scheme. In first phase, EMDBUTMF works in all possible cases. The second phase is executed after the execution of first phase to preserve the fine edge details of the picture. The steps of proposed algorithm are listed as follows:

Phase I

Step 1: Take 2-D window of size 3×3. The processing pixel of the processing window (W_p) is presumed as W_{ij}.

Step 2: If 0 < 255 < W_{ij}, then W_{ij} is weigh as an uncorrupted picture element and is left over unaltered. This is exam plumed in case 1) of section IV.

Step 3: If W_{ij} = 0 or W_{ij} = 255, then W_{ij} is a corrupted pixel. The possible stages are given below:

Step 4: It consists of four possible cases, which are as follows:

Case 1) If W_p bears entire picture elements as noisy, and then W_{ij} remains unchanged.

Case 2) If W_p bears entire picture elements as 0's exclusively, then W_{ij} is exchanged with salt noise trimmed global mean value of the picture.

Case 3) If W_p bears entire picture elements as 255's, then W_{ij} is exchanged with pepper noise trimmed global mean value of the picture.

Case 4) If W_p bears not entire pixels as noisy, then W_{ij} is exchanged with Winsorized mean value of W_p.

Phase II

It is considered at high noise density up to 99% when the output of phase I still consists of noisy pixel, then W_{ij} is replaced with the calculated Winsorized mean of the second ordered neighborhood pixels of the processing window otherwise W_{ij} is remains unaltered.

IV. ILLUSTRATION OF PROPOSED ALGORITHM

Case 1) If W_p bears entire pixels as noise free pixels and W_{ij} is also a noise free picture element then it is left unaltered.

For example, if W_{ij} is 45 then it is $0 < 45 < 255$

$$\begin{bmatrix} 44 & 68 & 71 \\ 56 & <45> & 90 \\ 87 & 82 & 67 \end{bmatrix}$$

Case 2) if the processing window bears entire pixels as noisy (0 and 255), then still the processing pixel remains unaltered and it will be processed in the second phase.

$$\begin{bmatrix} 255 & 0 & 255 \\ 255 & <0> & 0 \\ 0 & 255 & 0 \end{bmatrix}$$

The process pixel remains unaltered.

Case 3) if the processing window bears entire pixels as 0's exclusively, then W_{ij} is replaced with the partial global mean of the image, excluding 0's

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & <0> & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Case 4) If the processing window bears entire pixels as 255's exclusively, then W_{ij} is replaced the partial global mean of the image, excluding 255's

For example, if W_{ij} is 255

$$\begin{bmatrix} 255 & 255 & 255 \\ 255 & <255> & 255 \\ 255 & 255 & 255 \end{bmatrix}$$

Case 5) If the processing window bears not entire pixels as 0's and 255's.

For example, if W_{ij} is 255

$$\begin{bmatrix} 79 & 91 & 255 \\ 121 & <255> & 0 \\ 98 & 0 & 74 \end{bmatrix}$$

Now wipe out all the 0's and 255's then 2-D array is converted into 1-D array. After sorting resultant array is [74 79 91 98 121]. Now replicate the minor and major value of the resultant array and calculate the Winsorized mean of the array. Hence W_{ij} is 94.

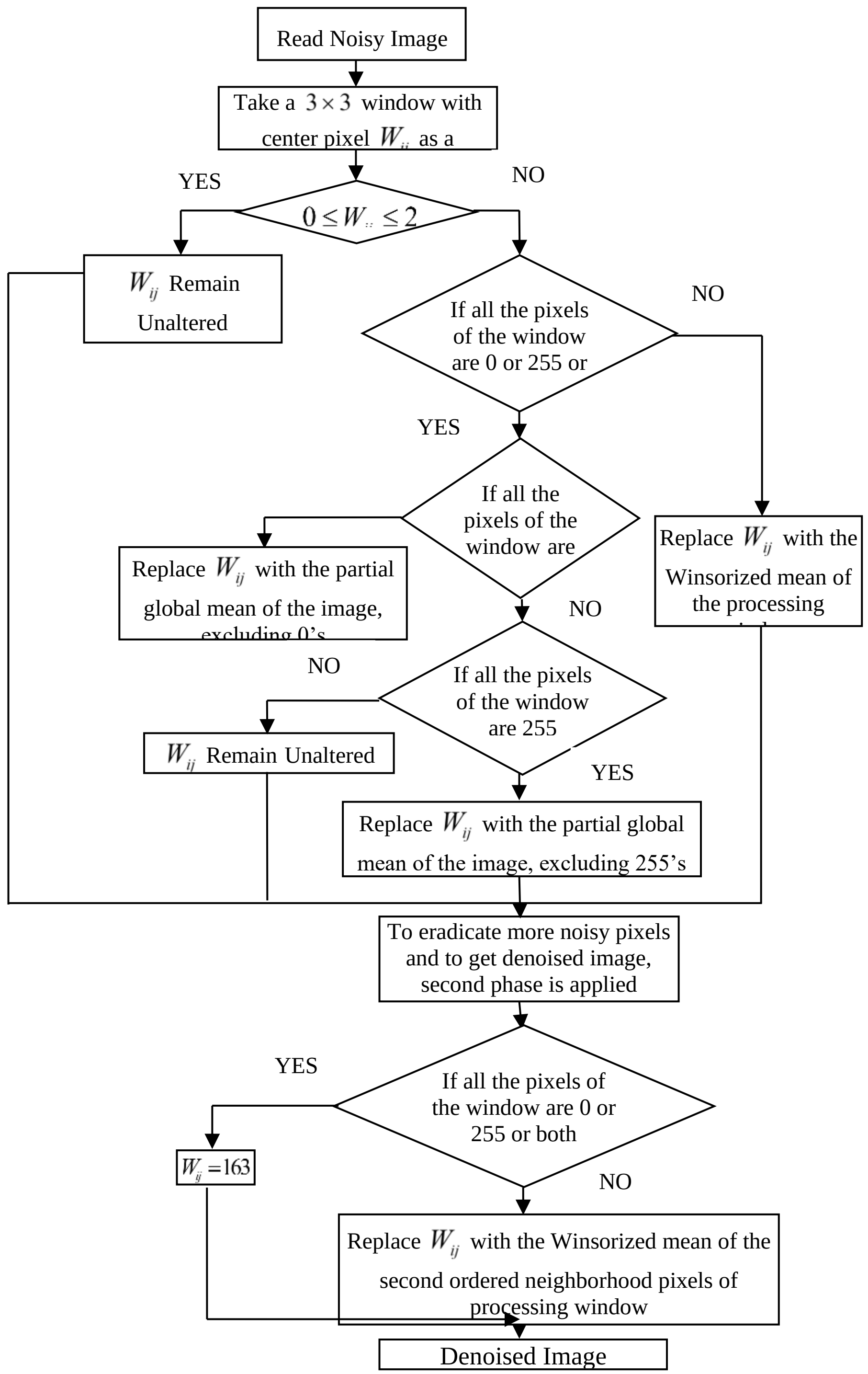

Fig. 1: Represents the flow chart of proposed algorithm

Case 6) If noise is still present after the phase I, then W_{ij} =0 or 255 is replaced with the calculated Winsorized mean of the second ordered neighborhood pixels of processing window.

For example, if W_{ij} is 255

$$\begin{bmatrix} 115 & 255 & 0 \\ 60 & <0> & 255 \\ 67 & 255 & 255 \end{bmatrix}$$

By calculation Winsorized mean is evaluated as 91. Hence W_{ij} is 91.

V. SIMULATION RESULTS

The performance of the proposed algorithm is tested using several grayscale images. The noise intensity is varied from 10% to 99%. Performance of proposed algorithm is quantitatively evaluated and based on PSNR, MSE, IEF and SSIM as defined in equations 1 to 4 respectively.

$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right) \tag{1}$$

$$MSE = \frac{\sum_i \sum_j \left(o_{ij} - r_{ij}\right)^2}{M \times N} \tag{2}$$

$$IEF = \frac{\sum_i \sum_j \left(n_{ij} - o_{ij}\right)^2}{\sum_i \sum_j \left(r_{ij} - o_{ij}\right)^2} \tag{3}$$

$$SSIM\,(p,q) = \frac{\left(2\mu_p\mu_q + c_1\right)\left(2\sigma pq + c_2\right)}{\left(\mu_p^2\mu_q^2 + c_1\right)\left(\sigma p^2 + \sigma q^2 + c_2\right)} \tag{4}$$

where o, n, r denote the original image, noisy image and the restored image respectively; M and N are the image width and height respectively; μ_p and μ_q are the mean intensity of original image and restored image respectively; σ_p and σ_q are the standard deviation of original image and restored image respectively; σ_{pq} is the covariance of the original image and recovered image; c_1 and c_2 variables are used to avoid division with zero where $c_1 = (0.01L)^2$; $c_2 = (0.03L)^2$ and L is the dynamic range of the pixel values. For gray scale image L = 255.

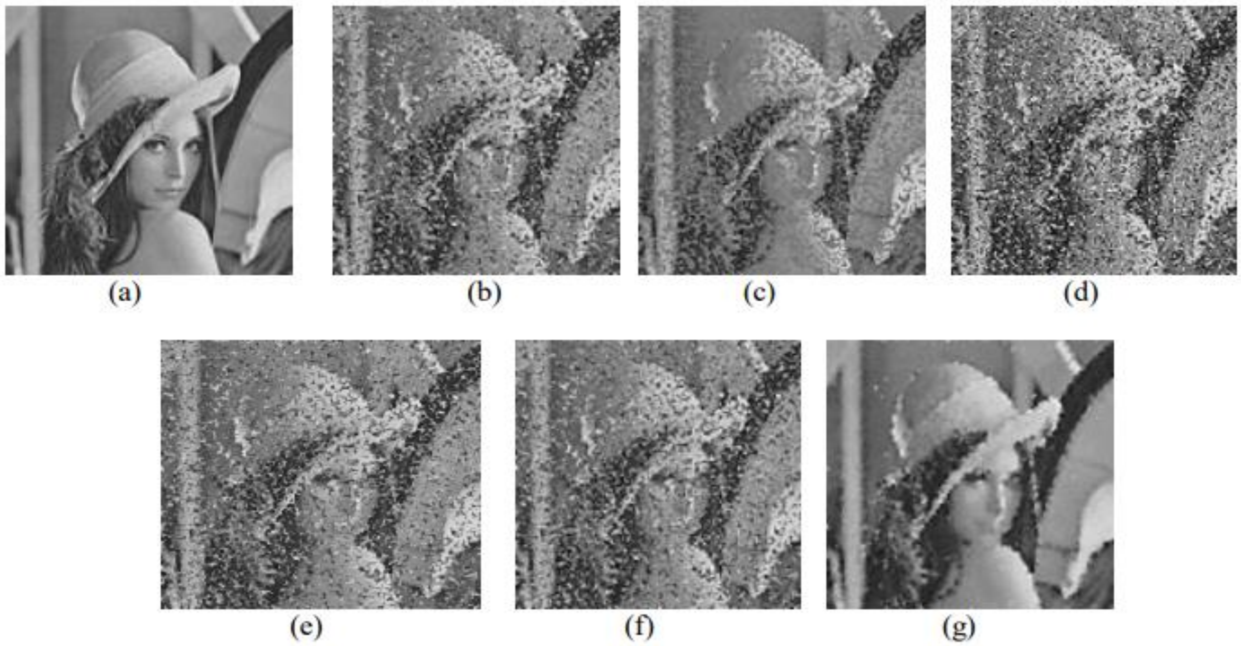

Figure-2: Qualitative results of Lena Image at noise density 90% for different algorithm (a) Original image (b) MDBUTMF [12] (c) MDBUMF_GM [18] (d) EMDBUTMF [19] (e) DBANMF [33] (f) DBUTWMF [32] (g) Proposed algorithm.

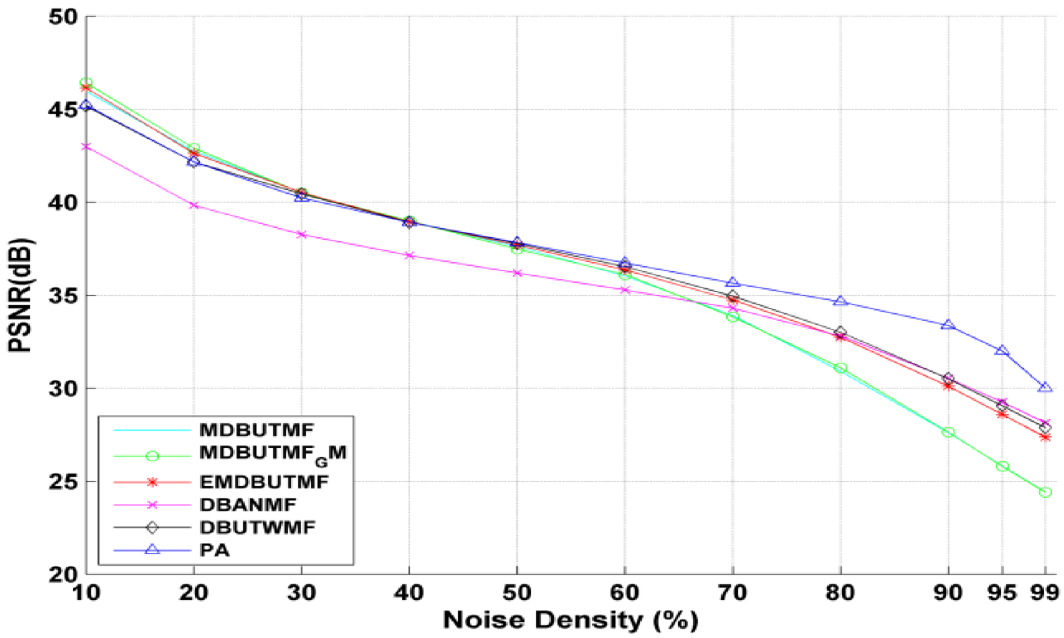

Figure-3: Graphical representation of PSNR (dB) versus Noise Density (%)

The simulation is done on Intel i3 processor 4005 with operating frequency 1.70GHz and 4GB RAM capability. Table 1 represents the comparative analysis of proposed algorithm tested on Lena picture for different noise densities and obtained values of PSNR, IEF and SSIM. Figure 2 depicts the pictorial representation of proposed algorithm against the existing non-linear filters for Lena picture at 90% noise density. From the graphical representation of PSNR for 'Lena' picture at variable noise levels as shown in Figure 3 it has been proved that the proposed algorithm gives its best performance at high noise density.

Table-1: PSNR, IEF and SSIM values on comparison with different noise removal algorithm at 10%-99% levels for "Lena" image.

Noise in %	Performance Metrics	MDBUTMF	MDBUTMF_GM	EMDBUTMF	DBANMF	DBUTWMF	PA
10	PSNR	46.1444	**46.1445**	46.1455	42.9818	45.2038	45.2038
	IEF	**268.8181**	268.7884	268.7884	72.1933	204.4259	204.4259
	SSIM	**0.9915**	0.9915	0.9915	0.9746	0.9900	0.9900
20	PSNR	42.6186	**42.6229**	42.6229	39.8290	42.1529	42.1529
	IEF	234.0061	**234.3253**	234.3253	64.8487	206.4870	206.4870
	SSIM	**0.9813**	0.9813	0.9813	0.9485	0.9800	0.9800
30	PSNR	40.5057	**40.5099**	40.5069	38.2555	40.2179	40.2184
	IEF	207.9888	**208.1232**	207.2444	74.4534	194.4167	194.4218
	SSIM	**0.9685**	0.9684	0.9682	0.9270	0.9674	0.9674
40	PSNR	38.9345	38.9344	**38.9385**	37.1208	38.9051	38.9098
	IEF	161.9051	136.1395	154.8807	73.8274	170.0109	**171.9323**
	SSIM	0.9487	0.9489	0.9471	0.9031	0.9508	**0.9520**
50	PSNR	37.6721	37.6667	37.6811	36.1774	37.7352	**37.8246**
	IEF	135.4539	141.0073	104.1682	70.4751	143.6788	**159.7256**
	SSIM	0.9198	0.9229	0.9025	0.8693	0.9232	**0.9353**
60	PSNR	36.3769	36.3750	36.3527	35.2658	36.4839	**36.7277**
	IEF	86.4293	103.5041	43.0704	59.9427	92.9984	**142.8235**
	SSIM	0.8475	0.8650	0.7720	0.8045	0.8541	**0.9130**
70	PSNR	34.9399	34.8080	34.7360	34.2974	34.9970	**35.6396**
	IEF	47.8988	65.9726	14.9242	40.2021	49.0636	**114.2092**
	SSIM	0.6852	0.7458	0.4914	0.6530	0.6883	**0.8768**
80	PSNR	33.0977	32.9209	32.7299	32.7972	33.1010	**34.6372**
	IEF	21.3878	34.1227	5.6899	20.0699	21.5227	**83.5829**
	SSIM	0.4440	0.5690	0.2189	0.4197	0.4455	**0.8297**
90	PSNR	30.5779	30.3324	30.0945	30.5207	30.5762	**33.3653**
	IEF	9.9122	16.5189	2.3015	9.5785	9.9176	**55.5146**
	SSIM	0.2149	0.3813	0.0622	0.2025	0.2149	**0.7533**
95	PSNR	29.2475	28.8630	28.5822	29.2454	29.2464	**31.9834**
	IEF	6.5102	11.3477	1.4868	6.3616	6.5107	**35.1553**
	SSIM	0.1162	0.3150	0.0263	0.1086	0.1162	**0.6630**
99	PSNR	28.1047	28.1713	27.3605	28.1335	28.1046	**29.9946**
	IEF	4.8763	8.6032	1.0785	4.7547	4.8763	**11.1820**
	SSIM	0.0550	0.3756	0.0072	0.0497	0.0550	**0.4679**

VI. CONCLUSIONS AND FUTURE SCOPE

In this paper, a novel two phase scheme technique for removal of high density SPN has been proposed. The proposed technique is tested for very high noise density SPN for different performance metrics and compared the results with the literature and it has been observed that proposed algorithm is worthy for removal of high density SPN. As a future scope, more work can be done for removal of high density SPN from videos and different advanced techniques can be applied to preserve the fine details of the edges.

REFERENCES

[1] A. C. Bovik, Handbook of Image and Video Processing (Communication, Networking and Multimedia), New York: Academic Press, 2005.

[2] R. H. Chan, C. W. Ho and M. Nikolova, "Salt and pepper noise removal by median type noise detectors and detail preserving regularization," IEEE Transactions on Image Processing, vol. 14, no. 10, pp. 1479-1485, Oct. 2005.

[3] K. Aishwarya, V. Jayaraj and D. Ebenezed, "A new and efficient algorithm for removal of high density salt and pepper noise in images and videos," in Second International Conference Computer Modeling and Simulation, pp.409-413, 2010.

[4] T. Veerakumar, S. Esakkirajan, I. Vennila, "An approach to minimize very high density salt and pepper noise through trimmed global mean," International Journal of Computer Applications, vol. 39, no. 12, Febuary 2012.

[5] K. Vasant, T. G. Manjunath and S. N. Raj, "A decision based unsymmetric trimmed modified winsorized mean filter for the removal of high density salt and pepper noise in images and and videos," in Eleventh International Multi-Conference on Information Processing, pp. 595-604, 2015.

[6] T. S. Huang, G. J. Yang and G. Y. Tang, "Fast two dimensional median filtering algorithm," IEEE Transactions Acoustics, Speech, Signal Processing, vol. ASSP-1, no. 1, pp. 13-18, January 1979.

[7] T. A. Nodes and N. C. Gallagher, "The output distribution of median type filters ," IEEE Transactions Communication, vol. COM-32, no. 5, pp. 532-541, May 1984.

[8] H. Hwang and R. A. Haddad, "Adaptive median filters: New algorithms and results," IEEE Transactions Image Processing, vol. 4, no. 4, pp. 499-502, April 1995.

[9] P. E. Ng and K. K. Ma, "A switching median filter with boundary discriminative noise detection for extremely corrupted images," IEEE Transactions Image Processing, vol. 15, no. 6, pp. 1506- 1516, June 2006.

[10] K. S. Srinivasan and D. Ebenezer, "A new fast and efficient decision based algorithm for removal of high density impulse noise," IEEE Signal Processing Letters, vol. 14, pp. 189-192, March 2007.

[11] V. Jayaraj and D. Ebenezer, "A switching- based median filtering scheme and algorithm for removal of high density salt and pepper noise in image," EURASIP J. Advance Signal Processing, 2020.

[12] S. Esakkirajan, T. Veerakumar, A. N. Subarmanyam and C. H. PremChand, "Removal of high density salt and pepper noise through modified decision based unsymmetric trimmed median filter," IEEE Signal Processing Letter, vol. 18, no. 5, pp. 287-290, May 2021.

[13] E. J. Leavline and D. A. A. G. Singh, "Enhanced modified decision based unsymmetric trimmed mean filter for salt and pepper noise removal," International Journal of Imaging and Robotics, vol. 11, no. 3, pp. 46-56, 2019.

[14] M. T. Raza and S. Sawant, "High density salt and pepper noise removal through decision based partial trimmed global mean filter," Nirma University International Conference on Engineering (NUiCONE), pp. 1-5, 2021.

[15] A. K. Samantaray and P. Mallick, "Decision based adaptive neighborhood median filter," Procedia Computer Science, vol. 48, pp. 222-227, 2020.