Analysis of Non-Linear Filtering Techniques based on Quantitative Metrics using Different Images

RAJKUMAR S
SCSE
VIT University, Vellore

SHREY GUPTA
SCSE
VIT University, Vellore

VIJAYARAJAN V
SCSE
VIT University, Vellore

ANIL KUMAR
SCSE
VIT University, Vellore

ABSTRACT
Image filter is the process of removing various types of noise from the images. The ensuant image is an information rich image than the original input image. The filtering finds its application in many fields from medical imagery, face detection, robot navigation, object detection, aircraft maintenance to image enhancement and image restoration. In the field of medical sciences the filter serves the purpose of image enhancement for efficient disease diagnosis, in aircraft maintenance for the purpose of detection of faults during takeoff, in case of face detection, object recognition and robot navigation used for object detection. This paper uses different quantitative metrics to analyze the result of different filtering techniques on an image. Initially, well known registered images from various aspects of science and nature are taken such as one image ct.jpg from medical sciences, two images Lighthouse.jpg, Penguins.jpg of natural scenery, two images of faces Koala.jpg, lena.jpg and a picture of naturally grown flowers Tulips.jpg are taken as input. Filtering techniques namely Median Filter (MF), Adaptive Filter (AF), New Adaptive Median Filer (NAMF), New Adaptive Spatial Filter (NASF), Edge Preserving Smooth Filter (EPSF) are applied on them. Further the filtered images are analyzed using five quantitative metrics such as Entropy (EN), Standard Deviation (SD), Peak Signal to Noise Ratio (PSNR), Mean Square Error (MSE), and Mean Absolute Error (MAE). From the experimental result and the corresponding metrics used we observed that the resultant image is more informative than the original source images.

General Terms
Image Enhancement, Salt and Pepper noise, Gaussian noise, Quantitative Analysis.

1. INTRODUCTION
In recent years, noises can occur from a number of resources. While capturing image from a sensor or transmitting image through a communication channel, image can get frequently contaminated by several noises namely Salt- and- pepper noise, Gaussian noise etc. As a result of this, such corrupted images while being used for processing viz. image segmentation, object recognition, image fusion, and edge detection tend to provide unexpected results. Henceforth, before any processing, improving the quality of images play a vital role. The required improvement can be achieved using filtering techniques. Filtering techniques are classified into two types - linear filters and non-linear filters. Linear filters have a drawback that it fails to recover sharp pixel edge corrupted by noise. In addition, impulse noise cannot be abridged adequately. Salt-and–pepper noise occurs when there is a fault or malfunction in the equipment capturing the picture or an error in the digitization process of the image or an error in storing the image to memory location. The performance of linear filters on Salt-and-pepper noise is unsatisfactory hence it cannot be used frequently [7, 8, 9].

Generally, Non-linear filter show better results than linear filter for removing such noise. To overcome these problems we have proposed the following non-linear filters namely Median Filter, Adaptive Filter, New Adaptive Median Filter, New Adaptive Spatial Filter, and Edge Preserving Smooth Filter. The main aim of these filters is to enhance the quality of image for richer image quality and better feature extraction.

Median filter (MF) utilizes median of the neighborhood of a pixel to smoothen the image [1].

The Adaptive Filter (AF) works in two stages, namely adaptive median-based filtering and statistics-based estimation. Adaptive median-based filter detects corrupted pixel in an image through estimation techniques and then statistical analysis corrects the corrupted pixel through the local neighborhood [2].

The New Adaptive Median Filter (NAMF) performs spatial processing to determine which pixels in an image have been affected by impulse noise i.e. this filter sets a threshold intensity value derived from a region of uniform amplitude. Thus it can detect impulse noise in a pixel when there is a large difference in the value of the pixel in consideration and threshold value set, replacing only noise corrupted pixels and not the noise-less ones. As a result reduces noise without blurring the image [3].

The New Adaptive Spatial Filter (NASF) uses discrete features of an image. It constructs a mask using the uniform regions of the image resulting in a better mask than the conventional spatial filter. It calculates a threshold value in an image and replaces the pixel data if and only if the difference between the highest pixel intensity and the lowest pixel intensity is less than the threshold value. The result is an enhanced image with reduced noise and better sharpness [4].

The Edge Preserving Smooth Filter (EPSF) reduces the noise and provides better edge preservation property [5].

The performances of the filtering techniques are evaluated based on the different quantitative metrics such as EN, SD, PSNR, MSE, and MAE.

The remaining sections of this paper organized as follows. In section II discusses the type of noise, section III system design briefly reviewed, section IV describes experimental results and evaluates the performance of the proposed methods based on the quantitative metrics. Discussion and future work are summarized at the end.

2. TYPES OF NOISE

Normally, an image is affected by different types of noise. The most familiar types of noise are Impulse (Salt-and-Pepper) noise, Gaussian (Normal) noise, Uniform noise, Exponential noise, Gamma noise etc. In this paper, mainly Impulse (Salt-and-Pepper) noise is discussed. The following subsection briefly explains the noise detail.

2.1 *Impulse Noise*

The Probability Density Function (PDF) of Bipolar impulse noise is defined by [12]

$$p(z) = \begin{cases} P_a & for \quad z = a \\ P_b & for \quad z = b \\ 0 & otherwise \end{cases} \qquad (1)$$

If b > a, gray-level 'a' appears as a dark dot in the image. 'b' appears like a light dot in the image. If either P_a or P_b is zero, the impulse noise called as unipolar noise. If neither P_a or P_b is zero, and particularly if they are roughly equal, impulse noise values resemble Salt and Pepper granules randomly distributed over the image. The noise impulses can be either negative or positive. Negative impulses appear as black (Pepper)points in an image and positive impulses appear as white (Salt)noises. For an 8 bit image this means that $a = 0$ (black) and $b = 255$(white). The pictorial representation of impulse noise is shown in fig.1-2 [11].

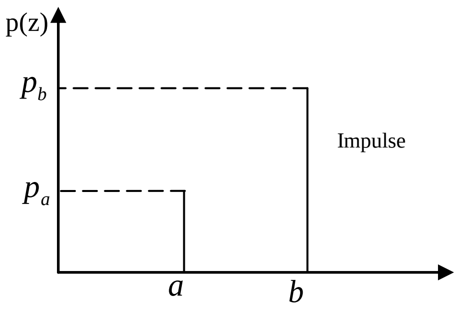


Fig 1: Probability Density Function of Impulse Noise

Fig 2: Representation of Salt-and-Pepper Noise

3. SYSTEM DESIGN

In this system initially registered gray scale image taken as input. Noise is applied to the input image and then the noisy image is filtered using different filtering techniques. Finally, the output filtered image is validated using quantitative measures. The system design is shown in Fig. 3.

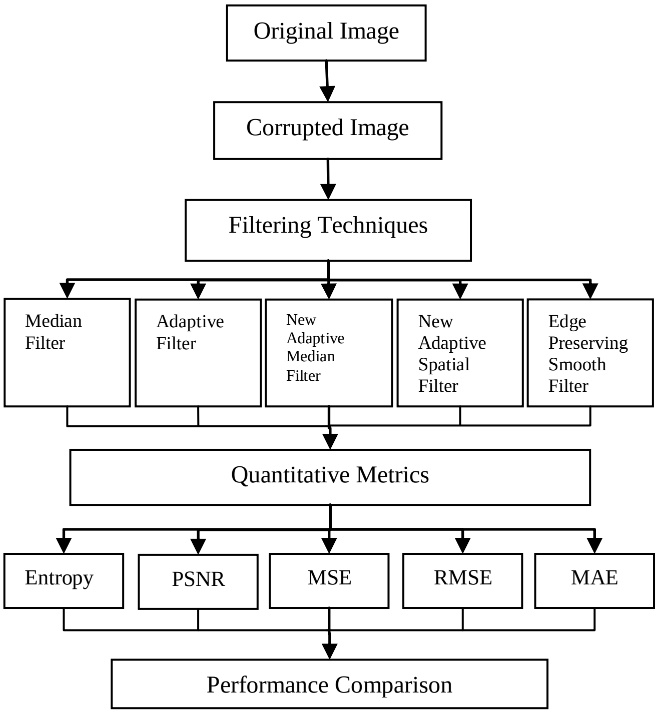


Fig 3: An overall view of system design

3.1 *Median Filter*

Median filter [1] is a non-linear filter that is often described in the spatial domain. It actually, removes Salt-Pepper noise from distorted image, but this image while filtering suffers the blurring effect. The median filter works by utilizing the median value of neighboring pixels. The calculation of median filter performs following task to find each pixel value in the processed image:

- All pixels in the neighborhood of the pixel in the original image which are recognized by the mask are stored in the descending (or) ascending order.
- The median of the stored value is computed and is chosen as the pixel value for the processed image.

Thus even the unaffected pixels are replaced by median value resulting in image distortion.

3.2 Adaptive Filter

The Adaptive Filter [2] works in two stages, namely adaptive median-based filtering and statistics-based estimation. In first stage, the adaptive median-based filter aims to detect corrupted pixel in order to replace its value with the noise-free median of local neighborhood. If it fails to obtain a noise-free pixel then statistics-based estimation is used to find the noise-free pixel. In second stage, statistical analysis corrects the corrupted pixel through the local neighborhood.

3.2.1 Adaptive Median-Based Filter: Given an image A corrupted with salt-and-pepper noise, this algorithm first square window $W_{2d+1}(i, j)$ with odd $(2d+1)*(2d+1)$ dimensions centered on a corrupted pixel (i, j) and calculates the median from the corresponding neighborhood. Formally, define the processing window as

$$W_{2d+1}(i, j) = A(i + x, j + y) \qquad (2)$$

Where x, y belongs to $\{-d....0,1....d\}$.

Initially, the algorithm start with d=1, and checks the median of the local neighborhood is noise-free or not. If a noise-free median is found, then the value of the corrupted pixel (i, j) is replaced with median value of local neighborhood. Suppose, the median value is noisy then the window size is expanded by incrementing the value of d by 1 and again algorithm checks for the noise-free median in local neighborhood. This process is repeated until a noise-free median is found or d reaches the pre-defined maximum window size $(d = 3)$.

If a noise-free median is found, the corrupted pixel is replaced by median value. If a noise-free median is not found, the corrupted pixel is subjected to further statistical analysis of the local neighborhood in order to estimate an accurate correction term.

3.2.2 Statistics-Based Estimation: Statistics-Based Estimation algorithm is used when the adaptive filter cannot find the noise-free median through the maximum size processing window. This can happen when the entire pixels in the image are corrupted. To solve this problem, Statistics-Based-Estimation checks the value of last processed pixel.

If the value of the last processed pixel is not 0 or 255, then the current pixel is considered as a noisy pixel. However in this case, basically using last processed pixel to replace the noise pixel may not be consistent for the property of local region.

In order to perform well this method first checks the property of the region defined by window size is $9*9$. If a noise-free median is found in the neighborhood through the processing window then the noisy pixel is replaced with the last processed pixel value. Otherwise, the noisy pixel is replaced by the mode of the local neighborhood.

3.3 New Adaptive Median Filter

This filter detects the impulse noise, for this it makes the assumption that a noisy pixel takes a gray value which is different from the neighboring pixels in the filtering window. Here, the difference of the median value of pixels in the filtering window and the current pixel value is compared with a threshold to make a decision about presence of the noise [3].

The New Adaptive Median Filter algorithm as follow as:

Step 1: Take a sub window of size $w \times w$ around the current pixel where $w = 3$.

Step 2: Move to step 1 if the difference between the maximum and minimum value of pixel is not above the threshold value under the window limit.

Step 3: Count all other pixel values excluding minimum and maximum values under the window limit.

Step 4: If the count value calculated in step 3 is greater than or equal to w than calculate the median value of pixels which are not the part of maximum and minimum value; else increase the size of the window by $w = w + 2$ and go to step 1.

Step 5: Replace the current pixel by excluded median value if the current pixel is equal to maximum and minimum value; else leave it and move to next pixel and go to step 1.

3.4 New Adaptive Spatial Filter

This filter removes the Gaussian and impulse noise to produce a good quality image [4].

The New Adaptive Spatial Filter given below:

Consider an image f of size $M \times N$ and L gray levels.

Step 1: Construct a matrix from the entire image so that pixel values (X_{ij}), of order $M \times N$ can be stored which is also called image resolution.

Step 2: Add dummy rows or columns(whichever is suitable) if M & N are not the multiple of 3 to make them so.

Step 3: Make smaller matrixes of order 3×3 from the above obtained matrix.

Step 4: Consider a 3×3 matrix (T_k) and operate upon this k^{th} matrix, $(1 \le k \le (M \times N)/9)$ as follows:

Step 4.1: Identify the maximum (Max_k) and minimum (Min_k) from the entire given pixel values.

Step 4.2: If $|Max_k - Min_k|$ is less than a threshold values (optimal value 200 obtained by trial and error) then calculate mean of the pixel values (X_{ij}) over matrix (T_k) as:

$$\Delta = (1/9) x \sum_{i=1}^{i=3} \sum_{j=1}^{j=3} X_{ij} \qquad (3)$$

else consider the next matrix (T_{k+1}).

Step 4.3: Construct a difference matrix $\delta[i][j]$ whose value is calculated as $\delta[i][j] = |X_{ij} - \Delta|$

Step 4.4: Replace the value in the given 3×3 matrix with the pixel value corresponding to $\min(\delta[i][j])$.

3.5 Edge Preserving Smoothing Filter

Sharpening and smoothing are two effects induced by this filter. The following steps calculate approximately sharpening and smoothing values [5].

Step 1: Plot a scattergram: A scattergram is plotted of the pixel of the gradient magnitude of the original image versus those of the gradient magnitude of the filtered image.

$$\Delta f = \begin{matrix} \dfrac{\partial F}{\partial x} \\ \dfrac{\partial F}{\partial y} \end{matrix} \qquad (4)$$

$$\Delta F = mag(\Delta f) = \sqrt{(\partial F/\partial x)^2 + (\partial F/\delta y)^2} \qquad (5)$$

where ΔF denotes the gradient magnitude of function $I(x, y)$

Δf denotes the gradient vector of function $I(x, y)$

Step 2: Fit Lines: Fit line $y = ax + b$ used through the two sets to achieving density-independent factors in which edges are sharpened and flat regions are smoothed.

$$(a_A, b_A) = \arg\min_{(a,b)} \sum_{(x,y)\in A} \left| y - (ax + b) \right| \qquad (6)$$

$$(a_B, b_B) = \arg\min_{(a,b)} \sum_{(x,y)\in B} \left| y - (ax + b) \right| \qquad (7)$$

where a_A denotes the smoothing of the filter

a_B indicate the sharpening of the filter

$a_A \leq 1$ and $a_B \geq 1$ are needed because values are cut at 1 is necessary.

Step 3: Weight Slopes: The slopes found are weighted with the relative number of points used to specify the number of pixels actually used to estimate these values.

$$\text{Smoothing}(F, I) = (a_A^1 - 1)|A| / |A| + |B| \qquad (8)$$

$$\text{Sharpening}(F, I) = (a_B^1 - 1)|B| / |A| + |B| \qquad (9)$$

where $a_A^1 = 1/a_A$

The above two equation values can be considered to be an amplification factor of edges, and an attenuation factor of flat regions, respectively.

3.6 Quantitative Analysis on Filtered Image

The quantitative measurement (Performance Evaluation) is done on the filtered images using some objective and subjective quality measures. It helps better in assessing the information of images. This section explains the quantitative metrics used in the analysis of this system.

3.6.1 *Entopy(EN)*: Entropy can effectively reflect the amount of information in certain image. A larger value indicates that a better fusion result is obtained [6]

$$EN = -\sum_{i=0}^{L-1} P_F(i) \log_2 P_F(i) \qquad (10)$$

where P_F is the normalized histogram of the fused image to be evaluated, L is the maximum gray level for a pixel in the image. In our tests, L is set to 255.

3.6.2 *Peak Signal to Noise Ratio (PSNR)*: PSNR is used to measure the quality of the image with respect to the original input image. It is defined as: [10]

$$MSE = 1/pq \sum_{i=0}^{p-1} \sum_{j=0}^{q-1} [A(i, j) - B(i, j)]^2 \qquad (11)$$

$$PSNR = 10 \log_{10} \left(MAX^2 / MSE \right) \qquad (12)$$

where MAX is the maximum value in an image. p, q are the height and weight of an image. $A(i, j)$ is the value of input image and $B(i, j)$ is the value of filtered image.

3.6.3 Standard Deviation (SD): Standard deviation defines the contrast information of an image. Image with more contrast has high value of standard deviation whereas images with low contrast have low values of standard deviation: [6]

$$\sigma = \sqrt{1/N \sum_{i=1}^{N} (x_i - x)^2} \qquad (13)$$

Where x is defined as a summation

$$x = 1/N \sum_{i=1}^{N} x_1 + x_2 + \ldots\ldots + x_N / N \qquad (14)$$

3.6.4 Mean Square Error (MSE):

$$MSE = 1/MN \sum_{i,j} (y_{ij} - x_{ij})^2 \qquad (15)$$

where y_{ij} denotes the corrupted image, x_{ij} denotes the filtered image, MN total number of pixel in the image [2].

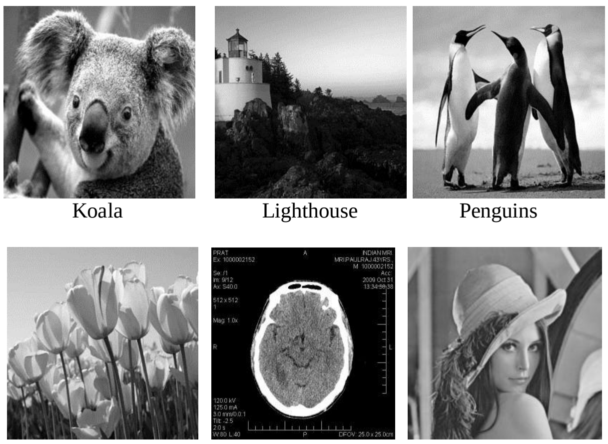
3.6.5 Mean Absolute Error (MAE):

$$MSE = 1/MN \sum_{i,j} \left| y_{ij} - x_{ij} \right| \qquad (16)$$

where y_{ij} denotes the corrupted image, x_{ij} denotes the filtered image, MN total number of pixel in the image [2].

4. EXPERIMENTAL RESULTS AND PERFORMANCE EVALUATION

The experimental results of the filtering techniques are analyzed with six different types of gray scale images with jpg file extension. All images have the same size of $256 * 256$ pixels, with 256-level gray scale. Input images are shown in Fig. 4.



Koala Lighthouse Penguins

Tulips CT Lena

Fig 4: Input Images

The above input images are corrupted with 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, 100% with the fixed impulse noise. Some of the corrupted images with 40%, 70% and 90% impulse noise Shown in Fig. 5-7.

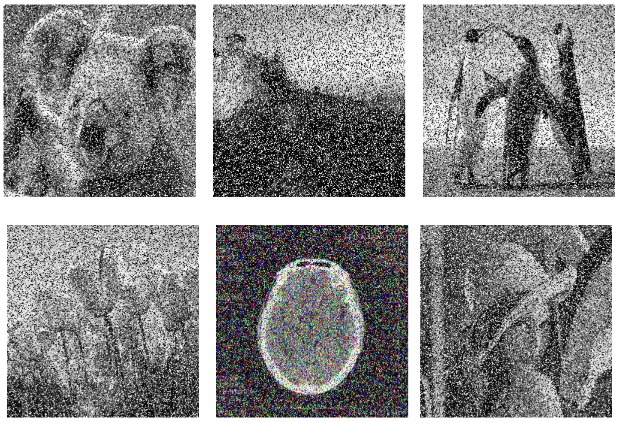


Fig 5: All the input images corrupted with noise level 40%

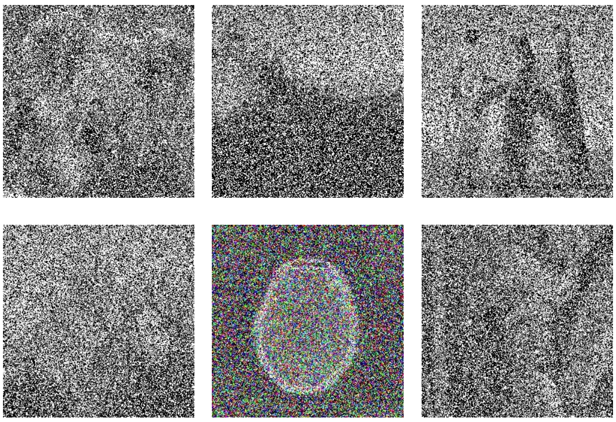


Fig 6: All the input images corrupted with noise level 70%

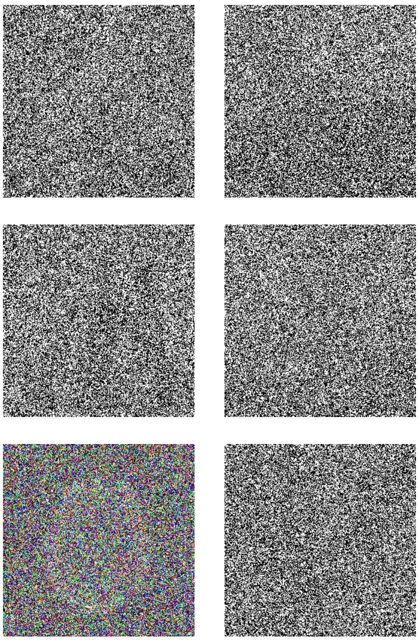


Fig 7: All the input images corrupted with noise level 90%

The resultant of the koala image derived from the 90% corrupted noise after applying different filters are shown in Fig. 8.

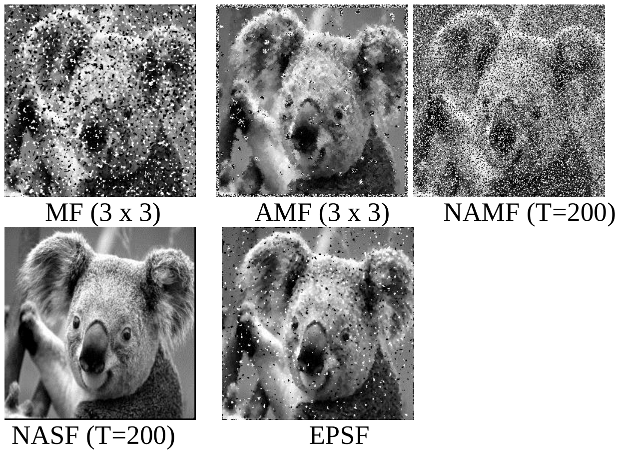


Fig 8: Shows enhanced Koala image after filtering from noise level = 90%

The resultant of the penguins image derived from the 90% corrupted noise after applying different filters are shown in Fig. 9.

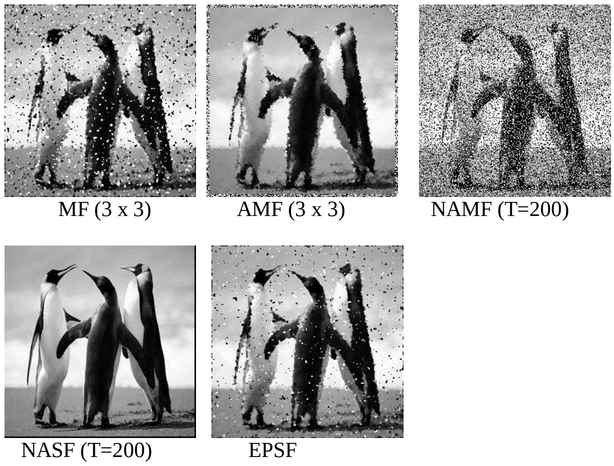


Fig 9: Shows enhanced Penguins image after filtering from noise level = 70%

The resultant of the CT image derived from the 40% corrupted noise after applying different filters are shown in Fig. 10.

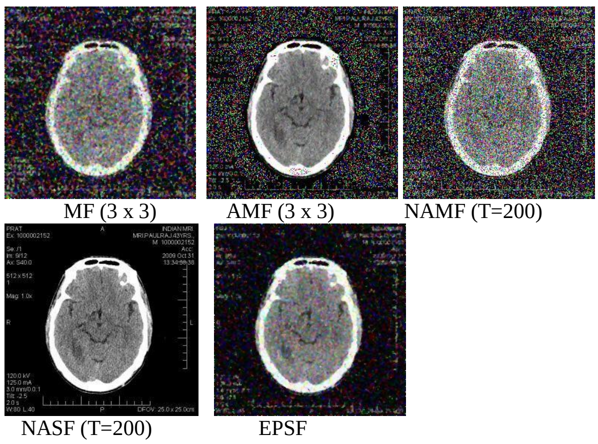


Fig 10: Shows enhanced Ct image after filtering from noise level = 40%

The filtered image obtained from each technique is analyzed with quantitative metrics discussed in the section III. The results of all the techniques, analyzed for the filtered image

(Tulips and Lena) with the metrics are shown in the TABLE I-II and 90% of corrupted images resultant value for each metrics shown as graph in Fig. 11-15.

In each graph x axis specifies all test images and y axis denotes the corresponding value derived from the exact metric for each of the proposed method.

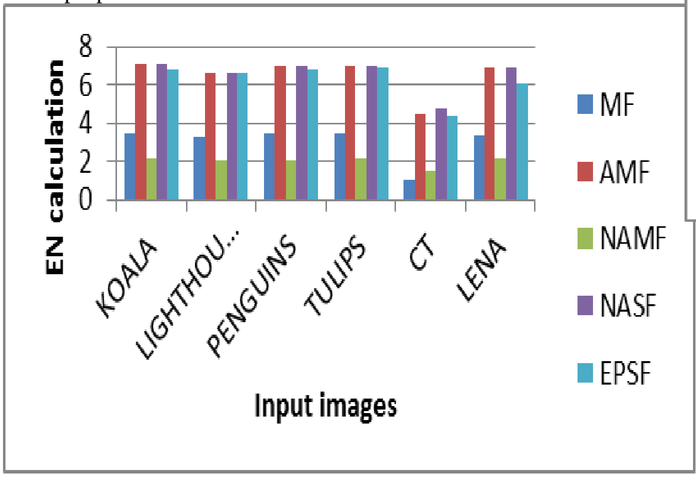


Fig 11: Graph -1 Entropy Calculation

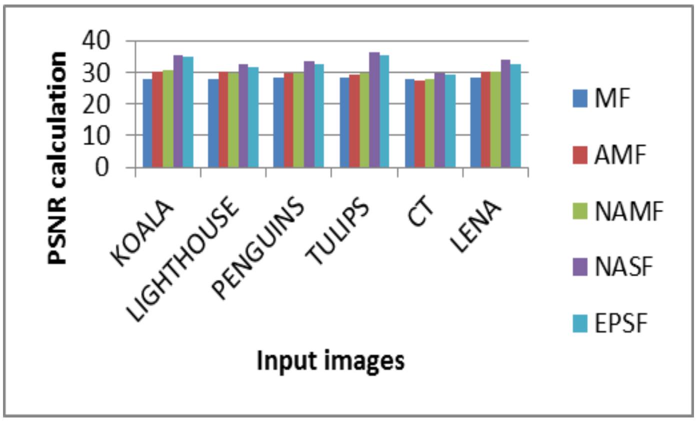


Fig 12: Graph -2 PSNR Calculation

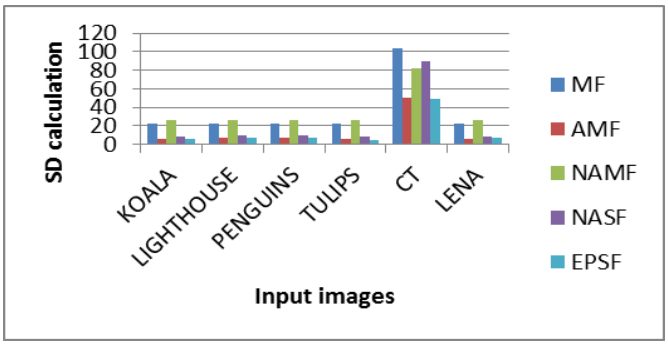


Fig 13: Graph -3 SD Calculation

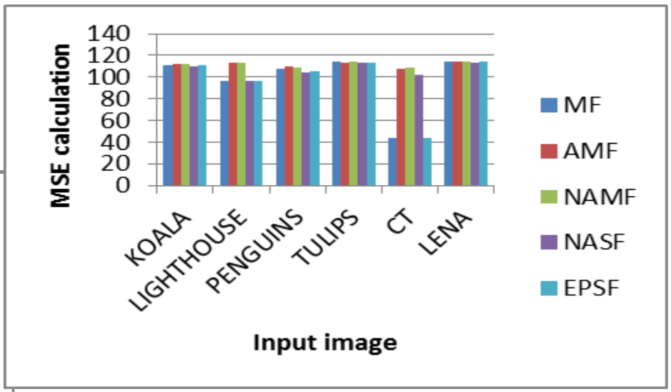


Fig 14: Graph-4 MSE Calculation

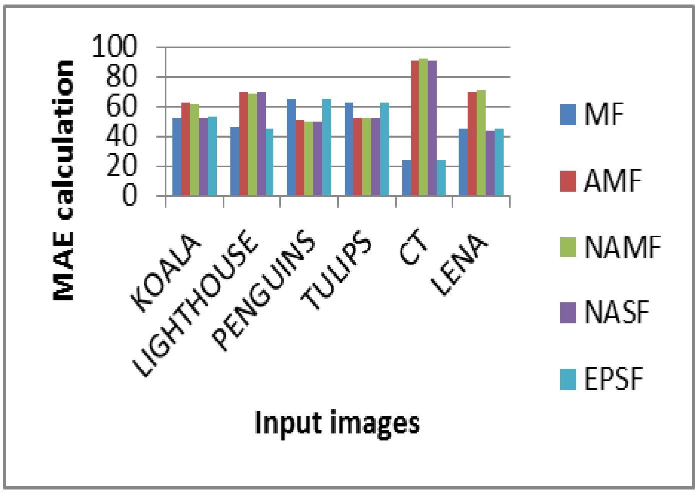


Fig 15: Graph -5 MAE Calculation

TABLE I: COMPARISON OF IMAGE FILTER ALGORITHM FOR TULIPS.JPG

Noise	Metrics	MF	AMF	NAMF	NASF	EPSF
0.1	EN	7.6383	7.6690	7.4618	7.6623	6.7099
	PSNR	37.026	39.214	39.383	45.642	36.2253
	SD	1.6990	1.6733	3.0941	1.7404	1.1403
	MSE	12.775	12.806	12.822	12.402	12.5731
	MAE	7.0674	5.8722	5.8603	5.7662	6.8819
0.2	EN	7.6504	7.6651	7.0446	7.6075	6.9213
	PSNR	34.933	37.188	36.8707	44.378	37.514
	SD	1.7085	1.6733	5.7809	2.2163	1.1645
	MSE	25.437	25.820	25.328	25.197	25.238
	MAE	14.020	11.617	11.443	11.374	13.920
0.3	EN	7.6518	7.6624	6.5417	7.5341	6.9839
	PSNR	33.259	34.901	34.799	40.532	37.914
	SD	1.6818	1.6902	8.5845	2.9557	1.1762
	MSE	38.663	38.393	38.220	38.120	38.195
	MAE	21.367	17.334	17.319	17.302	20.965
0.4	EN	7.6154	7.6547	5.9901	7.4515	6.9333
	PSNR	32.358	33.663	32.823	38.726	36.914
	SD	1.9425	1.7199	11.346	3.8168	1.1811
	MSE	50.582	50.423	50.337	50.249	50.740
	MAE	27.640	22.702	22.815	22.210	28.105
0.5	EN	7.4275	7.6421	5.3260	7.3633	6.7756
	PSNR	31.068	32.998	32.522	38.993	36.794
	SD	3.2682	1.7769	14.346	3.8168	1.1904
	MSE	63.882	63.232	63.240	62.552	64.652
	MAE	35.040	28.547	28.786	27.774	28.105
0.6	EN	7.0164	7.6333	4.6251	7.2717	6.4685
	PSNR	29.061	31.601	31.5052	38.012	37.619
	SD	5.9094	1.8374	17.251	4.7407	1.1875
	MSE	76.557	76.700	75.566	75.292	75.516
	MAE	42.080	34.701	34.300	34.156	35.6727
0.7	EN	6.2695	7.6156	3.8816	7.1781	6.7490
	PSNR	28.785	30.773	30.687	37.990	37.665
	SD	9.9598	1.9378	20.115	6.6919	1.1606
	MSE	89.658	89.233	88.363	88.197	88.869
	MAE	49.388	40.315	39.976	39.216	41.556
0.8	EN	5.0464	7.5376	3.0418	7.0833	6.2001
	PSNR	29.266	30.468	30.381	37.562	37.387
	SD	15.526	2.4224	23.072	7.7019	1.0837
	MSE	101.98	101.51	102.678	101.35	101.241
	MAE	56.376	45.889	46.361	46.089	55.662
0.9	EN	3.4131	6.9460	2.1156	6.9880	6.8625
	PSNR	28.265	29.382	29.928	36.452	35.587
	SD	21.774	6.2466	25.988	8.7296	4.3757
	MSE	114.23	113.80	114.879	113.51	113.658
	MAE	62.597	51.552	51.9651	51.494	62.725
1.0	EN	0.9995	0.9999	0.9999	6.8927	6.0712
	PSNR	27.870	29.471	29.543	35.984	35.872
	SD	28.745	28.737	28.737	9.7732	8.6900
	MSE	127.53	126.41	127.898	126.13	127.215
	MAE	70.326	57.210	57.943	57.100	69.906

TABLE II: COMPARISON OF IMAGE FILTER ALGORITHM FOR LENA.JPG

Noise	Metrics	MF	AMF	NAMF	NASF	EPSF
0.1	EN	7.5379	7.5849	7.3937	7.5855	7.0805
	PSNR	39.443	39.062	38.919	44.632	38.744
	SD	1.6990	1.6733	3.0941	1.7404	1.1403
	MSE	12.578	12.793	12.855	12.076	12.445
	MAE	7.8910	7.8285	7.8746	7.0080	8.9504
0.2	EN	7.5467	7.5897	6.9982	7.5315	7.112
	PSNR	36.590	36.699	37.578	43.347	37.607
	SD	1.7085	1.6756	5.7809	2.2163	1.1645
	MSE	25.025	25.703	24.898	24.174	24.776
	MAE	15.929	15.706	15.244	15.321	15.7127
0.3	EN	7.5619	7.5849	6.4962	7.4595	7.0791
	PSNR	33.558	35.412	35.608	41.283	40.568
	SD	1.6818	1.6902	8.5845	2.9557	1.1811
	MSE	37.892	37.797	37.812	37.583	37.620
	MAE	23.033	23.149	23.156	23.021	23.769
0.4	EN	7.5345	7.5825	5.9443	7.3782	6.9880
	PSNR	31.945	35.251	35.113	39.286	38.689
	SD	1.9425	1.7199	11.346	3.8168	1.1904
	MSE	49.799	50.964	50.271	49.750	50.855
	MAE	19.575	31.321	30.835	19.305	19.915
0.5	EN	7.3532	7.5765	5.3042	7.2914	6.8094
	PSNR	31.058	32.351	35.927	38.470	37.297
	SD	3.2682	1.7769	14.348	4.7407	1.1875
	MSE	63.407	63.571	63.286	63.082	63.710
	MAE	38.979	38.974	38.667	38.092	38.158
0.6	EN	6.9618	7.5664	4.6184	7.2013	6.4899
	PSNR	30.158	32.219	32.483	36.201	35.491
	SD	5.9094	1.8374	17.251	5.7028	1.1606
	MSE	75.717	75.722	76.282	75.157	76.714
	MAE	29.842	46.309	46.777	29.044	29.351
0.7	EN	6.1564	7.5542	3.8582	7.1090	6.7972
	PSNR	29.754	31.430	31.694	35.632	34.463
	SD	9.9598	1.9378	20.115	6.6919	1.0837
	MSE	88.390	88.839	89.080	87.590	87.903
	MAE	34.868	54.495	54.430	34.210	34.391
0.8	EN	5.0221	7.4890	3.02846	7.0156	6.1124
	PSNR	28.279	31.056	30.871	34.001	33.377
	SD	15.526	2.4224	23.072	7.7019	9.3757
	MSE	101.65	101.10	103.63	100.05	100.53
	MAE	40.026	61.826	63.363	39.079	39.645
0.9	EN	3.3443	6.9031	2.1355	6.9217	6.8198
	PSNR	28.279	30.128	30.028	33.990	32.463
	SD	21.774	6.2466	25.988	8.7296	7.4885
	MSE	114.78	113.95	114.78	113.48	113.93
	MAE	44.678	69.686	70.235	44.024	44.658
1.0	EN	0.9999	0.9999	0.9999	6.8277	6.0780
	PSNR	27.520	29.789	29.729	32.892	31.508
	SD	28.745	28.737	28.737	9.7732	8.6900
	MSE	126.95	127.93	126.95	125.95	126.61
	MAE	49.282	78.181	77.655	48.343	49.846

5. CONCLUSIONS

In this paper, five different filtering techniques with quantitative metrics (performance evaluation) are analyzed with six different types of images contaminated with salt-and-pepper noise of varying densities such as CT, Koala, Penguins, Tulips, Lighthouse and Lena. The result shows that NASF technique outperforms than remaining filter techniques from the visual perspective which is also verified with the quantitative metrics. From Table I, II and figure 11, 12, 13, 14, 15 observed that higher values of EN and PSNR provides enhanced information for NASF, lower values of SD ensure that contrast information for EPSF, a lower values of MSE that is a better reconstruction of original image from the noisy image, similarly MAE is much better predictor of image quality with a lower value resulting in a good image. These filtering techniques lead to many computational merits in reality which includes: efficient retrieval, reorganization objects easily, and can able to diagnosis disease.

In future work, we are interested to discover the effective other filter techniques for different types of noisy images and improve the performance of the system.

6. REFERENCES

[1] Sanjay Singh , Chauhan R.P.S, Devendra Singh "Comparative Study of Image Enhancement using Median and High Pass Filter-ing Methods", Journal of Information and Operations Management, vol. 3,pp.96-98, 2012.

[2] Muhammad Mizanur Rahman, Faisal Ahmed, Mohammed Imrul JubairSyed Ashfaqueuddin Priom, Imtiaz Masud Ziko "An Enhanced Non-Linear Adaptive Filteing Technique for Removing High Density Salt-and-Pepper Noise", International Journal of Computer Applications, vol.38, no.11, pp.7-12, January 2012.

[3] Punyaban Patel, Banshidhar Majhi, Abhishek Tripathi, C.R. Tripathy, "A New Adaptive Median Filtering Technique for Removal of Impulse Noise from Images", ICCCS'11, pp.462-467, February 2011.

[4] Amiya Halder, Sandeep Shekhar, Shashi Kant, Musheer Ahmed Mubarki, Anand Pandey "A New Efficient Adaptive Spatial Filter for Image Enhancement", Second International Conference on Computer Engineering and Applications, pp.244-246, 2010.

[5] Shawn Chen, Tian-Yuan Shih, "On the evaluation of edge preserving smoothing filter, In Proceeding Geoinformatics, p.p C43, 2002.

[6] S.Rajkumar, S.Kavitha, " Redundancy Discrete Wavelet Transform and Contourlet Transform for Multimodality Medical Image Fusion with fQuantitative Analysis", 3rd International Conference on Emerging Trends in Engineering and Technology, November 2010.

[7] Shuyue Chen, Yun Qiu, Jun Feng, "A Nonlinear Filter Based on Row and Column Operation for Positive Impulsive Noise Reduction", Asia-Pacific Conference on Information Processing, pp.44-46, 2009.

[8] Zhou Dengwen and Shen Xiaoliu, "Image Denoising Using Weighted Averaging", International Conference on Communications and Mobile Computing, pp.400-403, 2009.

[9] Banshidhar Majhi," Soft Computing Techniques for Image Restoration", PhD Thesis, Sambalpur University, 2003.

[10] Krishan Kant Lavania, Shivali, Rajiv Kumar, " Image Enhancement using Filtering Techniques", International Journal of Computer Science and Engineering, vol.4, pp.14-20, January 2012.

[11] A.K.Jain, "Fundamentals of digital image processing", Prentice-Hall of India,1989

[12] Gonzalea R.C, Woods R.E., "Digital Image Processing", 3rd edition, Pearson Education, 2009.