Image Denoising Using Cloud Model Filter

R. Arun Kumar[1] D. Dedeepya[2] D. Sai Kumar[3]
[3]Assistant Professor
[1,2,3]Department of Computer & Science Engineering
[1,2]Chaitanya Bharathi Institute of Technology, Telangana [3]lords Institute of Engineering and Technology, Telangana

Abstract— Image processing is the way of making the digital image to resemble the original image. It can be achieved by identifying the pixels that are different from the original image and then correcting the pixels to the required intensity. It is observed that techniques which follow the two stage process of detection of noise and filtering of noisy pixels achieve better performance than others. Classification is used to separate uncorrupted pixels from corrupted pixels. Reconstruction involves replacing the corrupted pixels by certain approximation technique. The performance of the proposed Cloud Model Filter is measured using the Peak Signal to Noise Ratio (PSNR) values. The PSNR values obtained for the restored image using the proposed filter is compared with the median filter. The performance of the filter is tested by taking various combinations of noisy images and varying the noise levels from 1percent to 100 percent. With any combination of noises, the proposed filter's performance proved better than the median filter.

Key words: Classification, PSNR

I. INTRODUCTION

An image may be defined as a two dimensional function, f(x, y), where x and y are spatial coordinates, and the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y and the amplitude values of f are all finite, discrete quantities, we call the image a digital image. The field of digital image processing refers to processing digital images by means of a digital computer.

Image restoration is a fundamental step of digital image processing [1]. The entire process of image processes and analysis starting from the receiving of visual information to the giving out description of the scene, may be divided into three major stages which are also considered as major sub-areas, and are given below:

A. *Discretization and Representation:*

converting visual information into a discrete form, suitable for computer processing, approximating visual information to save storage space as well as time requirement in subsequent processing.

B. *Processing:*

improving image quality by filtering etc.; compressing data to save storage and channel capacity during transmission.

C. *Analysis:*

extracting image features; quantifying shapes, registration and recognition.

In the initial stage, the input is a scene (visual information), and the output is corresponding digital Image. In the secondary stage, both the input and the output are images where the output is an improved version of the input. And, in the final stage, the input is still an image but the output is a description of the contents of that image [5].

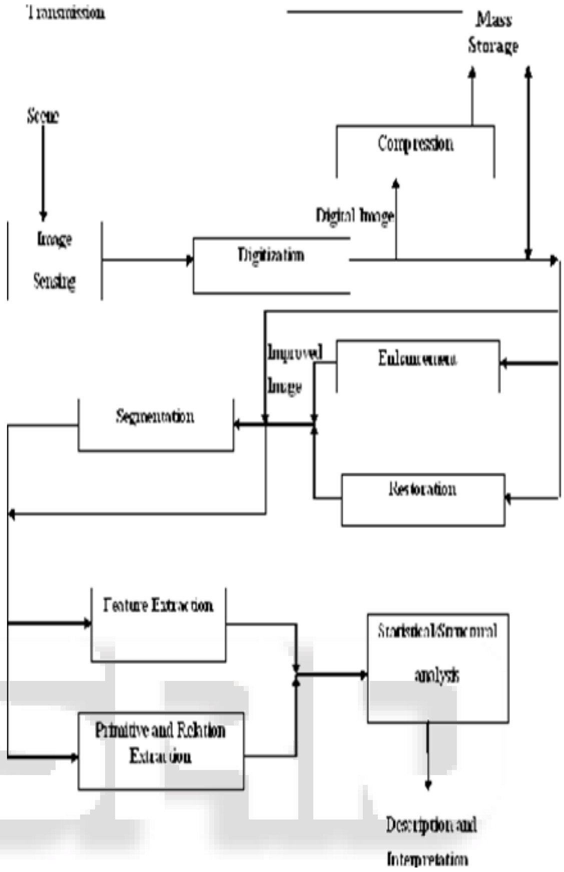

Fig. 1: Different stages of image processing and analysis scheme

II. DESIGN AND IMPLEMENTATION

A. *Cloud Model Filter Characteristics:*

The Cloud Model is a natural-language cognitive model with uncertainty. It combines the fuzziness and the randomness, and forms an intermapping between the qualitative and quantitative information. The definition is given as follows.

Let U be a universal set expressed by exact numbers, and C be the qualitative concept associated with U. If number $x \in U$ exists, which is the random realization of concept C, and the certainty degree of x for C , i.e., $\mu(x) \in [0,1]$, is a random value with stabilization tendency, i.e.,

$$\mu : U \rightarrow [0,1] \; \forall x \in U \qquad x \rightarrow \mu(x)$$

then the distribution of x on U is called the cloud, and each x is called a drop .The cloud can be characterized by three parameters, i.e., the expected value Ex, entropy En, and hyper entropy He. Ex is the expectation of the cloud drops' distribution in the domain.

It points out which drops can best represent the concept and reflects the distinguished feature of the concept. En is the uncertainty measurement of the qualitative concept, which is determined by both the randomness and the fuzziness of the concept. It represents the value region in which the drop is acceptable by the concept, while reflecting the correlation of the randomness and the fuzziness of the concept

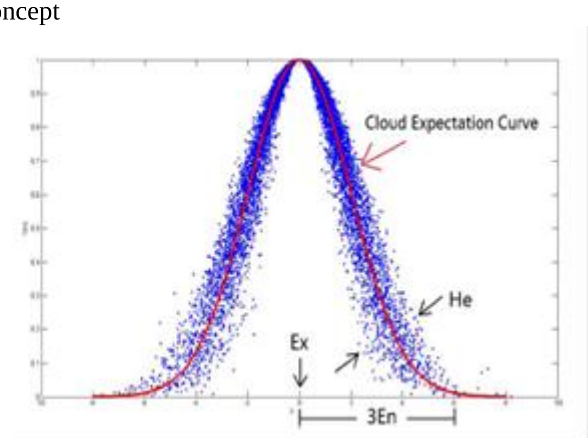

Fig. 2: Representation of Cloud Expectation Curve

The cloud employs its three parameters to represent the qualitative concept. For example, cloud C (Ex=0 , En=2 He=0.3) is shown in Fig.

B. Contribution of Cloud Drops to the Concept:

The drops compose the cloud. When the drops are approaching Ex, the certainty degrees and the contribution degrees of the drops are increasing. Therefore, in the cloud, the drop communities contribute to the concept with the different contribution degrees. In fact, the drops located within [Ex-3En, Ex+3En] take up to 99.99% of the whole quantity and contribute 99.74% to the concept. Thus, the drops are located out of domain [Ex-3En, x+3En], and then, their contributions to the concept can be neglected. This is "the 3En rule." [12]

C. Implementation Details:

The CM Filter calculates the weights of each pixel based on the expected and entropy values and the corresponding weights are multiplied with the pixel intensity if it is identified as a corrupted pixel.

Let $W_{i,j}^{2N+1}$ be a window of size (2N+1)×(2N+1) centered at location (i,j) , and $x_{i,j}$ denotes the gray value of the pixel at location (i,j); S_{max} and S_{min} denote the maximum and minimum gray values in the noise image, respectively. W_{max}^{2N+1} And W_{min}^{2N+1} denote the maximum or minimum values in , $W_{i,j}^{2N+1}$ respectively.

D. Algorithm for Noise Pixels Detection:

(1) Initialize N=1 and denote n as the number of uncorrupted pixels in $W_{i,j}^{2N+1}$.

(2) Calculate Ex of all the pixels in $W_{i,j}^{2N+1}$, i.e.,

$$Ex = \frac{1}{n} \sum_{x_{i+s,i+t} \in W_{i,j}^{2N+1}} x_{i+s,j+t}$$

(3) Calculate En, i.e.,

$$En = \sqrt{\frac{\pi}{2}} \times \frac{1}{n} \sum_{x_{i+s,j+t} \in W_{i,j}^{2N+1}} |x_{i+s,j+t} - Ex|$$

(4) Calculate W_{max}^{2N+1} and W_{min}^{2N+1} , respectively, i.e.,
W_{max}^{2N+1} =Min(S_{max},Ex+3En)
W_{min}^{2N+1} =Max(S_{min},Ex-3En)

where Min and Max are the extreme operations to recover the smallest and the largest of the two values, respectively.

(5) If $W_{min}^{2N+1} < x_{i,j} < W_{max}^{2N+1}$ and , then $x_{i,j}$ is an uncorrupted pixel; otherwise, go to step 6.

(6) Identify if the other pixels in $W_{i,j}^{2N+1}$ are the noise pixels or not. For pixel $x_{i+s,j+t}$, if $W_{min}^{2N+1} < x_{i+s,j+t} < W_{max}^{2N+1}$, then $x_{i+s,j+t}$ will remain, and n=n+1.

(7) If $W_{min}^{2N+1} >= x_{i,j}$ or $x_{i,j} >= W_{max}^{2N+1}$, then set N=N+1, and go to step 2; otherwise, $x_{i,j}$ is a noise candidate.

E. Algorithm for Correction of Noise Pixels:

(1) Calculate Ex of each uncorrupted pixel in , i.e.,

$$Ex = \frac{1}{n} \sum_{x_{i+s,i+t} \in N_{i,j}^{2N+1}} x_{i+s,j+t}$$

(2) Calculate En, i.e.,

$$En = \sqrt{\frac{\pi}{2}} \times \frac{1}{n} \sum_{x_{i+s,j+t} \in N_{i,j}^{2N+1}} |x_{i+s,j+t} - Ex|$$

(3) Calculate weights for , i.e.,

$$w_{i+s,j+t} = \exp(-(x_{i+s,j+t} - Ex)^2 / 2En^2)$$

(4) Calculate the weighted mean, i.e.,

$$y_{i,j} = \sum_{x_{i+s,i+t} \in N_{i,j}^{2N+1}} w_{i+s,j+t} x_{i+s,j+t}$$

Based on the expected and entropy values calculated, the noisy pixels are detected at each window and pixel intensities are multiplied by their weights if it is a noisy pixel .Thus , WFM filter replaces the noise pixel by using the weighted mean of the remaining pixels, and their weights are the certainty degrees of them.

To detect the type of noise in the image, the number of pixels with intensities 0 and 255 are calculated. Based on the count of the pixels, the type of the noise can be detected.

For better results the existing CM Filter is combined with the Gaussian filter to restore the image after applying the CM Filter. To test the performance of the proposed filter, MATLAB predefined median filter is also implemented. To compare the performances, the input noisy image is passed as a parameter to the implemented restoration function and the same input is passed as a parameter to the median filter function .Thus, same image is restored simultaneously by the proposed and median filter. The Peak Signal to Noise ratio(PSNR) values are calculated for the restored images obtained from both the filters and are compared against each other.

III. RESULTS AND ANALYSIS

For comparing the performances, commonly tested grayscale image, Lena is selected in the simulations. For comparative purposes, the implemented filter's performance is compared against the already existing median filter .The restoration performance by both the filters is compared by calculating the PSNR values for the restored images obtained from the filters. It is observed that the implemented filter performs well in any type of situation, even when the noise corruption levels are high. For simulations, different kinds of images corrupted with different kinds of noise (Salt and Pepper noise, Gaussian noise, Speckle noise, Poisson noise) with various noise levels are compared and tested.

The restoration performances are quantified by the peak signal-to-noise ratio (PSNR)

$$\mathrm{PSNR} = \log_{10}\left(\frac{255^2}{\frac{1}{MN}\sum_{i,j}(y_{i,j}-x_{i,j})^2}\right)\times 10$$

where $y_{i,j}$ and $x_{i,j}$ denote the pixel values of the restored image and the original image, respectively.

First, the filters apply on the noise images that are affected with four different kinds of noise (namely Gaussian, Speckle, Salt and Pepper noise and Poisson noise) and used the commonly tested image Lena.jpg with both the colored and gray versions. The obtained PSNR values for restored images from both the filters are compared and it is observed that for any kind of noise affected, the performance of the implemented filter is better than the compared median filter. For clear comparisons, the same image is used to test the four types of noise and the results are tabulated and presented along with screenshots at each step of execution. For precise comparisons, bar charts are also presented with the type of noise on x-axis and the PSNR values obtained after the restoration on the y-axis.

A. *Lena.Png Color Image as Input:*

Firstly, lena.png image is used as input image and tested the performance of both the filters by imparting the four types of noise. The screen shots at each step , i.e, image with particular amount of noise , and the restored image obtained from the proposed and the median filter are presented below:

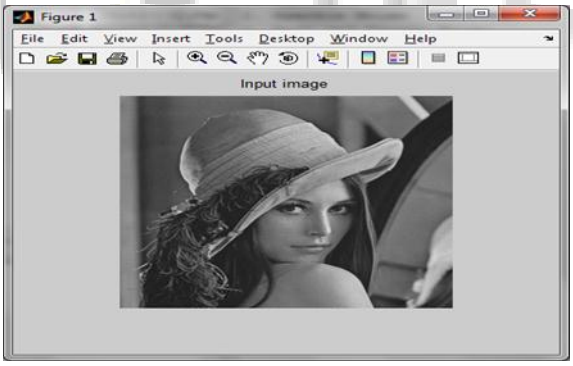

Fig. 3: Input image

B. *Adding Gaussian Noise:*

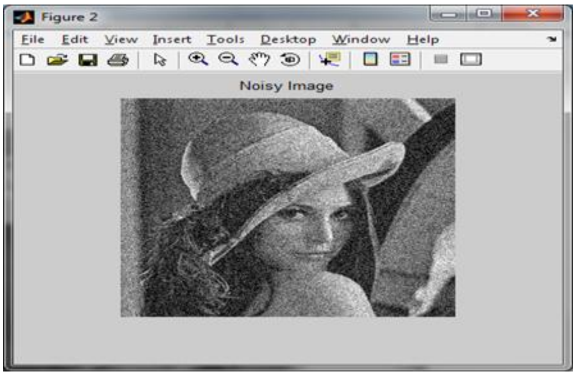

Fig. 4: Noisy image affected with Gaussian noise

Figure shows the image that is obtained after imparting the Gaussian Noise which is used as input for further processing.

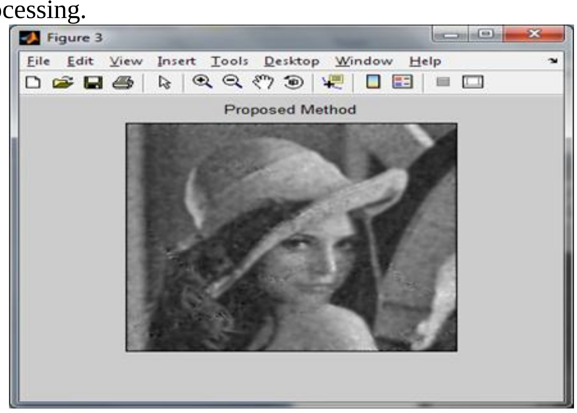

Fig. 5: Restored image using the proposed Filter

Figure shows the restored image obtained from the proposed cloud model filter.

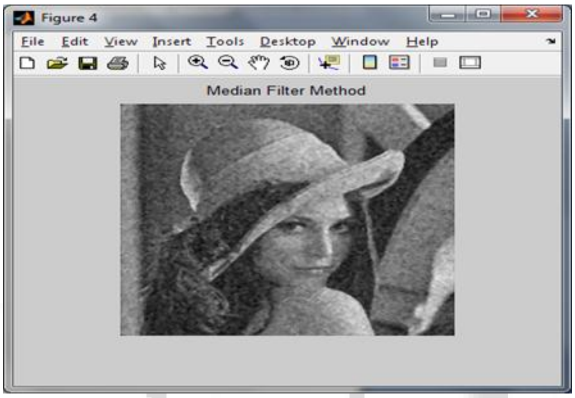

Fig. 6: Restored Image using median Filter.

Fig. 7: Restored image using the proposed filter

Figure shows the restored image obtained from the proposed cloud model filter.

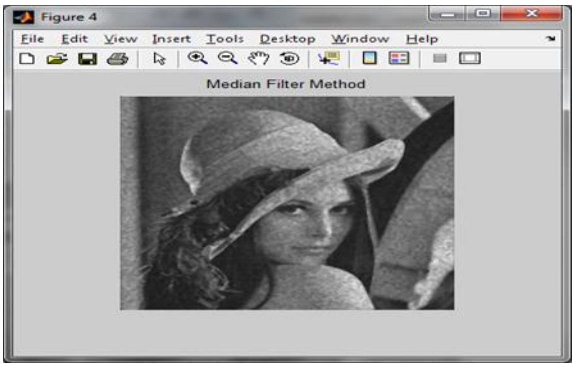

Fig. 8: Restored Image using the median filter

Figure shows the restored image obtained from the median filter.

C. Image with Salt and Pepper Noise:

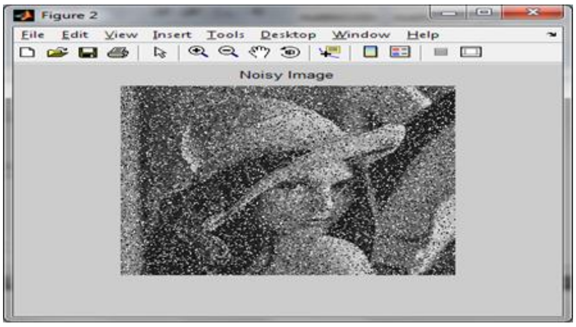

Fig. 9: Image affected with Salt and Pepper noise

Figure shows the image that is obtained after imparting the Salt and Pepper Noise which is used as input for further processing.

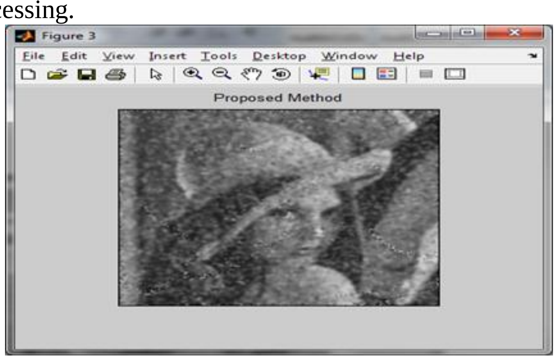

Fig. 10: Restored image using proposed filter

Figure shows the restored image obtained from the proposed cloud model filter.

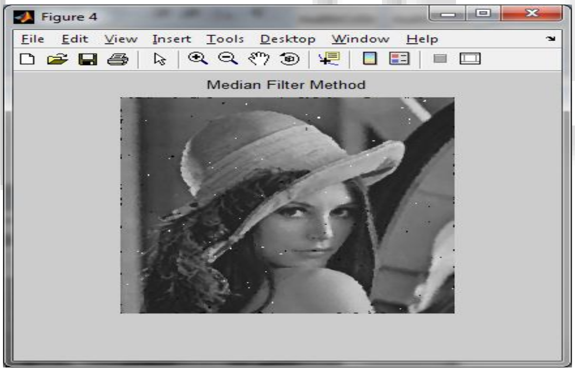

Fig. 11: Restored image using median filter

Figure shows the restored image obtained from the median filter.

D. Image with Poisson Noise:

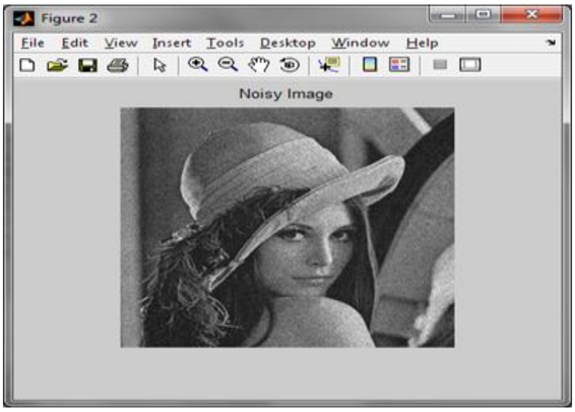

Fig. 12: Image affected with poisson noise

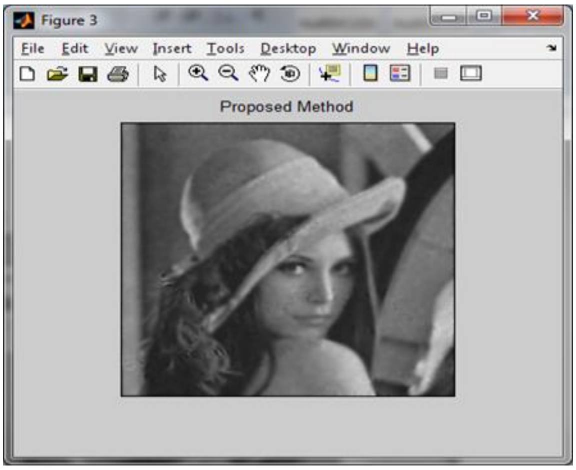

Fig. 13: Restored image using proposed filter

Figure shows the restored image obtained from the proposed cloud model filter.

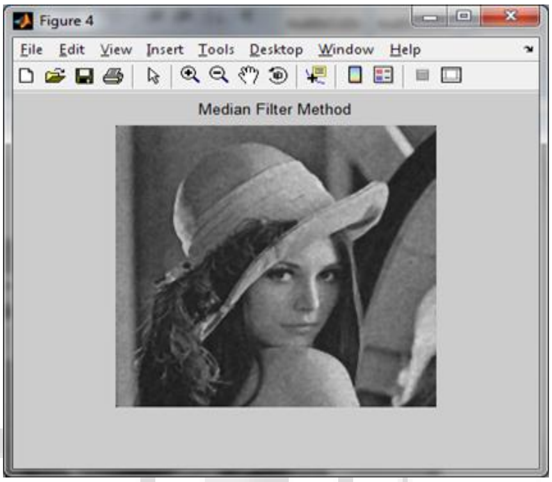

Fig. 14: Restored image using median filter

Figure shows the restored image obtained from the median filter.

The values obtained are tabulated below where the first column contains the type of noise, second column contains PSNR values obtained using the median filter and the third column contains PSNR values obtained using the proposed novel filter. The below table shows the values obtained with the Lena.png color image.

Type Of Noise	**PSNR from Median Filter**	**PSNR from Proposed Filter**
Gaussian Noise	55.8473	56.5938
Speckle Noise	51.6641	53.7403
Salt And Pepper Noise	63.5401	63.0194
Poisson Noise	48.5567	52.2957

Table 1: Results obtained with varying noises on colored image

From the above Table, it can be observed that , given the same image and the type of noise , the proposed filter works well in any kind of situation and when corrupted with any type of noise except Salt and Pepper noise.

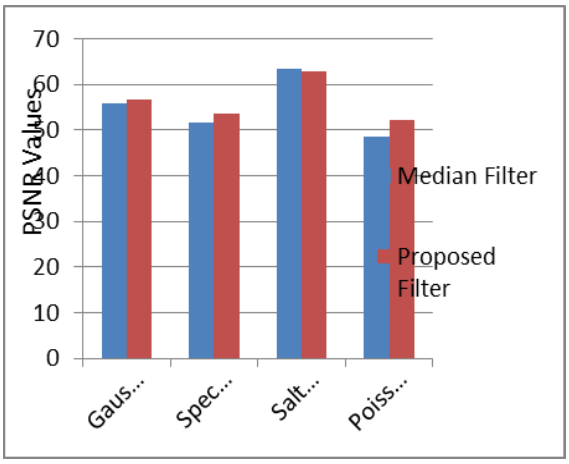

Fig. 15: Chart depicting the colored Lena image PSNR values

Figure is the graph which is plotted by taking the type of the noise on the X-axis and the PSNR values obtained using the proposed and the Median Filter on the Y-axis. It can be observed from the graph that the performance of the proposed filter is better in terms of the PSNR values than the existing median filter.

E. Results Obtained By Varying the Intensity Level of Speckle Noise:

To test the performance of the filter when the intensity of noise level is high , a sample colored lena image is taken and tested by varying the percentage of Speckle noise on the image

Percentage of Speckle Noise	PSNR from Median Filter	PSNR from Proposed Filter
2	51.6224	53.6927
4	54.3583	55.3862
6	54.0442	56.6278
8	57.1634	57.4745
10	58.0392	58.1692
20	60.8429	60.9815
30	62.5775	62.7675
40	63.5672	63.6469
50	64.3180	64.7404
60	64.7900	64.7960
70	65.1929	64.5034
80	65.5484	64.7891
90	65.8066	65.0010
100	66.1158	65.2619

Table 2: Results obtained with varying Speckle noise

It can be observed that, given the same image with any combination of noise, either it is combination of two or three or four types of noises, the proposed Cloud Model filter works well in any kind of situation.

IV. CONCLUSION

An efficient noise removal algorithm using the Cloud Model Filter is implemented. The CM Filter is applied to the noisy image. The CM Filter restores the pixel values based on the threshold calculated using the expected values. To compare the performance of the proposed filter, existing median filter is also implemented. Simulations are done by taking various types of noise affected images and compared with the median filter's performance.

A regularly used image is used to test the performance of both the filters. Noise detection is performed based on the number of pixels with intensity 0 and 255. Four types of noise are taken into consideration Gaussian Noise, Speckle Noise, Salt and Pepper Noise, Poisson Noise. When the same image is affected with different kinds of noise, the performance of the proposed filter is found to be better. For testing with same kind of noise affected images, Gaussian noise and Speckle noise is varied in levels of 2 percent till 10 percent and then increased the noise intensity in levels of 10 percent till 100 percent. The filters are also tested by giving input noisy image corrupted with multiple noises and observed that proposed filter performs better than the median filter. The results are compared using the PSNR values obtained from the restored images using the median filter and the proposed filter. Even when the noise levels of the same type of noise is varied and tested, at different noise levels, the proposed filter performs better than the existing median filter.

REFERENCES

[1] Zhe Zhou, "Cognition and Removal of Impulse Noise With Uncertainty", IEEE Transactions on Image Processing, VOL. 21, NO. 7, JULY 2012.
[2] Polesel, G.Ramponi, and V.J. Mathews, "Adaptive Unsharp Masking for Contrast Enhancement", IEEE Transactions on Consumer Electronics, 1997.
[3] Abreu, M. Lightstone, S. K. Mitra, and K. Arakawa, "A new efficient approach for the removal of impulse noise from highly corrupted images," IEEE Trans. Image Process, vol. 5, Jun. 1996.
[4] Picinbono, "Quadratic Filters", IEEE Transactions on Accoustics, Speech and Signal Processing IEEE International Conference on ICASSP, May 1982.
[5] Banshidhar Majhi, "Soft Computing Techniques for Image Restoration", PhD thesis, Sambalpur University, 2000.
[6] B Chandra and D Dutta Majumder, "Digital Image Processing and Analysis", Prentice-Hall, India, first edition edition, 2007.
[7] Brownrigg, "The weighted median filter," Commun. Assoc. Comput., vol. 27, Aug. 1984.
[8] L. Yin, R. Yang, M. Gabbouj, and Y. Neuvo, "Weighted median filters: A tutorial," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 43, Mar. 1996.
[9] P. Maragos and R.W. Schafer, "Morphological Filters Part–i: Their Set Theoritic Analysis and Relations to Linear Shift Invariant Filters", IEEE Transactions on Accoustics, Speech, Signal process, June 1987.