International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:2682
Vol.5, No.7, July 2016 DOI:10.15693/ijaist/2016.v5i7.118-128

Impulse Noise Removal Using Enhanced Efficient Leading diagonal Sorting Algorithm

R.Rajkumar
Research Scholar, ECS,
Karpagam University, India

Dr.S.Saira Banu
AssociateProfessor,ECS,
Karpagam University,India

Dr. K.G.Padmasine
Asst. Professor, ECS,
Karpagam University, India

***Abstract*—The paper aims to make the perfect noise free images. The images which are captured by the digital camera and the noise presented in the image are affect the degradation in the quality of images. At present various techniques are used to remove the noise from the images. But the output images are blurred due to the loss of true pixel values present in the images. While removing the noise, the true pixel values are erased by the old techniques. The new enhanced efficient algorithm proposes a new solution for the degradation of the quality images. This technique removes the noise with the better quality and manages the loss of true pixel values.**

***Index Terms* –noise free images, digital camera, blurred, degradation, pixel values.**

I. Introduction

The noise is affected in the images during transmission and acquiring images from cameras. Electrical interferences and unwanted signals and blurred image while capturing, environmental conditions and sampling and quantization error could be considered as noise. The main task for image processing is removing the noise from the captured image. Visual interpretation of image is degraded by salt and pepper. The noise pixels are described by maximum and minimum value. '255' is the maximum and '0' is the minimum value of 8 bit pixel. The noises are represented by two categories; they are linear and Non-Linear. In conventional noise removal algorithms are repeated replacement of neighboring pixel produces streaking effect. To minimize the streaking effect by Modified Decision Based Unsymmetric trimmed Median Filter (MDBUTMF) (1). In the drawback of EDBA, IDBA and FIDRM filters are leave jagged edges over the processed image when the noise level is very high. The drawbacks of the above filters are eliminated by edge preserving method (2). Absent of sorting in the Impulse Noise Removal method which increase the speed of operation and better reconstruction of images (3). The Adaptive Weighted Mean Filter is detected impulses and removes noise and to get higher computational efficiency and better detail preservation ability (4). In image processing further improved the application of videos and generated noise is minimized by the spatiotemporal order-static information about the directional samples, the video pixels is classified into noise free and noisy (5). To detect the noise and to remove noise by either median filter or mean filter (6). Leading Diagonal Sorting Algorithm only 3 pixels are involved and hence reduce the complexity of computational and Hardware (7). The algorithm contains three steps of noise detection, refinement, and impulse noise cancellation. It replaces the noisy pixel with estimated ones (8).

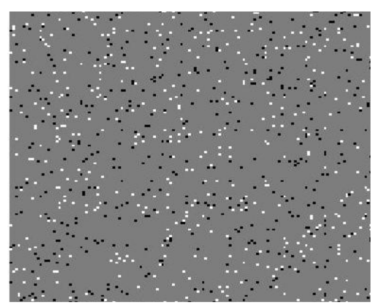


Figure1. Salt and pepper noise

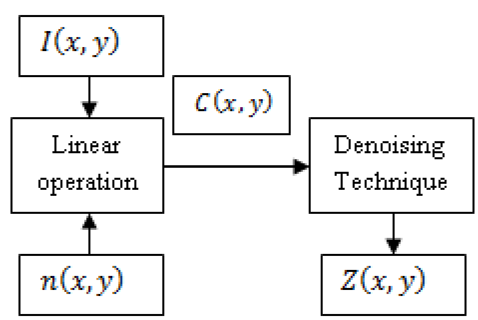


Figure 2. Denoising concept

Figure 2 illustrates the denoising concept; where $n(x, y)$ denotes the percentage of noises added to an image $I(x, y)$. The original image is corrupted due to mixing of noise. $C(x, y)$ is represent as corrupted image.

II. Materials Methods

The proposed sorting algorithm reduces the noise in the given image. Figure 1 shows the flowchart for proposed framework. The steps of the proposed framework and two case (1) and (2) are used for identified and eliminate the noises is given as follows

Case 1: If all the selected element in the sliding window is affected by noise then replace the noisy elements by mean of the selected elements.

Case 2: If any one or two elements are affected by noise then replace the noisy elements by the median value of the selected elements.

Step 1: Select a window size of 3×3 from the given image.

Step 2: If $0 < p_{ij} < 255$ where p_{ij} is the fifth pixel of the sliding window and it is considered as an uncorrupted pixel. If $0 = p_{ij} = 255$, then p_{ij} is a corrupted pixel and it is replaced by median of the given sliding window.

Step 3: Check the left diagonal element of the given sliding window is corrupted by noisy element or non-noisy element in case (1) and (2)

Step 4: Check the right diagonal element of the given sliding window is corrupted by noisy element or non-noisy element in case (1) and (2)

Step 5: If $center\ row\ element\ = 0\ or\ 255$ then it is considered as corrupted pixels then case (1) and (2) are used to eliminate the noise.

Step 6: If $center\ column\ element\ = 0\ or\ 255$ then it is considered as corrupted pixels case (1) and (2) are used to eliminate the noise.

III. Example Illustration For Proposed Algorithm:

Step 1: To check whether fifth pixel is corrupted by noise element 0 or 255. In below sliding window fifth pixel is '255' and is considered as corrupted pixel.

$$\begin{bmatrix} 12 & 142 & 76 \\ 0 & \langle 255 \rangle & 26 \\ 0 & 255 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 12 & 142 & 76 \\ 0 & (26) & 26 \\ 0 & 12 & 0 \end{bmatrix}$$

Then the processing pixel is replaced by median of the sliding window. Here the median value is 26 and is replaced the value 255.

Step 2: To check whether left diagonal element is corrupted by noise element 0 or 255.

$$\begin{bmatrix} (12) & 142 & 76 \\ 0 & (26) & 26 \\ 0 & 255 & (0) \end{bmatrix} \rightarrow \begin{bmatrix} (12) & 142 & 76 \\ 0 & (12) & 26 \\ 0 & 12 & (12) \end{bmatrix}$$

Left diagonal elements are 0, 12 and 26 is represented in the above sliding window. Where '0' is the corrupted pixel and it is replaced by median of the diagonal elements it is given in case 2. Here the median value is 12 and is replaced the value 0.

Step 3: To check whether right diagonal element is corrupted by noise element 0 or 255.

$$\begin{bmatrix} 12 & 142 & (76) \\ 0 & (12) & 26 \\ (0) & 255 & 12 \end{bmatrix} \rightarrow \begin{bmatrix} 12 & 142 & (76) \\ 0 & (12) & 26 \\ (12) & 12 & 12 \end{bmatrix}$$

Right diagonal elements are 0, 12 and 76 is represented in the above sliding window. Where '0' is the corrupted pixel and it is replaced by median of the diagonal elements it is given in case 2. Here the median value is 12 and is replaced the value 0.

Step 4: To check center row elements are corrupted by noise 0 or 255.

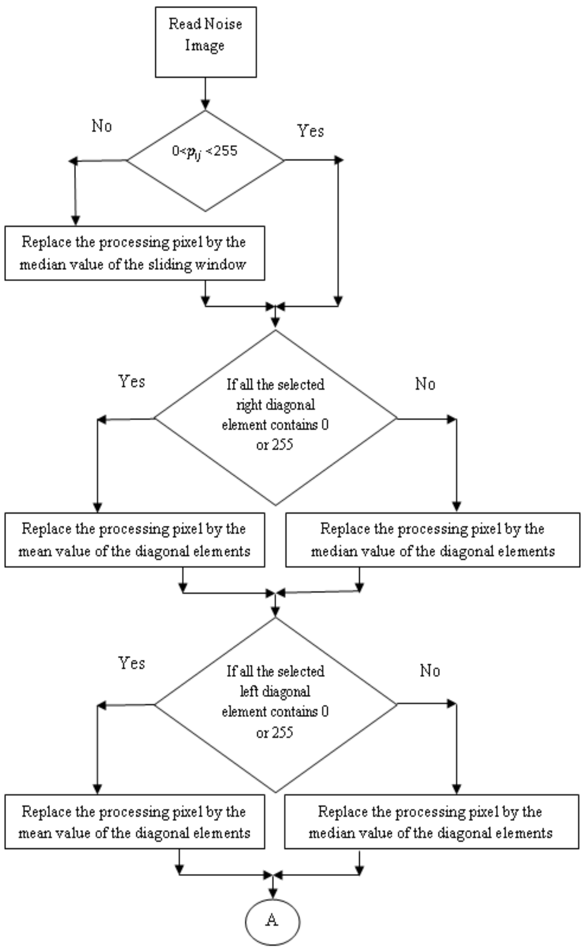
$$\begin{bmatrix} 12 & (142) & 76 \\ 0 & (12) & 26 \\ 12 & (255) & 12 \end{bmatrix} \rightarrow \begin{bmatrix} 12 & (142) & 76 \\ 0 & (12) & 26 \\ 12 & (142) & 12 \end{bmatrix}$$

Center row elements are 12, 142 and 255 is represented in the above sliding window. Where '255' is the corrupted pixel and it is replaced by median of the diagonal elements it is given in case 2. Here the median value is 142 and is replaced the value 255.

Step 5: To check center column elements are corrupted by noise 0 or 255

$$\begin{bmatrix} 12 & 142 & 76 \\ (0) & (12) & (26) \\ 12 & 142 & 12 \end{bmatrix} \rightarrow \begin{bmatrix} 12 & 142 & 76 \\ (12) & (12) & (26) \\ 12 & 12 & 12 \end{bmatrix}$$

Center row elements are 0, 12 and 26 is represented in the above sliding window. Where '0' is the corrupted pixel and it is replaced by median of the diagonal elements it is given in case 2. Here the median value is 12 and is replaced the value 0.



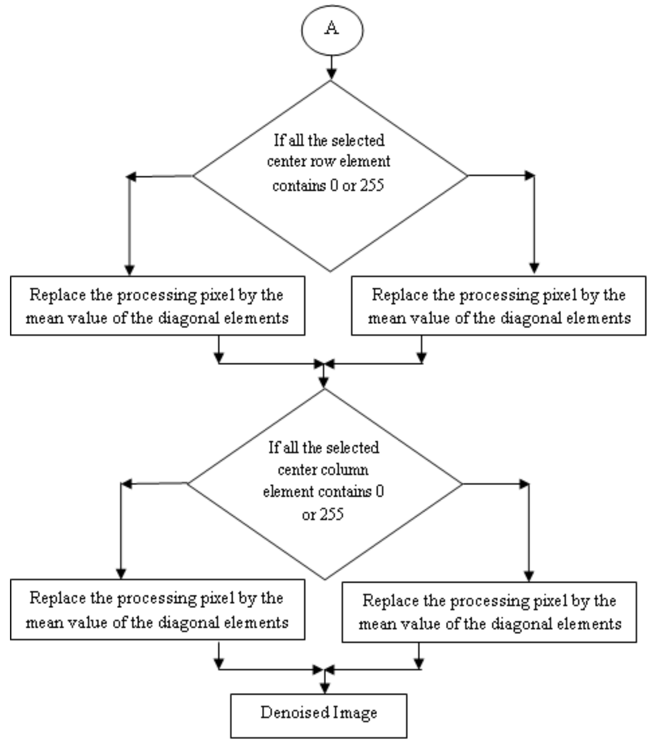


Figure 3. Flow Chart of Proposed Algorithm

I. EXPERIMENTAL RESULTS:

The performance is tested with different gray images such as Lena, Baboon and Lady Image with the size of 256×256. The simulation is corrupted by salt and pepper noise with equal probability of noise level. The noise density level is varied from 10% to 90% with the increment of 10%. The restoration results are quantitatively measured by Peak Signal-to-Noise Ratio (PSNR) and Mean Square Error (MSE).

Mean Square Error (MSE): Mean square error is defined by below equation

$$\mathrm{MSE} = \frac{1}{xy} \sum_{i=0}^{x-1} \sum_{j=0}^{y-1} \left(I_{ij} - D_{ij}\right)^2$$

Smaller the MSE better the result.

Peak Signal-to-Noise Ratio (PSNR): PSNR is defined by below equation:

$$\mathrm{PSNR} = 10 \times \log_{10} \frac{255^2}{\mathrm{MSE}}$$

PSNR is opposite to MSE, higher the PSNR obtained the better result.

Table 1: Comparison of PSNR and MSE value between Existing and Proposed Algorithm for Lena Image

Noise density	TMF algorithm (Existing)		LDS algorithm (Existing)		EELDS algorithm (Proposed)	
	PSNR	MSE	PSNR	MSE	PSNR	MSE
10% noise	15.14812	2002.938	15.23513	1963.212	15.43369	1875.475
20% noise	12.10082	4040.153	12.13583	4007.714	12.32368	3838.06
30% noise	10.35051	6045.454	10.22646	6220.625	10.61212	5692.046
40 %noise	9.10075	8061.298	8.824653	8590.425	9.351546	7608.96
50% noise	8.138781	10060.09	7.614107	11351.92	8.428818	9410.184
60% noise	7.42298	11862.66	6.534521	14555.52	7.71955	11079.62
70% noise	6.852175	13528.9	5.577583	18143.53	7.209461	12460.45
80% noise	6.317526	15301.26	4.717221	22118.61	6.872044	13467.14
90% noise	6.030733	16345.82	3.887578	26774.55	6.708181	13984.98

Table 2: Comparison of PSNR and MSE value between Existing and Proposed Algorithm for Baboon Image

Noise density	TMF algorithm (Existing)		LDS algorithm (Existing)		EELDS algorithm (Proposed)	
	PSNR	MSE	PSNR	MSE	PSNR	MSE
10%noise	6.030733	16345.82	15.52711	1835.56	16.27111	1546.568
20%noise	12.01275	4122.92	12.40292	3768.665	13.21111	3128.737
30%noise	10.47031	5880.972	10.55865	5762.55	10.0383	6496.059
40%noise	9.334868	7638.236	9.080487	8098.997	10.0383	6496.059
50%noise	8.440099	9385.773	7.866093	10712	8.992137	8265.445
60%noise	7.756941	10984.64	6.760758	13816.69	8.199197	9921.11
70%noise	7.109296	12751.18	5.783749	17302.35	7.58921	11417.18
80%noise	6.586906	14381.01	4.769251	21855.2	7.118788	12723.34
90%noise	6.134398	15960.27	3.947564	26407.27	6.831572	13593.23

Table 3: Comparison of PSNR and MSE value between Existing and Proposed Algorithm for Lady Image

Noise density	TMF algorithm (Existing)		LDS algorithm (Existing)		EELDS algorithm (Proposed)	
	PSNR	MSE	PSNR	MSE	PSNR	MSE
10%noise	14.90214	2119.658	15.32875	1921.343	15.7364	1749.201
20%noise	12.08949	4050.712	12.22874	3922.884	12.64851	3561.468
30%noise	10.41955	5950.11	10.28688	6134.684	10.84992	5388.754
40%noise	9.20538	7869.407	8.862775	8515.349	9.594293	7195.329
50%noise	8.261825	9779.067	7.69912	11131.87	8.651954	8938.91
60%noise	7.549623	11521.73	6.599257	14340.17	7.888884	10655.93
70%noise	6.967839	13173.34	5.594114	18074.59	7.321745	12142.43
80%noise	6.484549	14723.97	4.74086	21998.54	6.94631	13238.81
90%noise	5.964135	16598.41	4.74086	21998.54	6.759648	13820.23

Table 4: Comparison of PSNR and MSE value between Existing and Proposed Algorithm for Cameraman Image

Noise density	TMF algorithm (Existing)		LDS algorithm (Existing)		EELDS algorithm (Proposed)	
	PSNR	MSE	PSNR	MSE	PSNR	MSE
10% noise	14.71507	2212.955	15.16635	1994.552	15.43481	1874.991
20% noise	11.89414	4237.07	12.05731	4080.835	12.41021	3762.35
30% noise	10.15209	6328.061	10.21571	6236.047	10.61092	5693.606
40 % noise	8.932543	8379.645	8.680352	8880.649	9.402842	7519.617
50% noise	7.987673	10416.28	7.561494	11490.28	8.411472	9447.845
60% noise	7.209184	12461.25	6.511505	14632.87	7.690992	11152.72
70% noise	6.650283	14172.67	5.629496	17927.94	7.150405	12631.05
80% noise	6.101103	16083.09	4.668416	22368.58	6.842593	13558.78
90% noise	5.899137	16848.7	3.890816	26754.59	6.698291	14016.86

Table 5: Comparison of PSNR and MSE value between Existing and Proposed Algorithm for Gold hill Image

Noise density	TMF algorithm (Existing)		LDS algorithm (Existing)		EELDS algorithm (Proposed)	
	PSNR	MSE	PSNR	MSE	PSNR	MSE
10% noise	15.25554	1954.008	15.42589	1878.844	15.64488	1786.456
20% noise	12.3173	3843.707	12.29424	3864.167	12.63618	3571.596
30% noise	10.58042	5733.738	10.41326	5958.738	10.84741	5391.87
40 % noise	9.339063	7630.862	8.947825	8350.211	9.556657	7257.954
50% noise	8.370327	9537.779	7.719895	11078.74	8.662309	8917.621
60% noise	7.602852	11381.37	6.658973	14144.34	7.934613	10544.32
70% noise	6.955055	13212.18	5.677785	17729.7	7.382207	11974.55
80% noise	6.396522	15025.46	4.749969	21952.45	6.988727	13110.14
90% noise	6.028001	16356.1	3.900508	26694.95	6.783801	13743.58

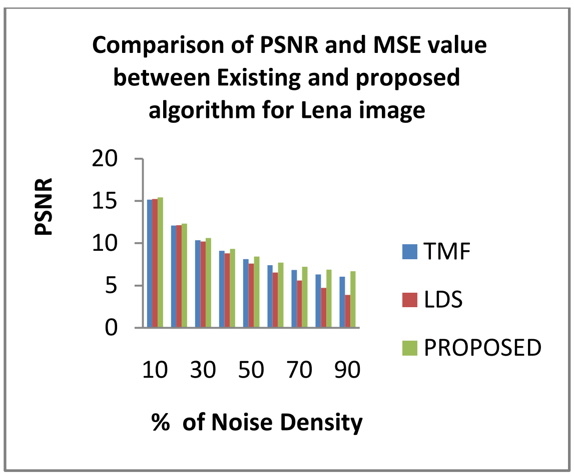


Figure 4.Comparison of PSNR value for Lena Image

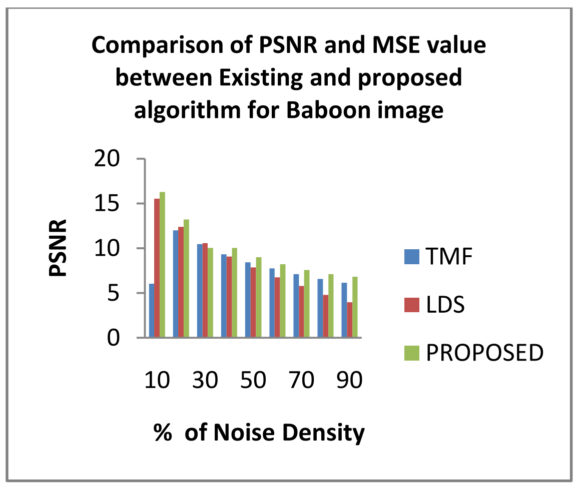


Figure 5. Comparison of PSNR value for Baboon Image

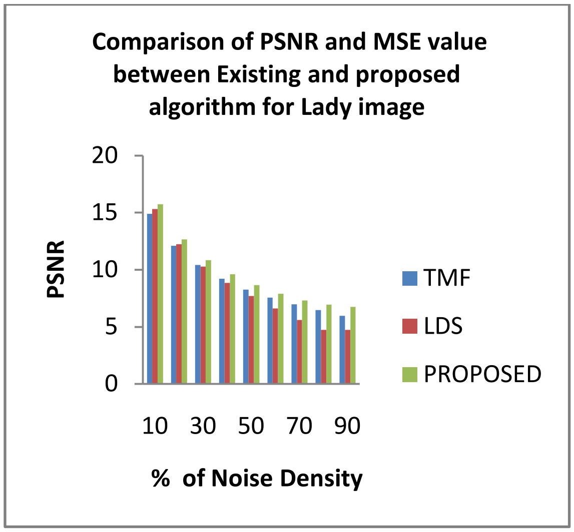


Figure 6. Comparison of PSNR value for Lady Image

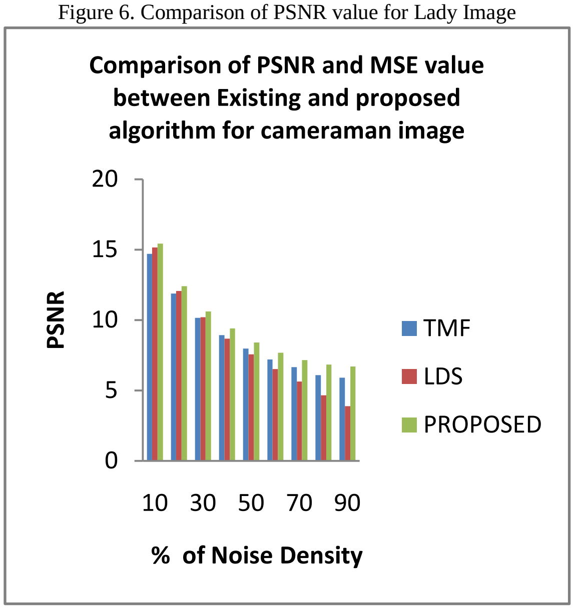


Figure 7. Comparison of PSNR value for Cameraman Image

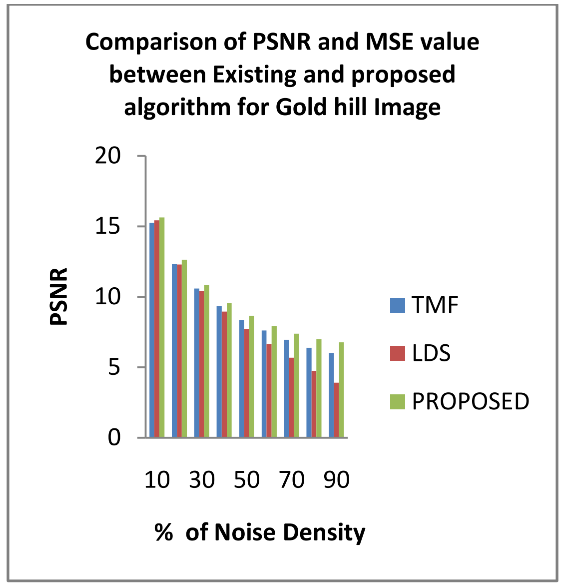


Figure 8. Comparison of PSNR value for Gold hill Image

The PSNR and MSE are evaluated for simulation and comparison of performance of TMF, LDS & EELDS algorithm is presented in the table 1-5. It is observed that the proposed algorithm can provide better performance in terms of PSNR and MSE. For every noise level proposed EELDS provides better achievement than TMF and Existing LDS algorithm. Fig 4-8 shows the visual qualities of filtered images using TMF, ELDS and EELDS algorithm for the images “Lena” “Baboon” “Lady” “Cameraman” “Gold Hill” corrupted by impulse noise level by 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80% and 90%. It can be seen that for different noise level proposed EELDS suppresses the impulses and also preserves the edges than the TMF & LDS algorithms. TMF algorithm suppresses the impulse noise, but it produced blur effect. The proposed EELDS algorithm achieves better PSNR value compared to TMF and LDS algorithms and preserving the image quality for a wide range of noise density.

Fig 8-12 illustrates the comparison of PSNR value for Lena, baboon, Lady Image, Cameraman and Gold Hill. From the graph, it is observed that proposed EELDS algorithm provides better PSNR value in image quality than TMF and LDS algorithm. These results reveal that the proposed algorithm exhibits better visual quality.

Figure 9. Original images
(a) Lena image (b) baboon image (c) Lady image (d) Camera man image (e) Gold hill image

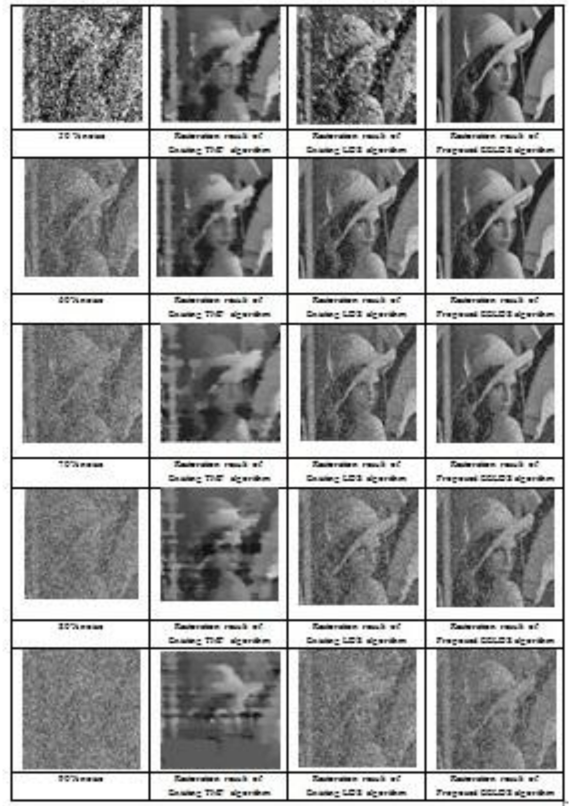
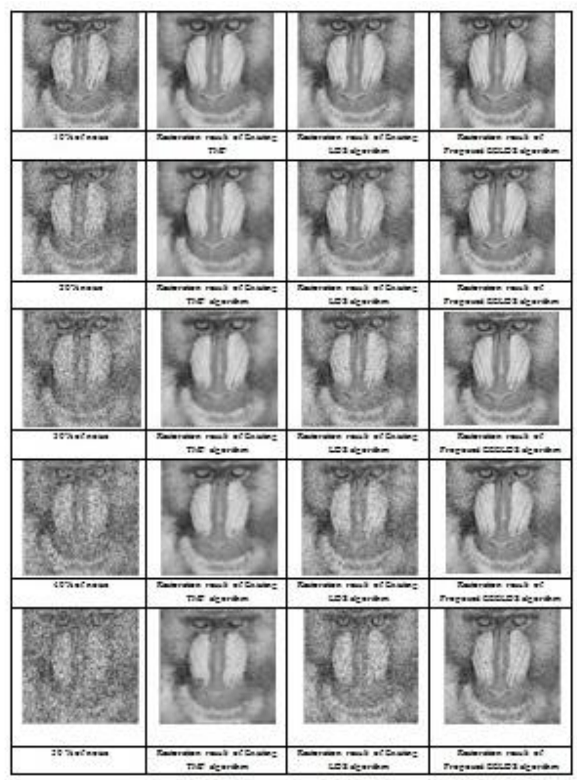


Figure 10. Image restoration result of the Lena Image



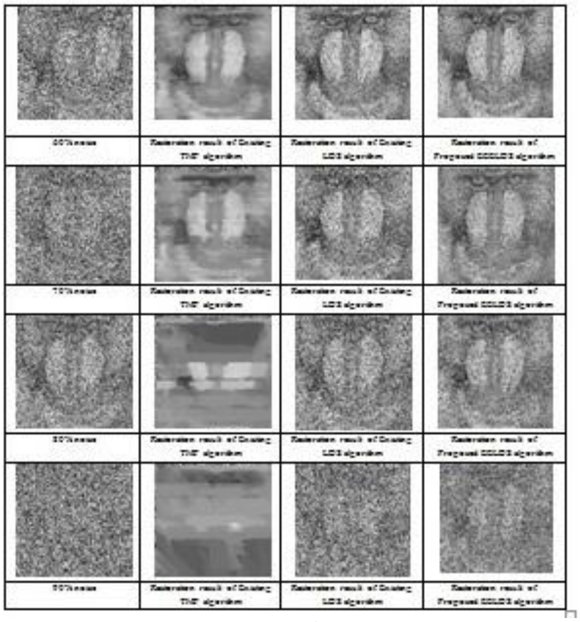


Figure 11. Image restoration result of the Baboon Image

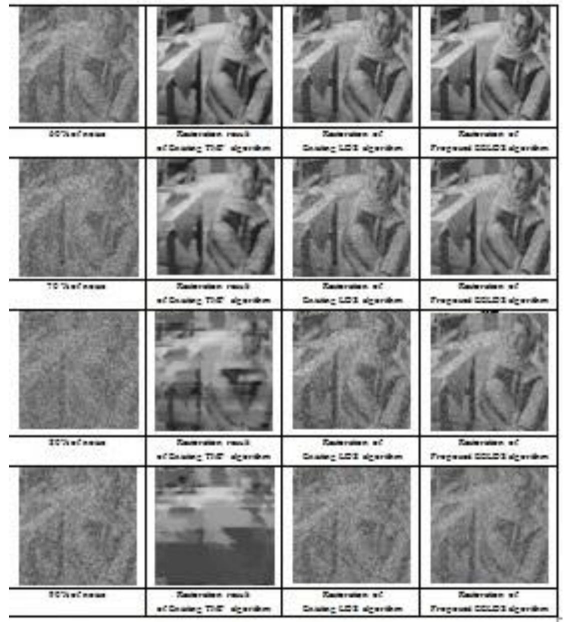
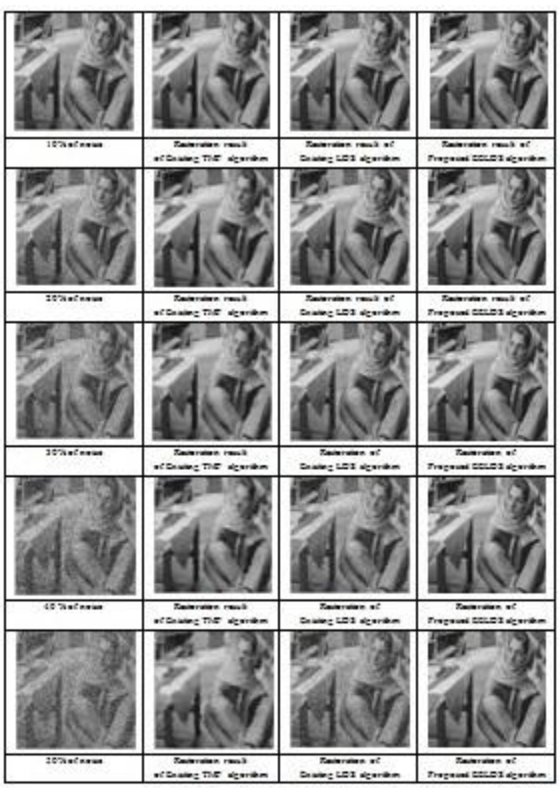


Figure 12: Image restoration result of the Lady Image



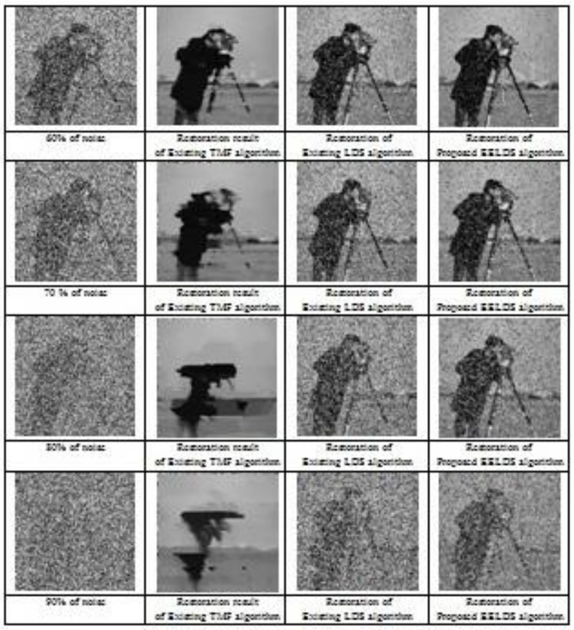


Figure 13. Image restoration result of the Cameraman Image

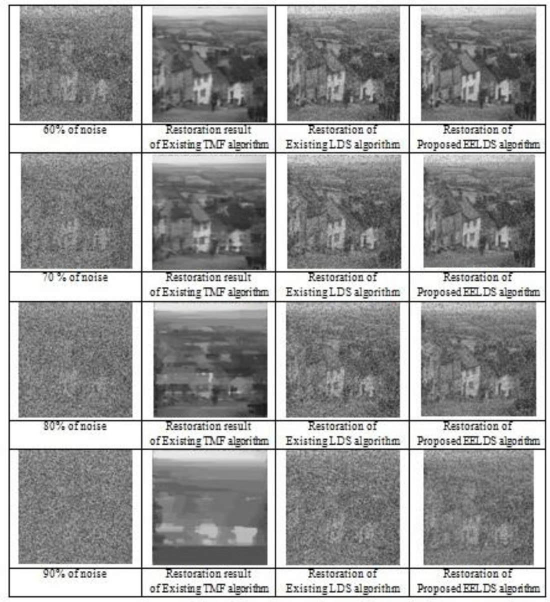


Figure 14. Image restoration result of the Gold Hill Image

Conclusion: In this paper, an Enhanced Efficient Leading Diagonal Sorting (EELDS) algorithm is proposed for the impulse noise detection and filtering process of the noise image in an efficient manner. The perform metric of PSNR and MSE values are compared with the different sampled images. It shows the proposed algorithm of EELDS provides better quality; with the high performance and preserve the edges and also more appreciable information in the image data which are captured by the human eye than the existing algorithms in the noise removal (TMF and LDS). The proposed algorithm gives better and efficient performance in the noise density increased as the density increases from 10% to 90% and in every stage, the noise density increases in 10%.

REFERENCE

1. S Esakkirajan and T Veerakumar, "Removal of High Density Salt and Pepper Noise Through Modified Decision Based Unsymmetric Trimmed Medain Filter," I E E E Signal Processing Letters, vol. 18, no. 5, May 2011
2. Jian Wu and Chen Tang, "An Efficient decision based and edge preserving method for salt and pepper noise removal," ELSEVIER, no. 32, 2011.
3. A Rajamani and V Krishnaveni, "Impulse Denoising Algorithm for Gray and RGB Image," IJCA, Vol 70, no. 2, May 2013.

4. Xuming Zhang and Youlun Xiong, Impulse Noise Removal Using Directional Difference Based Noise Detector and Adaptive Weighted Mean Filter, IEEE Signal Processing Letters, Vol.16, No.4, April 2009.
5. Lianghai Jin, Hong Liu , Xiangyang Xu , Enmin Song ,Quaternion-Based Impulse Noise Removal from Color Video Sequences, IEEE Transactions on Circuits and Systems for Video Technology, Vol.23, No.5, May 2013.
6. Vijaykumar V.R, G. Santhana Mari, D. Ebenezer, Fast Switching Based Median–Mean Filter for High Density Salt and Pepper Noise Removal, International Journal of Electronics and Communications 68(12), June 2014
7. A. Rajamani, V. Krishnaveni, An Efficient Denoising Algorithm for Impulse Noise Removal, Journal of Computer Science 01/2015, 11(1)
8. Wenbin Luo, An efficient Detail-Preserving approach for Removing Impulse Noise in Images, IEEE Signal Processing Letters, Vol.13, No.13, April 2006

Authors Profile

R. Rajkumar received the M.Phil degree in electronics from the Karpagam University. Currently doing Ph.D in electronics in Karpagam University, Coimbatore, India. His research interest includes Image processing, PLC, Embedded Systems.

Dr.S.SairaBanu received the Ph.D degree in Electronics and communication systems, Bharathiar University, Coimbatore,India. She is Currently an Associate Professor in dept. of electronics and communication systems in Karpagam University, Coimbatore, India. His research interest includes wireless communication (WiFi,WiMax),Sensor Networks ,Neural Networks and fuzzy logic, Communication networks, Image Processing.

Dr.K.G.Padmasine received the Ph.D degree in Electronics and Instrumentation, Bharathiar University, Coimbatore, India. She is currently an Assistance Professor in dept. of electronics and communication systems in Karpagam University, Coimbatore, India. His research interest includes wireless communication (WiFi,WiMax), Mobile Ad hoc networks ,Sensor Networks ,MEMS, Embedded System.