International Journal of Future Generation Communication and Networking
Vol. 13, No. 2, 2020 pp.1629-1641

Hardware Implementation of Image Denoising Using AIQR Technique

Dr. P. Anil Kumar[1], Dr. K.Sateesh Kumar[2], Dr. S.V.Padmaja Rani[3]

[1]Associate Professor, Department of Electronics & Communication Engineering, CVR College of Engineering, Hyderabad, Telangana, India

[2]Assistant Professor, Department of Electronics & Communication Engineering, Vignan's LARA Institute of Technology and Science, Guntur, Andhra Pradesh, India

[3]Principal, Sri Venkateswara College of Engineering, Nellore, Andhra Pradesh, India

[1]anilkumar.417@gmail.com, [2]sateeshkumarkanagala@gmail.com, [3]sv.padmajarani@gmail.com

Corresponding Author: P. Anil Kumar

Email: anilkumar.417@gmail.com

Abstract

Image denoising is one of the fundamental steps in the processing of images. The defects in the image acquisition process result in adding the salt and pepper noise to the image at some fixed points leading to the degradation of image quality. Adaptive Median filter is one of the most widely used filters, in the literature, for the removal of salt and pepper noise which was implemented in FPGA. The Inter Quartile Range filter is a nonlinear spatial filter aims at removing the noise by preserving the edge information. In this paper, modified Adaptive Inter Quartile Range (AIQR) filter is presented for the removal of salt and pepper noise. Real time image processing requires huge amount of operations and high throughput rate. Due to the limitations in the hardware, the algorithm is modified making it feasible for the FPGA implementation while retaining the original features of the AIQR. The hardware implementation of the proposed algorithm aims at achieving high speed with minimum area. To prove the robustness of the proposed filter, it is done state of art comparison with the existing spatial domain denoising techniques. The results demonstrate superiority of the proposed filter over other competing filter topologies with respect to resource utilization, PSNR and SSIM.

Keywords: *Adaptive Median Filter, FPGA, Image denoising, nonlinear filter, Salt and pepper noise.*

I Introduction

Owing to the technological advancements, there has been a significant increase in the transmission of digital images when compared to the text messages. As these images are transmitted over the channel, they prone to noise and hence the quality will degrade, called as noisy image [I]. The sources of noise can be due to either internal or external sources. Of all the different types of noise presented in the literature, salt and pepper noise is investigated by considering its complex denoising procedure. Salt and pepper noise arise due to the sensor defects in the camera due to alteration in the original pixel value by the maximum or minimum gray level value [II,III,IV]. Besides the salt and pepper noise, sharp edges also contribute to the high frequency content in the image. Low variation between the successive pixels contributes to the low frequency content of the image, while the sharp edges contribute to the high frequency content of the image. The edge and the noise should be de-lineated properly using image denoising algorithms. The image denoising algorithm works on the principle of removal of noise by preserving the edge information in the image [V]. Image denoising is involved in medical image processing, object detection, pattern recognition and stereo vision etc. FPGA based denoising techniques suffer with execution time and power problems because of more number of computations and space needed to perform in real-time. Parallel processing made the faster system but not much progress in the

ISSN: 2233-7857 IJFGCN

power variations. This proposed technique developed the faster system for denoising spatial techniques. In this paper, salt and pepper noise is considered for denoising real time digital images. On a general note, proper selection of threshold for denoising is vital role for implementation. In the proposed denoising technique soft thresholding is used. In addition to the implementation in FPGA, MATLAB implementation was also done for comparison of results at different block sizes and various noise levels. PSNR, and SSIM and Entropy are the performance metrics chosen for state of art comparison over existing implementations. The database used in this paper is from different textures and different features. This proposed method has an advantage of implementing spatial domain based denoising, which will directly operate on pixels. So comparatively it needs minimum time with frequency based denoising techniques. This denoising can be done on gray/colour images.

II Related Work

Literature review pertaining to salt and pepper noise in FPGA environment, even though median filter(including Adaptive Median Filtering, AMF) is used to deal salt and pepper noise from last few decades, so many reaches are finding best solution even in recent times [I,II,VI,VII,VIII].In recent times, Fuzzy based salt and pepper noise removal techniques have been drawn so much of attention among researchers but for the FPGA implementation wise it is not up to the level because of the complexity wise. [IX,X,XI,XII,XIII,XIV] have developed prominent technique for the replacement of noisy pixels for medical images which are binary images. This can't be extended to grey level images. On the digital image processing domain, adaptive median filter with alpha trimming was also implemented as a successful denoising technique, in software but for the FPGA implementation, it requires lot of memory and LUT to approximate the noisy pixels in real domain [XV]. The main problem with the existing median filter based denoising techniques is similarity (nonlocal) among different portions in the image [XVI,XVII,XVIII]. This can be overcome with the help of proposed method of denoising. In the existing frequency and spatial FPGA based image denoising techniques are suffering with Complexity, execution time and memory issues because of shrink techniques they follow [XIX,XX,XXI].

The Inter Quartile Range (IQR) filter [XVII] is considered as a solution to the aforementioned problem. The hardware implementation of the IQR can help in improving the throughput of the signal using the pipelining techniques. The adaptive IQR shows better performance when compared to the previous methods while maintain the minimum area and delay.

This paper is organized as follows: Section II deals with the state-of-the-art denoising techniques. Section III provides the detailed explanation the IQR algorithm. Section IV demonstrates the hardware implementation of the proposed algorithm. Section V provides the detailed analysis with simulation results and finally the conclusions are drawn in Section VI.

III Proposed Method

The proposed spatial domain based denoising technique is implemented in both MATLAB and FPGA. Initially mask is taken of size 3X3, 5X5, 7X7 or 9x9. This mask is applied as overlapping blocks on input image whose size is like 128X128, 256X256, 512X512 etc. Then values in the mask are divided into five categories based on the spatial distribution of the input image. The five statistical numbers obtained from the mask are as minimum value (Q_m), first quartile (Q_1), the median (Q_2), the third quartile (Q_3), and the maximum value (Q_M) [3]. The pixel values of the input image are in the window are arranged in either increasing/decreasing order. The expressions for calculating different quartile values are given in eq. 1, 2 & 3 respectively. In fact, these are indices, not values. The values at $Q_1, Q_2 \ and \ Q_3$ is referred as First Quartile, Median and Third Quartile Values respectively. The value of Interquartile range is mathematically represented as in eq. 4. As salt and Pepper noise generally lies at the edges i.e., either 0 or 255, the IQR never lies on edges. Statistically IQR is a range which indicates the probable range of actual data.

ISSN: 2233-7857 IJFGCN

$$\mathbf{Q_1 = 0.25 * (n+1)^{th}\ positioned\ value\ in\ the\ sequence} \quad (1)$$

$$\mathbf{Q_2 = 0.50 * (n+1)^{th}\ positioned\ value\ in\ the\ sequence} \quad (2)$$

$$\mathbf{Q_3 = 0.75 * (n+1)^{th}\ positioned\ value\ in\ the\ sequence} \quad (3)$$

$$\mathbf{IQR = (Q_3 - Q_1)} \quad (4)$$

A pixel is considered to be noisy (salt and pepper), if the pixel value is less than first quartile or more than the third quartile range. The IQR filter considers 25% of both top and bottom pixels as noisy while the remaining pixels are considered noise-free. A threshold is set to permit a range of pixels as shown in fig 1. The values below the $\mathbf{Q_1 - T_1}$ and $\mathbf{Q_3 - T_2}$ are considered as noisy pixels. Where, $\mathbf{T_1}$**and** $\mathbf{T_2}$ are the hard thresholds used to imposing stringent constraints.

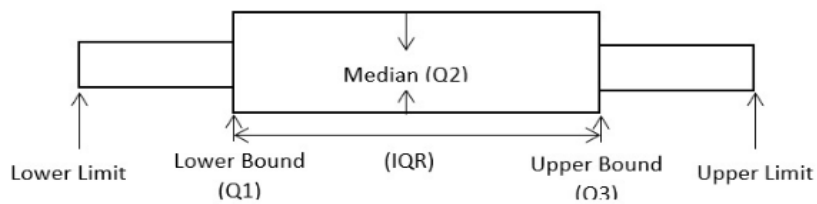

Fig 1. IQR with threshold [XII]

The following conditions are the possibility of getting a noisy pixel in an input image.

(i) If the centre pixel is noisy, it is replaced by the average of surrounding 8 non noisy pixels.

(ii) If the noisy pixel is on the edge of the mask, it is replaced by the average of the surrounding 5 non noisy pixels.

(iii) If the noisy pixel is at the corner, it is replaced by the average of the three surrounding noise free pixels [5, 6].

The noisy pixels in the mask are identified and then replaced by the average of the surrounding non noisy pixels. The IQR algorithm is modified in this paper for the feasibility of the hardware implementation without compromising the efficiency. Here, only the centre pixel is tested for the noisy condition. For this reason, if the pixel is noised on either corners of the image, to approximate with 3x3 mask, an extra row and extra column padded along x and y directions (on all four sides). If the size of the mask is 5, the image is padded by 2 rows and columns on all the 4 sides.

The steps, describing the image denoising process in the modified IQR are as follows.

1. Pad the image on all the four borders with the boundary pixel values.
2. Subdivide the image into overlapping odd sized blocks.
3. Sort the pixel values in the window either in the increasing or decreasing order.
4. Find the first, second and third quartile values by using the equation 1, 2, 3.
5. Find the interquartile range using eq. 4.
6. Select a suitable threshold value and find the lower and upper margin. Lower bound is obtained by subtracting the threshold from Q_1. Upper bound is obtained by adding the threshold to Q_3.
7. If the centre pixel is less than lower bound or greater than upper bound, it is considered as a noisy pixel. Otherwise leave the pixel as it is and slide the window to the next position and go to step 1.

ISSN: 2233-7857 IJFGCN

8. For a noisy centre pixel, replace the centre pixel with the average of the surrounding 8 non noisy pixels.
9. Repeat the same procedure for all the pixels in the image.

Selection of the threshold

In the proposed algorithm novelty lies in the selection of the threshold.

As impulse noise effect the image then the value may be modified as 0 or 255 but interestingly the pixels having the value of 0/255 are need not to be noisy always. So, in this proposed algorithm centre pixel is considered for evaluation of noise in an image.

(a) If the centre pixel of the mask is either 0/255 consider it as suspect pixel.

(b) If the values of $Q_1 or\ Q_3$ is 0/255, then centre pixel is connecting component of edge/border otherwise this can be considered as noise.

(c) Replace the noisy pixel with average of all 8 surrounding pixels of the mask.

Hardware Implementation

The test bench setup for the FPGA implementation of the image denoising using the modified IQR algorithm is shown in Fig 2. The external interface, such as, reading the image and displaying the image on the display are done with the help of MATLAB. The image denoising algorithm was implemented on the FPGA. The image is read with the help of MATLAB and salt and pepper noise of fixed noise density is added to the image. The corrupted pixel values of the image are copied into the memory initialization file of the input memory. The denoising algorithm is implemented on the FPGA by reading the values from the input memory and the denoised output image is stored in the output memory. The denoised outputs are taken back to the MATLAB for displaying.

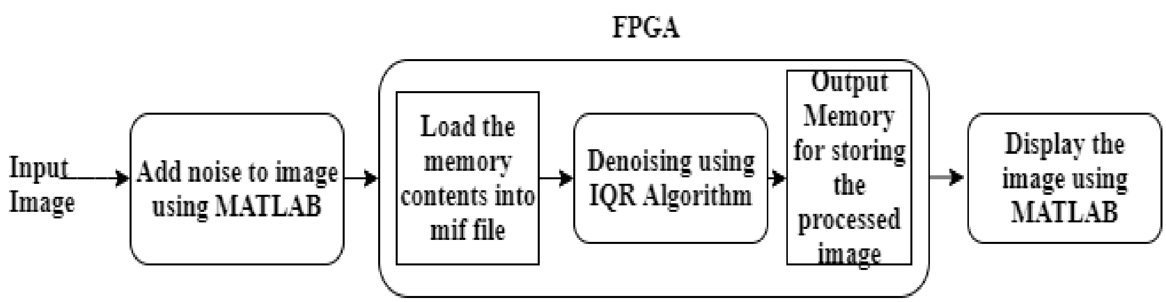

Fig 2. Test setup for the image denoising using modified IQR on FPGA

The block diagram for a real time hardware implementation of the image denoising using modified IQR is shown in fig 3. Here, Stratix II FPGA is chosen for implementation of the algorithm where the image is read with the help of camera. The analog inputs are converted to digital inputs with the help of an ADC. The input values are stored in SRAM1 which is the external memory provided on the board. The image is displayed on the monitor with the help of DAC after processing it [XI]. The hardware setup for the image denoising is shown in fig 4.

ISSN: 2233-7857 IJFGCN

International Journal of Future Generation Communication and Networking
Vol. 13, No. 2, 2020 pp.1629-1641

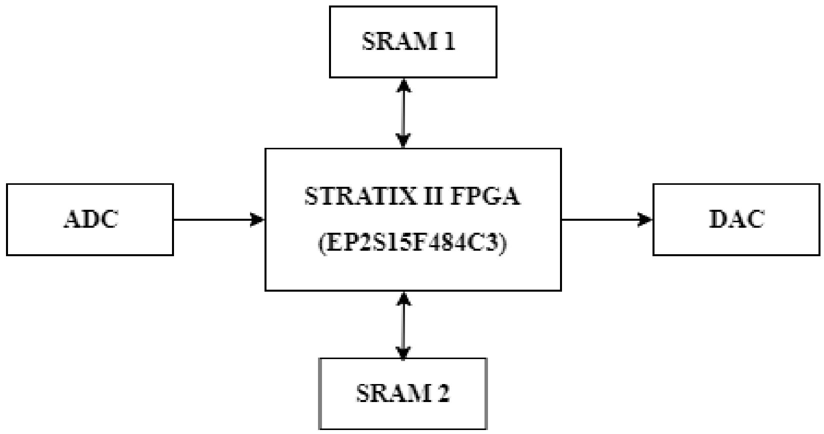

Fig 3. Hardware implementation of image denoising using modified IQR algorithm

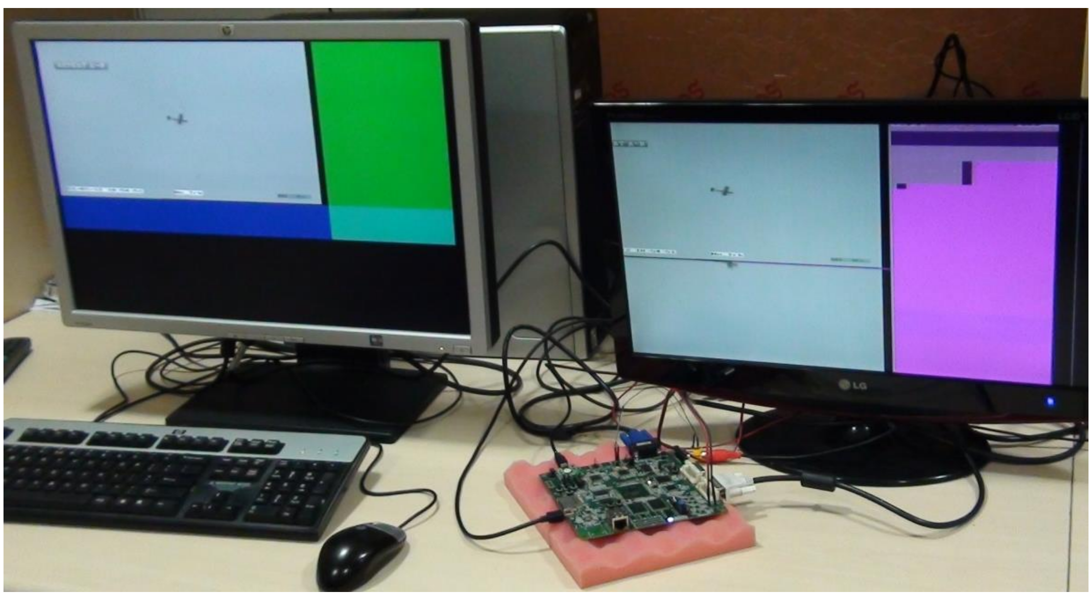

Fig 4. Image denoising using FPGA

IV Result and Discussions

The proposed algorithm is implemented in both MATLAB and FPGA and compared the results for various grey and colour images. Fundamentally, algorithm is proposed for gray or 2D images and has been extended to colour images also by implementing individually on each colur plane of R,G and B. The algorithm was tested on grey scale image of resolution 128 X 128. Salt and pepper noise is added to the image with a varying noise density from 1% to 90%. The performance of the IQR filter is compared with standard median filter, adaptive median filters and existing IQR. The size of the window of IQR and adaptive median filter is varied from 3X3 to 9X9. The state of art comparison PSNR and SSIM were used to measure the amount of mismatch between the original image and the denoised output image. The comparison of the PSNR and SSIM values for different levels of noise on various input images using diferent filters are tabulated in table 1.

ISSN: 2233-7857 IJFGCN

Table 2: (a) PSNR (dB) values for Peppers image

Noise level	**Median**	**AMF 3**	**AMF 5**	**AMF 7**	**IQR 3**	**IQR 5**	**IQR 7**	**IQR 9**
1	37.75	25.77	25.77	25.77	**41.41**	35.32	32.11	30.01
2	37.61	25.78	25.78	25.78	**41.14**	35.25	32.08	30.01
3	37.19	25.77	25.77	25.77	**41.31**	35.22	32.08	30
5	36.81	25.76	25.76	25.76	**40.5**	34.9	31.95	29.98
10	34.76	25.77	25.77	25.77	**39.73**	34.74	31.87	29.91
20	29.71	24.92	25.64	25.64	**37.86**	34.08	31.59	29.84
30	24.37	23.15	25.42	25.47	**36.42**	33.4	31.1	29.6
40	18.79	19.38	25.36	25.41	**34**	32.91	30.8	29.35
50	15.22	16.11	23.88	25.091	30.51	**31.92**	30.2	28.86
60	12.06	12.98	20.47	24.22	25.53	**31.27**	29.64	28.31
70	9.5	10.26	15.71	20.8	21.03	**30.01**	28.73	27.49
80	7.8	8.33	11.83	15.65	16.96	27.61	**27.83**	26.69
90	6.26	6.55	8.15	10.11	12.38	18.61	24.49	**25.49**

Table 1. (b) PSNR(dB) values for Cameraman image

Noise level	**Median**	**AMF 3**	**AMF 5**	**AMF 7**	**IQR 3**	**IQR 5**	**IQR 7**	**IQR 9**
1	27.137	18.938	18.939	18.939	**29.289**	25.159	23.573	22.725
2	27.008	18.932	18.932	18.932	**29.207**	25.171	23.556	22.725
5	26.586	18.891	18.894	18.894	**29.142**	25.216	23.595	22.738
10	25.755	18.903	18.910	18.910	**28.928**	25.207	23.609	22.792
20	23.744	18.762	18.872	18.872	**28.319**	25.039	23.659	22.861
30	20.727	18.099	18.813	18.817	**27.665**	25.277	23.885	23.158
40	17.529	16.671	18.710	18.750	**26.630**	25.183	24.140	23.359
50	14.322	14.480	18.418	18.619	24.646	**24.753**	23.903	23.318
60	11.614	12.211	17.095	18.328	21.834	**24.128**	23.455	22.877
70	9.474	10.068	14.538	17.198	17.539	**23.467**	22.837	22.172
80	7.788	8.293	11.551	14.549	13.466	21.568	**21.927**	21.457
90	6.184	6.468	7.962	9.730	9.373	15.511	19.888	**20.496**

Table 1. (c) PSNR(dB) values for Barbara image

ISSN: 2233-7857 IJFGCN

Noise level	**Median**	**AMF 3**	**AMF 5**	**AMF 7**	**IQR 3**	**IQR 5**	**IQR 7**	**IQR 9**
1	29.445	20.949	20.949	20.949	**32.538**	29.617	27.443	25.908
2	29.339	20.953	20.953	20.953	**32.439**	29.568	27.439	25.895
5	28.947	20.922	20.922	20.922	**32.307**	29.526	27.408	25.874
10	28.258	20.879	20.880	20.880	**32.045**	29.419	27.373	25.920
20	25.916	20.639	20.793	20.793	**31.426**	29.174	27.361	25.927
30	21.714	19.635	20.740	20.749	**30.486**	28.856	27.288	25.976
40	18.341	17.866	20.605	20.646	**29.339**	28.361	26.938	25.765
50	15.005	15.458	20.082	20.513	26.752	**27.832**	26.549	25.488
60	11.887	12.577	18.085	19.925	22.808	**27.089**	25.897	24.854
70	9.810	10.493	15.375	18.506	18.471	**26.073**	25.092	24.078
80	7.912	8.393	11.618	14.978	13.781	23.537	**24.286**	23.412
90	6.495	6.765	8.395	10.319	9.951	16.836	21.932	**22.332**

Table 1. (d) PSNR(dB) values for Lena image

Noise level	**Median**	**AMF 3**	**AMF 5**	**AMF 7**	**IQR 3**	**IQR 5**	**IQR 7**	**IQR 9**
1	32.846	21.643	21.643	21.643	**35.709**	30.413	27.933	26.216
2	32.588	21.654	21.654	21.654	**35.525**	30.404	27.920	26.244
5	31.612	21.632	21.636	21.636	**35.282**	30.309	27.931	26.285
10	30.789	21.625	21.631	21.631	**34.586**	30.139	27.792	26.189
20	26.500	21.267	21.555	21.555	**33.856**	29.929	27.766	26.268
30	22.504	20.295	21.479	21.510	**31.842**	29.356	27.488	26.176
40	18.729	18.331	21.368	21.448	**30.457**	29.059	27.390	26.183
50	15.244	15.679	20.902	21.296	27.204	**28.517**	27.092	26.023
60	12.281	13.020	18.949	20.839	22.913	**27.701**	26.447	25.363
70	9.954	10.662	15.615	19.154	18.241	**26.723**	25.734	24.732
80	8.206	8.703	12.068	15.453	13.820	24.017	**24.765**	23.960
90	6.618	6.897	8.422	10.287	9.528	16.247	22.089	**22.684**

Table 3. (a) SSIM values for Peppers image

ISSN: 2233-7857 IJFGCN

Noise level	Median	AMF 3	AMF 5	AMF 7	IQR 3	IQR 5	IQR 7	IQR 9
1	0.975	0.823	0.823	0.823	**0.993**	0.972	0.944	0.914
2	0.974	0.823	0.823	0.823	**0.993**	0.972	0.943	0.913
5	0.971	0.822	0.822	0.822	**0.991**	0.971	0.943	0.912
10	0.962	0.820	0.821	0.821	**0.990**	0.969	0.940	0.911
20	0.911	0.798	0.817	0.817	**0.985**	0.964	0.938	0.910
30	0.771	0.712	0.812	0.813	**0.978**	0.958	0.932	0.906
40	0.505	0.514	0.803	0.807	**0.960**	0.950	0.925	0.899
50	0.260	0.288	0.755	0.794	0.918	**0.941**	0.912	0.884
60	0.117	0.131	0.619	0.768	0.798	**0.930**	0.901	0.870
70	0.052	0.057	0.367	0.651	0.556	**0.906**	0.879	0.846
80	0.027	0.029	0.137	0.388	0.298	0.840	**0.854**	0.821
90	0.012	0.012	0.034	0.095	0.112	0.499	0.768	**0.778**

Table 2. (b) SSIM values for Cameraman image

Noise level	Median	AMF 3	AMF 5	AMF 7	IQR 3	IQR 5	IQR 7	IQR 9
1	0.867	0.660	0.660	0.660	**0.939**	0.861	0.814	0.787
2	0.866	0.660	0.660	0.660	**0.938**	0.863	0.814	0.788
5	0.860	0.655	0.655	0.655	**0.937**	0.864	0.816	0.788
10	0.846	0.652	0.653	0.653	**0.935**	0.865	0.818	0.791
20	0.788	0.629	0.648	0.648	**0.929**	0.867	0.825	0.798
30	0.647	0.541	0.643	0.644	**0.920**	0.873	0.834	0.810
40	0.434	0.396	0.638	0.642	**0.894**	0.869	0.840	0.817
50	0.234	0.224	0.611	0.638	0.827	**0.859**	0.834	0.814
60	0.122	0.114	0.498	0.619	0.704	**0.833**	0.809	0.788
70	0.064	0.061	0.300	0.549	0.471	**0.801**	0.778	0.752
80	0.034	0.038	0.131	0.332	0.274	0.712	**0.730**	0.709
90	0.013	0.014	0.034	0.081	0.151	0.400	0.632	**0.654**

ISSN: 2233-7857 IJFGCN

Table 2. (c) SSIM values for Barbara image

Noise level	**Median**	**AMF 3**	**AMF 5**	**AMF 7**	**IQR 3**	**IQR 5**	**IQR 7**	**IQR 9**
1	0.868	0.668	0.668	0.668	**0.948**	0.903	0.854	0.812
2	0.866	0.667	0.667	0.667	**0.948**	0.902	0.854	0.813
5	0.859	0.666	0.666	0.666	**0.946**	0.902	0.854	0.813
10	0.850	0.662	0.662	0.662	**0.943**	0.902	0.856	0.816
20	0.804	0.645	0.656	0.656	**0.936**	0.899	0.859	0.818
30	0.675	0.577	0.651	0.651	**0.921**	0.894	0.857	0.818
40	0.482	0.449	0.638	0.641	**0.903**	0.884	0.848	0.811
50	0.295	0.290	0.608	0.628	0.846	**0.868**	0.831	0.793
60	0.145	0.151	0.498	0.596	0.719	**0.842**	0.802	0.761
70	0.079	0.081	0.335	0.527	0.508	**0.806**	0.764	0.718
80	0.036	0.038	0.135	0.325	0.266	0.719	**0.715**	0.668
90	0.016	0.016	0.043	0.101	0.113	0.435	0.617	**0.604**

Table 2. (d) SSIM values for Lena image

	Median	**AMF 3**	**AMF 5**	**AMF 7**	**IQR 3**	**IQR 5**	**IQR 7**	**IQR 9**
1	0.944	0.704	0.704	0.704	**0.980**	0.928	0.882	0.842
2	0.942	0.704	0.704	0.704	**0.979**	0.928	0.882	0.842
5	0.935	0.702	0.702	0.702	**0.978**	0.928	0.882	0.844
10	0.926	0.701	0.701	0.701	**0.974**	0.926	0.881	0.842
20	0.855	0.674	0.697	0.697	**0.968**	0.924	0.882	0.844
30	0.712	0.597	0.692	0.692	**0.954**	0.918	0.880	0.846
40	0.496	0.448	0.685	0.688	**0.935**	0.911	0.873	0.839
50	0.276	0.267	0.652	0.678	0.867	**0.898**	0.862	0.829
60	0.138	0.136	0.534	0.653	0.710	**0.876**	0.839	0.801
70	0.073	0.072	0.322	0.561	0.473	**0.848**	0.812	0.772
80	0.035	0.034	0.130	0.337	0.232	0.758	**0.770**	0.732
90	0.016	0.016	0.039	0.088	0.083	0.392	0.671	**0.672**

For the noise level of 40%, the denoised outputs using median, IQR of window size 3,5,7,9 and Adaptive median filter of window size 3, 5, and 7 is shown in fig 5. For the noise level of 90%, the denoised outputs using median, IQR of window size 3,5,7,9 and Adaptive median filter of window size 3, 5, and 7 is shown in fig 6.

ISSN: 2233-7857 IJFGCN

Fig 5. (a) Original Image (b) Median Filter Output (c) IQR output of window size 3 (d) IQR output of window size 5 (e) IQR output of window size 7 (f) IQR output of window size 9 (g) Adaptive median filter output of window size 3 (h) Adaptive median filter output of window size 5 (i) Adaptive median filter output of window size 7 (proposed)

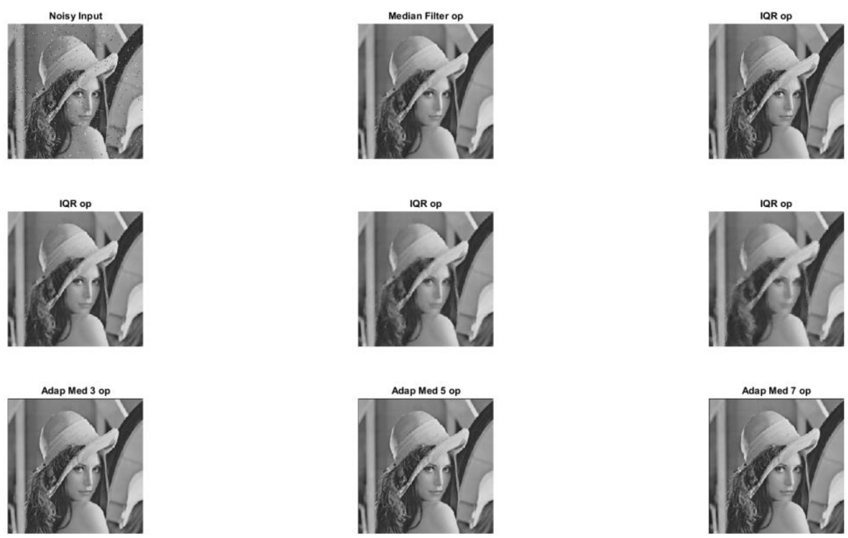

Fig 6. (a) Original Image (b) Median Filter Output (c) IQR output of window size 3 (d) IQR output of window size 5 (e) IQR output of window size 7 (f) IQR output of window size 9 (g) Adaptive median filter output of window size 3 (h) Adaptive median filter output of window size 5 (i) Adaptive median filter output of window size 3 (proposed)

ISSN: 2233-7857 IJFGCN

The baby image is subjected to the 50 Noise level as shown in fig 7(a). The denoised outputs using median, IQR of window size 3, 5, 7, 9 and Adaptive median filter of window size 3, 5, and 7 is shown in fig 7(b) – 7(i).

Fig 7. (a) Original Image (b) Median Filter Output (c) IQR output of window size 3 (d) IQR output of window size 5 (e) IQR output of window size 7 (f) IQR output of window size 9 (g) Adaptive median filter output of window size 3 (h) Adaptive median filter output of window size 5 (i) Adaptive median filter output of window size 7 (proposed)

The compilation report for the hardware implementation of the IQR is shown in Table 3. The architecture works at a design frequency of 78.38 MHz with a minimum number of lookup tables on a stratix II board. The algorithm requires 10 clock cycles to process a single a pixel. The amount of time required to process a single image of size 128 X 128 is 2.12 ms which includes all the algorithm implementation and control process. The same process can be extended to a video which can handle nearly 45 frames per second.

Table 3. Compilation Report

S. No	Resources	Utilization
1.	Number of ALUT	555/12480 (4.45%)
2.	Block RAM	131072/419328(31.26%)
3.	DSP Blocks	0 / 576 (0 %)

ISSN: 2233-7857 IJFGCN

4.	Number of flip flops	214/12480 (1.71%)
5.	Frequency	78.38 MHz

V. Conclusion

The Salt and pepper noise with high noise density degrades the overall quality of image. Median filter performs better when the noise density is low. As the noise density increases, the existing method removes the noise as well as the high frequency edge information. The existing method fails to work for high density noise. The modified Inter Quartile Range Filter helps in overcoming the problem. The modified Inter Quartile Range Filter performs better compared to the other state-of-the-art methods. The modified IQR not only removes the noise but also preserves the edges information. The performance of the algorithm was tested with the help of PSNR. For a low noise density, the smaller window size modified IQR filter performs better. For an increasing noise density, higher order window size IQR presents the better results. The algorithm was implemented on a Stratix II FPGA. The size of the image was restricted to 128 X 128 due to the limitation of the inbuilt memory in the FPGA. The real time implementation of the algorithm works at a frequency of 78.38 MHz by consuming very little hardware. The architecture can be used for denoising a video of resolution 128 X 128 at 45 frames per second. Finally, the superiority in the performance of higher window sized IQR with high noise. The superiority in the performance of higher window sized IQR with high noise. Similarly, for low noise density, IQR of smaller window size performs better compared to the other competing methods.

References

I. T. A. Nodes and N. C. Gallagher, “Median filters: Some modifications and their properties,” *IEEE Trans. Acoust., Speech, Signal Process.*,vol. ASSP-30, no. 5, pp. 739–746, Oct. 1982.

II. O. Yli-Harja, J. Astola, and Y. Neuvo, “Analysis of the properties of median and weighted median filters using threshold logic and stack filter representation,” *IEEE Trans. Signal Process.*, vol. 39, no. 2, pp. 395–410, Feb. 1991.

III. T. Sun and Y. Neuvo, “Detail-preserving median based filters in image processing,” *Pattern Recognit. Lett.*, vol. 15, pp. 341–347, 1994

IV. F. Russo and G. Ramponi, “A fuzzy filter for images corrupted by impulse noise,” *IEEE Signal Process. Lett.*, vol. 3, no. 6, pp. 168–170, Jun. 1996.

V. H. L. Eng and K. Ma, “Noise adaptive soft-switching median filter," IEEE Trans. Image Process., vol. 10, no. 2, pp. 242-251, 2001.

VI. Ruizhen Liu and Tieniu Tan, Senior Member, IEEE 2002 An SVD-Based Watermarking Scheme for Protecting Rightful Ownership. IEEE Transactions On Multimedia, Vol. 4, No. 1, March 2002.

VII. Gouchol Pok, Jyh-Charn Liu, and Attoor Sanju Nair, “Selective Removal of Impulse Noise Based on Homogeneity Level Information,” IEEE Transactions on Image Processing, vol. 12, no. 1, January 2003.

VIII. R. Oten and R. J. P de Figueiredo, “Adaptive alpha-trimmed mean filters under deviations from assumed noise model,” *IEEE Trans. Image Process.*, vol. 13, no. 5, pp. 627–639, May 2004

IX. R. H. Chan, C.-W. Ho, and M. Nikolova, “Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization,” *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1479–1485, Oct. 2005.

ISSN: 2233-7857 IJFGCN

X. S. Schulte, M. Nachtegael,V. DeWitte, D.Van derWeken, and E. E.Kerre,"A fuzzy impulse noise detection and reduction method," *IEEE Trans.Image Process.*, vol. 15, no. 5, pp. 1153–1162, May 2006.

XI. K. S. Srinivasan and D. Ebenezer, "A New Fast and Efficient Decision-Based Algorithm for Removal of High-Density Impulse Noises", *IEEE signal processing letters*, VOL. 14, NO. 3, March 2007.

XII. U. Ghanekar,A. K. Singh, and R. Pandey, "A contrast enhancement-based filter for removal of random valued impulse noise," *IEEE Signal Process. Lett.*, vol. 17, no. 1, pp. 47–50, Jan. 2010.

XIII. Karabchevsky, S., Kahana, D., Ben-Harush, O. and Guterman, H,"FPGA-based adaptive speckle suppression filter for underwater imaging sonar", *IEEE Journal of Oceanic Engineering*, Vol. 36, No. 4, pp.646–657.,2011

XIV. B. Ergen (2012). "Signal and Image Denoising Using Wavelet Transform", Advances in Wavelet Theory and Their Applications in Engineering, Physics and Technology, Dr. Dumitru Baleanu (Ed.), ISBN: 978-953-51-0494-0,

XV. Chih-Yuan Lien, Chien-Chuan Huang, Pei-Yin Chen, "An Efficient Denoising Architecture for Removal of Impulse Noise in Images", *IEEE transactions on computers*, vol. 62, no. 4, April 2013.

XVI. Faruk Ahmed and Swagatam Das,"Removal of High-Density Salt-and-Pepper Noise in Images With an Iterative Adaptive Fuzzy Filter Using Alpha-Trimmed Mean", *IEEE TRANSACTIONS ON FUZZY SYSTEMS*, Vol. 22, no. 5, October 2014

XVII. Firas Ajil Jassim, International Journal of Soft Computing and Engineering, ISSN: 2231-2307, Volume-2, Issue-6, January 2013

XVIII. K. Sateesh Kumar, G. Sreenivasulu, "Adaptive Inter-Quartile Range approach for denoising of Landsat Satellite Imagery ", International Journal of Modern Electronics and Communication Engineering (IJMECE), ISSN: 2321-2152, Vol-4,Issue-5, pp-46-49, 2016.

XIX. Jia, X., Liu, S., Feng, X., & Zhang, L. (2019). Focnet: A fractional optimal control network for image denoising. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 6054–6063)

XX. Kumar, S., Chauhan, S., Sundaram, G., Chouksey, M., & Jha, R. K. (2019a). An FPGA based practical implementation of stochastic resonance for image enhancement. In 25th international conference on noise and fluctuations (ICNF 2019), CONF.

XXI. Kumar, S., & Jha, R. K. (2019). FD-based detector for medical image watermarking. IET Image Processing, 13(10), 1773–1782.

ISSN: 2233-7857 IJFGCN