iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 **Issue:** II **Month of publication:** February 2020

DOI: http://doi.org/10.22214/ijraset.2020.2100

Low Complexity Random Noise Denoising Method for Medical Image Analysis

Dr. G Vimala Kumari[1], Prof. G Sasi Bhushana Rao[2], Prof. B. Prabhakara Rao[3]

[1]*Assistant Professor, ECE, MVGR College of Engineering, Vizianagaram*
[2]*Professor, Department of Electronics & Communication Engineering, Andhra University College of Engineering, Visakhapatnam, Andhra Pradesh, India.*
[3]*Programme Director, School of Nanotechnology, JNTU, Kakinada, Andhra Pradesh, India.*

Abstract: The medical images are often corrupted by noise during acquisition and transmission due to patient movement, inaccurate instrumental setup and surrounding noise. The noise usually reduces the visual quality of the medical images that complicates diagnosis and treatment. Hence, the need for an efficient denoising method has led to extensive research and development of various innovative methods to remove the random valued impulse noise. For this, a method which detects and filters random valued impulse noise in medical images is employed. The method proposed in this paper uses a decision tree based impulse detector and an edge preserving filter to reconstruct noise free images. The method requires less storage space due to its lower complexity and is more efficient than the existing techniques. Different Magnetic Resonance Imaging (MRI) images are tested by using the algorithm and it gave better Peak Signal to Noise Ratio (PSNR) than the other lower complexity techniques.
Keywords: Impulse noise; Salt and Pepper noise; Random valued Impulse noise; Effective noise Removal; Decision tree; Edge preserving filter

I. INTRODUCTION

This With the advent of digital images, productivity in science and technology has greatly increased due to better understanding capabilities, analysis, visualization, and interpretation. In today's world digital images and their processing are not only limited to the scientific or defense communities but are used in commercial applications also which include entertainment, advertising, etc., as well as fields such as medical applications, education, infotainment, etc. However, with these advantages came the limitations of using digital images. Prominent among these include removing noise and storage problems. Noise develops in images while the image is acquired and transmitted. Presence of noise in images cripples all the advantages with using digital images as it deteriorates the quality of images making it difficult to analyze, interpret or visualize [1]. This led to the development of effective denoising algorithms which target various types of noises in images. Today, image processing not only concentrates on image interpretation, analysis, and visualization; but also on denoising of images. Thus, denoising is a key part of image processing which involves noise detection followed by its removal. Among the discovered image noises, most arduous are the impulse noises which affect the images during their acquisition and transmission [2-3]. Impulse noises generally involve random occurrence or distribution of noisy pixels over the image. However based on the values of its noisy pixels, these are divided into two categories as fixed value impulse noise and random valued impulse noise, both of which involve random distribution of noisy pixels over the image but have different pixel values [4-5]. Fixed value impulse noise is limited to only two noisy pixel values i.e., salt - 255 and pepper - 0 which are distributed randomly over an image. On the other hand, random valued impulse noise involves noisy pixels of any value within the range [0 to 255] which are distributed randomly over the image [6-7].

It is evident that denoising random valued impulse noise is far more challenging than denoising Salt and Pepper noise due to its random pixel values. Of all the various methods available to denoise impulse noise very few address the random valued impulse noise. Thus, there exists a need for development of an efficient denoising method for removal of random valued impulse noise in images. Mean filters perform denoising effectively, but at the cost of heavy information loss due to blurring or smoothening of the image. Median filters are preferred to preserve several important details even after denoising. Although not as much as mean filters, median filters result in information loss too [8-10]. Hence, to avoid this switching median concept is opted. Here, the median filter is provided with an impulse detector which detects the noisy pixels prior to the filter [11-14]. Thus by detecting the noisy pixels before filtering, the filter may be conditioned such that it filters only those noisy pixels that are detected; thereby resulting in less information loss and better preservation of details in the image which amount to better quality in terms of image characteristics as well as visual perception. Thus for an effective implementation of the technology, an efficient denoising method is required which is not just effective in their performance but are of lower complexity too; enabling to implement it to as many real time applications as possible. The complexity of a denoising method can be determined by three important factors that affect its performance: a) The size of the window used b) The memory required to store the code c) The total number of iterations involved. Keeping in view the above requirements, in this paper, a model which is of low complexity that employs a new impulse detector based on decision tree and a filter to preserve the edge to denoise randomly valued impulse noise in an image.

II. SYSTEM MODEL

Figure. 1 illustrates the total system model for the proposed algorithm. The figure shows how an image goes through various steps and gets reconstructed at the end.

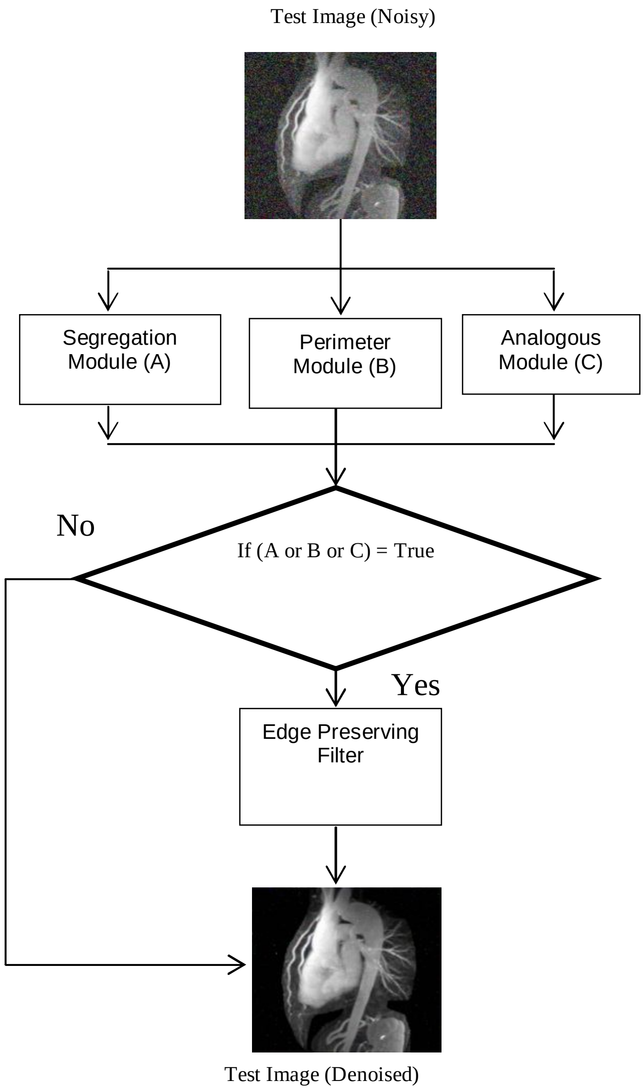


Fig. 1. Flow chart for removing Random Noise

III.METHODOLOGY

A. Forming a Mask

In this method a 3×3 mask is considered using which noise detection and filtering is performed. The mask considered is made up of nine pixels, of which the centre pixel is of the utmost importance as it is only the centre pixel onto which the final filtering is performed. The algorithm is so designed that the mask moves over the entire image, thereby, covering as many pixels as possible. However pixels at the border of the image fail to form masks as they do not possess enough neighbors. To address this problem border correction technique may be used, wherein the border pixels are replaced with their immediate neighbors that fall under the range of filtering. The centre pixel is denoted by 'pi,j', while, its luminance is denoted by 'si,j'. The neighbours are named accordingly in the range of (i-1 : i+1, j-1 : j+1).

B. Components of DTBDM

This method involves two main components viz., impulse detector and edge preserving filter. The impulse detector makes use of a 3×3 mask, to identify the noisy pixels and to activate the filter accordingly.

C. Impulse Detector

The impulse detector detects the corrupted pixels so as to activate the filter only for those pixels that are corrupted, thereby avoiding unnecessary filtering and the resulting information loss. Here, an impulse detector is designed based on a decision tree shown in Fig. 2. A decision tree is a way of simplifying a complex decision by resolving it into multiple simpler decisions, used mainly in multiple variable analyses. This decision is now resolved into three simpler decisions; (i) "Is the pixel Isolated?" (ii) "Is it an edge pixel?" (iii) " How is the similarity of the pixel with its neighbourhood?” Thus, the impulse detector comprises of three modules based on the above decisions viz., Segregation, Perimeter, and Analogous. Each of these modules is independent of each other and can segregate corrupted pixels. However, to increase the efficiency of the detector, all the three modules are used in a parallel fashion, wherein, if any of the three modules detect the pixel to be corrupted it is filtered. The pixel is considered noise-free only if all three modules judge it to be uncorrupted. Several thresholds viz., Th_1, Th_2, Th_3, Th_4, Th_5 and Th_6 are used at various places in the modules. Each of these thresholds possesses a predefined value and is found as a result of experimentation and research. The values of Th_1, Th_2, Th_3, Th_4, Th_5 and Th_6 are 15, 20, 45, 70, 10 and 50 respectively.

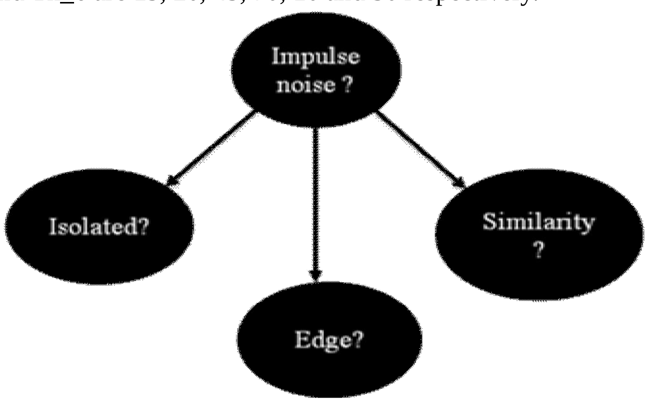


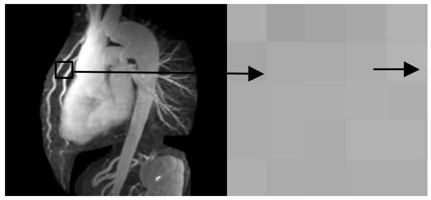
Fig. 2. Decision tree in impulse detector

D. Segregation Module

The segregation module is the first among the three modules, for the fact that it is the easiest and the most obvious way of determining whether a pixel is corrupted or not. Here, detecting the existence of noise is possible because it has a huge difference in intensity when compared with its neighbours. Any isolated pixel can be identified by computing the intensity differences of the pixel with its neighbours. However, this logic is not always valid, because the pixel values in a uniform surface are very close as given in Fig.3. If there are any noisy pixels and edges in this surface, distribution of the gray scale values is different as given in Fig. 4. So segregation logic is based on the prime assumption that the centre pixel is located on a smooth surface. Thus, to test the pixel for segregation, first, it has to be on a uniform surface. Fig. 5 shows an example of corrupted image. In this gray level distribution of uniform surface is significantly different between neighbouring pixels and hence called segregation point. Practically, the impulse module can be easily implemented using the 3×3 mask. The mask is first split into two halves UT-Half and UB-Half. Now, the nine pixels in the mask can be named as 'a, b, c, d, e, f, g, h and si, j', where, si, j" denotes the centre pixel and the variables 'a to h' denote the eight neighbours of the centre pixel. 'UT-Half' comprises of the top four neighbours 'a to d', while, 'UB-Half' comprises of the four neighbours 'e to h', at the bottom of the mask. They are represented as

$$U_{T-Half} = \{a, b, c, d\} \quad (1)$$

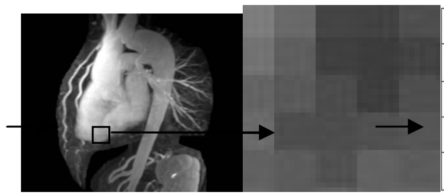
$$U_{B-Half} = \{e, f, g, h\} \quad (2)$$



172	164	164	170	176
166	170	170	174	178
170	173	174	175	176
171	173	172	177	177
175	170	171	174	173

3.a.Original image 3.b.Region in the Square 3.c.Gray Scale values

Fig. 3. Uniform surface in Angio Image



119	96	67	66	75
108	88	64	61	67
98	84	73	66	78
98	78	77	83	86
96	89	85	96	93

4.a.Original image 4.b.Region in the Square 4.c.Gray Scale values

Fig. **4.** Non-uniform surface in Angio Image

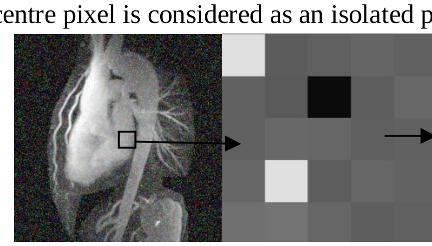
The test for uniformity can be performed by finding out the maximum possible difference in intensities of both the top and bottom halves of the mask. If either of these differences in intensity is very high i.e., greater than a threshold, then, the region is considered as a non-uniform region and the test for Segregation cannot be performed. The test can be represented using the following equations

$$U_{T-Half} = U_{T-Half\max} - U_{T-Half\min} \quad \textbf{(3)}$$

$$U_{B-Half} = U_{B-Half\max} - U_{B-Half\min} \quad (4)$$

$$Decision\ 1 = \begin{cases} true, if\ (U_{T-Halfdif} \geq Th\_1) \\ false, otherwise \end{cases} \quad (5)$$

Decision1 determines whether the centre pixel lies in a uniform region or a non-uniform region. If the region is found to be uniform then proceed to the test for segregation. The centre pixel is now taken into consideration and the intensity difference between the centre pixel and UT-Halfmax, UT-Halfmin, UB-Halfmax, and UB-Halfmin, are computed respectively. If any of these differences is higher than the threshold, then, the centre pixel is considered as an isolated pixel.



195	100	89	99	98
110	85	35	89	103
96	108	94	95	98
112	188	106	99	98
112	125	102	95	96

5.a.Corrupted image 5.b.Region in the Square 5.c.Gray Scale values

Fig. 5. Mask of containing isolated pixels in AngioIimage

Isolated pixel can be represented as

$$I_{T-Ha;f} = \begin{cases} true, if\ (S_{i,j} - U_{T-Half\max} | \geq Th\_2) \\ or (S_{i,j} - U_{T-Half\min} | \geq Th\_2) \\ false, otherwise \end{cases} \quad \textbf{(6)}$$

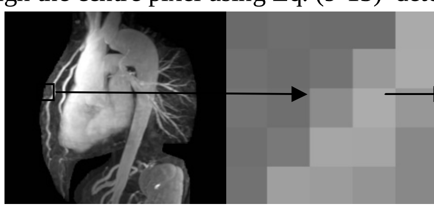
$$I_{B-Ha;f} = \begin{cases} true, if\,(S_{i,j} - U_{B-Half\max} | \geq Th\_2) \\ or(S_{i,j} - U_{B-Half\min} | \geq Th\_2) \\ false, otherwise \end{cases} \tag{7}$$

$$Decision2 = \begin{cases} true, if\,(I_{T-Half} = true) \\ or\,(I_{B-Half} = true) \\ false, otherwise \end{cases} \tag{8}$$

Decision2 determines whether the centre pixel is isolated or not. It is to be noted that Decison2 is arrived at only when Decision1 is false.

E. Perimeter Module

Edges can be regarded as the most important feature of an image as in most of the cases; they tend to give more information than any other feature of an image. The perimeter module in this algorithm is one such effective mechanism that detects the edges and avoids their filtering unnecessarily. Detection of edges in an image and differentiating them from noise is not always easy, which is due to the fact that most of the edge pixels always end up being the high frequency content of an image i.e., they always seem to be isolated when compared to their neighbouring non-edge pixels as shown in Fig. 6. But edge pixels are always of lesser intensity differences computed along the direction of an edge. Since, the only concern is about the centre pixel, arrive at four directions E1 to E4, as shown in Fig. 7 passing through the centre pixel using Eq. (9-13) determine whether the centre pixel is an edge pixel or not.



107	110	108	115	164
112	109	118	150	169
114	113	140	167	154
112	125	160	164	140
118	153	157	150	137

6a.Original image 6b.Region in Square 6c.GrayScale values (edge)

Fig. 6**.** Mask of containing an edge in the Angio Image.

The directional differences are computed as follows

$$F_{E1} = \begin{cases} false\;, if\,(| a - S_{i,j} | \geq Th\_3) \\ or\,(| h - S_{i,j} | \geq Th\_3) \\ or\,(| a - h \;| \geq Th\_4) \\ true\;, otherwise \end{cases} \tag{9}$$

$$F_{E2} = \begin{cases} false\;, if\,(| c - S_{i,j} | \geq Th\_3) \\ or\,(| f - S_{i,j} | \geq Th\_3) \\ or\,(| c - f \;| \geq Th\_4) \\ true\;, otherwise \end{cases} \tag{10}$$

$$F_{E3} = \begin{cases} false\;, if\,(| b - S_{i,j} | \geq Th\_3) \\ or\,(| b - S_{i,j} | \geq Th\_3) \\ or\,(| b - g \;| \geq Th\_4) \\ true\;, otherwise \end{cases} \tag{11}$$

$$F_{E4} = \begin{cases} false \ , if \ (| \ a - S_{i,j} \ | \geq Th\_3) \\ or \ (| \ e - S_{i,j} \ | \geq Th\_3) \\ or \ (| \ d - e \ | \geq Th\_4) \\ true \ , otherwise \end{cases} \quad (12)$$

$$Decision3 = \begin{cases} false, if \ (F_{E1}) or (F_{E2}) \\ or (F_{E3}) or (F_{E4}) istrue \\ true, otherwise \end{cases} \quad (13)$$

As per decision 3, if any of the directional differences FE1 to FE4 is true i.e., shows existence of an edge, then, decision 3 is made false i.e., the centre pixel is uncorrupted, else, the centre pixel is considered as corrupted. Since a parallel logic is used, each module is expected to be independent and self-sufficient, so as to deliver the judgment onto the centre pixel.

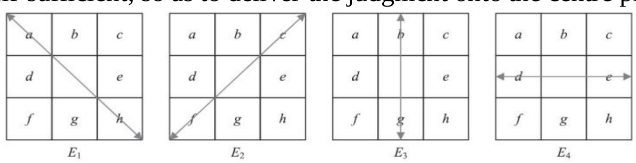


Fig. 7**.** Four directions along which directional differences are calculated

F. Analogous Module

The analogous module basically determines whether the centre pixel belongs to the general range of pixel values within the image or not, thereby, it concludes whether the pixel is noisy or uncorrupted. Thus, only those pixels that have intensity values either greater or lesser than the general intensity range are considered as noise. They always end up at the extreme ends when all the pixels of a mask are arranged in either ascending or descending order. In this module, to identify the noisy pixels first consider two sets of thresholds, out of which only those thresholds that are closer to the general pixel range are chosen. This improves the accuracy of the detection mechanism. To determine the thresholds, first the nine pixels within 3x3 mask is sorted in ascending order. In ascending order the fifth value obtained is the median represented as Kij, the value preceding it is the fourth value represented as Ji,j and the value succeeding it is the sixth value represented as Lij. Now use these values to obtain the thresholds as

$$\begin{aligned} W_{\max} &= J_{i,j} + Th\_5 \\ W_{\min} &= J_{i,j} - Th\_5 \end{aligned} \quad (14)$$

These are the first set of thresholds obtained by adding and subtracting a fixed threshold from the 6th and 4th pixels respectively. Now, the following set of equations give the final set of thresholds

$$T_{\max} = \begin{cases} W_{\max} \ if \ (W_{\max} \leq K_{i,j} + Th\_6) \\ K_{i,j} + Th\_6, otherwise \end{cases} \quad (15)$$

$$T_{\min} = \begin{cases} W_{\min} \ if \ (W_{\max} \geq K_{i,j} - Th\_6) \\ K_{i,j} - Th\_6, otherwise \end{cases} \quad (16)$$

Here, Tmax, Tmin are the final thresholds that are used to test the centre pixel, while, Ki,j +Th_6 and Ki,j -Th_6 are the second set of thresholds as shown in Fig. 8. Only those thresholds that are closer to the general pixel range are chosen here.

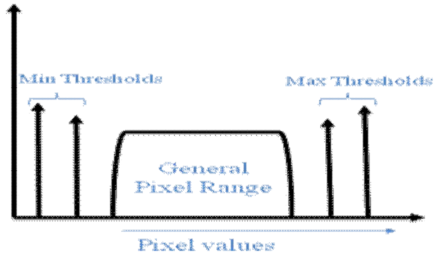


Fig. 8. Explanation of choice of thresholds closer to the pixel range of the image.

Now, the final decision, 'whether the centre pixel is noisy or uncorrupted', is arrived at by comparing the centre pixel value with the two final thresholds Tmax and Tmin. If the centre pixel value lies in between the thresholds Tmax and Tmin, then, it is considered as uncorrupted; else, it is considered as corrupted and is filtered. This decision can be expressed as

$$Decision1 = \begin{cases} true, if\,(S_{i,j} \geq T_{\max}) or (S_{i,j} \leq T_{\min}) \\ false, otherwise \end{cases} \quad \textbf{(17)}$$

G. Edge-Preserving Filter

The basic logic behind the working of the edge preserving filter is that, if there is an edge in any particular direction passing through the centre pixel in a mask or not. Then, replacing the centre pixel with a mean value of only those pixels that are involved in the edge will not break the edge but preserves it. To detect the presence of edge and to locate eight directions are within the 3×3 mask and their corresponding directional differences D1 to D8 as shown in Fig. 9. The direction with the least directional difference is more likely to have an edge. Thus, the pixels that make up that direction are used to find the mean, which in turn is used to replace the centre pixel value. To avoid wrong calculation of directional difference, omit all those directions from D1 to D8 that include a noisy pixel during the calculation of directional differences. To determine which pixels are noisy, the Wmax and Wmin from the analogous module are used. In the case where all the neighbours of the centre pixel are suspected to be noisy, the weighted average of a, b, and c is to be found. The estimated gray scale value of the centre pixel is given by

$$\hat{S}_{i,j} = (a + b \times 2 + c)/4 \quad \textbf{(18)}$$

Here, $\hat{S}_{i,j}$ denotes the estimated value of the center pixel. While, 'a, b and c' denote the upper row neighbors of the centre pixel that are filtered previously.

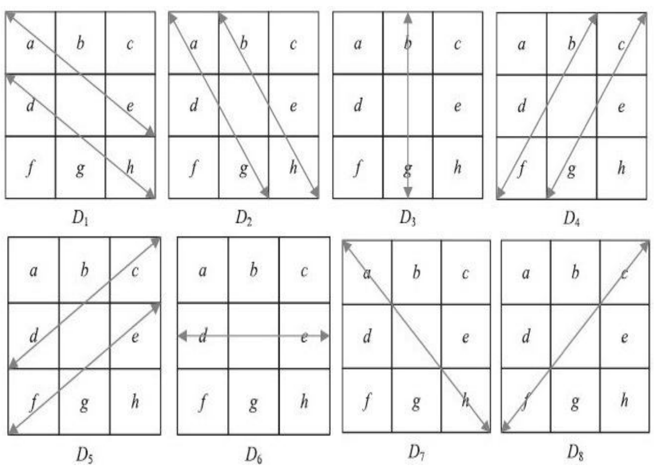


Fig. 9 Eight directions along which directional differences are calculated

Generally, if none of the neighbours were found to be noisy calculate the edge distances using the equations below

$$\begin{aligned} D_1 &= |d-h| + |a-e|, \\ D_2 &= |a-g| + |b-h|, \\ D_3 &= |b-g| \times 2, \\ D_4 &= |b-f| + |c-g|, \\ D_5 &= |c-d| + |e-f|, \\ D_6 &= |d-e| \times 2, \\ D_7 &= |a-h| \times 2, \\ D_8 &= |c-f| \times 2, \end{aligned} \tag{19}$$

Based on which of the directional differences ends up being the Dmin, the estimated gray scale value of the centre pixel is calculated as given under

$$\hat{S}_{i,j} = \begin{cases} (a+d+e+h)/4, ifD_{\min} = D_1 \\ (a+b+g+h)/4, ifD_{\min} = D_2 \\ (b+g)/4, ifD_{\min} = D_3 \\ (b+c+f+g)/4, ifD_{\min} = D_4 \\ (c+d+e+f)/4, ifD_{\min} = D_5 \\ (d+e)/4, ifD_{\min} = D_6 \\ (a+h)/4, ifD_{\min} = D_7 \\ (c+f)/4, ifD_{\min} = D_8 \end{cases} \tag{20}$$

It is to be noted that whenever the median of the pixels b, d, e and g is computed, it always results in a value equal to that of the estimated value. In the case of a wrong detection of an edge or any other causes of error, the estimated value is wrongly calculated. To prevent this, median of the pixels b, d, e & g along with estimated value is computed and use the resultant median to replace the centre pixel value. This not only prevents the error but also acts as a checking mechanism for the proper functioning of the edge preserving filter. The final median value obtained, which replaces the centre pixel, is given by

$$\hat{S}_{i,j} = Median(\hat{S}_{i,j}, b, d, e, g) \tag{21}$$

IV. RESULT AND DISCUSSIONS

Above proposed algorithm has been implemented on three standard medical test images; Angio, Lungs and Abdomen. The algorithm has been implemented using MATLAB R2013a. While simulating, random noise will be added to image to distort it. The density of noise applied on the image vary from 5%-20%. The following are the tabulated results of the method proposed. When compared with techniques presently used the proposed method is better when applied on various medical images using peak signal to noise ratio (PSNR) expressed in decibels as the comparison parameter is given as

$$PSNR = 10\log\frac{255^2}{MSE} \tag{22}$$

Where, MSE stands for mean square error. The obtained values of PSNR for a Angio, Lungs and Abdomen medical images for various noise densities from 5%-20% in the steps of 5% have been tabulated and are shown in Table 1-3 respectively and have been compared and it is evident by the comparison that the proposed method has achieved better results than existing algorithms namely Median, Adaptive Median Filter (AMF) and Decision Tree Based Denoising Method (DTBDM). The bar charts show the performance of the proposed method when compared with existing techniques of low complexity algorithms.

Table I
Comparison Of Restored Angio Image For Various Noise Densities

Method	PSNR(dB) values for Angio image			
	5%	10%	15%	20%
Noisy	21.42	19.67	17.11	15.21
Median	30.74	30.30	29.85	29.30
AMF	33.66	32.59	31.28	30.22
DTBDM	37.26	35.14	33.60	32.21
Proposed	37.84	36.01	34.43	33.04

TABLE III
Comparison of Restored Lungs image for various noise densities

Method	PSNR(dB) values for Lungs image			
	5%	10%	15%	20%
Noisy	21.12	18.06	16.32	15.05
Median	32.66	31.78	31.78	29.90
AMF	36.26	33.01	32.78	30.23
DTBDM	38.63	36.30	34.40	32.73
Proposed	39.46	37.13	35.27	33.59

TABLE IIIII
Comparison of Restored Abdomen image for various noise densities

Method	PSNR(dB) values for Abdomen image			
	5%	10%	15%	20%
Noisy	22.33	19.45	17.74	15.40
Median	33.81	33.04	32.13	31.29
AMF	37.30	34.33	33.19	31.49
DTBDM	39.84	37.18	35.47	33.70
Proposed	40.54	37.76	36.23	34.53

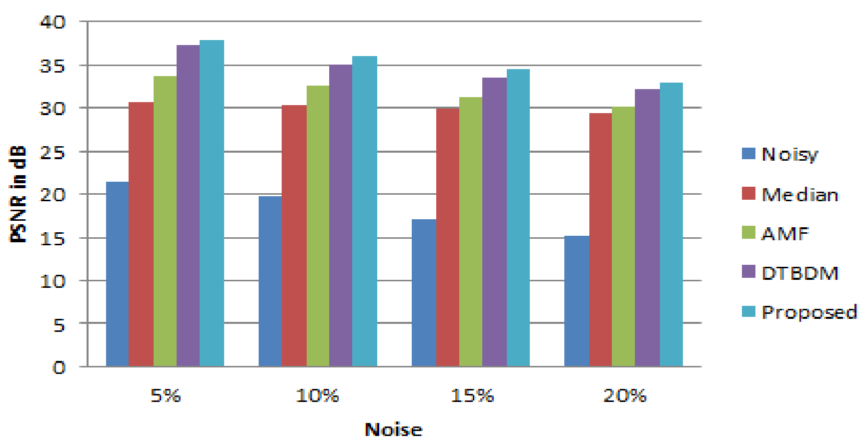


Fig. 10. Comparison of restoration results for Angio image

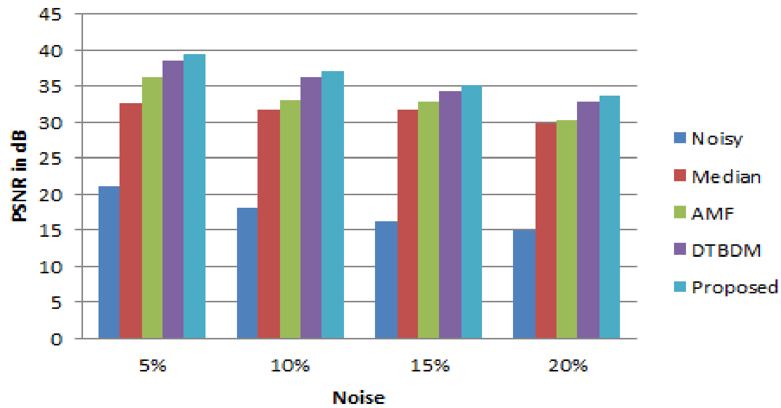


Fig. 11. Comparison of restoration results for Lung image

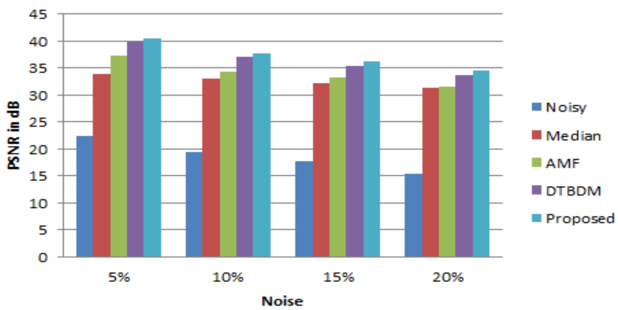
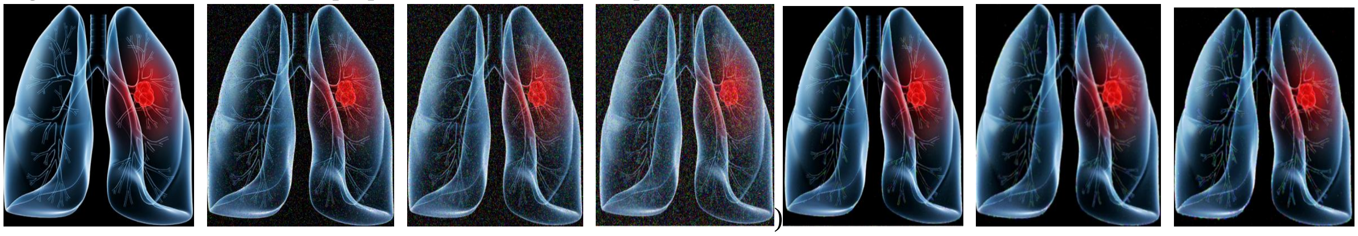


Fig. 12. Comparison of restoration results for Abdomen image

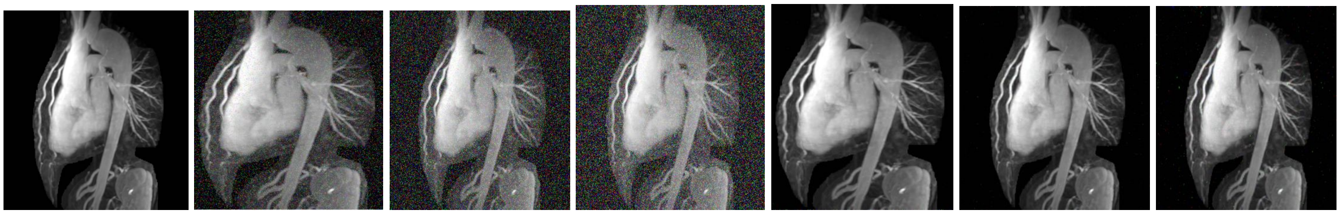
Figures (10-12) shows bar charts of different methods on various images in case of PSNR with respect to various noise levels. From the figures, it is observed that the proposed method has better performance than other methods in all noise levels.



(a) (b) (c) (d) (e) (f) (g)

Fig. 13. Performance of proposed algorithm for Lung image: (a) Original image (b-d) Noisy images at a noise density of 10%, 15% and 20% respectively

(e-g) Denoised images at a noise density of 10%, 15% and 20% respectively

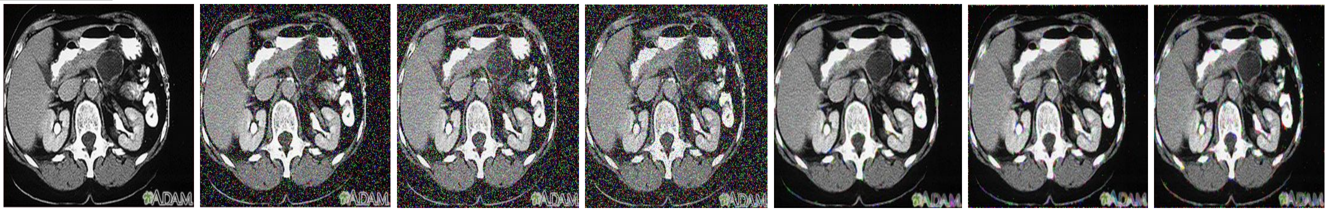


(a) (b) (c) (d) (e) (f) (g)

Fig. 14. Performance of proposed algorithm for Angio image: (a) Original image (b-d) Noisy images at a noise density of 10%, 15% and 20% respectively

(e-f) Denoised images at a noise density of 10%, 15% and 20% respectively

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 8 Issue II Feb 2020- Available at www.ijraset.com



(a) (b) (c) (d) (e) (f) (g)

Fig. 15. Performance of proposed algorithm for Abdomen image: (a) Original image (b-d) Noisy images at a noise density of 10%, 15% and 20% respectively (e-f) Denoised images at a noise density of 10%, 15% and 20% respectively.

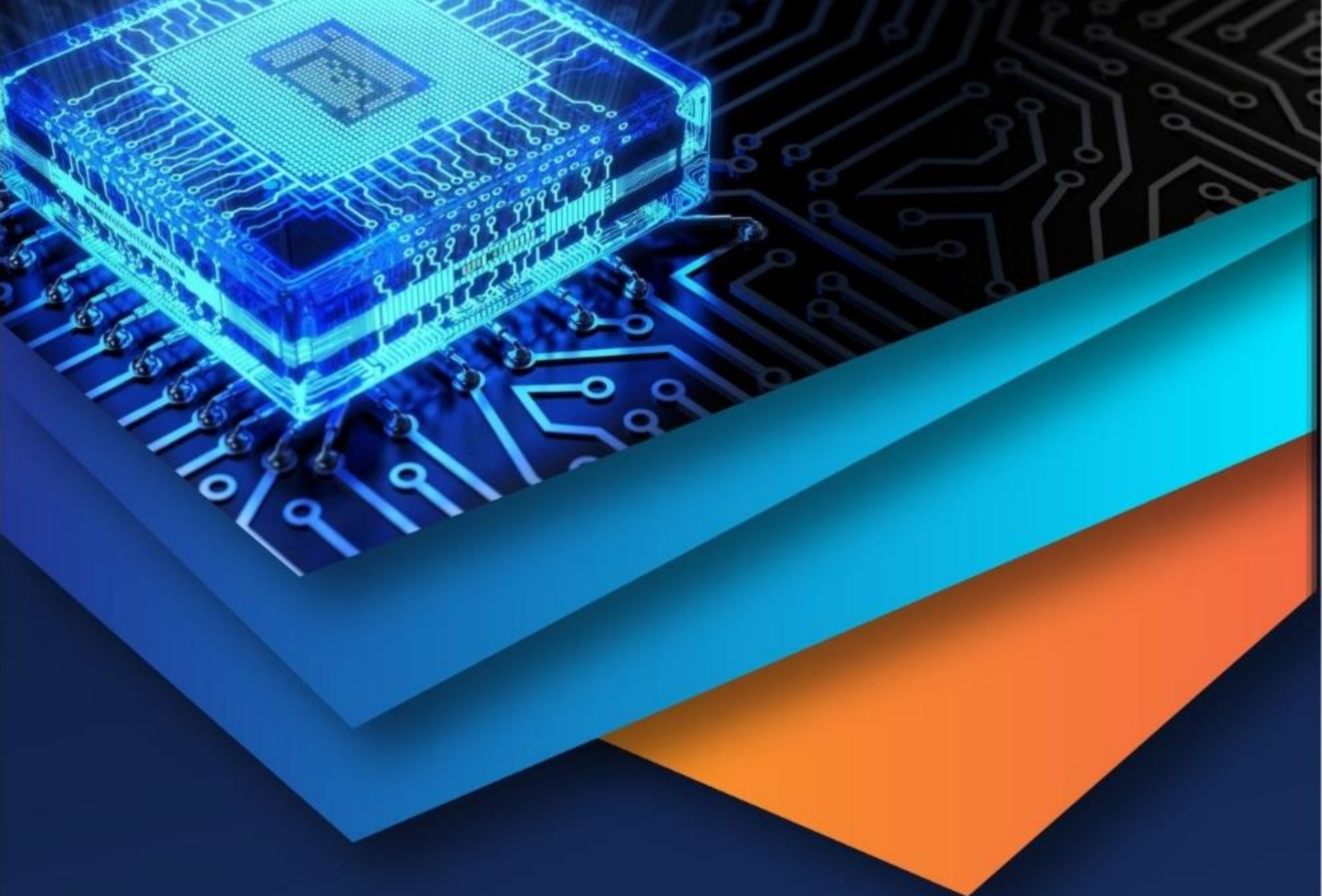
Figures (13-15) illustrate performance of proposed de-noising algorithm at noise density 10, 15 and 20 percentages of random noise for Medical images Lungs, Angio and Abdomen respectively to explore the visual quality. Figures 13, 14 and 15 (a) represent original image, Figures 13, 14 and 15 (b-d) represent the noisy images at a noise density of 10%, 15% and 20% resspectively and Figures 13, 14 and 15 (e-g) represent reconstructed images of proposed de-noising algorithm. From the figures 13, 14 and 15 (e-g) it is observed that reformed images obtained by applying the proposed algorithm preserves the edges in process of de-noising random valued impulse noise in an image.

V. CONCLUSIONS

In this paper, an effective algorithm to denoise randomly valued impulse noise in medical images is proposed. The proposed method uses an impulse detector based on a decision tree and uses an edge preserving filter to denoise the corrupted pixels. The method uses simple mathematical expressions to detect and denoise impulse noise in various Magnetic Resonance Imaging (MRI) images. Due to this simple approach the proposed method falls under the category of low complexity techniques and provides a better performance in terms of PSNR than that of the low complexity techniques, making it very suitable for real time applications. This algorithm without disturbing the edges can de-noise the noisy images, so it would be very helpful in biomedical applications.

REFERENCES

[1] Dong Yiqiu and Xu Shufang, ”A New Directional Weighted Median Filter for Removal of Random Valued Noise”, IEEE Signal Processing Letters, Vol.14, No.3, 2007.
[2] Chen Pei-Yin and Lien Chih-Yuan, “An efficient Edge-Preserving Algorithm for Removal of Salt and Pepper Noise”, IEEE Signal Processing Letters, Vol.15, 2008.
[3] Yu, L. Zhao, and H. Wang, “An efficient Procedure for Removing Random Valued Impulse Noise in Images”, IEEE Signal Processing Letters, Vol. 15, 2008.
[4] Wenbin Luo, "An Efficient Detail-Preserving Approach for Removing Impulse Noise in Images", IEEE Signal Processing Letters, Vol. 13, No.7, July 2006.
[5] Ng Pei-Eng and Ma Kai-Kuang, "A Switching Median Filter With Boundary Discriminative Noise Detection for Extremely Corrupted Images", IEEE Transactions On Image Processing, Vol. 15, No. 6, 2006.
[6] Aiswarya.K, "A New and Efficient algorithm for the removal of High Density Salt and Pepper Noise in images and videos", Second International Conference on Computer Modeling and Simulation, 2010.
[7] Md. Imrul Jubair, "An Enhanced Decision Based Adaptive Median Filtering Technique to Remove Salt and Pepper Noise in Digital Images", Proceedings of 14th International Conference on Computer and Information Technology, 2011.
[8] Dr. Santhanam T and Ms. Chithra K, "A New Decision Based Un-symmetric Trimmed Median Filter using Euclidean Distance Measure for Removal of High Density Salt and Pepper Noise from Images", International Conference on Information Communication & Embedded Systems, 2014.
[9] Dash Aand SathuaS. Kumar, "High Density Noise Removal By Using Cascading Algorithms", Fifth International Conference on Advanced Computing & Communication Technologies, 2015.
[10] Sreejith das L. and S. Vishaga, "A Survey on Switching Median Filters for Impulse Noise Removal", International Conference on Circuit, Power and Computing Technologies, 2015.
[11] Chan R.H., Ho C.W., and Nikolova M, “Salt-and-Pepper Noise Removal by Median-Type Noise Detectors and Detail-Preserving Regularization,” IEEE Trans. Image Processing, Vol. 14, No. 10, pp. 1479-1485, 2005.
[12] Sun Tand Neuvo Y, “Detail-Preserving Median Based Filters in Image Processing,” Pattern Recognition Letters, Vol. 15, pp. 341-347, 1994.
[13] Hwang H. and Haddad R.A., “Adaptive Median Filters: New Algorithms and Results,” IEEE Trans. Image Processing, Vol. 4, No. 4, pp. 499-502, Apr. 1995.
[14] Zhang S. and Karim M.A., “A New Impulse Detector for Switching Median Filter,” IEEE Signal Processing Letters, Vol. 9, No. 11, pp. 360-363, 2002.



10.22214/IJRASET

INDEX COPERNICUS 45.98

IMPACT FACTOR: 7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY