Cascaded Sorting Approach for Fixed Impulse Denoising

A. Rajamani
Department of ECE
PSG Polytechnic College
Coimbatore, India
Email: rajamani_saranya [AT] rediffmail.com

N. Nandhini Priya
Department of Biomedical Engg
PSG College of Technology
Coimbatore, India

Abstract— **Noise removal plays an important role in image processing applications. In this paper an algorithm is designed and implemented to remove the fixed impulse noise (salt and pepper) from corrupted digital images and video frames. This paper consists of three stages for removing the noise. In the first stage, a decision based median filter is used to identify noisy pixels. If the processed pixel is noisy, then it is replaced by the median value of the pixels of the selected window. In the second stage the noisy pixels are replaced diagonally by the median value. Finally, in third stage, the noisy pixels are replaced in horizontal plane wise. The experimental results show that the proposed cascaded sorting approach gives excellent results when compared to existing filters. The proposed algorithm is tested against different gray scale, color digital images and video frames. It gives better results for the quantitative metrics such as Peak Signal-to-Noise Ratio (PSNR), Mean Absolute Error (MAE), Normalized Correlation Coefficient (NCC), Normalized Absolute Error (NAE) and Quality Index (QI) with good edge preservation capabilities even at higher noise densities.**

Keywords- Decision based median filter; Salt and pepper noise; Noisy pixel; Cascaded sorting

I. INTRODUCTION

Processing on the digital images [2] and its applications play an important role in every day to day life. It is a vast domain where image is composed of many pixels and various operations are performed on images. When an image is obtained by the image acquisition devices, lighting and camera characteristics cause changes in the appearance and quality of the image, and is called noise. Moreover, noise hides the important details of images.

Image filtering techniques are mainly used for removing noise, sharpening contrast, Highlighting contours and detecting edges. Images in general are frequently [3] corrupted by various types of noise such as random noise, gaussian noise and impulse noise to name a few. The two types of impulse noise namely random valued impulse noise and fixed valued impulse noise. If the intensity value lies within a predetermined range then it is called random valued impulse noise. If it is fixed then it is said to be fixed impulse noise. Fixed impulse noise otherwise known as salt and pepper noise can corrupt the images by taking the pixel value either maximum or minimum. Impulse noise mostly occurs in digital images because of bit change, data transmission errors and data compression. [4]. Impulse noise reduction in images plays dominant task, since it is important preprocessing step for further analysis on images [5].

II. SOME EXISTING DENOISING TECHNIQUES

Image filters are broadly classified into two types namely linear and non linear filters. Linear filters are also known as convolution filters as they can be represented using matrix multiplication, whereas thresholding and image equalization are examples of nonlinear operations, as in the median filter. Linear filters are generally tends to blur edges and other image detail and it is performing poor for non-Gaussian noise.

Non-linear filters depend on the pixels of selected window of the image segment under consideration, and which is covered by the filter and then the center pixel's value is replaced with the value obtained from the ranking or sorting result. The famous nonlinear median filter [6] plays dominant role in impulse denoising and works by substituting the pixel value by median value of the neighborhood pixel in the current filter window under consideration. Some of the existing denoising techniques are discussed as follows

Standard Median Filter (SMF) is a popular non linear filter for removing salt and pepper noise, because of its computational efficiency and good denoising power [7].The limitation of this filter is that the pixel value is replaced by median value of the corresponding neighborhood pixel regardless of whether it is corrupted or not. But, it has other advantage as simplest procedure with better edge preservation. This type of filtering needs the arrangement of the pixel values either by increasing or decreasing order followed by the replacement of the noisy pixel by the median or mean of its neighborhood values. The main demerit of the SMF is that it changes both noisy and non-noisy pixel values. Thereby, it removes some fine details of the image.

Adaptive Median Filter (AMF) is used to eliminate the problems occurred in SMF. These two filters di No y having the adaptive median filter window size as variable. This variation depends on the noise density level in the processed 3x3 window. If the centre pixel value is an impulse noise, then the size of the window is expanded, otherwise it's left unchanged. Thus, if the pixel being considered is not an impulse, then the gray scale value of the pixel in the filtered image is same as that of input image. That is it can perform filtering only when the corrupted image with impulse noise of probability greater than 0.2. AMF meant for both noise removal and resolving distortion in the image. Additionally, this filter smoothes other types of noise with much better results compared to standard median filter [9]. It has the limitations such as AMF works well at low noise level but not at high noise level and moreover the window size is to be maximized which in turn causes blurring effect on the image.

Progressive Switching Median Filter (PSMF) is a median based filter [8] mainly works on the basis of two steps. In first step, an impulse search algorithm is employed to provide a chain of binary flag images and this flag computes the location of noise in the image under consideration. Noise filtering is done in the second step through much iteration. This filter can perform better only for fixed valued impulse noise but performs worst on random valued impulse noise. The merits of this method are, it saves the image boundaries, useful for visual examination and measurement. But, it has limitation as it performs better only up to 40% noise level.

The Decision Based Algorithm (DBA) is also median based filter [4] in which the pixel value present inside the window is arranged in ascending order and the middle element of the window is taken as median value of window. Firstly, the decision is taken to know whether the pixel under consideration is noisy or not and after that filtering could be done if the processing pixel is noisy. The disadvantage of this method is that when the noise density level is high, the pixel which is used to replace the noisy pixel is one of the neighbor pixel, and which is repeatedly utilized for replacing the noisy pixel that minimizes quality of denoised images called streaking effect. In order to avoid this drawback cascaded sorting algorithm is proposed in this paper for images and video frames.

III. PROPOSED ALGORITHM

The proposed cascaded sorting algorithm for images and video frames works to remove fixed impulse noise or salt and pepper noise. The proposed algorithm first detects the salt and pepper noise present in the image. The corrupted and uncorrupted pixels in the image are detected by checking the pixel element value against the maximum and minimum values (0,255) in the selected window. For a gray scale image, the minimum and maximum value of the salt and pepper noise respectively as 0 and 255. If the pixel value of the processed window is within $0 < p_{2\times2} < 25$ then it is considered as a noise free pixel, and it is left unchanged. If the pixel value of the processed window is equal to 0 or 255 then it is considered as noisy pixel and replaced by the median value of the processed window.

- This paper has three cascaded steps of noise reduction scheme as follows:
- The processing pixel is checked for noise and replaced by median value of the selected 3x3 window
- Selecting the diagonal elements of the processing matrix , checks for the noisy pixel and replaced by median value
- Selecting the horizontal plane of the processing matrix, checks for noisy and replaced by median value.The steps of the cascaded sorting algorithm are elucidated as flowchart in Fig.2

A. *Proposed Cascaded Sorting Algorithm for Gray Scale Images*

- *Step 1:* Choose 2-D sliding window with size 3x3 as sub matrix in an image and centre pixel being processed as P2,2
- *Step 2*: If $0 < P_{2,2} < 255$, then $P_{2,2}$ is an uncorrupted pixel and its value is left unchanged.
- *Step 3:* If $0 = P_{2,2} = 255$then $P_{2,2}$ is a corrupted pixel then the current window is replaced with the median of the element of window
- *Step 4:* Check the noisy pixels for the diagonal elements of the selected window and is again replaced with the median of the element of window
- *Step 5:* Select the elements of horizontal plane of the selected window and check for the noisy pixels and is replaced with the median of the element of window
- *Step 6:* Repeat from step 1 to 5 for all sliding window of the image.
- Each step is explained with a graphical illustration in Fig.1

B. *Proposed Algorithm Implementation for RGB images*

The Proposed Algorithm is also implemented for the color images. The color image channels consist of three band color images (Red, Green, and Blue)[10]. First, red band is selected among the three bands. Selecting a sliding window of window size 3X3 as sub matrix in an image and further process are same (step 2 to step 6) as that of the above discussed gray scale algorithm. The above process is repeated consequently for the green and blue band and the process is carried over for all three RGB arrays to denoise the entire color image.

C. *Implementation for Video Sequence*

The video sequences are first converted into frames [11] and successively frames are converted into images. Then the proposed algorithm is applied to the images that are separated from the image. After the completion of the filtering process, the frames are transformed back to the original motion picture. The algorithm is as follows:

- *Step 1:* The video sequences affected by impulse noise is converted into .avi format, which is an uncompressed format and the individual frames are extracted from the video
- *Step 2:* For further processing, the extracted frames are converted into images.

- *Step 3:* Then the images which are affected by impulse noise are denoised by the proposed algorithm
- *Step 4*: After completion of the above process, the denoised frames are finally collected back to get the original movie or video.

Fig.1 Graphical illustration of steps

Φιγ.2 Οϖεραλλ φλοωχηαρτ οφ τηε Προποσεδ Αλγοριτημ

IV. PERFORMANCE ANALYSIS

In order to estimate the performance of the proposed noise removal algorithm quantitative wise, the following performance metrics are analyzed:

- Peak Signal-to-Noise Ratio (PSNR)
- Mean Absolute Error (MAE)
- Normalized Correlation Coefficient (NCC)
- Normalized Absolute Error (NAE)
- Quality Index (QI)

The results of these parameters are analyzed and based on the inferences drawn from the performance; the proposed filter performance is analyzed.

A. Peak Signal-to-Noise Ratio (PSNR):

Peak Signal-to-Noise Ratio is ratio between the highest feasible power of a signal and the power of noise that causes the fidelity of its representation.

$$PSNR = 20.\log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right)$$

B. Mean Absolute Error (MAE):

Mean Absolute Error is a quantity used to measure the sum of absolute difference between input image and the filtered image to the number of pixels present in the image

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|f_i - y_i|$$

Mean absolute error is the average of an absolute error, where is the filtered image and is the input image

C. Normalized Cross Correlation (NCC):

Normalized Cross Correlation can be computed by

$$NCC = \sum_{i=1}^{M}\sum_{j=1}^{N}[g(i,j).g'(i,j)] / \sum_{i=1}^{M}\sum_{j=1}^{N}(g(i,j))2$$

If the Normalized Cross Correlation tends to 1, then the image quality seems to be better.

D. Normalized Absolute Error (NAE):

Normalized absolute error can be computed by

$$NAE = \sum_{i=1}^{M}\sum_{j=1}^{N}|[g(i,j).g'(i,j)]| / \sum_{i=1}^{M}\sum_{j=1}^{N} g(i,j)$$

Normalized absolute error should be minimum, in order to minimize the difference between original and filtered image. The smallest value of NAE means that the removal of noise in image provides better result

E. Quality Index (QI):

Quality index is calculated by rating each characteristics using following formulae:-

$$C(j) = \frac{\sum_{i=N(j)} S(i)}{N \cdot j)}$$

where (j) is the rating of quality characteristics (i) is the rating of quality Sub-characteristics on which C(j) is judged,N(j) is the number of sub- characteristics not having sub- characteristics N(j) will be 1.

$$QI = -\frac{\sum_{j=n} C(j)}{n} * 100$$

where QI=quality index, N= number of quality characteristics.

The proposed algorithm is also tested qualitatively by using 512 512 size images Lena (Gray), Lena (RGB) and sample video for all noise levels (upto 90%). At each time the test image is corrupted by salt and pepper noise of different noise density ranging from 10%-90%.The qualitative performance of the proposed technique and its comparison with existing Lone Diagonal Sorting (LDS) algorithm is shown below and quantitative analysis metrics results of PSNR, MAE, NCC, NAE and QI have been tabulated as in Tables 1, 2, 3&4.

Fig.3 Results obtained for 80% and 90% noise corrupted Lena (Gray) image and their denoised results

Fig.3(a) shows original lena input image (512X512), 3(b) shows 80% (1) noise corrupted image, 3(c) denoised image of LDS algorithm, and 3(d) shows the denoised result of the proposed algorithm 3(e) shows 90% noise corrupted image 3(f) denoised image of LDS algorithm, and 3(g) shows the denoised result of the proposed algorithm.

Fig.4 Results obtained for 80% and 90% noise corrupted Lena (RGB) image and their denoised results (2)

Fig.4(a) shows original RGB lena input image (512X512), 4(b) shows 80% noise corrupted image, 4(c) denoised image of LDS algorithm, and 4(d) shows the denoised result of the proposed algorithm 4(e) shows 90% noise corrupted image, 4(f) denoised image of LDS algorithm, and 4(g) shows the denoised result of the proposed algorithm. (3)

TABLE I. PERFORMANCE RESULTS OF PSNR

Table1 and Fig.5 presents the comparison of PSNR values for existing and proposed technique at various noise densities ranging from 10%-90%, From the graph, it is observed that the proposed sorting algorithm shows higher PSNR value even at high noise densities compared to other filters. (4)

Fig.5 PSNR graph of gray scale image (Lena)

TABLE II. PERFORMANCE RESULTS OF MAE

Table2 and Fig.6 presents the comparison of MAE values for existing and proposed technique at various noise densities ranging from 10%-90% and it is observed that the proposed algorithm shows lower MAE value even at high noise densities compared to other filters.

Fig.6 MAE graph of gray scale image (Lena)

TABLE III PERFORMANCE RESULTS OF NCC

Table3 and Fig.7 presents the comparison of NCC value for existing and Proposed technique at various noise densities ranging from 10%-90%. From the graph, it is observed that the proposed algorithm shows higher NCC value even at high noise densities compared to other filters.

Fig.7 NCC graph of gray scale image (Lena)

TABLE IV PERFORMANCE RESULTS OF NAE

Table4 and Fig.8 presents the comparison of NAE values for existing and proposed technique at various noise densities ranging from 10%-90%, and from the graph, it is observed that the proposed algorithm shows higher NAE value even at high noise densities compared to other filters.

Fig.8 NAE graph of gray scale image (Lena)

TABLE V PERFORMANCE RESULTS OF QUALITY INDEX (QI)

Table5 and Fig.9 presents the comparison of Quality Index (QI) value for existing and proposed technique at various noise densities ranging from 10%-90%. From the graph, it is observed that the proposed algorithm shows higher QI value even at high noise densities compared to other filters.

Fig.9 QI graph of gray scale image (Lena)

TABLE VI PERFORMANCE RESULTS OF PSNR FOR RGB LENA IMAGE

Table6 and Fig.10 presents the comparison of PSNR value for existing and Proposed technique at various noise densities ranging from 10%-90%.The plotted graph clearly shows that the proposed algorithm shows higher PSNR value even at high noise densities compared to other filters.

Fig.10 PSNR for RGB image (Lena)

TABLE VII PERFORMANCE RESULTS OF MAE

Table7 and Fig.11 presents the comparison of MAE value for different filters at various noise densities ranging from 10%-90%. The plotted graph clearly shows that the proposed algorithm shows lower MAE value even at high noise densities compared to other filters.

Fig.11 MAE for RGB image (Lena)

TABLE VIII PERFORMANCE RESULTS OF NCC

Table8 and Fig.12, presents the comparison of NCC value for different filters at various noise densities ranging from 10%-90%. The plotted graph clearly shows that the proposed algorithm shows higher NCC values even at high noise densities compared to other filters.

Fig.12 NCC for RGB image (Lena)

TABLE IX PERFORMANCE RESULTS OF NAE

Table9 and the Fig.13 presents the comparison of NAE value for different filters at various noise densities ranging from 10%-90%. The plotted graph clearly shows that the proposed algorithm shows higher NAE value even at high noise densities compared to other filters.

Fig.13 NAE for RGB image (Lena)

TABLE X PERFORMANCE RESULTS OF QI

Table10 and the Fig.14 presents the comparison of QI value for different filters at various noise densities ranging from 10%-90%. The plotted graph clearly shows that the proposed algorithm shows higher QI value even at high noise densities compared to other filters.

Fig.14 QI for RGB image (Lena)

TABLE XI PERFORMANCE RESULTS OF PSNR FOR VIDEO

Table11 and the Fig.15 presents the comparison of PSNR values for different filters and the plotted graph clearly shows that the proposed algorithm shows higher PSNR value even at high noise densities compared with existing technique.

Fig.15 PSNR for video

TABLE XII PERFORMANCE RESULTS OF MAE FOR VIDEO

Table12 and Fig.16 presents the comparison of MAE value of different filters for video. The plotted graph clearly shows that the proposed algorithm shows lower MAE value even at high noise densities compared with existing technique.

Fig.16 MAE for video

TABLE XIII PERFORMANCE RESULTS OF NCC FOR VIDEO

Table13 and Fig.17 presents the NCC value of different filters for video and the plotted graph clearly shows that the proposed algorithm shows higher NCC value even at high noise densities compared with existing technique.

Fig.17 NCC for video

TABLE XIV PERFORMANCE RESULTS OF NAE

Table14 and Fig.18 presents the comparison of NAE value for existing and proposed technique at various noise densities ranging from 10%-90%. The plotted graph clearly shows that the proposed algorithm shows higher NAE value even at high noise densities compared with existing techniques.

Fig.18 NAE for video

TABLE XV PERFORMANCE RESULTS OF QI

Table15 and Fig.19 presents the comparison of QI value for existing and proposed technique at various noise densities ranging from 10%-90%. The plotted graph clearly shows that the proposed algorithm shows higher QI value even at high noise densities compared with existing techniques

Fig.19 QI for video

V. CONCLUSION

In this paper, a modified cascaded sorting algorithm is proposed to remove noise in gray, color images and video sequences. It gives better performance when compared to existing noise removal algorithms in terms of performance metrics PSNR, MAE, NCC, NAE, and QI etc. The proposed cascaded sorting algorithm has been tested at low, medium and high noise densities on gray-scale, color images and video also. When compared to existing algorithms, the cascaded sorting algorithm gives better results even at high noise density levels. In future, this work can be extended for other type of noise such as Gaussian and Speckle noise, which also affects the image abundantly.

REFERENCES

[1]. C.Rafael,Gonzalez, Richard and E. Woods, "Digital Image Processing" Prentice-Hall, India, second edition, 2007

[2]. ChristyldaAngelin Hannah. J, Natheldha and S.Mary Navina," A Survey on Image Denoising" ,International Journal of Engineering Research and Applications (IJERA), vol. 3, Issue 1, January-February 2013, pp.153-156

[3]. R.Avadhoot Telepatil,S.A.Patil,andP..Vishal.Paramane," A Survey on Median Filters for Removal of High Density Salt & Pepper Noise in Noisy Image", IOSR Journal of Electronics and communication Engineering (IOSR-JECE) ISSN: 2278-2834, ISBN: 2278-8735

[4]. N.Ramanaiah and V.Satish Kumar ," Removal Of High Density Salt And Pepper Noise In Images And Videos Using Denoising Methods", IJCSMC, Vol. 2, Issue. 10, October 2013

[5]. Rosebell John, V. Karunakaran," A Survey on Denoising Methods", IOSR Journal of Engineering (IOSRJEN), e-ISSN: 2250-3021, p-ISSN: 2278-8719, Volume 2, Issue 10,October 2012

[6]. A.Rajamani, and V. Krishna veni,"A New Denoising Approach for the Removal of Impulse noise from color images and video sequences", Image Anal Stereol, 2012.

[7]. R. Rohini, Varade,M. R. Dhotre, and B. Archana Pahurka," A Survey on Various Median Filtering Techniques for Removal of Impulse Noise from Digital Images.", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 2, Issue 2, February 2013

[8]. Zhou Wang and David Zhang, " Progressive Switching Median Filter for the Removal of Impulse Noise from Highly Corrupted Images" IEEE Transactions on Circuits and Systems, Analog and Digital Signal Processing, Vol. 46, No. 1, January 1999.

[9]. V.Krishnaveni,and A.Rajamani,"Denoising by a Modified Sorting and Trimming Median Filter",International Journal of Electronic and Communication Research,vol.5,no.1,2014.

[10]. V.Krishnaveni,and A.Rajamani,"Impulse Denoising Algorithm for Gray and RGB images",International Journal of Computer Applications,vol.70,no.2,2013

APPENDIX

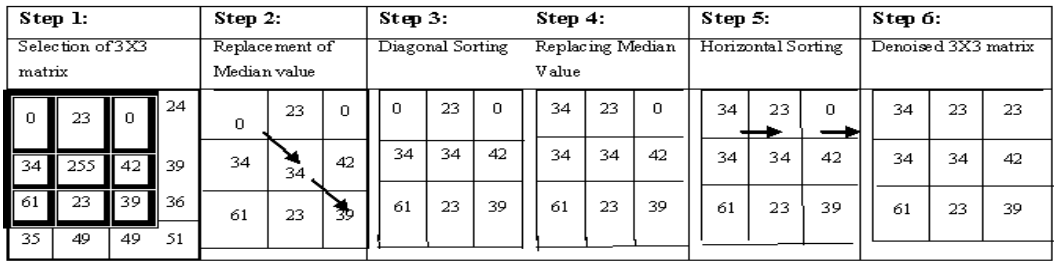

Fig.1Graphical illustration of steps

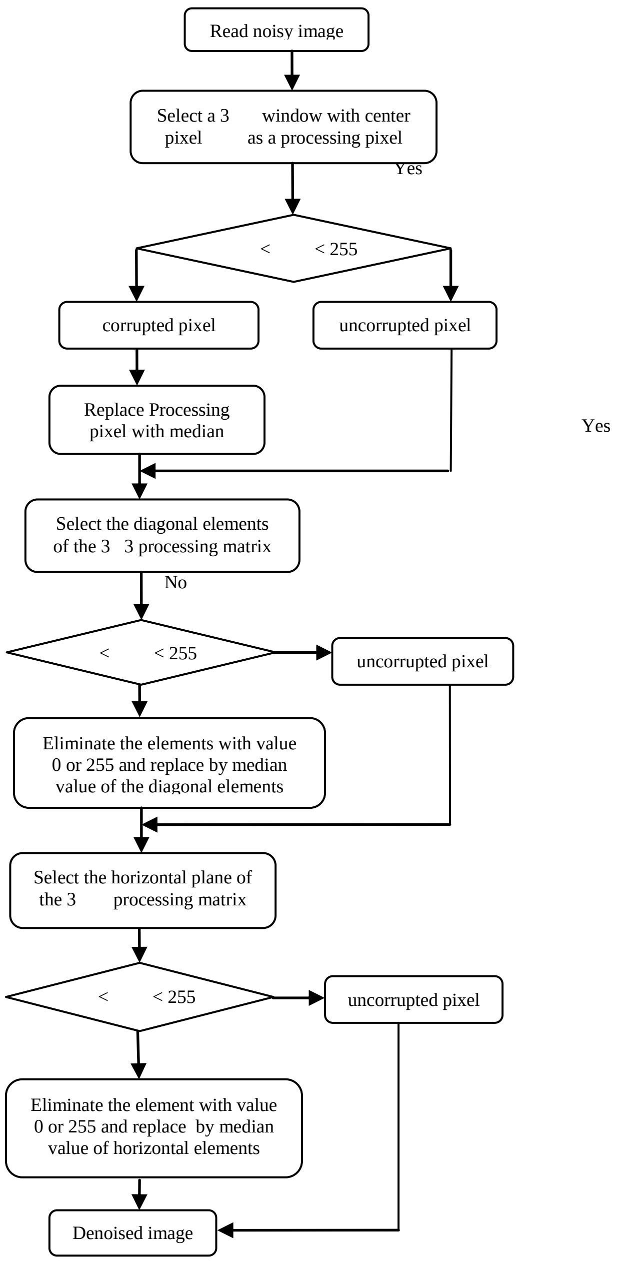
Read noisy image
Select a 3 window with center pixel as a processing pixel
Yes
< < 255
corrupted pixel
uncorrupted pixel
Replace Processing pixel with median
Yes
Select the diagonal elements of the 3 3 processing matrix
No
< < 255
uncorrupted pixel
Eliminate the elements with value 0 or 255 and replace by median value of the diagonal elements
Select the horizontal plane of the 3 processing matrix
< < 255
uncorrupted pixel
Eliminate the element with value 0 or 255 and replace by median value of horizontal elements
Denoised image

Fig.2 Overall flowchart of the Proposed Algorithm

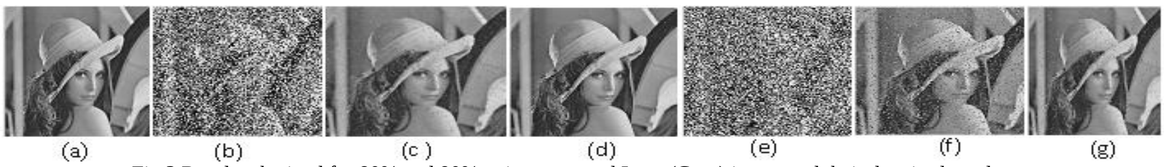

Fig.3 Results obtained for 80% and 90% noise corrupted Lena (Gray) image and their denoised results

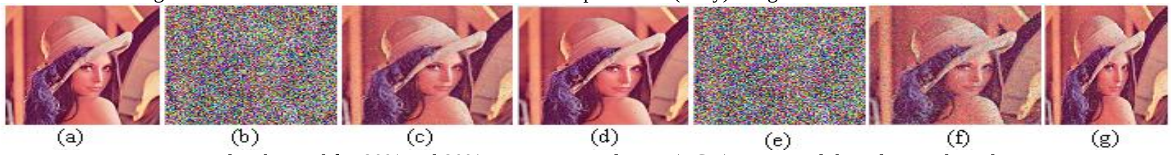

Fig.4 Results obtained for 80% and 90% noise corrupted Lena (RGB) image and their denoised results

TABLEI PERFORMANCE RESULTS OF PSNR

Noise(%)	MF	SMF	AMF	PSMF	WMF	AMWF	TDF	DBA	MDBA	MDB UTMF	MAUT MPF	LDS	Proposed Cascaded Sorting
10	26.34	22.75	29,48	30.22	34.22	34.90	35.53	38.43	36.94	37.91	42.53	45.3895	49.8337
30	21.86	15.30	27.1	25.52	21.66	28.06	31.73	35.92	30.41	32.29	37.99	39.6542	43.2366
50	15.04	11.82	24.04	19.13	14,22	25.62	29.78	32.21	26.52	28.18	35.37	32.6879	38.6583
80	8.68	9.08	11.60	8.02	7.90	20.67	22.51	26.23	20.44	21.70	30.70	22.4639	26.3949
90	6.65	8.25	8.002	6.57	6.56	19.04	20.04	23.94	17.56	18.40	28.17	19.2101	22.1020

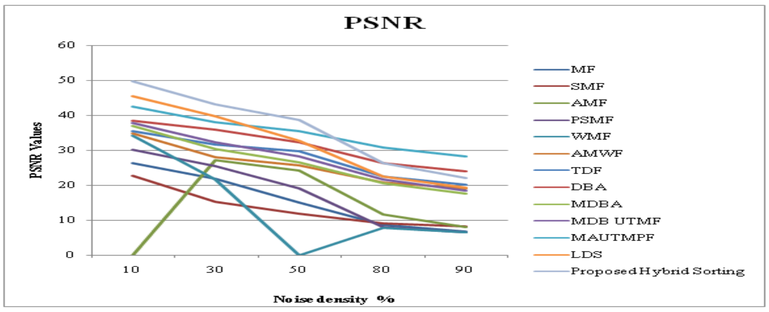

Fig.5 PSNR graph of gray scale image (Lena)

TABLE II: PERFORMANCE RESULTS OF MAE

Noise (%)	MF	SMF	AMF	PSMF	WMF	AMWF	TDF	DBA	MDBA	MDB UTMF	MAUT MPF	LDS	Proposed Cascaded Sorting
10	32.55	25.90	33.76	24.90	20.34	21.52	23,54	20.64	9.30	4.56	3.63	1.0256	0.2848
30	125.80	117.50	83.53	43.40	179.56	180.99	184.49	56.10	29.10	26.34	10.33	3.0132	1.0046
50	787.01	677.04	147.34	352.14	895.50	897.64	898.67	113.12	41.58	41.23	18.87	4.5687	2.0298
80	4565.10	3464.50	517.56	2205.32	3672.36	3675.98	3679.68	305.39	169.61	118.90	55.36	12.3654	10.1889
90	5514.21	4883.21	1041.99	3987.35	5031.06	5099.1	5132.98	730.72	240.925	134.00	99.00	25.6542	22.6244

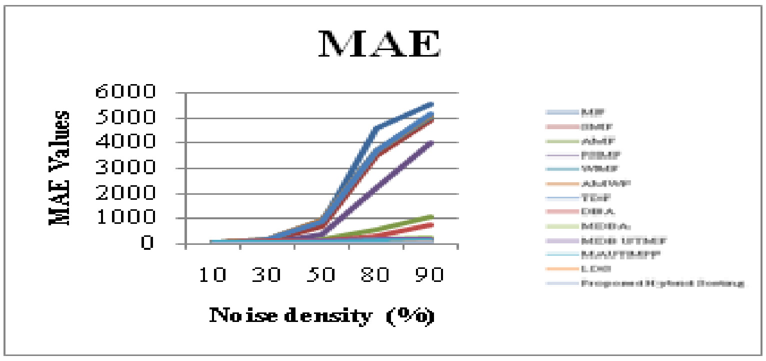

Fig.6 MAE graph of gray scale image (Lena)

TABLE III PERFORMANCE RESULTS OF NCC

Noise Density (%)	MF	SMF	AMF	PSMF	WMF	AMWF	TDF	DBA	MDBA	MDB UTMF	MAUT MPF	LDS	Proposed Cascaded Sorting
10	0.0952	0.1014	0.2299	0.3011	0.3190	0.3452	0.4012	0.4800	0.5277	0.5410	0.7201	0.7856	1.0000
30	0.1163	0.1741	0.2578	0.3241	0.3251	0.3895	0.4029	0.4711	0.5441	0.5517	0.6891	0.7352	1.0002
50	0.1265	0.1841	0.2981	0.3341	0.3397	0.3971	0.4264	0.4899	0.5529	0.5674	0.6821	0.6999	0.9999
80	0.1385	0.1920	0.3121	0.3401	0.3481	0.4112	0.4398	0.5011	0.5671	0.5912	0.6800	0.6878	0.9896
90	0.1487	0.2014	0.3541	0.3490	0.3754	0.4210	0.4419	0.5214	0.5780	0.6100	0.6241	0.6547	0.9649

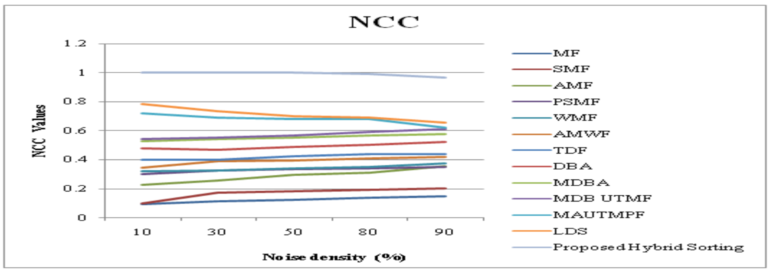

Fig.7 NCC graph of gray scale image (Lena)

TABLE IV PERFORMANCE RESULTS OF NAE

Noise Density (%)	MF	SMF	AMF	PSMF	WMF	AMWF	TDF	DBA	MDBA	MDB UTMF	MAUT MPF	LDS	Proposed Cascaded Sorting
10	0.0010	0.0040	0.0067	0.0072	0.0099	0.0108	0.0120	0.0135	0.0190	0.0228	0.0254	0.0265	0.0023
30	0.0121	0.0241	0.0299	0.0310	0.0390	0.0410	0.0499	0.0540	0.0587	0.0599	0.0720	0.0984	0.0081
50	0.0300	0.0320	0.0428	0.0440	0.0480	0.0520	0.0549	0.0680	0.0720	0.0894	0.0998	0.1123	0.0164
80	0.0321	0.0350	0.0432	0.0467	0.0499	0.0532	0.0587	0.0691	0.0731	0.0762	0.0865	0.0952	0.0412
90	0.0487	0.0512	0.0987	0.1088	0.1582	0.1969	0.2198	0.2671	0.2986	0.3214	0.3364	0.3562	0.1824

Fig.8 NAE graph of gray scale image (Lena)

TABLE V: PERFORMANCE RESULTS OF QUALITY INDEX (QI)

Noise Density (%)	MF	SMF	AMF	PSMF	WMF	AMWF	TDF	DBA	MDBA	MDB UTMF	MAUT MPF	LDS	Proposed Cascaded Sorting
10	0.1654	0.2111	0.2354	0.2874	0.3099	0.3478	0.3874	0.4666	0.5010	0.5579	0.5988	0.6213	0.9878
30	0.1541	0.1984	0.2224	0.2741	0.2984	0.3154	0.3654	0.4541	0.4987	0.5387	0.5945	0.6421	0.9544
50	0.0985	0.1055	0.1978	0.2150	0.2589	0.2987	0.3423	0.3899	0.4155	0.4660	0.4945	0.5214	0.8761
80	0.0100	0.0123	0.0135	0.0154	0.0211	0.0232	0.0354	0.0397	0.0439	0.0589	0.0645	0.0721	0.2856
90	0.0055	0.0096	0.0121	0.0159	0.0231	0.0264	0.0287	0.0321	0.0369	0.0375	0.0409	0.0413	0.1277

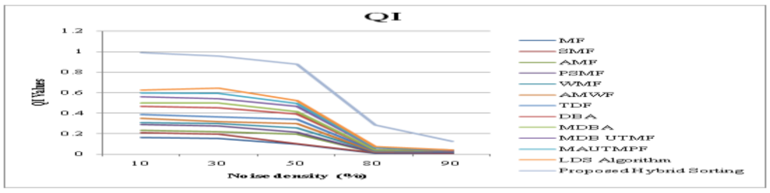

Fig.9 QI graph of gray scale image (Lena)

TABLE VI PERFORMANCE RESULTS OF PSNR FOR RGB LENA IMAGE

Noise Density (%)	MF	SMF	AMF	PSMF	WMF	AMWF	TDF	DBA	MDBA	MDB UTMF	MAUT MPF	LDS	Proposed Cascaded Sorting
10	30.009	32.019	33.456	34.965	36.897	38.896	40.165	43.124	44.568	45.256	47.586	49.2536	53.5862
30	23.969	24.201	25.667	27.213	29.001	30.998	33.567	35.235	36.986	38.012	39.986	42.2135	47.4925
50	15.968	17.008	19.216	20.002	22.364	24.654	26.879	29.996	32.896	34.985	36.256	37.1204	42.6794
80	7.856	8.241	9.300	10.548	12.789	14.985	16.2006	17.758	18.156	20.879	22.356	25.1023	30.1373
90	4.896	5.475	7.698	9.857	10.658	12.965	13.602	15.285	16.789	18.201	19.458	20.1403	25.7365

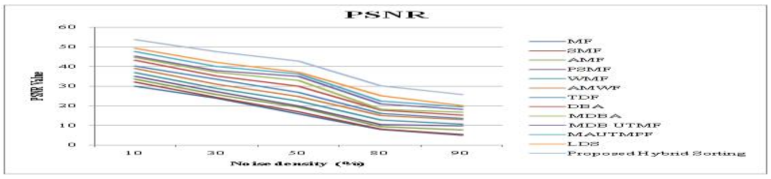

Fig.10 PSNR for RGB image (Lena)

TABLE VII PERFORMANCE RESULTS OF MAE

Noise Density (%)	MF	SMF	AMF	PSMF	WMF	AMWF	TDF	DBA	MDBA	MDB UTMF	MAUT MPF	LDS	Proposed Cascaded Sorting
10	2.9867	2.7984	2.5891	2.3269	2.0689	1.987	1.7892	1.5895	1.3256	0.8547	0.6889	0.4521	0.2886
30	3.9978	3.7894	3.5789	3.3789	3.1796	2.9784	2.8999	2.6048	2.4879	2.2478	2.0978	1.9875	0.9936
50	25.7894	23.1247	21.4785	19.7895	17.2246	15.4789	13.5897	11.2564	9.9999	8.4786	6.2456	4.0123	2.0200
80	34.2147	32.6745	30.5864	28.2134	26.6547	24.7894	22.6547	20.1117	18.0004	17.4789	15.7894	13.2546	11.0750
90	51.6532	49.2146	47.5684	45.0004	43.8975	41.3265	39.4562	37.6359	35.8975	33.6547	31.4578	28.0123	25.6593

Fig.11 MAE for RGB image (Lena)

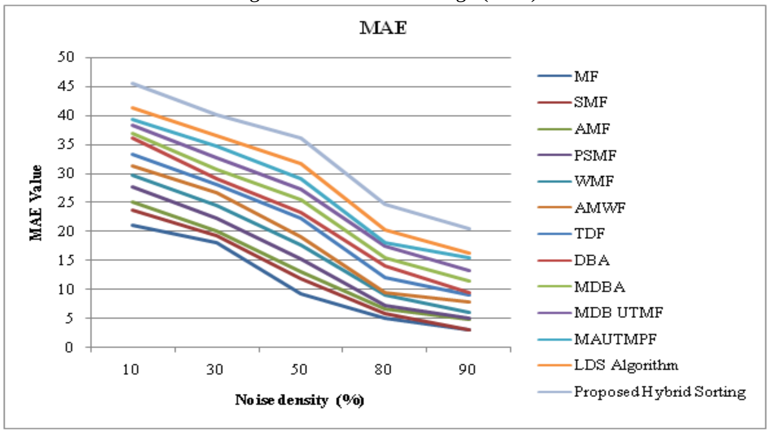

TABLE VIII PERFORMANCE RESULTS OF NCC

Noise Density (%)	MF	SMF	AMF	PSMF	WMF	AMWF	TDF	DBA	MDBA	MDB UTMF	MAUT MPF	LDS	Proposed Cascaded Sorting
10	0.2068	0.2987	0.3214	0.3987	0.4654	0.5178	0.5247	0.6025	0.6.478	0.7.1456	0.8145	0.8207	1.0003
30	0.1989	0.2332	0.2978	0.3124	0.3987	0.40003	0.4698	0.5247	0.5987	0.6012	0.7096	0.7102	1.0004
50	0.2000	0.2869	0.3102	0.3687	0.4001	0.4320	0.4987	0.5256	0.5987	0.6214	0.6997	0.7031	1.0001
80	0.1654	0.2006	0.2564	0.2995	0.3214	0.3865	0.4321	0.4986	0.5201	0.5876	0.6024	0.6412	0.9883
90	0.0123	0.0567	0.0986	0.1231	0.1985	0.2456	0.2987	0.3236	0.3987	0.4069	0.4689	0.5412	0.9659

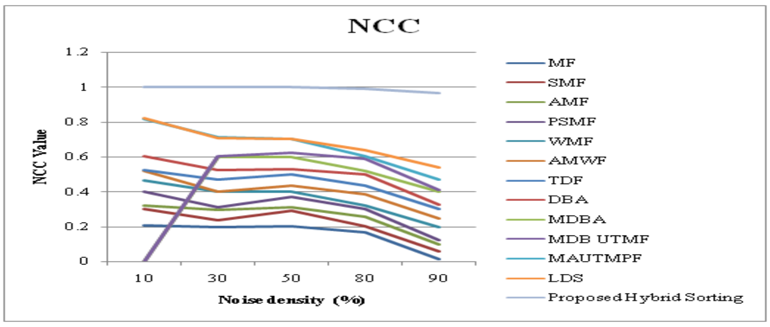

Fig.12 NCC for RGB image (Lena)

TABLE IX PERFORMANCE RESULTS OF NAE

Noise Density (%)	MF	SMF	AMF	PSMF	WMF	AMWF	TDF	DBA	MDBA	MDB UTMF	MAUT MPF	LDS	Proposed Cascaded Sorting
10	0.0009	0.0011	0.0026	0.0034	0.0048	0.0068	0.0982	0.0329	0.0489	0.0587	0.0689	0.0754	0.0023
30	0.0052	0.0070	0.0086	0.0098	0.0106	0.0346	0.0459	0.0596	0.0684	0.0768	0.0895	0.0941	0.0078
50	0.0065	0.0075	0.0086	0.0098	0.0103	0.0135	0.0156	0.0189	0.0287	0.0354	0.0498	0.0521	0.0158
80	0.0321	0.0441	0.0465	0.0635	0.0725	0.0897	0.0967	0.1165	0.1265	0.1348	0.1478	0.1541	0.0864
90	0.1032	0.1365	0.1876	0.2007	0.2867	0.3124	0.3785	0.4006	0.4548	0.5000	0.5102	0.5210	0.2001

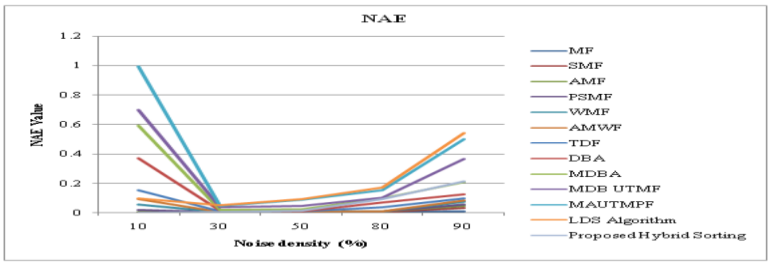

Fig.13 NAE for RGB image (Lena)

TABLE X PERFORMANCE RESULTS OF QI

Noise Density (%)	MF	SMF	AMF	PSMF	WMF	AMWF	TDF	DBA	MDBA	MDB UTMF	MAUT MPF	LDS	Proposed Cascaded Sorting
10	0.1864	0.2263	0.2987	0.3091	0.3865	0.4006	0.4967	0.5142	0.5915	0.6103	0.6925	0.7213	0.9978
30	0.0851	0.0978	0.1986	0.2007	0.2967	0.3456	0.4004	0.4879	0.5213	0.5989	0.6453	0.7421	0.9644
50	0.0958	0.0985	0.0999	0.1111	0.1897	0.2447	0.3124	0.3921	0.4005	0.4678	0.5847	0.6214	0.9261
80	0.0902	0.0945	0.0964	0.0968	0.0986	0.1006	0.1987	0.2297	0.3324	0.4007	0.4965	0.5721	0.8856
90	0.0802	0.0835	0.0921	0.0975	0.0983	0.1008	0.1769	0.2265	0.2975	0.3651	0.3961	0.4413	0.8277

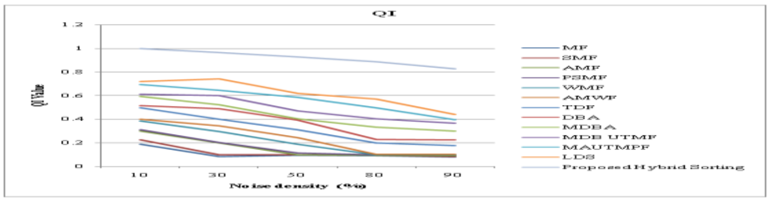

Fig.14 QI for RGB image (Lena)

TABLE XI PERFORMANCE RESULTS OF PSNR FOR VIDEO

Noise Density (%)	MF	SMF	AMF	PSMF	WMF	AMWF	TDF	DBA	MDBA	MDB UTMF	MAUT MPF	LDS	Propose d Cascade d Sorting
10	21.1116	23.6987	25.1236	27.6847	29.6875	31.2147	33.2647	35.9874	36.9648	38.2141	39.1964	41.2013	45.4274
30	18.0364	19.2586	20.0312	22.3697	24.4691	26.5684	28.1247	29.0456	30.7986	32.6547	34.5467	36.4132	40.0129
50	9.2564	11.8965	13.1032	15.2635	17.5864	19.0021	22.1325	23.2145	25.4781	27.2634	29.0364	31.5452	35.9265
80	5.0003	5.9647	6.7621	7.2365	8.9631	9.3965	11.9647	13.9654	15.3698	17.3654	18.0123	20.3165	24.5640
90	2.9362	3.0603	4.9036	5.0694	5.9321	7.7653	8.9364	9.3624	11.3654	13.2695	15.3621	16.2876	20.3605

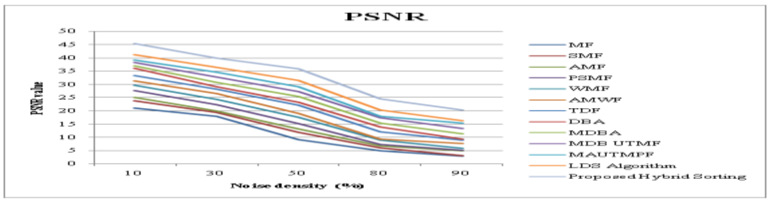

Fig.15 PSNR for video

TABLE XII PERFORMANCE RESULTS OF MAE FOR VIDEO

Noise Density (%)	MF	SMF	AMF	PSMF	WMF	AMWF	TDF	DBA	MDBA	MDB UTMF	MAUT MPF	LDS	Proposed Cascaded Sorting
10	3.7825	3.5986	3.2361	2.9687	2.7654	2.4658	2.0069	1.8632	1.5986	1.2369	0.9687	0.6428	0.3484
30	13.3201	12.0236	11.2147	10.9874	9.2364	8.6547	7.1547	6.3654	5.1247	4.3654	3.0314	2.1253	1.0896
50	19.3002	17.3020	15.6547	13.6506	12.3026	11.0302	9.9947	8.9678	8.1234	7.6987	6.3214	4.5127	2.1278
80	30.3621	29.6987	27.5763	25.3620	24.0630	23.9631	22.3625	20.6932	7.6589	19.3654	16.0369	14.582 4	11.2938
90	49.1235	46.3647	44.2569	42.3694	40.3265	39.3654	37.3256	35.6954	33.2469	31.2364	30.6310	28.304 1	25.7116

Fig.16 MAE for video

TABLE XIII PERFORMANCE RESULTS OF NCC FOR VIDEO

Noise Density (%)	MF	SMF	AMF	PSMF	WMF	AMWF	TDF	DBA	MDBA	MDB UTMF	MAUT MPF	LDS	Propose d Cascad ed Sorting
10	0.0810	0.1001	0.1965	0.2031	0.2963	0.3214	0.3961	0.4221	0.4996	0.5687	0.6935	0.8412	1.0000
30	0.0098	0.0109	0.0236	0.0569	0.0969	0.1201	0.1654	0.2004	0.2931	0.3624	0.4321	0.6123	0.9998
50	0.0075	0.0091	0.0164	0.0554	0.0968	0.1012	0.1968	0.2204	0.2964	0.3231	0.4136	0.5812	0.9993
80	0.0064	0.0085	0.0166	0.0456	0.0965	0.1002	0.1976	0.2101	0.2687	0.3124	0.3968	0.4541	0.9777
90	0.0042	0.0065	0.0120	0.0332	0.0869	0.0996	0.1020	0.1964	0.2221	0.2978	0.3645	0.4891	0.9430

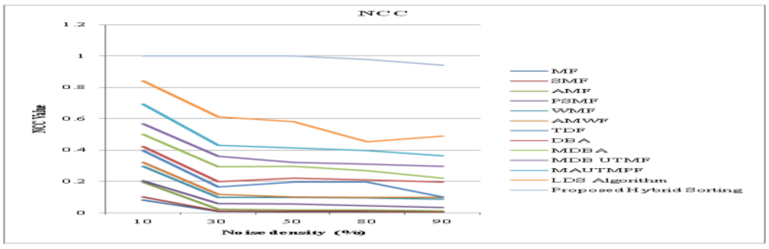

Fig.17 NCC for video

TABLE XIV PERFORMANCE RESULTS OF NAE

Noise Density (%)	MF	SMF	AMF	PSMF	WMF	AMWF	TDF	DBA	MDBA	MDB UTMF	MAUT MPF	LDS	Proposed Cascaded Sorting
10	0.0012	0.0026	0.0055	0.0168	0.0543	0.0964	0.1524	0.3697	0.5924	0.7012	0.9965	0.1001	0.0029
30	0.0026	0.0036	0.0046	0.0056	0.0065	0.0098	0.0102	0.0164	0.0214	0.0374	0.0461	0.0541	0.0090
50	0.0031	0.0038	0.0046	0.0052	0.0064	0.0085	0.0096	0.0116	0.0264	0.0456	0.0874	0.0942	0.0177
80	0.0046	0.0063	0.0086	0.0098	0.0102	0.0146	0.0365	0.0698	0.0986	0.1023	0.1546	0.1721	0.0938
90	0.0112	0.0321	0.0498	0.0561	0.0745	0.0874	0.0976	0.1254	0.2123	0.3687	0.5001	0.5412	0.2135

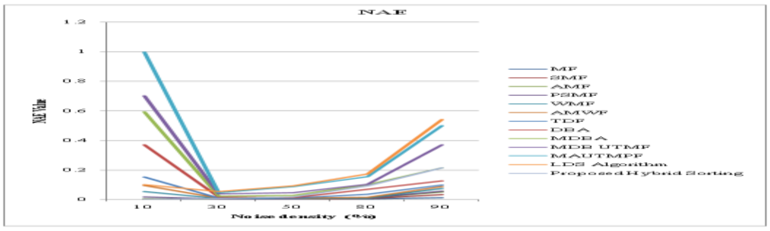

Fig.18 NAE for video

TABLE XV PERFORMANCE RESULTS OF QI

Noise Density (%)	MF	SMF	AMF	PSMF	WMF	AMWF	TDF	DBA	MDBA	MDB UTMF	MAUT MPF	LDS	Proposed Cascaded Sorting
10	0.3124	0.6345	0.9870	1.0231	2.3147	3.1140	3.6541	4.2136	5.0122	0.5213	0.5965	0.6213	0.8978
30	0.0086	0.0096	0.0103	0.0365	0.0654	0.0965	0.1006	0.1926	0.2964	0.3624	0.4654	0.5421	0.8644
50	0.0336	0.0546	0.0764	0.0965	0.1016	0.1936	0.2164	0.2964	0.3031	0.3496	0.4964	0.5214	0.8261
80	0.0085	0.0099	0.0164	0.0264	0.0365	0.0564	0.0789	0.0945	0.1467	0.2465	0.3456	0.4721	0.7856
90	0.0025	0.0048	0.0065	0.0098	0.0102	0.0212	0.0524	0.0685	0.0968	0.1263	0.3124	0.4413	0.7277

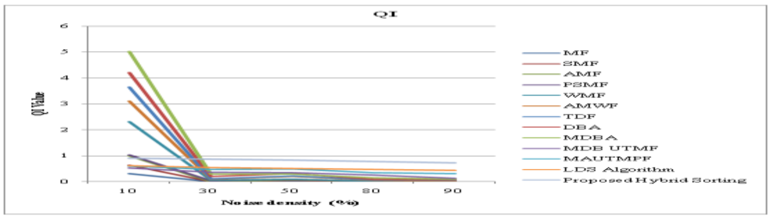

Fig.19 QI for video