International Journal of Engineering & Technology, 7 (4) (2018) 5026-5031

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet. v7i4.26950
Research paper

Development of Algorithm for Dynamic Hybrid Mean/Median Filter using Adaptive Window Selection Approach to Eliminate Salt & Pepper Noise

Neha Patil [1] *, Dr. V. R. Udupi [2]

[1] *Research Scholar, CSE Department, GIT, VTU Belagavi, 590008, India*
[2] *Professor, ECE Department, MMEC, VTU Belagavi, 591113, India*
**Corresponding author E-mail: neha.nsp@gmail.com*

Abstract

The research work aimed to develop a new Dynamic Hybrid Mean/Median Filter (DHMMF) algorithm to eliminate salt & pepper noise. The proposed DHMMF algorithm decides window size dynamically during runtime, also adaptively adjusts window size based on the non-noisy pixels present in a local window. Window size is limited to 9 X 9. This reduces blurring and computational complexity. Filter is designed with two stages, noise detection succeeded by filtering strategy. During the noise detection stage, if pixel intensity value is in-between 1 to 254, it is classified as a non-noisy pixel and left unchanged. Pixel having 0 or 255 intensity value is classified as a noisy pixel. During the filtering stage, a noisy pixel is replaced with mean, median or trimmed values within a local window depending on various algorithmic conditions. Performance of DHMMF algorithm is compared with various existing methods. Performance is tested for low, medium and high density noise. Simulation results demonstrate that image quality is retained by preserving fine details and edges which results in better visual quality. Quantitative and qualitative analysis is carried out using PSNR and SSIM respectively.

Keywords: *Adaptive; Filter; Multimedia Data; Noisy Data; Salt Pepper Noise.*

1. Introduction

Salt & Pepper noise gets introduced in images during the acquisition process, faulty memory location, atmosphere conditions, bit error during transmission, synchronization error and malfunction in capturing devices like misaligned lenses, camera sensors and weak focal length [1], [2]. Such noisy image pixels exhibit either minimum 0 or maximum 255 gray level intensity value. This causes the image quality degradation and loss of fine edges [3], [4].

In multimedia data, de-noising is a necessary pre-processing step for image processing operations. It is required to eliminate the noise to restore the data. There are various linear and non-linear filtering methods exists [5 - 12]. For additive noise, linear filtering is useful, but it blurs the image and losses fine details. Hence most methods use non-liner filter to preserve fine details and edges. Median filter is simple robust which replaces all pixels with median of neighborhood pixels present in a current local window. This achieves acceptable results in low noise densities but results in a blur image and it also process non-noisy pixels. The proposed DHMMF algorithm developed to retain a high-quality image and it is compared with various existing methods.

In this research paper, Section 2 explains the noise model, Section 3 describes a new DHMMF algorithm, Section 4 gives various adaptive window scenarios, Section 5 explains performance measures, Section 6 demonstrate comparison and simulation results, Section 7 the conclusion of research objective.

2. Noise Model

Salt & Pepper noise consists of pixels with minimum value $r_{min} = 0$ and maximum value $r_{max} = 255$. Progressively white pixel values will be present in the dark region and vice versa. A noisy image with Salt & pepper noise [2] defined as:

$$Z(i,j) = \begin{cases} r_{min} & with\ probability\ p_1 \\ p(i,j) & with\ probability\ 1 - p_1 - p_2 \\ r_{max} & with\ probability\ p_2 \end{cases} \qquad (1)$$

where

$Z(i,j)$ is a noisy image

$p(i,j)$ is a non-noisy pixel.

197	123	110
180	156	77
56	109	94

→

197	123	110
180	0	77
56	109	94

Fig. 1: Image Metrics 3 X 3 with a Corrupted Central Pixel.

Consider 3 X 3 image metrics as shown in Figure 1. Suppose salt & pepper noise is introduced in the image, a central pixel value i.e. 156, is changed to 0.

3. Proposed DHMMF Algorithm

The proposed DHMMF algorithm combines the advantages of both mean and median filter. Current local window size is considered as threshold value. If the number of non-noisy pixels <

threshold, then window size is increased by W=W+2 till it reaches the maximum window size limit W=9 to avoid blurring and computational complexity. A noisy pixel is detected and replaced with below algorithmic conditions.

$Input:$ $The\ noisy\ image\ Z$
$Output:$ $The\ filtered\ image\ X$

Step 1 : Set $W = 3, h = 2, W_{max} = 9$

Step 2 : Processing starts with a sub-window $W\ X\ W$. Consider the center pixel as Z_{ij}

Step 3 : If $0 < Z_{ij} < 255$, then keep Z_{ij} unchanged. Shift to next sub-window.
Go to Step 1

Step 4 : If the center pixel $Z_{ij} = 0$ or $Z_{ij} = 255$ then it is considered as corrupted pixels.

Step 5 : Pixels except 0 and 255 are collected in S

Step 6 : If $size(S) \geq W_{current}$ and $W_{current} \leq W_{max}$ then replace Z_{ij} with $Median(S)$
Go to Step 10

Step 7 : If $size(S) < W_{current}$ and $W_{current} \leq W_{max}$ then window size is increased as $W = W_{current} + h$
Go to Step 2

Step 8 : If $size(S) < W_{current}$ and $W_{current} = W_{max}$ then replace the center pixel Z_{ij} as per following cases:
Case 1 : If $size(S) < 7$ then replace Z_{ij} with $Mean(S)$
Go to Step 10
Case 2 : If $size(S) \geq 7$ then replace Z_{ij} with $Median(S)$
Go to Step 10

Step 9 : If $W_{current} \geq W_{max}$ then replace the center pixel Z_{ij} as per following cases:
Let $A = \{0\}$ and $B = \{255\}$
Case 1 : If $v \in A$ then replace Z_{ij} with 255
Go to Step 10
Case 2 : If $v \in B$ then replace Z_{ij} with 0
Go to Step 10
Case 3 : If $v \in (A \cup B)$ then replaceZ_{ij} with $Mean(v)$
Go to Step 10

Step 10 : Select next sub-window $W\ X\ W$. Repeat Step 2 to Step 7

Summary of DHMMF Filter algorithm implementation approach:

- Proposed algorithm is applied on a noise image, $Z(i,j)$

$$g(i,j) = EDHMMF\ (Z(i,j)) \quad (2)$$

- Filtered image is restored as,

$$h(i,j) = \begin{cases} Z(i,j) & (i,j)\ uncurrupted\ pixels \\ g(i,j) & other\ pixels \end{cases} \quad (3)$$

4. Adaptive Window Scenarios

1) Center pixel intensity value is except 0 and 255. It will considered as a non-noisy pixel and keep it unchanged.

197	123	110
180	84	77
56	109	94

→

197	123	110
180	84	77
56	109	94

Fig. 2: Scenario 1.

2) Center pixel intensity value is 0, indicates noisy pixel. Current window size is W = 3.
S= {197, 123, 110, 180, 77, 56, 109, 94}, size (S) = 8
As size (S) > W, the center pixel is replaced with median(S).

197	123	110
180	0	77
56	109	94

→

197	123	110
180	110	77
56	109	94

Fig. 3: Scenario 2.

3) Center pixel intensity value is 0, indicates noisy pixel. Current window size is W = 3.
S= {197,123}, size (S) = 2
Here size (S) < W, W is increased to W+2.
S={67,99,102,197,123,100,10,6,12,120,111}
As size(S) > W, the center pixel is replaced with median(S).

197	123	0
0	0	255
0	255	255

→

0	0	67	99	255
102	197	123	0	0
100	0	100	255	10
6	0	255	255	12
120	0	0	255	111

Fig. 4: Scenario 3.

4) Center pixel intensity value is 0, indicates noisy pixel. Current window size W is expanded till W=9 until size (S) > W. Suppose maximum window size is reached.
S= {98,52,8,4,155 ,6} , size(S) = 6
As size(S) < 7, the center pixel is replaced with $Mean(S)$

0	98	0	0	0	0	0	0	0
255	0	0	0	0	52	0	0	0
8	0	0	255	0	0	0	0	0
0	0	0	0	0	255	255	4	0
0	255	0	0	0	0	0	155	0
0	0	0	0	0	0	0	0	255
0	0	255	0	6	0	0	0	0
0	0	0	0	0	0	255	0	0
0	255	0	0	255	0	0	0	0

↓

0	98	0	0	0	0	0	0	0
255	0	0	0	0	52	0	0	0
8	0	0	255	0	0	0	0	0
0	0	0	0	0	255	255	4	0
0	255	0	0	53.8	0	0	155	0
0	0	0	0	0	0	0	0	255
0	0	255	0	6	0	0	0	0
0	0	0	0	0	0	255	0	0
0	255	0	0	255	0	0	0	0

Fig. 5: Scenario 4.

5) Center pixel intensity value is 0, indicates noisy pixel. Current window size W is expanded till W=9 until size (S) > W. Suppose maximum window size limit is reached.
S = {8, 98, 89, 44, 100, 66, 6, 52, 102}, size(S) = 9

As size(S) > 7, the center pixel is replaced with $Median(S)$

0	98	0	0	0	0	0	0	0
255	0	100	0	0	52	0	0	0
8	0	0	255	0	0	0	0	0
0	0	0	0	0	255	255	0	0
0	255	0	66	0	0	0	255	0
0	0	0	0	0	102	0	0	255
0	89	255	0	6	0	0	0	0
0	0	0	0	0	0	255	0	0
0	44	0	0	255	0	0	0	0

↓

0	98	0	0	0	0	0	0	0
255	0	100	0	0	52	0	0	0
8	0	0	255	0	0	0	0	0
0	0	0	0	0	255	255	0	0
0	255	0	66	66	0	0	255	0
0	0	0	0	0	102	0	0	255
0	89	255	0	6	0	0	0	0
0	0	0	0	0	0	255	0	0
0	44	0	0	255	0	0	0	0

Fig. 6: Scenario 5.

6) Center pixel intensity value is 0, indicates a noisy pixel. Current window size W is expanded till W=9 until size (S) > W. Suppose maximum window size limit is reached and all pixels are with intensity value 0, then replace the center pixel with intensity value 255.

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

↓

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	255	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Fig. 7: Scenario 6.

7) Center pixel intensity value is 255, indicates noisy pixel. Current window size W is expanded till W=9 until size (S) > W. Suppose maximum window size limit is reached and all pixels are with intensity value 255, then replace the center pixel with intensity value 0.

255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255

↓

255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255
255	255	255	255	0	255	255	255	255
255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255

Fig. 8: Scenario 7.

8) Center pixel intensity value is 255, indicates a noisy pixel. A current window size W is expanded till W=9 until size (S) > W. Suppose maximum window size limit is reached and all pixels having intensity value 0 or 255, then replace the center pixel with $Mean(v)$.

255	0	255	0	0	0	0	0	255
0	0	0	0	255	0	0	0	0
0	0	0	0	255	0	0	0	0
255	0	0	0	0	0	0	255	255
0	0	0	0	255	0	0	0	0
0	0	0	0	0	0	255	00	0
255	0	0	0	0	0	0	255	255
0	255	0	0	255	255	0	0	255
255	255	255	0	0	0	0	255	0

↓

255	0	255	0	0	0	0	0	255
0	0	0	0	255	0	0	0	0
0	0	0	0	255	0	0	0	0
255	0	0	0	0	0	0	255	255
0	0	0	0	62.9	0	0	0	0
0	0	0	0	0	0	255	00	0
255	0	0	0	0	0	0	255	255
0	255	0	0	255	255	0	0	255
255	255	255	0	0	0	0	255	0

Fig. 9: Scenario 8.

5. Performance Measures

1) PSNR: Watermarked image quality is estimated with respect to an original image using PSNR.

$$PSNR = 10\ log_{10}\ \frac{Z_{max}^{2}}{\frac{1}{MN}\sum_{i,j}[q(i,j)-p(i,j)]^{2}} \quad (4)$$

where $Z_{max} = 255$

$p(i,j)$ is intensity values of an input image
$q(i,j)$ is intensity values of the watermarked image

2) SSIM: Structured similarity is measured by considering luminance, correlation and contrast.

$$SSIM = \frac{(2\mu_f \mu_h + C_1)(2\sigma_{fh} + C_2)}{(\mu_f^2 + \mu_h^2 + C_1)(\sigma_f^2 + \sigma_h^2 + C_2)} \quad (5)$$

where $C_1 = (P_1 * G)^2, C_2 = (P_2 * G)^2$
$G = 255, P_1 = 0.01, P_2 = 0.03$
μ_f and μ_h are average of f and h respectively
σ_f and σ_h are variance of f and h respectively

6. Experimental Results

Performance of DHMMF algorithm is tested for low, medium and high noise density. Fifteen different 512 X 512 size images are taken for the experiment. Simulations are performed using MATLAB 2018b for noise density ranging from 10% - 90% shown in Figure 10. SSIM index map is shown in Figure 11. DHMMF algorithm compared with various algorithms. Qualitative results in terms of PSNR values and quantitative results in terms of SSIM are shown in Table 1 and Table 2. Figure 12 and Figure 13 demonstrate that DHMMF algorithm gives better performance with high PSNR and SSIM by retaining image quality.

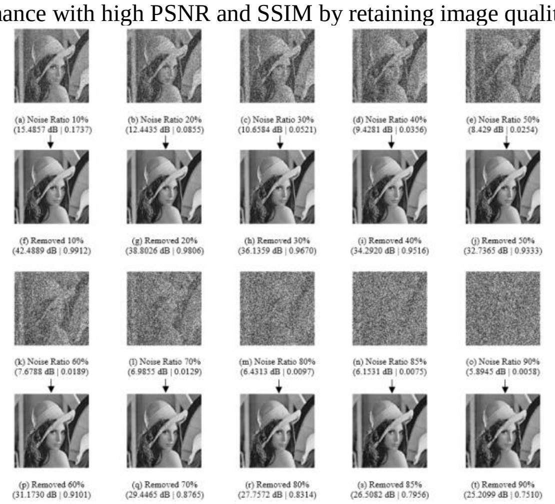

Fig. 10: The Simulation Result of Noisy Images (‘a’, ‘b’, ‘c’, ‘d’, ‘e’, ‘k’, ‘l’, ‘n’, ‘o’) and Respective Filtered Images (‘f’, ‘g’, ‘h’, ‘i’, ‘j’, ‘p’, ‘q’, ‘r’, ‘s’, ‘t’).

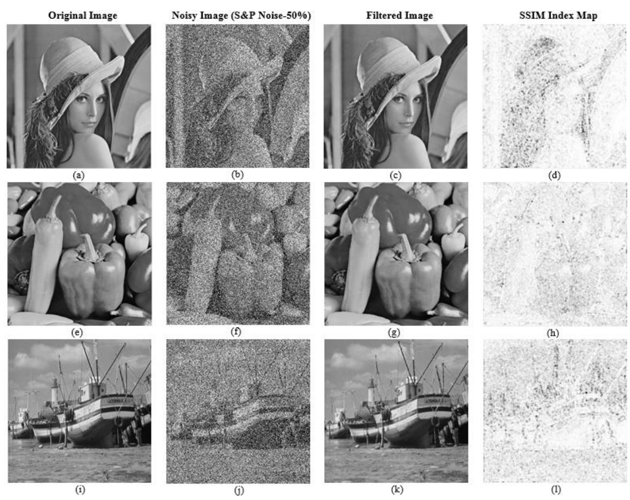

Fig. 11: Original Images - (‘a’, ‘e’, ‘i’), Noisy Images - (‘b’, ‘f’, ‘j’), Filtered Images - (‘c’, ‘g’, ‘k’), SSIM Index Map - (‘d’, ‘h’, ‘l’).

Table 1: Comparison of PSNR And SSIM Values of Various Methods for Fifteen Test Images with 50% Noise Density

Test Images	Proposed DHMMF Algorithm		BPDF [5]		MDBUTMF [6]		EMF1[7]		EMF2[7]		NAFSM [8]	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Leena	32.7365	0.9333	29.261	0.8835	23.6417	0.5717	26.7887	0.8436	29.745	0.8975	31.5015	0.9042
Pepper	31.2554	0.9079	27.7760	0.8563	30.0711	0.8654	26.5080	0.8280	29.4104	0.8827	30.6568	0.9031
Boat	29.4990	0.8759	26.6415	0.8143	28.7695	0.8416	25.2998	0.7937	27.8536	0.8553	27.7739	0.8423
House	32.1294	0.9135	28.2535	0.8736	30.3745	0.8596	25.1824	0.8068	28.7354	0.8904	30.8505	0.9110
Cameraman	30.3267	0.9419	28.4402	0.9143	28.8821	0.8896	25.4995	0.8524	27.8638	0.9200	28.9774	0.9113
Barbara	26.6011	0.8528	24.3695	0.8006	24.9448	0.8121	23.2070	0.7743	24.7918	0.8232	25.8259	0.8409
Baboon	23.7598	0.7797	22.3741	0.7134	23.4030	0.7527	22.0483	0.7115	23.4306	0.7517	23.1697	0.7307
Airplane	30.2453	0.9405	26.6940	0.8861	29.3028	0.8917	25.5516	0.8601	28.6201	0.9217	28.4296	0.9121
Goldhill	31.1138	0.8820	27.7181	0.8137	29.7459	0.8426	25.5310	0.7905	29.3670	0.8578	29.7861	0.8467
Monarch	31.1634	0.9567	26.7293	0.9084	29.8345	0.9176	25.7981	0.8651	28.3486	0.9322	28.7462	0.9377
Coin	30.9867	0.9449	26.9453	0.8997	29.1156	0.8853	25.9575	0.8436	27.7125	0.9146	30.2165	0.9263
Forest	24.1140	0.884	22.7481	0.842	23.3383	0.81	21.6434	0.81	22.5062	0.85	22.8106	0.84

		7		5		27		80		35		77
Pears	36.6878	0.9311	32.8577	0.8856	34.3468	0.8969	27.6739	0.8386	34.1967	0.9063	35.6153	0.9133
On-ion	33.3120	0.9501	27.7587	0.8689	31.9138	0.9124	26.0109	0.8357	29.0585	0.9078	31.7202	0.9242
Tape	32.6583	0.9198	29.2256	0.8786	29.9993	0.8557	27.2030	0.8549	31.7854	0.9170	31.9712	0.9056

Table 2: Comparison of PSNR and SSIM Values of Various Methods for 'Leena' Image with Noise Density 10-90%

Noise density (%)	Pro-posed DHMMF Algo-rithm		BPDF [5]		MDBUTMF [6]		EMF1[7]		EMF2[7]		NAFSM [8]	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
10	42.4889	0.9912	39.4907	0.9838	39.0720	0.9808	39.8843	0.9867	40.3588	0.9875	38.4412	0.9830
20	38.8026	0.9806	36.1618	0.9678	35.8430	0.9601	36.5511	0.9723	37.0726	0.9747	35.866	0.9676
30	36.1359	0.9670	33.7977	0.9476	32.7498	0.9192	32.9754	0.9482	34.3961	0.9555	34.0624	0.9519
40	34.2920	0.9516	31.3387	0.9200	28.5737	0.8020	29.8997	0.9100	32.0475	0.9307	32.6338	0.9339
50	32.7365	0.9333	29.261	0.8835	23.6417	0.5717	26.7887	0.8436	29.745	0.8975	31.5015	0.9042
60	31.1730	0.9101	26.4855	0.8266	19.6138	0.3363	23.1157	0.7261	27.0402	0.8443	29.3622	0.8625
70	29.4465	0.8765	23.5112	0.7399	16.4077	0.1799	19.2375	0.5470	24.788	0.7729	27.0579	0.8305
80	27.7572	0.8314	18.3111	0.5602	14.0997	0.0988	15.2989	0.3096	21.4987	0.6469	24.6401	0.7991
90	25.2099	0.7510	11.0140	0.2880	12.7481	0.0593	11.0059	0.1074	16.8297	0.4532	21.6192	0.6882

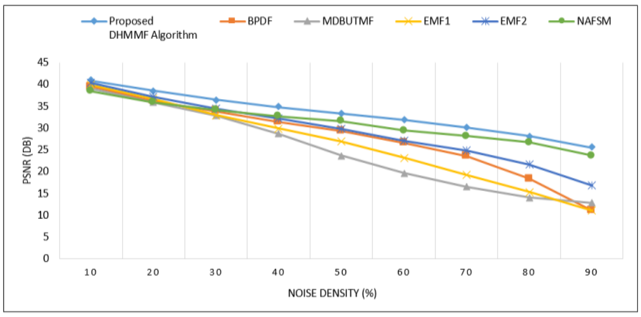

Fig. 12: Comparison of PSNR Values for Various Algorithms.

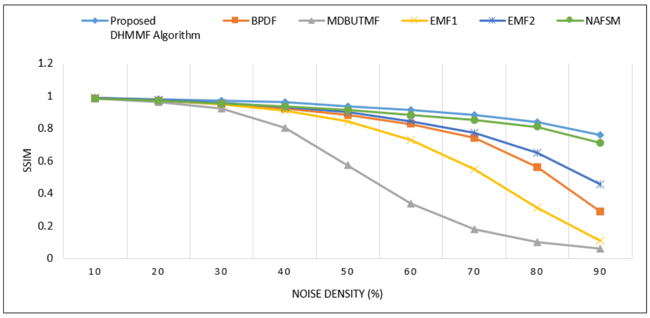

Fig. 13: Comparison of SSIM Values for Various Algorithms.

7. Conclusion

This paper conclude basic concepts and development of a new Dynamic Hybrid Mean/Median Filter (DHMMF) algorithm to eliminate salt & pepper noise. Filter is designed with two stages, noise detection succeeded by filtering strategy. DHMMF algorithm process only the noisy pixels. Filter decides window size dynamically during run time and also adaptively adjusts the window size based on non-noisy within a local window. Window size starts with 3 X 3 and expands up to 9 X 9. Window size is limited to 9 X 9 which results in a reduction of blurring and computational complexity. Threshold value chosen is the size of a current local window. Based on various algorithmic conditions DHMMF algorithm is tested for low, medium and high noise densities. Simulation results shows that image quality is retained by preserving details of an image. Quantitative and qualitative analysis shows higher PSNR and SSIM values, also results in better visual quality

References

[1] Yung-Yue Chen, Ching-Ta Lu, Pei-Yu Chang, "Enhancement of Salt-and-Pepper Noise Corrupted Images Using Fuzzy Filter Design", *Frontier Computing*, Vol. 422, (2017), pp. 691–701, available online: https://doi.org/10.1007/978-981-10-3187-8_65,

[2] Erkan, Uğur & Gökrem, Levent & Enginoglu, Serdar, "Different applied median filter in salt and pepper noise", *Computers & Electrical Engineering*, Vol. 70, (2018), pp. 789-798, , available online: https://doi.org/10.1016/j.compeleceng.2018.01.019

[3] Mousavi, Seyed Mojtaba, Syed Ab Rahman, "A robust medical image watermarking against salt and pepper noise for brain MRI images", *Multimedia Tools and Applications*, Vol.76, (2016), pp.10313-10342, available online: https://doi.org/10.1007/s11042-016-3622-9.

[4] Mújica-Vargas, Dante & Rubio, Jose de Jesus & Kinani, Jean Marie, "An efficient nonlinear approach for removing fixed-value impulse noise from grayscale images", *Journal of Real-Time Image Processing*, Vol. 14, (2017), pp. 617–633, , available online: https://doi.org/10.1007/s11554-017-0746-8.

[5] Uğur Erkan, Levent Gökrem, "A new method based on pixel density in salt and pepper noise removal", *Turkish Journal of Electrical Engineering & Computer Sciences*, Vol. 26, (2017), pp. 162-171, , available online: https://doi.org/10.3906/elk-1705-256.

[6] Esakkirajan, S & Thangaraj, Veerakumar & N. Subramanyam, Adabala & PremChand, C.H., "Removal of High Density Salt and Pepper Noise Through Modified Decision Based Unsymmetric Trimmed Median Filter", *IEEE Signal Processing Letters*, Vol.

18,(2011), pp.287-290, available online: https://doi.org/10.1109/LSP.2011.2122333.

[7] Uğur Erkan, Adem Kilicman, "Two new methods for removing salt-and-pepper noise from digital images", *Science Asia*, Vol. 42, (2016), pp. 28-32, , available online: https://doi.org/10.2306/scienceasia1513-1874.2016.42.028.

[8] Kenny Toh, Nor Ashidi Mat Isa, "Noise Adaptive Fuzzy Switching Median Filter for Salt-and-Pepper Noise Reduction", *IEEE Signal Processing Letters*, Vol. 17, (2010), pp. 281-284, , available online: https://doi.org/10.1109/LSP.2009.2038769.

[9] Igor Djurović, "BM3D filter in salt-and-pepper noise removal", *Image Video Processing*, (2016), Vol.13, available online: https://doi.org/10.1186/s13640-016-0113-x.

[10] Ke TuHongbo Li, Fuchun Sun, "A statistical learning based image denoising approach", *Frontiers of Computer Science*, Vol.9, No.5, (2015), pp.713–719, available online: https://doi.org/10.1007/s11704-015-4224-9.

[11] Abdol Hamid Pilevar, Soudeh Saien, Mina Khandel, "A new filter to remove salt and pepper noise in color images", *Signal, Image and Video Processing*, (2015), Vol.9, No.4, pp.779–786, available online: https://doi.org/10.1007/s11760-013-0514-6.

[12] B. K., Shreyamsha Kumar, "Image denoising based on non-local means filter and its method noise thresholding", *Signal, Image and Video Processing*, Vol.7, (2013), pp.1211-1227, available online: https://doi.org/10.1007/s11760-012-0389-y.