Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/170119

Improving the Efficiency of Median Filters Using Special Generated Windows

M. Abu-Faraj[1,], A. Al-Hyari[2], B. Al-Ahmad[1], Z. Alqadi[3], B. J. A. Ali[4] and K. Aldebe[5]*

[1] Department of Computer Information Systems, The University of Jordan, Aqaba 77110, Jordan
[2] Electrical Engineering Department, Al-Balqa Applied University, As-Salt 19117, Jordan
[3] Computers and Networks Engineering Department, Al-Balqa Applied University, Amman 15008, Jordan
[4] Accounting and Finance Department, Applied Science University, Kingdom of Bahrain
[5] Department of Information Technology, The University of Jordan, Aqaba 77110, Jordan

Received: 22 Nov. 2022, Revised: 13 Dec. 2022, Accepted: 22 Dec. 2022
Published online: 1 Jan. 2023.

Abstract: Digital images of various types, gray and color, are used in many vital applications, necessitating the need to rid them of the noise that can infect them during the messaging process. One of the most important types of noise that negatively affects the characteristics of the digital image is salt and pepper noise, which leads to changing some of the pixels in the digital image to values of 0 or 255. The negative effect of this noise increases with the increase in the noise ratio (the number of affected pixels). This paper will discuss a new method to reduce the adverse effects of salt and pepper noise. This research aims to provide an effective way to deal with the noise of salt and pepper, especially if the noise ratio is higher than 50%, which the rest of the filters cannot deal with. This method will be used to treat the affected pixels only by using six matrices with specific dimensions divided into two types: the examination (checking) matrix (WC) and the execution (processing) matrix (WP); where these two types of matrices are used to process the noise-forming pixels. The six generated matrices are used in the first round (3 of each type), while the first two matrices (one of each type) are used again in the second round to eliminate the adverse effects of noise. The proposed method will be implemented, and the obtained experimental results will be compared with median and average filters to show how the proposed method will enhance the quality of the processed noisy image; a visual and statistical analysis will be performed to prove the quality provided by the proposed method. The proposed method will be compared with other existing methods, such as MDBUT MF, MDBPT GMF, AWM F, and AAMF. MSE, PSNR, SSIM, and CC parameters will be used for comparison purposes.

Keywords: Salt and pepper noise, NR, median filter, average filter, MSE, PSNR, CC, SSIM, WC, WP.

1 Introduction:

Digital images, whether gray or colored, are used in many different vital applications, which require their clarity and not being affected by any external influences that negatively affect their clarity and characteristics, which will negatively influence the results of applications that use digital images in their processing [1].

The gray image can be represented by a 2D matrix, while the color image can be represented by a 3D matrix (one 2D matrix for each color: red, green, and blue, as shown in figure 1) [2-3].

The idea of representing the digital image using matrices will facilitate processing the digital image by implementing all arithmetic and logical processing operations. One of the most important of these operations is expanding (image padding) the matrix by increasing the number of columns and rows in it, as shown in the example illustrated in figure 2 [4-6].

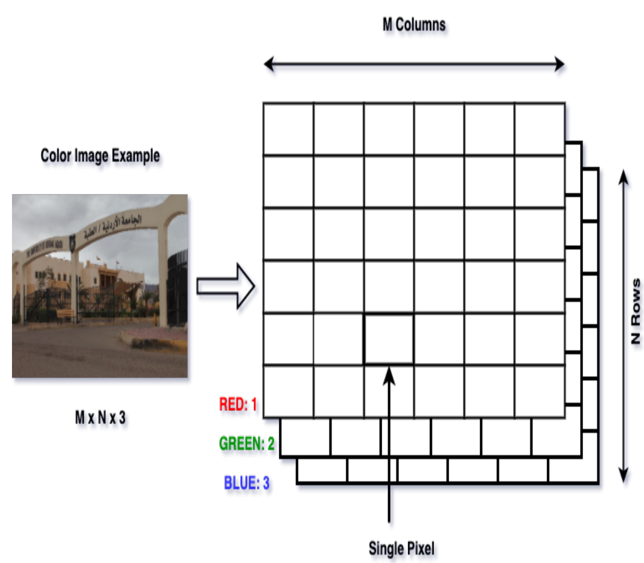

Fig. 1: Color image representation

*Corresponding author e-mail: m.abufaraj@ju.edu.jo

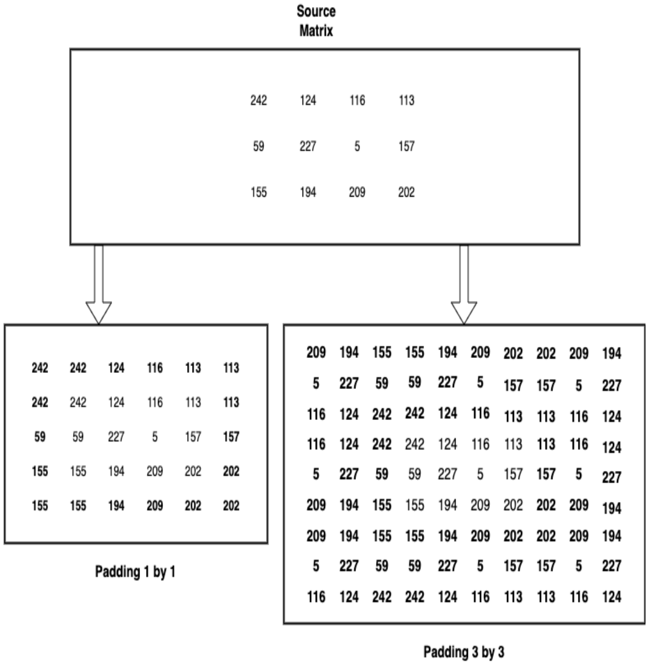

Fig. 2: Image matrix padding example

Salt and pepper noise is an impulse type of noise in images. Data transmission errors, memory cell failure, or analog-to-digital converter errors generally cause this noise. We consider salt-and-pepper noise, for which a certain amount of the pixels in the image are either zeros (black) or 255s (white) (black or white dots). Generally, if there are black dots in the image, we call it pepper noise, and if there are white dots in the image, we call it salt noise.

If we consider an 8-bit image, then salt and pepper noise randomly changes a certain number of pixels into two extreme values, either 0 or 255.

The noise significantly damages the image information, leading to difficulties in afterward image processing tasks such as edge detection or image segmentation and recognition tasks. This happens because the noise pixel differs from most of its local neighbors, as it has a considerable gradient value similar to the edge pixel.

Digital images are vulnerable to being affected by the salt and pepper noise, which leads to changing some values in the digital image to 0 or 255. Therefore, a notable increase in the two extreme values in an image histogram is depicted in figure 3.

The noise of salt and pepper negatively affects the digital image, whether gray or colored. Therefore, the image becomes blurred, and the negative effect increases with the increase in the noise ratio (NR), which is the ratio of the number of affected pixels to the total number of pixels in the digital image. Figure 4 shows how salt and pepper noise will affect the color image and how the adverse effects will increase when increasing NR.

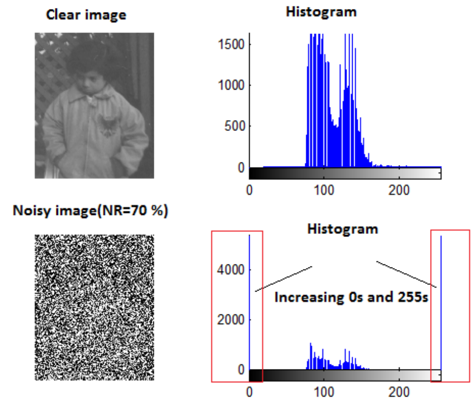

Fig. 3: Salt and pepper noise increases 0s and 255s in the image

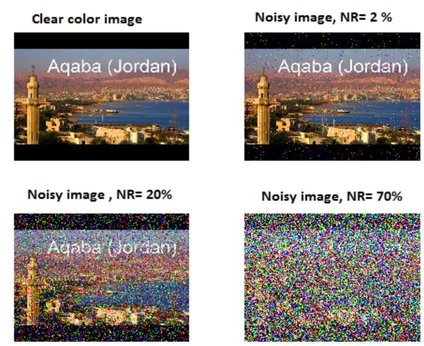

Fig. 4: The negative effects of various NR

Salt and pepper noise can be eliminated or reduced using digital filters. The efficiency of the used filter can be measured between the precise image and the denoised one using mean square error (MSE), peak signal-to-noise ratio (PSNR), correlation coefficient (CC), or structural similarity index measure (SSIM). The main gain of using SSIM is that it measures the quality of the processed images; it measures the similarity between two images by considering the relation between neighboring pixels, unlike PSNR and MSE, where they measure an absolute error. Smaller error values indicate high efficiency, while high values of CC and SSIM mean an enhancement in the denoised images, as is the case with large values of PSNR and CC. The quality parameters MSE, PSNR, and CC can be calculated between two data sets using equations 1, 2, 3, and 4 [6]:

MSE of x channel

$$MSE_x = \frac{1}{N}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[S(i,j)-R(i,j)]^2 \quad ,N = m*n \qquad (1)$$

Total MSE

$$MSE_t = MSE_R + MSE_G + MSE_B$$

Calculate PSNR

$$PSNR = 10*\log_{10}\frac{(MAX_I)^2}{MSE_t} \qquad (2)$$

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}} \quad (3)$$

Where:

r = correlation coefficient

x_i = values of first image matrix

$\bar{x}$ = mean of x matrix

y_i = values of second image matrix

$\bar{y}$ = mean of y matrix

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (4)$$

Where x and y are windows of the same size

- μ_x the average of x;
- μ_y the average of y;
- σ_x^2 the variance of x;
- σ_y^2 the variance of y;
- σ_{xy} the covariance of x and y;

The contribution of this research paper is to provide an easy method that can handle salt and pepper noise, which can affect all kinds of digital images; provided method will deal with all ratios of noise, especially high ones.

The organization of this research paper is as follows: Section 2 presents related work. Section 3 demonstrates the proposed method. Implementation and experimental results are conducted in Section 4, followed by the conclusions in Section 5.

2 Related Work

One of the most famous approaches for image denoising is spatial-based methods, in which a small filter is applied to the whole image taking advantage of the high correlation between the neighbor patches of the image.

Various digital filters are used to mitigate the noise of salt and pepper. The most important types of these filters are the median filter and the arithmetic mean (average) filter, and many digital filters depend on their work on these two filters. The median filter uses a mask to cover the pixel. The neighbor group, where the elements covered by the act are arranged in ascending order, and the value located in the middle is taken to become the new value of the pixel located in the center of the mask, as shown in figure 5 [7-8].

The average filter uses a mask with values of 1s, and each mask value is multiplied by the corresponding pixel value in the image. Then these values are combined and divided by nine to form the new pixel value in the image, as shown in figure 6 [9-10].

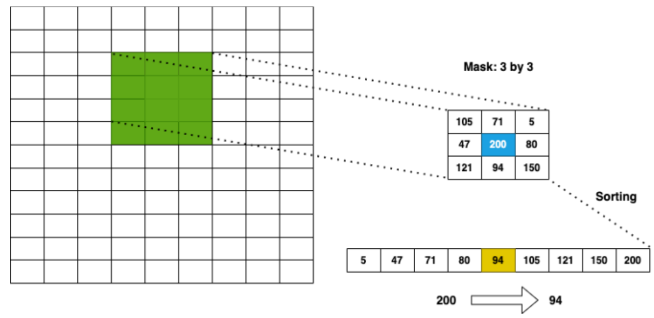

Fig. 5: Median filter operation

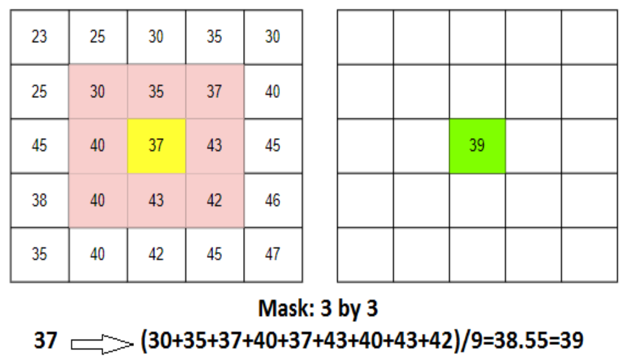

Fig. 6: Average filter operation

Median and average filters apply pixel operations, and all pixels in the noisy image must be treated (here, we must notice that pixels other than 0 or 255 are not noisy pixels); these filters can be used to reduce salt and pepper noise effects. They become inefficient when NR increases, as shown in figures 7 and 8 [11-17].

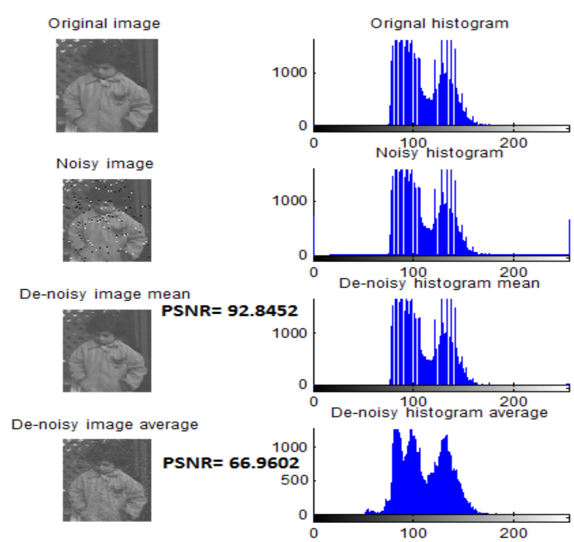

Fig. 7: Denoising salt and pepper noise with NR=2%

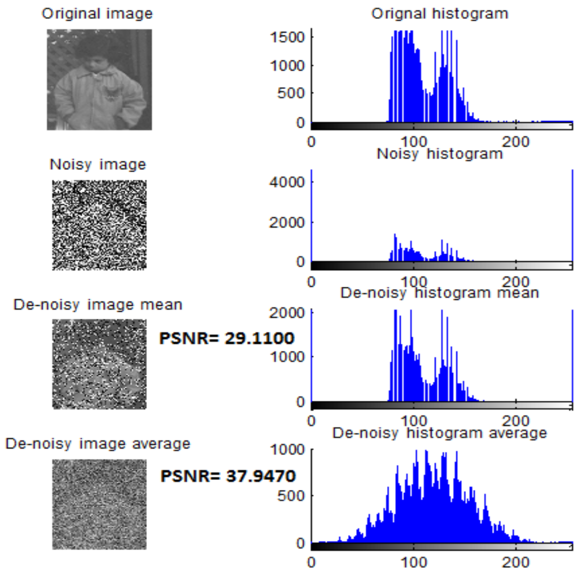

Fig. 8: Denoising salt and pepper noise with NR=60%

From figures 7 and 8, we can see the following:

- Median and average filters have sound effects on dealing with salt and pepper noise for small values of NR.
- The median filter is more efficient when NR is low.
- Median and average filters have poor efficiency when NR is high, and mostly, they fail to reduce the effects of salt and pepper noises with high noise ratios.
- Median and average filters treat all the pixels in the noisy image, even if the pixel is not noisy (values rather than 0 or 255).

Many methods based on median and average filters were proposed to reduce the negative effects of salt and pepper noise infecting the digital image [7-15]. These methods are Modified Decision-Based Unsymmetrical Median Filter (MDBUTMF) [13], Decision-Based Partially Trimmed Global Mean Filter (DBPTGMF) [14], Modified Decision Based Partially Trimmed Global Mean Filter (MDBPTGMF) [15], Adaptive Weighted Mean Filter (AWMF) [17] and Adaptive Approach (AAMF), [16]. These methods improve salt and pepper noise reduction and filtering efficiency by increasing PSNR between the clear and denoised images.

Many proposed approaches in the literature tackled the salt and pepper noise problem using mean and median filters to enhance the restored images' visual quality and increase the PSNR between the clear and denoised images. Table 1 below lists some of the research using a mean filter to resolve salt and pepper noise.

Table 1: Mean filter-related work

Technique	Summary
Decision-Based Partially Trimmed Global Mean Filter (DBPTGMF) [14]	DBPTGMF processes noisy pixels based on four scenarios: salt noise trimmed global mean, pepper noise trimmed global mean, both pepper and salt noise trimmed global mean, or window's median
Modified Decision Based Partially Trimmed Global Mean Filter (MDBPTGMF) [15]	The proposed method consists of two stages. In the first stage, the Decision base Median Filter (DMF) acts as the preliminary noise removal algorithm. The second stage is either Modified Decision Based Partial Trimmed Global Mean Filter (MDBPTGMF) or Modified Decision Based Unsymmetrical Trimmed Median Filter (MDBUTMF) to remove the remaining noise and enhance the image quality.
Adaptive Weighted Mean Filter (AWMF) [17]	AWMF is an adaptive window size method, and the noisy pixel is replaced by the weighted mean of the window
Adaptive Approach (AAMF), [16]	a new pixel value is determined by a fuzzy method reduces its success. Some of these methods have been designed to deal with high-density noise.
Adaptive Riesz Mean Filter (ARmF) [21]	ARmF uses Riessz mean instead of arithmetic mean to measure similarity between pixels, so the closest pixel is most effective ones
Improved Adaptive Weighted Mean Filter (IAWMF) [23]	In IAWMF, the window size is adaptive, and the weights of noiseless pixels are being considered.

While Table 2 lists some well-known work that uses a median filter in salt and pepper noise reduction.

Table 2: Median filter-related work

Technique	Summary
Modified Decision-Based Unsymmetrical Median Filter (MDBUTMF) [13]	MDBUTMF processes corrupted pixels. After the window is determined, the noisy pixels are removed from the window; then the median filter is applied as a final step
Adaptive Right Median Filter (ARMF) [18]	an adaptive nonlinear filter developed via the right median,
Detail-Aware Filter [19]	A two-step method is used. In the first step, a median-type filter was used to process the image corrupted by salt and pepper noise.

	Then, in the second step, a novel-designed adaptive nonlocal bilateral filter is used to weaken the error of the median-type filter.
Different Applied Median Filter (DAMF) [20]	
Noise Adaptive Fuzzy Switching Median Filter (NAFSMF) [22]	An adaptive window size is used instead of a fixed size; the suggested size can be up to 7x7

The filters based on the median filter concept were selected for the following reasons:

- The proposed method depends on the median filter.
- Methods based on the median filter efficiently deal with the salt and pepper noise, significantly if the noise ratio does not exceed 50%.
- The proposed filter is introduced to enhance the efficiency of median-based filters by improving the quality of the treated noisy image, even if the noise ratio exceeds 90%.

Besides the spatial domain approach for image denoising, other approaches use domain transformation-based techniques [24]. The most common practice in domain transformation is wavelet [25,26], in which the noisy image is decomposed into multiple sub-bands with different resolution scales and orientations. The primary benefit is that high-frequency components can be easily discarded from noisy images by thresholding the weight of these generated sub-bands. Hence the noise is deducted as it has a high-frequency component. In [25], the authors presented a clustering-based image restoration using a dictionary learning algorithm in the wavelet domain. They considered second-generation wavelets' main gain in clustering an image in a few subclasses and sparse coding.

In [26], Khmag et al. presented a combined approach for natural image denoising, in which second-generation wavelet and principal components analysis work together to remove noisy components. The noise component spreads all over the signal while the information of the image is clustered on a few subclasses in its domain. Moreover, the second-generation wavelet has the crucial gain of grouping the image's main energy features in small clusters. Thus, combining both techniques will facilitate the noise removal process.

The spatial-based method outperforms the domain transform-based method for salt and pepper impulsive noise because of high correlations between neighboring pixels [27-32].

3 The proposed method

The proposed method is based on a median filter, and it is implemented in two rounds; the pixels in the noisy image whose values are other than 0 and 255 do not need treatment and must be copied to the output clear (denoised) image. The proposed method uses some matrices described in detail in this section.

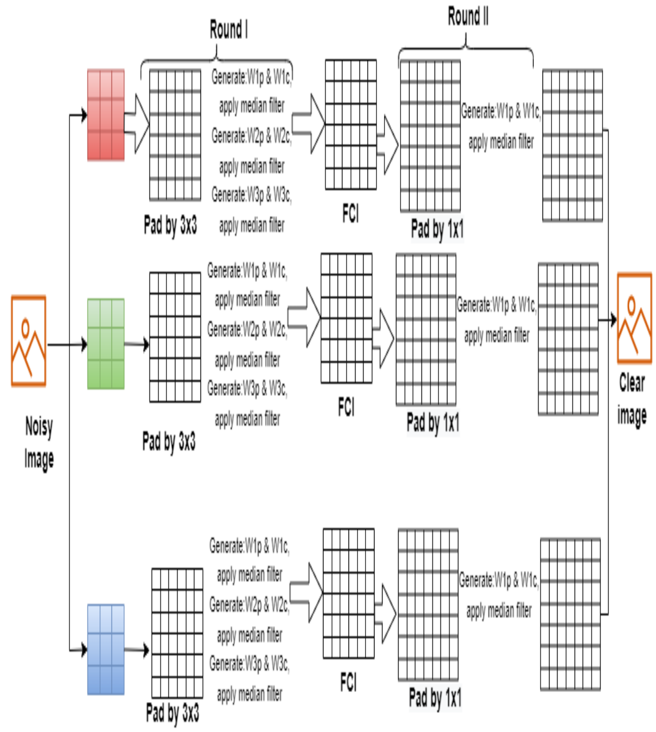

Fig. 9: Proposed Method

Figure 9 depicts the flow of the proposed method. Firstly, the noisy image is split into three different matrices, one for each color channel if the image is colored. On the other hand, if the image is gray, it can go under processing immediately. In the first round, each matrix is padded by 3x3, and six matrices are generated, which are W1c, W1p, W2c, W2p, W3c, and W3p. Half of the generated matrices, W1c, W2c, and W3c, denote the affected pixel by zero value since this method only targets the affected pixels, whose values are either 0 or 255. The processing matrices (W1p, W2p, and W3p) start scanning the noisy image and replacing each affected pixel with the matrix's median value, taking into consideration the importance of the neighbor's pixels. At the end of round 1, the first clear image (FCI) is generated. To further enhance the quality of the denoised image, a second round is applied in which only one matrix is used to eliminate any remaining noisy pixels. FCI image is padded by 1x1, and then the W1p matrix is only applied to it. The three matrices are finally combined to form the final clear image.

The noisy image (a) must be padded 3 by 3 to get the matrix PA. A reference matrix must be created for the noisy image (B); this matrix contains 0s or 1s; 0 indicates that the associated value in the noisy image is a noise pixel, while 1 means that the associated value in the noisy image is a transparent pixel, matrix B must be padded 3 by 3 to get the matrix PB as shown in figure 10.

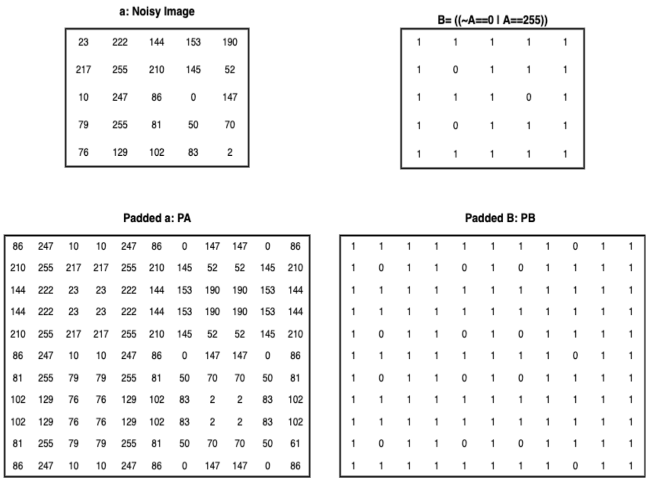

Fig. 10: Generating PA and PB matrices

Round 1 windows are to be created; this round requires three processing windows (WP) and three checking matrices (WC), as shown in figures 11, 12, and 13.

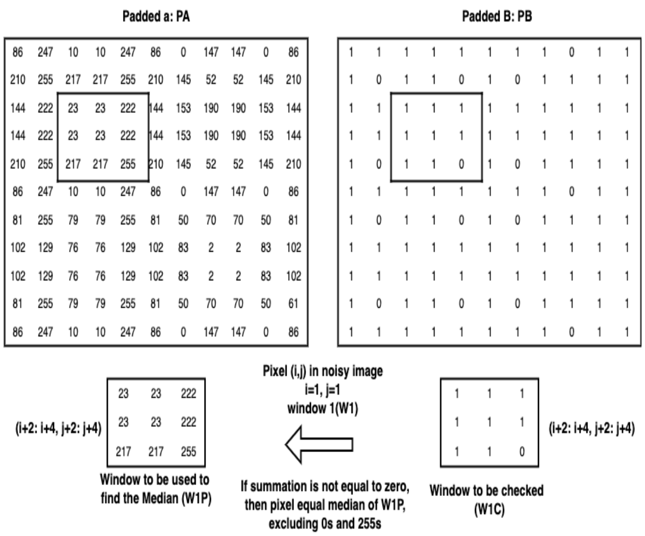

Fig. 11: Generating W1P and W1C

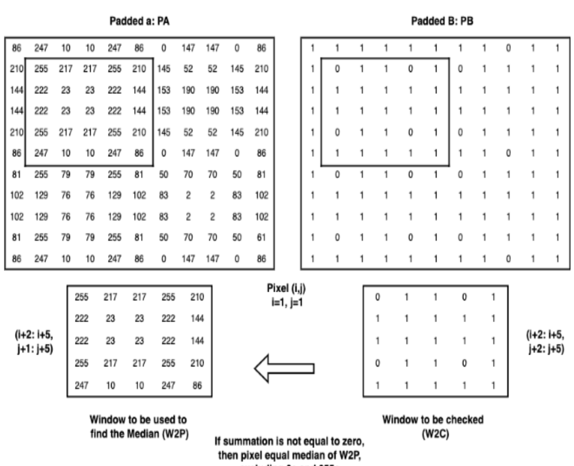

Fig. 12: Generating W2P and W2C

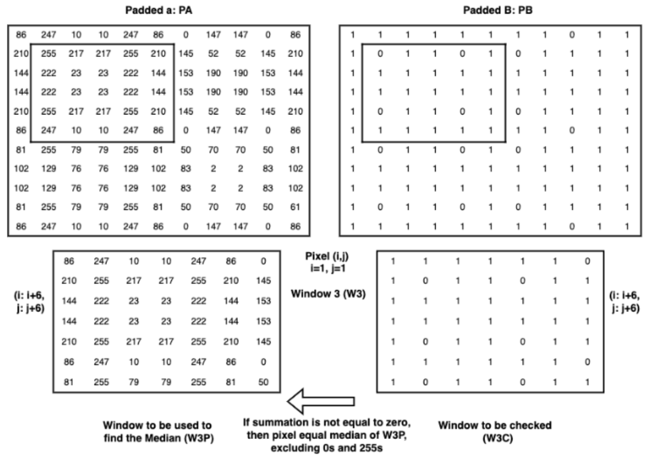

Fig. 13: Generating W3P and W3C

Round 2 matrices: this round will be implemented using the clear image obtained in round 1; this round uses the precise image and the generated PA and PB, but here we must pad the image 1 by 1. Window 1 here is to be used.

The algorithm of the proposed method is as follows (see appendix: the steps are easily changed into a sequence of MATLAB operations):

Inputs:

Noisy image (a)

Output:

Clear image (CI)

Process

Round 1:

1. *Get the noisy image (a) (a is a 2D matrix)*
2. *Retrieve the size of a*
3. *Pad (a) 3 by 3 to get PA*
4. *Calculate the reference matrix B*
5. *Pad (B) 3 by 3 to get PB*
6. *For each pixel (i, j) in a, do the following*
7. *If the pixel does not equal 0 or is not equivalent to 255, add this pixel to the precise image and continue to the next pixel. Else proceed to the next step.*

a) *Generate the W1C window.*

b) *IF the sum of the W1C window does not equal zero, perform the following steps:*

c) *Generate the W1P window.*

d) *Find the elements rather than 0 and 255 in W1P*

e) *Find the median of elements obtained in 2)*

f) *Let the pixel equal the results in 3)* *g) Add this pixel to CI*

h) Continue to another pixel

i) Generate W2C

j) IF the sum of the W2C window does not equal zero, perform the following steps:

k) Generate a W2P window.

l) Find the elements rather than 0 and 255 in W2P

m) Find the median of elements obtained in 2)

n) Let the pixel equal the results in 3)

o) Add this pixel to CI

p) Continue to another pixel

q) Generate W3C

r) IF the sum of the W3C window does not equal zero, perform the following steps:

s) Generate the W3P window.

t) Find the elements rather than 0 and 255 in W3P

u) Find the median of elements obtained in 2)

v) Let the pixel equal the results in 3)

w) Add this pixel to CI

x) Continue to another pixel

Round 2:

Inputs:

Clear image in round 1

Output

Final clear image (FCI)

Process

1. *Find the reference matrix of (FCI) (B)*
2. *Pad FCI to 1 by 1 to get PA.*
3. *Pad B to 1 by 1 to get PB.*
4. *For each pixel in FCI, do the following.*

a) If the pixel does not equal 0 or is not equivalent to 255, add this pixel to the clear image and continue to the next pixel. Else, proceed to the next step

b) Generate W1C

c) ` IF the sum of the W1C window does not equal zero, perform the following steps:

d) Generate the W1P window.

e) Find the elements rather than 0 and 255 in W1P

f) Find the median of elements obtained in 2)

g) Let the pixel equal the results in 3)

h) Add this pixel to CI

i) Continue to another pixel

The proposed method can be used to eliminate salt and pepper noise from a color image, here we must extract each 2D color matrix for red, green, and blue channels, and each of these matrices must be denoised separately using the proposed method; the denoised 2D matrices then must be combined to form the denoised color image.

4 Implementation and experimental results

The proposed method is to reduce the effects of salt and pepper noise on digital images; the method will enhance the quality of the treated noisy image by maximizing the PSNR values, minimizing the MSE values, and increasing SSIM values for any noise ratios (small, medium, and high). The increase in PSNR values and the decrease in MSE values mean an excellent enhancement of the filter performance. The following tests and implementations will be used to prove the enhancements provided by the proposed method.

The proposed method was implemented using various images and applying salt and pepper noises with multiple values of NR. To see the positive effects and the enhancements obtained, we can compare the example results shown in figures 14 and 15 with figures 7 and 8. The proposed method clears the image with a very high NR compared to traditional methods that fail with such a high NR.

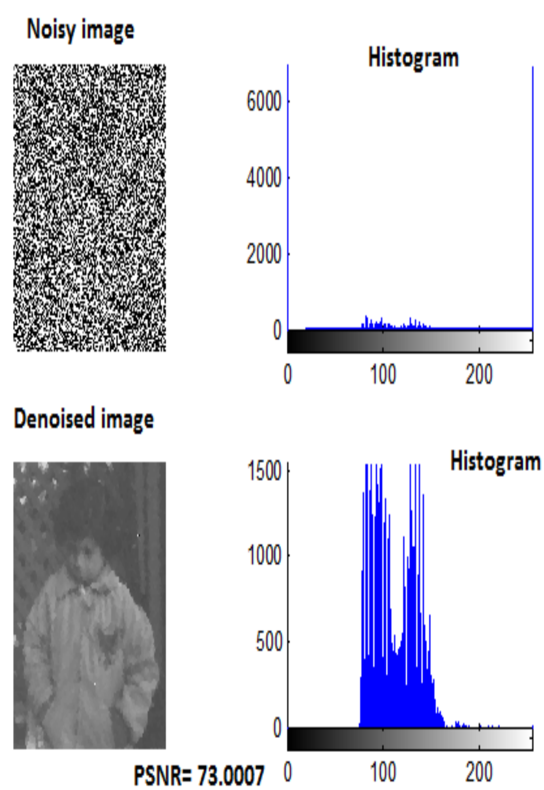

Fig. 14: Denoising 'pout.tif' with NR=90 %

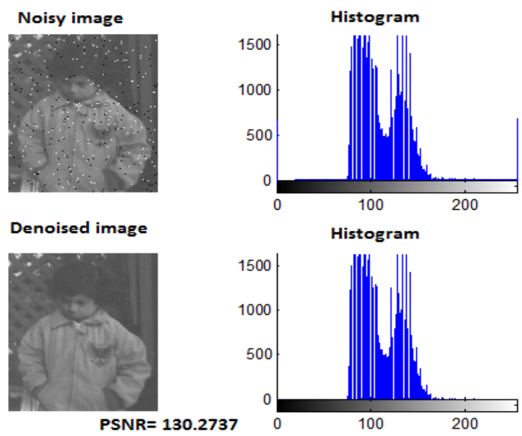

Fig. 15: Denoising 'pout.tif' with NR=2 %

One of the easiest methods of improving the quality of the proposed method is a visualization test analysis. The image histogram can be used for visual analysis of image quality. One of the selected images was affected with a 60% ratio of salt and pepper noise; then ,the proposed method was applied using the noisy image; figures show the obtained images with their color channels histograms.

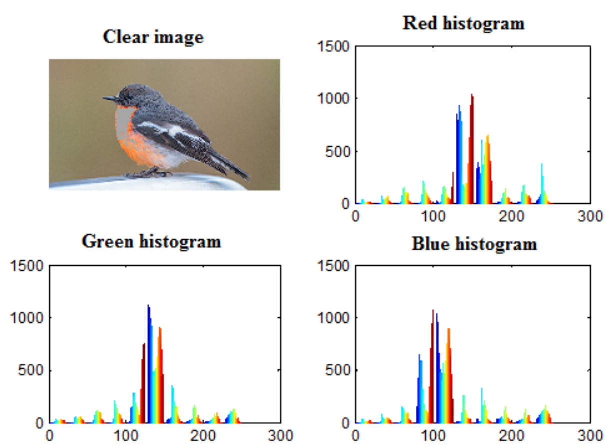

Fig. 16: Clean color image sample

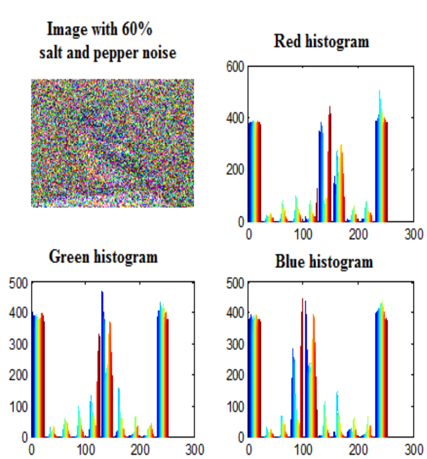

Fig. 17: Noisy color image

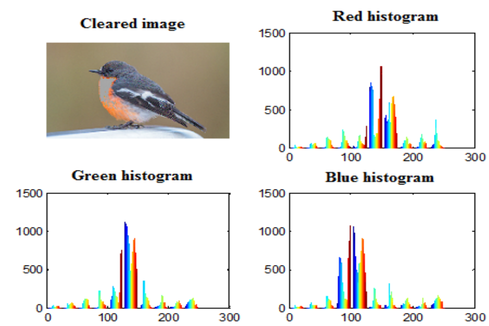

Fig. 18: Cleared color image

Visually we can see that the histograms and the image picture in figure 18 are very close to those in figure 16.

The proposed method was experimentally tested regarding many aspects; below are some of the performed experiments:

Experiment I: effect of two rounds of denoising

Round 2 was added to enhance the performance of the proposed filter. The obtained results are listed in Table 3. 'pout.tif' image with multiple values of NR was treated using the proposed method with one and two rounds. To measure the positive effect of adding a second round to the proposed method, an experiment comparing the impact of two rounds vs. one round with various values of NR is conducted.

Table 3: Denoising noisy images using round 1, and round 2

NR %	1st Round			2nd Rounds		
	MSE	PSNR	SSIM	MSE	PSNR	SSIM
1	0.0396	**140.5101**	0.9997	0.0623	**135.9978**	**0.9997**
3	0.2202	**123.3643**	0.9990	0.1760	**125.6055**	**0.9990**
8	0.4931	**115.3037**	0.9968	0.4691	**115.8014**	**0.9968**
10	0.6660	**112.2970**	0.9962	0.7170	**111.5596**	**0.9962**
20	1.8115	**102.2914**	0.9915	1.5063	**104.1364**	**0.9915**
30	2.9991	**97.2499**	0.9855	2.6107	**98.6366**	**0.9855**
40	4.6787	**92.8027**	0.9783	4.4636	**93.2734**	0.9783
50	6.1962	**89.9935**	0.9690	6.3842	**89.6947**	0.9690
60	9.6394	**85.5743**	0.9572	8.4476	**86.8941**	**0.9572**
70	14.3538	**81.5928**	0.9414	13.0028	**82.5813**	**0.9416**
80	21.2155	**80.2780**	0.9168	21.7925	**77.4173**	**0.9177**
90	160.7123	**60.0291**	0.8000	41.1032	**73.6644**	**0.8655**

Table 3 shows some enhancements using 2 rounds of denoising, as shown in figure 19. Adding an extra round positively impacts the performance and increases the PSNR even with very NR, reaching up to 90%; SSIM increases slightly, especially for high values of NRs. The results are shown in Table 3 and Figure 19.

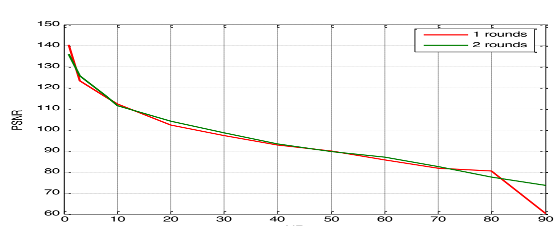

Fig. 19: PSNR using 1 and 2 rounds

Experiment II: Effect of increasing noise ratio

To compare the proposed method against traditional mean and median filters methods, the 'pout.tif' with various values (small values) of NRs was treated using these methods, and table 4 shows the obtained experimental results:

Table 4 : Median and average filters result for small NRs

NR (%)	Median filter				Average filter			
	MSE	**PSNR**	CC	SSIM	MSE	**PSNR**	CC	SSIM
0.5	0.0323	**142.5591**	0.9958	0.9737	33.8468	**73.0145**	0.9688	0.8802
1	4.6027	**92.9664**	0.9957	0.9734	43.0675	**70.6052**	0.9603	0.8270
2	5.4317	**91.3103**	0.9950	0.9729	60.8101	**67.1554**	0.9442	0.7561
3	6.0378	**90.2526**	0.9998	0.9725	80.7136	**64.3238**	0.9944	0.6789
4	6.6230	**89.3274**	0.9938	0.9704	99.0930	**62.2723**	0.9109	0.6176
5	6.3921	**89.6824**	0.9941	0.9683	124.2220	**60.0122**	0.8896	0.5732
6	8.5420	**86.7830**	0.9920	0.9694	143.3196	**58.6712**	0.8738	0.5236
7	8.7668	**86.5232**	0.9918	0.9667	160.6495	**57.4407**	0.8613	0.4939
8	10.9458	**86.8957**	0.9898	0.9636	185.4348	**56.0059**	0.8384	0.4667
9	12.7984	**85.3321**	0.9994	0.9606	204.6310	**55.4624**	0.8341	0.4369

The proposed method (filter) was tested using the same image with the same NR values; Table 5 shows the obtained results.

Table 5: Proposed filter result for minor NR

NR (%)	MSE	PSNR	CC	SSIM
0.5	0.0323	**142.5591**	1.0000	0.9999
1	0.0680	**135.1220**	0.9999	0.9996
2	0.1013	**131.1307**	0.9999	0.9993
3	0.1989	**124.3833**	0.9998	0.9990
4	0.2156	**123.5766**	0.9998	0.9986
5	0.2668	**121.4473**	0.9998	0.9981
6	0.4021	**117.3423**	0.9996	0.9976
7	0.4216	**116.8701**	0.9996	0.9974
8	0.5253	**114.6702**	0.9995	0.9967
9	0.5963	**113.4022**	0.9994	0.9968

Tables 4 and 5 show that the proposed filter increases the values of PSNR; thus, it maximizes the filtering efficiency, as shown in figure 20. In addition, SSIM maintains high values with various values of NRs, like the median filter, while SSIM for the average filter decreases drastically.

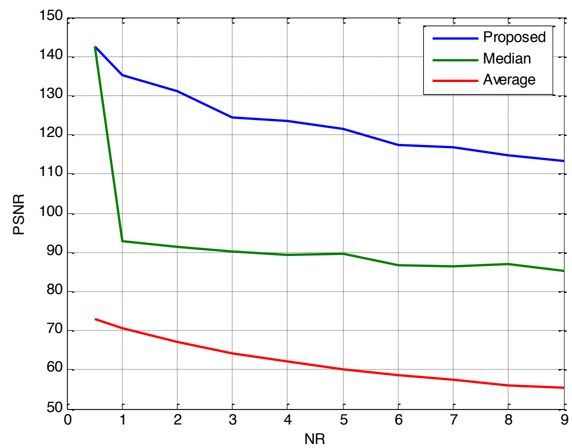

Fig. 20: PSNR comparisons for low NR noise

The previous experiment was repeated using larger values of NR; Tables 6 and 7showthe obtained experimental results:

Table 6 : Median and average filters result for big NR

NR (%)	Median filter				Average filter			
	MSE	**PSNR**	CC	SSIM	MSE	**PSNR**	CC	SSIM
10	**12.8332**	**85.3049**	0.9880	0.9604	221.1981	**54.3314**	0.8133	0.4093
20	**56.4754**	**70.4872**	0.9495	0.8909	433.8466	**47.9475**	0.6717	0.2557
30	**242.0302**	**55.9346**	0.8248	0.7006	662.6474	**43.7992**	0.5551	0.1815
40	**731.1006**	**44.8798**	0.6311	0.4176	883.1450	**41.5263**	0.4433	0.1413
50	1.7044e+003	**36.4155**	0.4431	**0.1744**	**1.1307e+003**	**39.4725**	0.3382	0.1111
60	3.4329e+003	**29.4136**	0.3189	**0.0669**	**1.4235e+003**	**38.2167**	0.2814	0.0901

70	5.7930e+003	**24.1811**	0.1976	**0.0257**	**1.6531e+003**	**36.7211**	0.1991	0.0727
80	9.1603e+003	**19.5990**	0.0992	**0.0141**	**2.0232e+003**	**34.7011**	0.1067	0.0611
90	1.2881e+004	**16.1899**	0.0613	**0.0066**	**2.3101e+003**	**33.3749**	0.0703	0.0494

Table 7 : Proposed filter result for big NR

NR (%)	MSE	PSNR	CC	SSIM
10	0.6151	**113.0928**	0.9994	0.9964
20	1.6723	**103.0907**	0.9984	0.9917
30	2.6829	**98.3638**	0.9975	0.9855
40	4.5516	**93.0781**	0.9958	0.9783
50	6.0841	**90.1761**	0.9943	0.9692
60	9.2801	**85.9542**	0.9913	0.9587
70	13.4360	**82.2535**	0.9874	0.9420
80	20.3219	**78.1159**	0.9810	0.9190
90	44.4632	**72.8787**	0.9586	0.8641

Also, here we can see that the proposed filter enhanced the efficiency of Denoising by maximizing PSNR, CC, and SSIM and minimizing MSE; this is shown in figure 21.

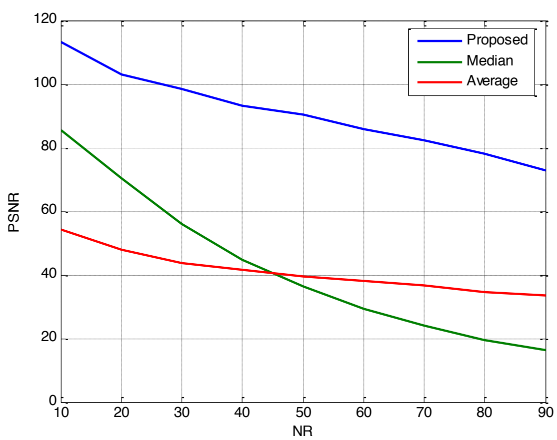

Fig. 21: PSNR comparisons for big NR noise

Figures 20 and 21 show that the proposed method is consistently better than the traditional mean and median filter methods, with different values of NR ranging from small to very high.

The proposed filter was compared with other filters mentioned in the related work; Table 8 shows the obtained results of these filters:

Table 8 : Filters comparison

NR %	PSNR				
	Proposed	MDBUT MF	MDBPT GMF	AWM F	AAMF
10	**113.0928**	106.0674	104.9845	111.4240	112.4038
20	**103.0907**	101.2922	102.0092	102.2062	109.6104
30	**98.3638**	81.2179	82.3682	85.8416	91.9312
40	**93.0781**	63.9605	65.0165	67.3288	70.7881
50	**90.1761**	50.3720	51.9859	52.4341	54.1376
60	**85.9542**	38.7028	40.1875	41.6846	44.4209
70	**82.2535**	30.1583	30.4933	32.0426	38.5340
80	**78.1159**	25.6423	27.2584	28.4141	30.3927
90	**72.8787**	24.5940	26.2044	27.5703	26.5740

From Table 8 ,we can see that the proposed filter has the best performance by maximizing the PSNR value ,as shown in figure 22.

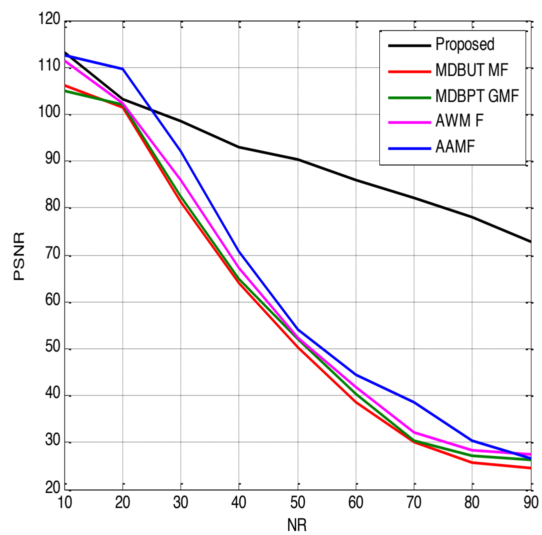

Fig. 22: Filters performance comparisons

Experiment III: color and gray images denoising

As we said, the proposed filter can be used to eliminate salt and pepper noise from the color image; here ,each color matrix must be treated alone, and figure 23 shows sample outputs of denoising the color image.

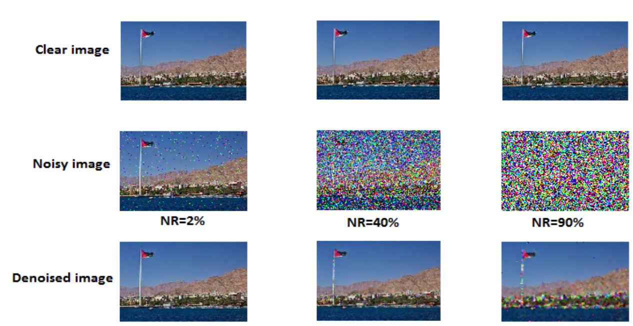

Fig. 23: Denoising color image

Low and significant NR values were used to affect the color image; Tables 9, 10 ,and 11 show the obtained results:

Table 9 : Denoising color image using the proposed filter

NR %	MSE	PSNR	CCR	CCG	CCB	SSIM
1	18.7210	**81.5288**	0.9969	0.9949	0.9936	0.9980
3	25.0759	**78.6062**	0.9956	0.9923	0.9921	0.9956
8	39.2879	**74.1161**	0.9935	0.9861	0.9883	0.9894
10	45.6989	**72.6045**	0.9914	0.9842	0.9873	0.9868
20	82.1979	**66.7340**	0.9829	0.9700	0.9796	0.9729
30	118.6333	**63.0649**	0.9763	0.9535	0.9710	0.9574
40	159.0226	**60.1348**	0.9680	0.9393	0.9597	0.9395
50	212.5231	**57.2348**	0.9577	0.9139	0.9477	0.9189
60	271.3848	**54.7899**	0.9448	0.8939	0.9314	0.8940
70	348.8373	**52.2792**	0.9287	0.8595	0.9141	0.8651
80	443.2053	**49.8849**	0.9089	0.8204	0.8926	0.8264
90	615.9854	**46.5930**	0.8714	0.7544	0.8521	0.7649

Table 10 : Denoising color image using the median filter

NR %	MSE	PSNR	CCR	CCG	CCB	SSIM
1	233.4400	**56.2960**	0.9530	0.9041	0.9434	0.8928
3	239.0592	**56.0582**	0.9515	0.9026	0.9417	0.8911
8	253.1031	**55.4873**	0.9497	0.8957	0.9373	0.8856
10	262.2732	**55.1314**	0.9473	0.8922	0.9353	0.8829
20	354.0093	**52.1320**	0.9270	0.8593	0.9125	0.8475
30	610.2096	**46.6872**	0.8704	0.7796	0.8595	0.7499
40	1.1842e+003	**40.0567**	0.7830	0.6461	0.7524	0.5852
50	2.2868e+003	**33.4763**	0.6423	0.4938	0.6146	0.4066
60	4.2620e+003	**27.2502**	0.4810	0.3501	0.4355	0.2587
70	7.0603e+003	**22.2028**	0.3175	0.2269	0.3085	0.1522
80	1.0267e+004	**18.4588**	0.2257	0.1569	0.1858	0.0785
90	1.4325e+004	**15.1279**	0.1020	0.0658	0.0963	0.0339

Table 11 : Denoising color image using average filter

NR %	MSE	PSNR	CCR	CCG	CCB	SSIM
1	279.2120	**54.5056**	0.9454	0.8869	0.9281	0.8582
3	326.0870	**52.9536**	0.9343	0.8673	0.9168	0.8129
8	443.3770	**49.8811**	0.9071	0.8192	0.8890	0.7297
10	502.2869	**48.6336**	0.8960	0.7967	0.8728	0.7027
20	796.0626	**44.0285**	0.8214	0.6922	0.8085	0.5888
30	1.1415e+003	**40.4245**	0.7402	0.6001	0.7172	0.4947
40	1.4844e+003	**37.7973**	0.6675	0.5046	0.6255	0.4104
50	1.8949e+003	**35.3559**	0.5612	0.4087	0.5322	0.3293
60	2.4092e+003	**32.9548**	0.4492	0.3132	0.3972	0.2599
70	2.8944e+003	**31.1200**	0.3252	0.2279	0.3077	0.1923
80	3.3984e+003	**29.5146**	0.2405	0.1691	0.1975	0.1279
90	3.9708e+003	**27.9582**	0.1192	0.0762	0.1103	0.0733

From tables 9, 10 ,and 11 ,we can see that the proposed filter added an enhancement to the process of color image denoising ,as shown in figure 24. SSIM decreases acutely for traditional median and average filters while maintaining an excellent value for the proposed method.

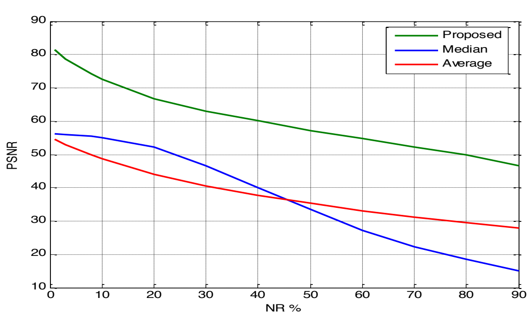

Fig. 24: PSNR comparisons of color image Denoising

Experiment IV: proposed method throughput

The proposed method does not require much time to remove the noise of salt and pepper from the digital image. The duration of denoising will depend on the size of the image and the noise ratio, and by increasing the value of any of them, the time will increase, and it should be noted that removing the noise from the color digital image would take time larger due to working with three-color matrices, as shown in figure 25.

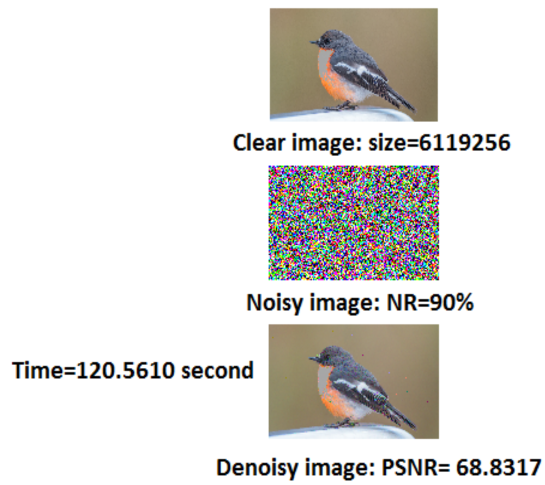

Fig. 25: Time to denoised large color image with high NR

Five gray images were affected by salt and pepper noise with NR of =3%; Table 12 shows the results of denoising these images:

Table 12 : Denoising salt and pepper noise with NR=3%

Image number	Resolution	Size (byte)	PSNR	Denoising time(second)
1	152 x 171	25992	79.4915	0.050000
2	151 x 333	50283	77.3515	0.089000
3	600 x 1050	630000	83.0705	1.406000
4	1071 x 1600	1713600	102.8204	2.712000
5	1144 x 1783	2039752	117.7031	3.192000
Average		891930		1.4898
Throughput (byte per second)			598690	

The denoising time will be increased if the value of NR increases, as shown in Table 13.

Table 13 : Denoising salt and pepper noise with NR=90%

Image number	Resolution	Size (byte)	PSNR	Denoising time(second)
1	152 x 171	25992	39.8209	0.503000
2	151 x 333	50283	35.8264	0.960000
3	600 x 1050	630000	46.0093	12.327000

4	1071 x 1600	1713600	54.8674	34.137000
5	1144 x 1783	2039752	70.4106	40.448000
Average		891930		17.6750
Throughput (byte per second)		50463		

From Tables 12 and 13, we can see that the proposed method (filter) has a good throughput; for small images affected by low-intensity salt and pepper noise, the required time for de-noising is very low, and the filters provided an excellent throughput, this throughput will be dropped when increasing the image size, or the NR or increasing both.

Experiment V: Benchmarking results

The appendix includes the code that is used to implement the proposed filter. Moreover, detailed experiments were conducted on some sample benchmark images embedded in MATLAB images.

The experiments compare the proposed filter against traditional mean and median filters. It compares these filters to MSE, PSNR, CC-red, CC-green, and CC-blue.

5 Conclusion

An efficient digital filter was proposed to mitigate the salt and pepper noise problem; it was shown that this filter increased the efficiency of filtering by increasing PSNR, CC, and decreasing MSE for both noisy gray and color images.

The proposed filter uses unique generated windows that target only the affected pixels by salt and pepper noise; the process of filtering was implemented by applying two rounds of filtering using the generated windows; this procedure increased the efficiency of the proposed filter by increasing PSNR and CC and decreasing MSE.

The proposed method (filter) was evaluated using various NR values, the obtained results were compared with the implementation results of median and average filters, and it was shown that the proposed filter added a significant improvement in the process of digital image filtering by minimizing the negative effects of salt and pepper noise, which affects digital images.

The famous digital filters (DBPTGMF, (MDBPTGMF), (AWMF) (and AAMF) were also evaluated, the evaluation results of these filters were also compared with the proposed filter results, and it was shown that the proposed filter provided the best possible efficiency. The proposed method can be easily used to treat any image, and it was revealed that it is very efficient in de-noising gray and color images.

The proposed method was tested using various ranges of salt and pepper noise ratios. They performed a quality test analysis proving the proposed method's quality for low and high salt and pepper noise ratios. It was confirmed that the provided method enhanced the filtering quality by sufficiently increasing the PSNR values and decreasing the MSE values, especially for the high-ratio noises.

It was shown that the proposed method provided an excellent throughput. The throughput depends on the image size and the noise ratio, decreasing when the image size and the noise ratio increase.

Conflict of interest

The authors declare that there is no conflict regarding the publication of this paper.

References

[1] M. Abu-Faraj, Z. Alqadi, B. Al-Ahmad, K. Aldebei, and B. Ali, "A Novel Approach to Extract Color Image Features using Image Thinning," Applied Mathematics & Information Sciences (AMIS), vol.16, no. 5, pp. 665-672, 2022, doi:10.18576/amis/160501.

[2] M. Abu-Faraj, Z. Alqadi, and M. Zubi, "Creating Color Image Feature Based on Morphology Image Processing," Traitement du Signal, vol. 39, no. 3, pp. 797-803, 2022, doi:10.18280/ts.390304.

[3] M. Abu-Faraj, K. Aldebei, and Z. Alqadi, "Simple, Efficient, Highly Secure, and Multiple Purposed Method on Data Cryptography," Traitement du Signal, vol. 39, no. 1, pp. 173-178, 2022, doi:10.18280/ts.390117.

[4] M. Abu-Faraj, A. Al-Hyari, and Z. Alqadi, "A Complex Matrix Private Key to Enhance the Security Level of Image Cryptography," Symmetry, vol. 14, Iss. 4, pp. 664-678, 2022, doi.org/10.3390/sym14040664.

[5] M. Abu-Faraj, and M. Zubi, "Analysis and Implementation of Kidney Stones Detection by applying Segmentation Techniques on Computerized Tomography Scans," Italian Journal of Pure and Applied Mathematics, is. 43, pp. 590-602, 2020.

[6] M. Abu-Faraj, A. Al-Hyari, K. Aldebei, B. Al-Ahmad, and Z. Alqadi, "Rotation Left Digits to Enhance the Security Level of Message Blocks Cryptography," IEEE Access, vol. 10, pp. 69388-69397, 2022.

[7] M. Abu-Faraj, A. Al-Hyari, I. Al-taharwa, B. Al-Ahmad, and Z. Alqadi, "CASDC: A Cryptographically Secure Data System Based on Two Private Key Images," IEEE Access, vol. 10, pp. 126304-126314, 2022.

[8] T. A. Nodes and N. C. Gallagher, "Median filters: some modifications and their properties," IEEE Trans.

Acoustics, Speech and Signal Processing, vol. 30, no.5, pp. 739-746, Oct. 1982.

[9] R. Yang, L. Lin, M. Gabbouj, J. Astola, and Y. Neuvo, "Optimal weighted median filters under structural constraints," IEEE Trans. Signal Processing, vol. 43, no.3, pp. 591-604, Mar. 1995.

[10] J. B. Bednar and T. K. Watt, "Alpha-trimmed means and their relationship to median filter," IEEE Trans. Acoustics, Speech, and Signal Processing, vol. 32, no.1, pp. 145-153, Jan. 1984.

[11] K. S. Srinivasan and D. Ebenezer, "A new fast and efficient decision-based algorithm for removal of high-density impulse noise," IEEE Signal Process. Lett., vol. 14, no. 6, pp. 1506-1516, Jun. 2006.

[12] K. Aiswarya, V. Jayaraj, and D. Ebenezer, " A new and efficient algorithm for the removal of high-density salt and pepper noise in images and videos," in Second Int. Conf. Computer Modeling and Simulation,2010, pp. 409- 413

[13] S. Esakkirajan, T. Veerakumar, A. N. Subramanyam, and C. H. PremChand, "Removal of high-density salt and pepper noise through a modified decision based unsymmetric trimmed median filter," IEEE Signal Process. Lett, vol. 18, pp. 287-290, May. 2011.

[14] M.T. Raza, And S. Sawant, "High-density salt and pepper noise removal through a decision based partially trimmed global mean filter," Engineering (NUiCONE), 2012 Nirma University International Conference on, pp.1-5, IEEE, Dec. 2012

[15] A. Dash and S. K. Sathua, "High-Density Noise Removal By Using Cascading Algorithms," 2015 Fifth International Conference on Advanced Computing & Communication Technologies, pp. 96-101, IEEE Computer Society, Feb. 2015

[16] Sujaya Kumar Sathua , Arabinda Dash , Aishwaryarani Behera , Removal of Salt and Pepper noise from Gray-Scale and Color Images: An Adaptive Approach, International Journal of Computer Science Trends and Technology (IJCST) – Volume 5 Issue 1, Jan – Feb 2017.

[17] P. Zhang and F. Li, "A New Adaptive Weighted Mean Filter for Removing Salt and Pepper Noise," IEEE Signal Process. Lett., vol. 21, no. 10, pp. 1280-1283, Oct. 2014.

[18] Erkan, U., Gökrem, L. and Enginoğlu, S., 2019. Adaptive Right Median Filter for Salt-and-Pepper Noise Removal. International journal of engineering research and development, 11(2), pp.542-550.

[19] Liang, H., Li, N. and Zhao, S., 2021. Salt and pepper noise removal method based on a detail-aware filter. Symmetry, 13(3), p.515.

[20] Erkan, U., Gökrem, L. and Enginoğlu, S., 2018. Different applied median filter in salt and pepper noise. Computers & Electrical Engineering, 70, pp.789-798.

[21] Enginoğlu, S., Erkan, U. and Memiş, S., 2019. Pixel similarity-based adaptive Riesz mean filter for salt-and-pepper noise removal. Multimedia Tools and Applications, 78(24), pp.35401-35418.

[22] Toh KKV, Isa NAM (2010) Noise Adaptive Fuzzy Switching Median Filter for Salt-and-Pepper Noise Reduction. 17:281–284

[23] Erkan, U., Thanh, D.N., Enginoğlu, S. and Memiş, S., 2020, June. Improved adaptive weighted mean filter for salt-and-pepper noise removal. In 2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE) (pp. 1-5). IEEE.

[24] Shao, L., Yan, R., Li, X. and Liu, Y., 2013. From heuristic optimization to dictionary learning: A review and comprehensive comparison of image denoising algorithms. IEEE transactions on cybernetics, 44(7), pp.1001-1013.

[25] Khmag, A., Ramli, A.R. and Kamarudin, N., 2019. Clustering-based natural image denoising using dictionary learning approach in wavelet domain. *Soft computing*, *23*(17), pp.8013-8027.

[26] Khmag, A., Ramli, A.R., Al-Haddad, S.A.R., Hashim, S.J., Noh, Z.M. and Najih, A.A., 2015. Design of natural image denoising filter based on second-generation wavelet transformation and principle component analysis. *Journal of medical imaging and health informatics*, *5*(6), pp.1261-1266.

[27] S. A. Nuaimi, H. Al-Ahmad, and M. Al-Mualla, "2D sub-optimum filters for sharpening interpolated satellite images by optimizing the structural similarity index measure," 2014 9th International Symposium on Communication Systems, Networks & Digital Sign (CSNDSP), 2014, pp. 668-672, doi: 10.1109/CSNDSP.2014.6923911.

[28] Y. Shahriari, R. Fidler, M. M. Pelter, Y. Bai, A. Villaroman and X. Hu, "Electrocardiogram Signal Quality Assessment Based on Structural Image Similarity Metric," in IEEE Transactions on Biomedical Engineering, vol. 65, no. 4, pp. 745-753, April 2018, doi: 10.1109/TBME.2017.2717876.

[29] K. Ma, Z. Duanmu, H. Yeganeh and Z. Wang, "Multi-Exposure Image Fusion by Optimizing A Structural Similarity Index," in IEEE Transactions on Computational Imaging, vol. 4, no. 1, pp. 60-72, March 2018, doi: 10.1109/TCI.2017.2786138.

[30] M. Abu-Faraj, and Z. Alqadi, "Rounds Reduction and Blocks Controlling to Enhance the performance of Standard Method of Data Cryptography,"

International Journal of Computer Science and Network Security (IJCSNS), vol. 21, no. 12, pp. 648-656, 2021, doi: 10.22937/IJCSNS.2021.21.12.89.

[31] M. Abu-Faraj, and Z. Alqadi, "Improving the Efficiency and Scalability of Standard Methods for Data Cryptography," International Journal of Computer Science and Network Security (IJCSNS), vol. 21, no.12, pp. 451-458, 2021, doi:10.22937/IJCSNS.2021.21.12.61.

[32] M. Abu-Faraj, and Z. Alqadi, "Using Highly Secure Data Encryption Method for Text File Cryptography," International Journal of Computer Science and Network Security (IJCSNS), vol. 21, no.12, pp. 53-60, 2021, doi:10.22937/IJCSNS.2021.21.12.8.