International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

An Efficient Low Complexity Salt & Pepper Noise Detection Method

Sajid Khan[1], Muhammad Asif Khan[2], Aeshah Alsughayyir[4], Abdullah Alshanqiti[5], Bandeh Ali Talpur[6], Abdul Rehman Gilal[3]

Submitted: 10/09/2022 **Accepted:** 20/12/2022

Abstract: A new impulse noise detection algorithm is proposed in this paper. The most popular method for detection of impulse noise is to classify every pixel with intensity 0 or 255 as noise. Other techniques involve the detection of noise candidates in the first stage followed by false-positive reduction process in the second stage. Both types of techniques have some problems such as the first approach fails to distinguish between a noisy pixel and pure white or black background region. The second type detection algorithm often results in the detection of an unwanted amount of false positives. They also demand more CPU elapsed time. The proposed method is a two-stage impulse noise detector that first detects all the true positives along with false positives that are the result of the appearance of pure white or pure black uniform regions in the image. It then applies morphological operators such as erosion and pixel connectivity to avoid the detection of uniform regions as noisy pixels. Simulation results show that the proposed impulse noise detector method outperformed existing noise detection methods. The proposed method can be applied as an initial noise detection step for the removal of salt & pepper noise using any spatial filter.

Keywords: *Impulse noise, Image restoration, salt & pepper noise, Morphological filtering*.

1. Introduction

Image processing is widely used almost everywhere. It has enormous applications in most of the scientific fields such as aerospace, astronomy, particle physics, biology, photogrammetry, geology, the science of material, and medical sciences [1]. Usually, images are corrupted by noise due to the usage of the corrupted sensor during the acquisition process or due to their transmission through a corrupted medium [2]. Noise can be of additive or multiplicative nature. Noise can be of different types according to the pdf that they are represented with. Some of the most occurring noise types are Gaussian noise, impulse noise, Laplacian noise, speckle noise and Rayleigh noise, impulsive noise, Gaussian noise, Rayleigh noise, and Laplacian noise, to name a few [3], [4], [5].

Salt & pepper noise is a type of impulse noise where corrupted pixels are replaced by the extreme intensities. Salt & pepper noise usually corrupt image due to the corrupted sensors. In salt & pepper noise, corrupted pixels are replaced by extreme intensity values. Mathematical model of salt & pepper noise is given in (1) as

$$Y = \begin{cases} 0, & with\,probability\; p_1 \\ L-1, & with\,probability\; p_2 \\ x, & with\,probability\; 1-p_1-p_2 \end{cases} \quad (1)$$

where x denote the uncorrupted intensity value that are not affected by the interference of noise.

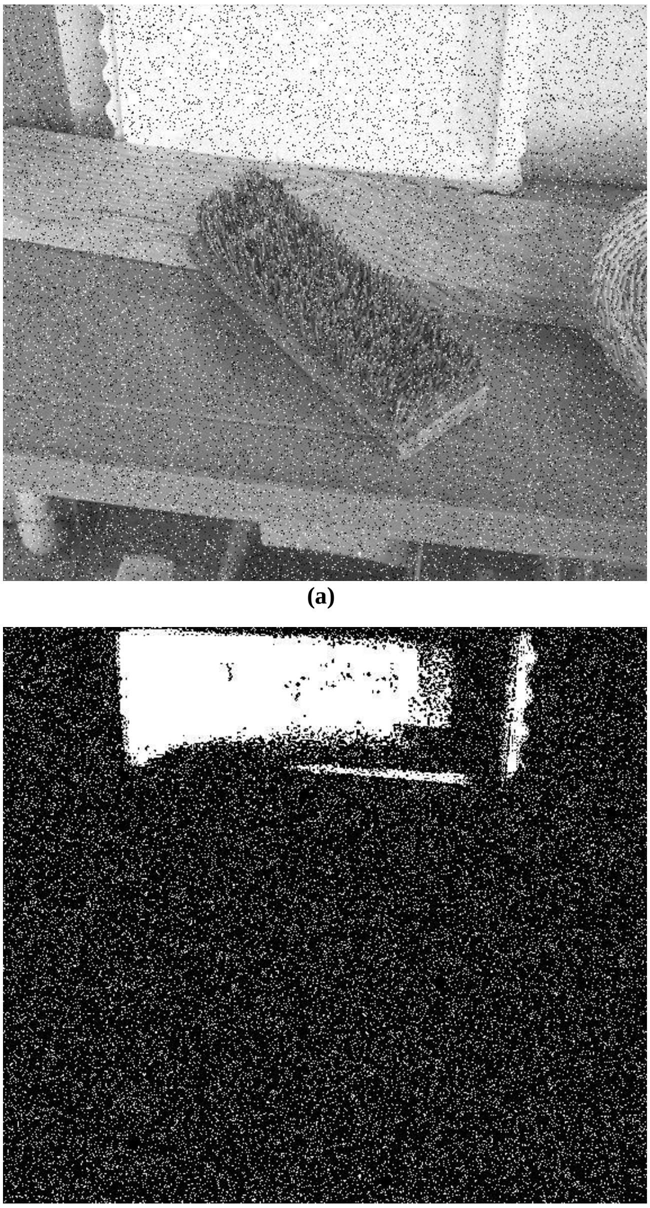
Salt & pepper addition is the problem that affect the output of MRI[3], CCTV [6], and TV[7]. Because salt & pepper noise seriously deteriorates picture quality and disrupts image processing, an efficient noise-removal algorithm is a core image processing technology. For removal of Salt & pepper noise, filters are applied to the image.

Applying filters help us to estimate the value of the pixel that is currently under consideration. Applying filters to all pixels in an image result in more blur and doesn't provide the desired output as uncorrupted pixels are also replaced by their estimated values. Instead, a common approach is to apply a two-stage image noise removal method that comprises of noise detection followed [8]-[17] by the application of the simple or adaptive filter.

The two-stage salt & pepper noise removal process produces less blur as selected pixels are filtered instead of applying filters to the whole image. The performance of such methods depends on the performance of both detection and filtering stage, however, in most of the case, detection matters the most. A detector that produces false negatives is of no use as some of the noisy pixels are left unprocessed. A detector that produces a higher percentage of false-positive also leads to the production of blur along edges and other details as many noise-free pixels are also processed for filtering.

[1] *Department of Computer Science, School of Engineering, Akfa University, Tashkent, Uzbekistan*
[2] *Department of Computer Science, Sukkur IBA University, Sukkur, Pakistan*
[3] *Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Seri Iskandar, Malaysia*
[4] *College of Computer Science and Engineering, Taibah University, Kingdom of Saudi Arabia*
[5] *Faculty of Computer and Information Systems, Islamic University, Kingdom of Saudi Arabia*
[6] *School of Computer Science and Statistics, Trinity College Dublin, Ireland*

Corresponding author: SAJID KHAN (e-mail: sajidkhan@iba-suk.edu.pk).



(a)

(b)

Figure 1. Results of considering pixels with intensities 0 or 255 as noisy: (a) input noisy images with 10% noise density; (b) output binary images.

To detect 100% true positives, few researchers [12]-[14] have proposed the simplest detection methods. They considered all those pixels noisy that have intensities of 0 or 255. This approach produces no or very few false positives, however, they fail on images that have pure white or pure black objects or uniform regions. Figure 1 shows examples of the application of such detection. It can be seen from the images, that although, their accuracy of true positives is 100%, however, it also detected a group of background non-noisy pixels as false positives. For any adaptive filtering algorithms [11], [12], [15], [17] that extends window size based on detected noise present in the scene will be unable to produce good results for such images as for most of the pixels detected as noisy pixels, any window size will contain all the false positives along with some true positives and hence adaptive filtering will either not be able to make a decision, or it may replace a false positive with the intensity value of a true positive.

Instead of considering maximum and minimum intensity values as noisy pixels in an image, researchers [15]-[17] have proposed other techniques that detect less false positives in extreme intensity regions. These methods also provide reasonable detection for random valued impulse noise as well in the presence of some unwanted false negatives.

This paper presents a simple and efficient two-stage salt & pepper noise detection method that detects 100% true positives in the presence of a fractional percentage of false positives. The proposed method first consider all the pixels with intensities 0 or 255 as noisy pixels, apply pixel connectivity based second stage to ensure that the detected pixels don’t belong to a uniform region. The algorithm produces encouraging results for all types of images and takes a fraction of time to execute. The proposed algorithm shows consistent performance for both low and high densities of noise. The proposed method can be combined with any type of adaptive filter to provide image restoration from salt & pepper noise in the presence of minimum blur.

The rest of the paper is organized as follows: Section 2 describes some of the conventional methods. Section 3 describes the proposed salt & pepper noise detection algorithm. Section 4 shows the results and discussion of the proposed filter followed by the conclusion in Section 5.

2. Conventional Methods

This section discusses some of the conventional state-of-the-art salt & pepper noise detection methods proposed by the researchers. Boundary discriminative noise detection (BDND) [16], Kumar [18], Ma [19], Ghanekar detector [15], and Khan detector [17] are discussed.

BDND [16] is a two stage noise detection method that has the following steps

i. Around every pixel of interest, impose a 21x21 window while keeping the pixel of interest as center.

ii. Find the median value “med” after sorting the window and storing it in vector v_0.

iii. Divide v_0 into two groups. First group will have sorted intensities from 0 to med whereas second group will have sorted intensities from med to 255.

iv. For both groups, find the difference between consecutive sorted intensities. Calculate b_1 and b_2 as sorted intensities that corresponds to maximum difference index of both groups.

v. If the pixel of interest is less than or equal to b_1 or is greater than or equal to b_2, it is marked as noisy candidate in the first stage of the algorithm. Otherwise, it is marked as non-noisy pixel.

vi. Impose a 3x3 window and repeat the process ii-v to see if the noisy candidate is still marked as noisy or not. If a noisy candidate is not marked as noisy in the second stage, mark it noise-free.

Kumar et al. [18] suggested a basic salt-and-pepper noise detector that chooses a pixel with an intensity of 0 or 255 as a candidate. The number of pixels in a 3x3 window with intensities of 0 or 255 is then counted. If the count passes a certain threshold, the pixel is considered noise-free because it could be in a flat region with intensities of 0 or 255. If the count is smaller than the threshold, the pixel is chosen as a noisy pixel.

Due to the use of a smaller window, the Kumar detector will produce a large number of false negatives for high density noise images. A similar solution was proposed by Ma and Nie [19], which addresses the window size issue. It determines if a pixel is noisy based on the number of dissimilarities between it and its neighbours. It counts the number of nearby pixels whose absolute differences G_k with the candidate pixels are larger than the threshold for a window of size nxn. If the count exceeds a certain threshold, the pixel is designated as a noisy pixel.

For noise levels less than 50%, Ma detector proposes using 3x3 or 5x5 windows, whereas for densities between 60 and 80 percent, 7x7 windows are employed. For a noise density of 90%, 9x9 windows are employed. The issue with the Ma detector is that it does not provide any way for estimating the image's noise density.

Ghanekar detector [15] is a two stage noise detection filter. The first stage comprises of the following steps

i. Create a binary image B of size of input image, mark all pixels in B as zero.

ii. Move 3x3 window across every pixel of interest $I(i,j)$.

iii. Sort all the pixels to form an array R such that $R(i,j) = [r_1(i,j), r_2(i,j), \ldots\ldots, r_9(i,j)]$.

iv. Mark the pixel of interest as noisy pixel if the following condition is satisfied if$\big(I(i,j) = r_1(i,j)\big)$ or $(I(i,j) = r_9(i,j))$, mark pixel of interest as noisy candidate by assigning value of 1 to $B(i,j)$.

The second stage of Ghanekar detector comprises of the following steps

i. For all noisy candidates $I(m,n)$, move 11x11 window across them, sort 11x11 window to get array R of size 1x121.

ii. Calculate difference between consecutive elements in R, and store them in array D such that $D(m,n) = [d_1(m,n), d_2(m,n), \ldots\ldots, d_{120}(m,n)]$

iii. Find four largest distances d_{max1},d_{max2}, d_{max3} and d_{max4} in D such that $d_{max1} = d_i(m,n)$; $d_{max2} = d_j(m,n)$; $d_{max3} = d_k(m,n)$; $d_{max4} = d_l(m,n)$; where $d_i(m,n) > d_j(m,n) > d_k(m,n) > d_l(m,n)$.

iv. Find the values of m and n such that

$$o = \min[i,j,k,l]\,; \qquad (2)$$

$$n = \max[i,j,k,l]\,; \qquad (3)$$

Define boundary values w_{min} and w_{max} for comparison such that

$$w_{min} = r_{o+1}(m,n) \qquad (4)$$

$$w_{max} = r_n(m,n) \qquad (5)$$

Noisy candidate is marked as noisy or non-noisy pixel on the basis of criteria

$$if(I(m,n) < w_{min}\ or\ I(m,n) > w_{max})$$
$$B(m,n) = 1$$
$$else$$
$$B(m,n) = 0$$

The problem with the Ghanekar method is that it produces false negatives during the second stage as it marks some of the pepper noisy pixels as non-noisy. Khan detector [17] modified the second stage of the Ghanekar detector to avoid the occurrence of false negatives in the detection process. The slight modification that Khan detector proposed in step v of the second stage of Ghanekar detector is

$$if(d_m == 0)$$
$$w_{min} = r_{o+1}(m,n) + 1$$

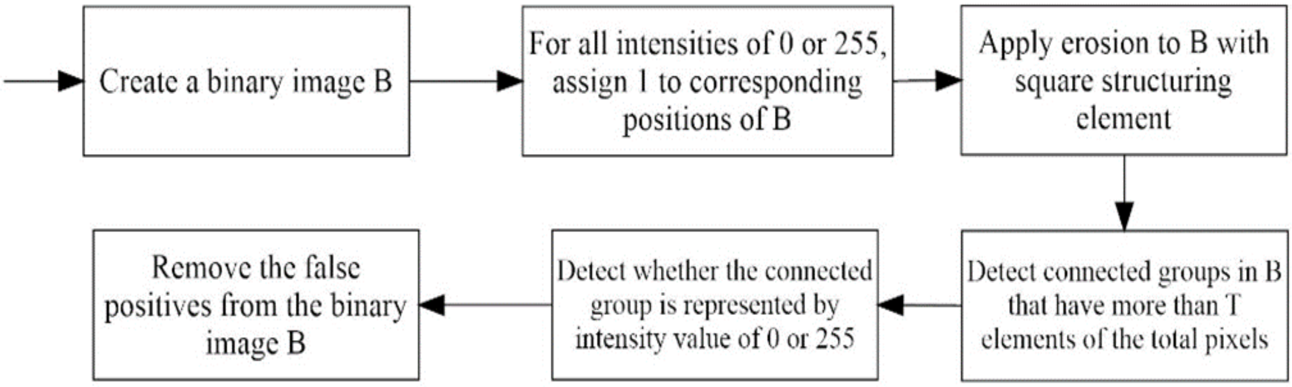

Create a binary image B
For all intensities of 0 or 255, assign 1 to corresponding positions of B
Apply erosion to B with square structuring element
Detect connected groups in B that have more than T elements of the total pixels
Detect whether the connected group is represented by intensity value of 0 or 255
Remove the false positives from the binary image B

Figure 2. Flowchart of the proposed salt & pepper noise detection algorithm.

3. Methodology

The objective of this paper is to propose a simple salt & pepper noise detection method that detects all the noisy pixels in the presence of negligible false positives. Marking all the pixels with intensity values of 0 or 255 will lead to the detection of 100% true positives, however, it will result in the detection of few percent of false positives as well. The percentage of false positives will be much higher for the images that have pure black or white regions as can be seen from Figure 1. This paper aims to reduce those false positives to a great extent.

Flowchart of the proposed impulse noise detection algorithm is shown in Figure 2. The proposed algorithm is a two-stage impulse noise detector. The steps for the first stage of the proposed algorithm are given below

i. Create a binary image B of size of input noisy image. Assign 0 to all the indices of B.

ii. If our pixel of interest at $I(i,j)$ have intensity value of either 0 or 255, detect it as noisy candidate by assigning value of 1 to $B(i,j)$. Otherwise, don't change the value of $B(i,j)$.

The output of the first stage will contain 100% true positives, however, false positives will be there as well. From Figure 1, it can be seen that white uniform regions are also detected as salt noise in this case. From Figure, it is clear that the false detection of the uniform region results in a group of connected pixels in the binary image. Therefore, the connectivity information can be used to reduce the detection of false positives in such cases. Therefore, the second stage of impulse noise detection involves few morphological image processing algorithms. The steps for the second stage are given below

i. Apply erosion with square structuring element of size 2 to image B. This step is applied in order to isolate the possible pepper or salt noise pixels attached to the boundaries of pure black or white uniform regions so that they cannot be selected as noise-free pixels in step iv.

ii. For every detected noise candidate at positions (m, n), detect all the connected pixels groups that have number of connected pixels (with 8-connected components) greater than or equal to T. Where T is given below

$$T = M \times N \times \frac{\delta}{100} \qquad (6)$$

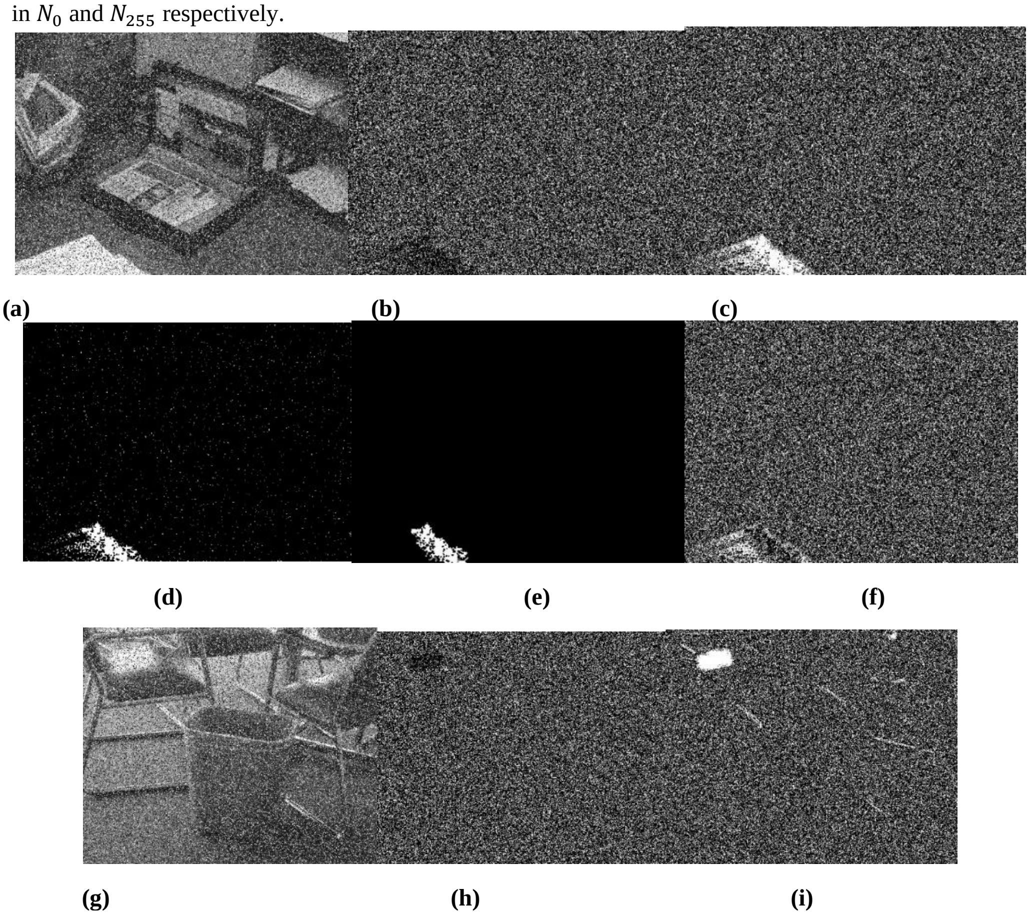
Where M and N are rows and column of the input image.

iii. For all the connected pixels group, detected in step ii, calculate the number of pixels that have intensity values of 0 and 255. Store these numbers in N_0 and N_{255} respectively.

iv. If $N_0 > N_{255}$ in a group, it means that the uniform region have intensities of 0 in the region of interest. Obtain the indices of all 0 intensity indices In_0 in that group, assign 0 values to all In_0 indices in binary image B. In case if $N_{255} > N_0$ for a group, repeat the process of step iv for intensities of 255.

Step i of the second stage is applied to ensure the removal of noisy pixels attached to the boundary of a connected group. For example, if a uniform region has intensities of 255, there is a strong possibility that few noisy salt pixels may be connected to the boundary of that region. Therefore, in order to retain the detection of those noisy pixels, connected boundary elements from all groups are removed using erosion. Step iv is used to detect whether the uniform region that is mistakenly detected as a group of a connected noisy pixel in the first stage contains intensity values of 0 or 255. If the uniform region has intensity values of 0, the number of black pixels will dominate the number of white pixels and vice versa.

Figure 3 shows the results of using multiple steps of the proposed technique to detect noisy pixels. With a salt and pepper noise density of 30%, Figure 3(a) and (g) display noisy briefcase and trashcan images. There are a considerable number of white pixels with intensities of 255 along the bottom left and top left corners of the briefcase and trashcan images, respectively. The ground truth in Figures 3(b) and (h) was determined using the equation below



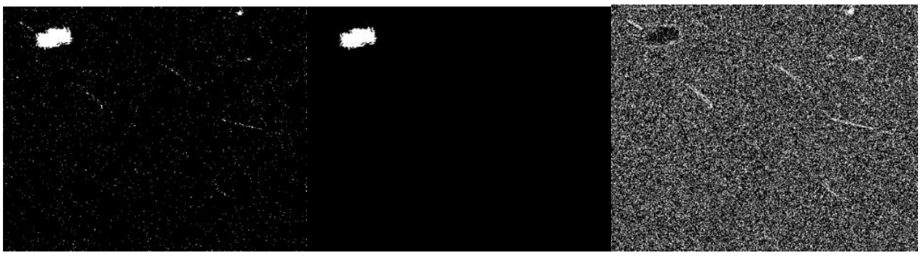


Figure 3. Results of application of the proposed algorithm to detect noisy pixel; (a, g) Image with 30% noise density, (b, h) ground tru th; (c, i) Results of first stage; (d, j) eroded image; (e, k) Detection of uniform non-noisy region; (f, l) Final binary image.

$$GT = abs\left(I_{noisy} - I_{orig}\right) > 0 \qquad (6)$$

where I_{noisy} is the noisy image and I_{orig} is the original image with no noise. Figure 3(c) and (i) illustrate the results of the first stage candidate selection, demonstrating that the image contains all noisy pixels as well as false positives, as the paper region in the briefcase and shining part of the chair in the trashcan images have intensities of 255. Figures 3(d) and (j) were created by using erosion to isolate any noisy pixels related to the paper and shining chair regions. This step ensures that boundary noisy pixels are removed when reducing false positives based on connectivity. Consequently, some false positives on the edge of the paper zone will be eliminated as well. The detected false positives white pixels with connection larger than and are segregated using the criteria given in stage two steps ii-iv are depicted in Figure 3(e) and (k). Figures 3(f) and (i) show the final binary pictures created.

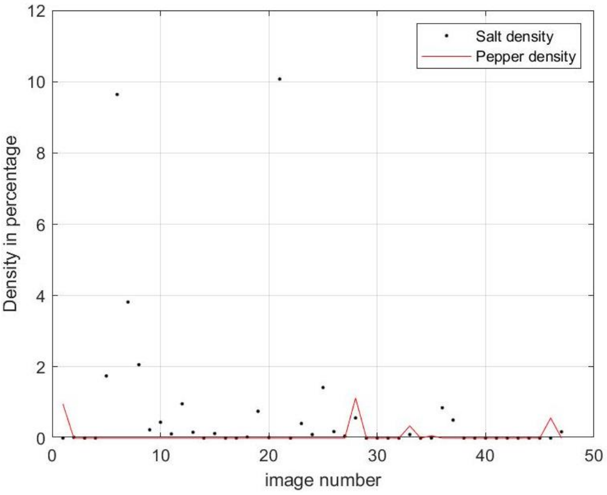


Figure 4. Densities of pixels with non-noisy 0 and 255 intensities in each of 47 images.

4. Results

For performance evaluation, a total of 47 images were used. Non-noisy pixels with intensities of 0 or 255 in the range of 0-10 percent can be found in the selected images. Figure 4 shows the distribution of 0 and 255 intensities in non-noisy images. As demonstrated in Figure 4, the selected images provide a good variety of examples to study. On the basis of extensive simulations, the value of δ=0.5 was chosen.

$$Density_{Ma} = \frac{N_0 + N_{255}}{M \times N} \qquad (7)$$

where M, N , N_0 and N_{255} are number of rows, columns, 0 intensity pixels and 255 intensity pixels respectively.

Figure 5-6 shows a comparison of true positive and false positive comparison for the detection of noise from 10%, 50%, and 80%. True positive percentage were calculated using (8) as

$$TP = \frac{N'_{TP}}{N_{TP}} \times 100 \qquad (8)$$

where N'_{TP} depicts number of true positives detected by algorithm for a specific image/case and N_{TP} are the true positives in the ground truth. False positives, on the other hand, are calculated using (9) as

$$FP = \frac{N'_{FP}}{N'_{total}} \times 100 \qquad (9)$$

where N'_{FP} are total number of false positives detected by the algorithm for a specific image/case and N'_{total} are the total number of noisy pixels detected by the algorithm.

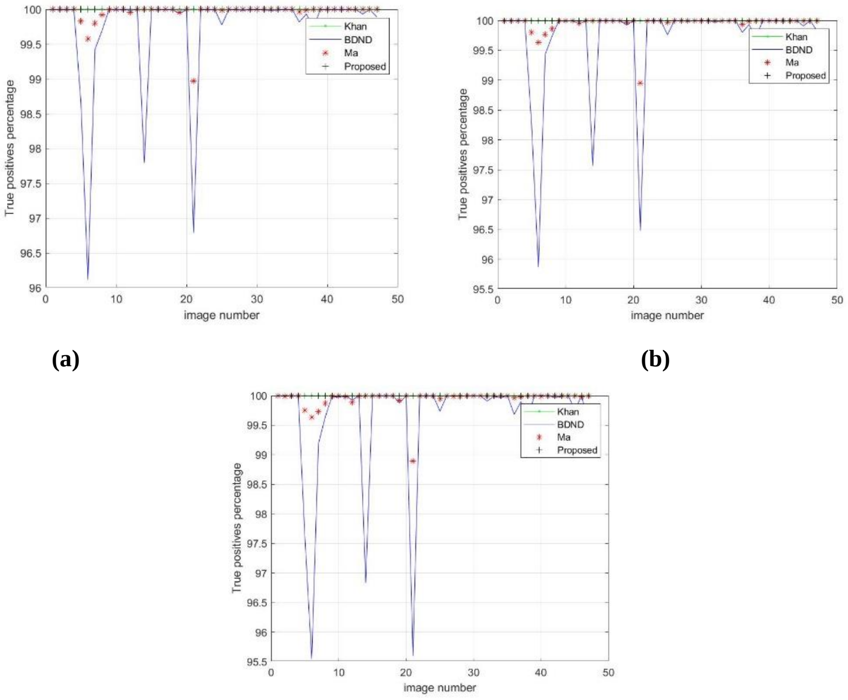
It can be seen from Figure 5 that both the proposed method and Khan detector detected 100% true positives for both low and high densities of noise. Ma detector also detected nearly 100% true positives for all 47 images with true positives in the range of 98.8-100%. BDND, however, shown true positive detection accuracy of 95.6-100%.

Figure 6 shows a false positive comparison for all four algorithms. It can be seen from Figure 6 that overall, BDND detected more false positives when compared to other algorithms. Khan detector performed better comparatively, however, for some images, false detection is more when compared to Ma. The proposed method, as can be seen, has overall outperformed all the other algorithms in the aspect of detection of less overall false positives. Table I shows the average percentage of true positives and false positives of all algorithms for 10%, 30%, 50%, 70%, and 80% noise densities. It can be seen from the table that the proposed algorithm and Khan detector detected 100% true positives. The proposed method detected very few false positives when compared to other methods.

If results discussed in Figure 6 and Table I are analyzed collectively, it can be seen from Figure 6 that for some cases, the percentages of false positives for some images are higher compared to the average false positive caught. These are the images that have a good proportion of pure white or black uniform regions with non-noisy extreme intensities. It is because of two reasons. One reason is ignoring the uniform white/black region pixels that are on the boundary due to the application of erosion. Another reason is that the cluster formed due to the non-noisy extreme values are either already smaller than the σ or they become smaller after the application of erosion. Erosion separates few of the regions from the bigger cluster by removing small bridges between them. However, for most of the images, false positive detection percentage is near to zero that results in very smaller percentage of average false positive detection. False positive detection becomes less as noise density increase as non-noisy extreme values forms bigger clusters that are easily isolated with the application of threshold σ. However, making the threshold σ extremely smaller by keeping the value of δ≪0.5 will result in detection of false negatives for high density of noise as noisy pixels seems to form clusters that may satisfy the threshold σ [22]-[25].

Figure 6 shows a false positive comparison for all four algorithms. It can be seen from Figure 6 that overall, BDND detected more false positives when compared to other algorithms. Khan detector performed better comparatively, however, for some images, false detection is more when compared to Ma. The proposed method, as can be seen, has overall outperformed all the other algorithms in the aspect of detection of less overall false positives. Table I shows the average percentage of true positives and false positives of all algorithms for 10%, 30%, 50%, 70%, and 80% noise densities. It can be seen from the table that the proposed algorithm and Khan detector detected 100% true positives. The proposed method detected very few false positives when compared to other methods.

If results discussed in Figure 6 and Table I are analyzed collectively, it can be seen from Figure 6 that for some cases, the percentages of false positives for some images are higher compared to the average false positive caught. These are the images that have a good proportion of pure white or black uniform regions with non-noisy extreme intensities. It is because of two reasons. One reason is ignoring the uniform white/black region pixels that are on the boundary due to the application of erosion. Another reason is that the cluster formed due to the non-noisy extreme values are either already smaller than the σ or they become smaller after the application of erosion. Erosion separates few of the regions from the bigger cluster by removing small bridges between them. However, for most of the images, false positive detection percentage is near to zero that results in very smaller percentage of average false positive detection. False positive detection becomes less as noise density increase as non-noisy extreme values forms bigger clusters that are easily isolated with the application of threshold σ. However, making the threshold σ extremely smaller by keeping the value of δ≪0.5 will result in detection of false negatives for high density of noise as noisy pixels seems to form clusters that may satisfy the threshold σ.

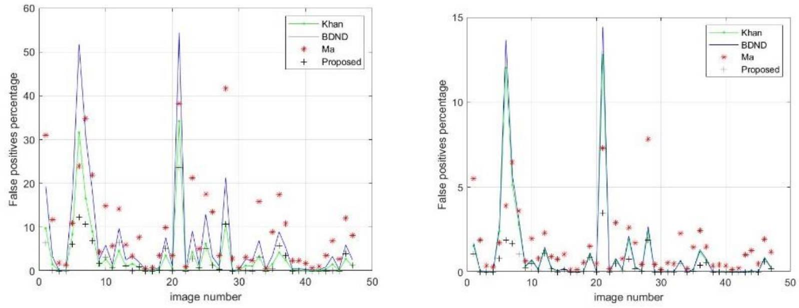


(a)

(b)

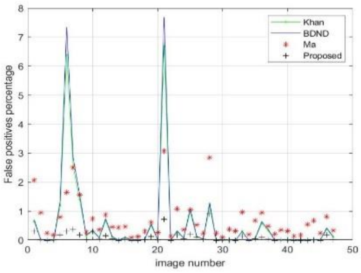
(c)

Figure 5. True positive comparison of Khan detector, BDND, Ma detector and proposed method for detection of ; (a) 10%, (b) 50%, (c) 80%noise densities.



(a)

(b)



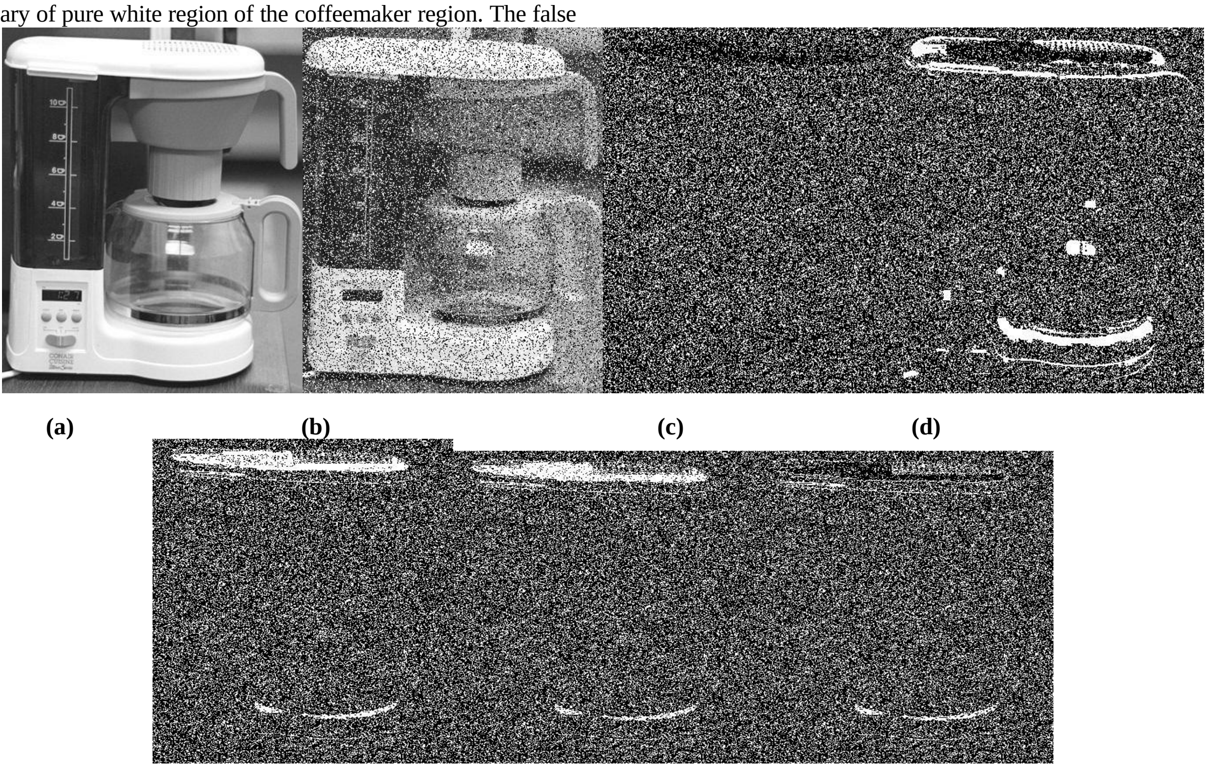
(c)

Figure 6. False positive comparison of Khan detector, BDND, Ma detector and proposed method for detection of; (a) 10%, (b) 50%, (c) 80% noise densities.

TABLE 1.
AVERAGE TRUE POSITIVE AND FALSE POSITIVE COMPARISON OF ALL ALGORITHMS FOR DIFFERENCE NOISE DENSITIES.

Noise density/method	Khan		Ma		BDND		Proposed	
	TP	FP	TP	FP	TP	FP	TP	FP
10%	100	3.72	99.74	6.99	99.962	9.76	100	2.47
30%	100	2.48	99.727	4.66	99.96	5.93	100	1.43
50%	100	1.07	99.70	1.20	99.957	1.62	100	0.41
70%	100	0.61	99.69	0.76	99.957	0.98	100	0.23
80%	100	0.54	99.68	0.61	99.954	0.69	100	0.101

Figure 7 shows comparison of application of all four algorithms to a crop patch of coffeemaker image. It can be seen from Figure 7(d) that BDND detected some solid dense regions of false negatives when it comes to distinguish the non-noisy white regions from the noisy one. Ma, on the other hand, detected false negatives that are less in numbers than both BDND and Khan detector. For some images, as can be seen from Figure 7, Khan detects more false positives compared to the Ma. As far as proposed method is concerned, the only false negatives that are detected are along the boundary of pure white region of the coffeemaker region. The false positives along the boundary regions are due to the application of erosion and cannot be removed as ignoring the application of erosion may result in false negatives along the boundary.



(a) **(b)** **(c)** **(d)**

(e) **(f)** **(g)**

Figure 7. Results of application of the different algorithms to detect noisy pixel; (a) Original noise-free image, (b) Image with 30% noise density, (c) ground truth; (d) BDND; (e) Ma; (f) Khan; (g) Proposed.

TABLE 2.
AVERAGE CPU ELAPSED TIME COMPARISON OF ALL ALGORITHMS FOR DIFFERENCE NOISE DENSITIES.

Noise density/method	Khan	Ma	BDND	Proposed
10%	1.778457	1.222167	6.67449	0.011233
30%	1.939439	1.662968	6.872705	0.011315
50%	2.947717	2.160008	6.654375	0.01237
70%	3.474102	2.695277	6.350044	0.014666
80%	3.960213	3.213243	6.969171	0.01953

As far as execution time is concerned, the proposed method outperformed all the other algorithms as it apply simplest two stage detection method among which the first stage maybe implemented using vectorization methods whereas second is all about applying basic image morphology. Also, the proposed method rely mainly on global processing instead of relying heavily on local processing, this is also one of the reason of faster execution of the proposed method.

5. Conclusion

A new impulse noise detection method is proposed in this paper. Conventional methods can be classified into two categories. The first category is one stage simple detectors that classify every pixel that have an intensity value of 0 or 255 as noisy. Second category detectors usually detect noise in two stages, i-e, first they detect noisy candidates, followed by the final selection of noisy pixels. The first approach results in some serious problems when it comes to detecting noise from images having pure white or pure black background. Second category detectors, however, produce an unwanted percentage of false detection. The proposed method overcomes problems mentioned above to ensure the detection of 100% true positives, in the presence of negligible detection of false positives. The proposed method is simple and takes very little CPU elapsed time to process a single image. The proposed method can be combined with any adaptive filter that depends on the input of the noise detector.

In the future, the aim is to propose an efficient adaptive filter that outperforms other linear and non-linear filters by getting the full benefit of the proposed noise detector method.

References

F. Russo, "Edge detection in noisy images using fuzzy reasoning," *IEEE Transactions on Instrumentation and Measurement*, vol. 47, no. 5, pp. 1102-1105, May 1998.

E. Abreu, M. Lightstone, S. K. Mitra, and K. Arakawa, "A new efficient approach for the removal of impulse noise from highly corrupted image," *IEEE Transactions on Image Processing*, vol. 5, no. 6, pp. 1012-1025, June 1996.

H. L. Eng, and K. K. Ma, "Noise adaptive soft-switching median filter," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 242-251, Feb. 2001.

T. Wang, J. Qiu, S. Fu, and W. Ji, "Distributed fuzzy H∞ filtering for nonlinear multirate networked double-layer industrial processes," *IEEE Transactions on Industrial Electronics*, vol. 16, no. 3, pp. 5203-5211, Oct. 2016.

P. Civicioglu, "Using uncorrupted neighborhoods of the pixels for impulsive noise suppression with ANFIS," *IEEE Transactions on Image Processing*, vol. 16, no. 3, pp. 759-773, Mar. 2007.

S. Kovaf, J. Valouch, H. Urbančoková, and M. Adámek, "Electromagnetic interference of CCTV," *IEEE International Conference on Information and Digital Technologies*, vol. 16, no. 3, pp. 172-177, Aug. 2015.

Q. Zhang, R. K. Ward, and J. Du, "Impulse noise correction in TV transmission," *IEEE Transactions on Consumer Electronics*, vol. 41, no. 3, pp. 731-737, Aug. 1995.

T. Lin, and P. T. Yu, "Salt-pepper impulse noise detection and removal using multiple thresholds for image restoration," *Journal of Information science and Engineering*, vol. 22, no. 1, pp. 189-198, Jan. 2006.

V. P. Ananthi, P. Balasubramaniam, and P. Raveendran, "Impulse noise detection technique based on fuzzy set," *IET Signal Processing*, vol. 12, no. 1, pp. 12-21, Feb. 2018.

M. Biswas, "Impulse Noise Detection and Removal Method Based on Modified Weighted Median," *International Journal of Software Innovation*, vol. 8, no. 2, pp. 38-53, 2020.

N. Iqbal, S. Ali, I. Khan, and B. M. Lee, "Adaptive edge preserving weighted mean filter for removing random-valued impulse noise," *Mdpi Symmetry*, vol. 11, no. 3, 2019.

K. Pritamdas, K. M. Singh, and L. L. Singh, "An adaptive switching filter based on approximated variance for detection of impulse noise from color images," *SpringerPlus*, vol. 5, no. 1, 2016.

U. Erkan, L. Gökrem, and S. Enginoglu, "Different applied median filter in salt and pepper noise," *Elsevier Computers and Electrical Engineering*, vol. 70, no. 5, pp. 789-798, 2018.

S. K. Sathua, A. Dash, A. Behera, "Removal of salt and pepper noise from gray-scale and color images: an adaptive approach," *Computer Vision and Pattern Recognition*, vol. 5, no. 1, pp. 117-126, 2017.

U. Ghanekar, "A novel impulse detector for filtering of highly corrupted images," *World Academy of Science, Engineering and Technology*, vol. 14, pp. 353–355, 2008.

H. Ibrahim, T. F. Ng, and S. H. Teoh, "An efficient implementation of switching median filter with boundary discriminative noise detection for image corrupted by impulse noise," *Scientific Research and Essays*, vol. 6, no. 26, pp. 5523–5533, 2011.

S. Khan, and D. H. Lee, "An adaptive dynamically weighted median filter for impulse noise removal," *EURASIP Journal on Advances in Signal Processing*, vol. 2017, no. 1 , 2017.

A. K. Samantaray, P. Kanungo, and B. Mohanty, "Neighbourhood decision based impulse noise filter," *IET Image Processing*, vol. 12, no. 7, pp. 1222-1227, 2018.

H. Ma, and Y. Nie, "A two-stage filter for removing salt-and-pepper noise using noise detector based on characteristic difference parameter and adaptive directional mean filter," *PloS one*, vol. 13, no. 10, pp. e0205736-1- e0205736-24, 2018.

M. A. Almomani, S. Basri, and A. R. Gilal, ‘‘Empirical study of software process improvement in Malaysian small and medium enterprises: The human aspects,’’ J. Softw. Evol. Process, vol. 30, no. 10, p. e1953, Oct. 2018.

Edge Detector Comparison. Accessed: Jan. 27. 2022. [Online]. Available: http://www.eng.usf.edu/cvprg/edge/edge_detection.html.

A. R. Gilal, J. Jaafar, S. Basri, M. Omar, and M. Z. Tunio, “Making Programmer Suitable for Team-Leader: Software Team Composition Based on Personality Types,” in International Symposium on Mathematic,

M. Z. Tunio, H. Luo, C. Wang, F. Zhao, A. R. Gilal, and W. Shao, “Task Assignment Model for crowdsourcing software development: TAM,” J. Inf. Process. Syst., 2018.

J. Jaafar, A. R. Gilal, M. Omar, S. Basri, I. A. Aziz, and M. H. Hasan, A Rough-Fuzzy Inference System for Selecting Team Leader for Software Development Teams, vol. 661. 2018.

S. M. Jameel, A. R Gilal, S. S. Hussain Rizvi, M. Rehman, and M. A. Hashmani, “Practical Implications and Challenges of Multispectral Image Analysis,” in 2020 3rd International Conference on Computing, Mathematics and Engineering Technologies: Idea to Innovation for Building the Knowledge Economy, iCoMET 2020, 2020.