Tabriz Journal of Electrical Engineering (TJEE), vol. 55, no. 1, 2025 Serial no. 111
DOI: 10.22034/tjee.2024.59623.4778

Investigation of the noisy image edge detection based on the GWO algorithm

Aref Eslami Mehdi Abadi, Farahnaz Mohanna*

Communication Engineering Department, University of Sistan and Baluchestan, Zahedan, Iran
Aref.eslami1992@gmail.com, f_mohanna@ece.usb.ac.ir
*Corresponding author

Received: 16/12/2023, Revised: 19/08/2024, Accepted: 28/09/2024.

Abstract
Edge detection is one of the basis of image segmentation, feature extraction, and object recognition processes. So far, many edge detectors have been introduced. However, even the best edge detectors lose their effectiveness in the presence of noise. Therefore, the correct detection of edges in the noisy image is still one of the challenging issues in the image processing. Various algorithms have been presented to solve this challenge, of which the meta-heuristic optimization algorithms are examples that can effectively search the space of possible solutions and easily work in the complex and unconditional problems. In this research, a method is proposed for the edge detection of the noisy images based on the grey wolf optimization whose new objective function is proposed based on the homogeneity, uniformity, and Kirsch edge detector masks. The proposed method has been simulated on the BSDS500 database including 500 images along with their Ground Truth images. In the simulation, the Gaussian, and salt-and-pepper noises have been applied. The evaluation has been done according to the mean square error, peak signal-to-noise ratio, precision, F-score, and accuracy criteria. The simulation results show the proposed method mean accuracy on the BSDS500 database images has achieved respectively 0.915, and 0.898 with the salt-and-pepper noise with density of 0.01, and the Gaussian noise with zero mean, and variance of 0.01. The proposed method average execution time with 80 runs for each BSDS500 database image has also obtained at 50.01, and 50.02 seconds in the presence of the mentioned noises respectively.

Keywords
Edge detection, noise, Meta-heuristics optimization algorithm, Grey wolf optimization algorithm, Object recognition.

1. Introduction

Using the edge image instead of the image itself in the digital image processing algorithms significantly reduces the data volume, and omits the useless information, while maintaining the necessary features of an image. In the digital image processing algorithms, the edge detection process must be efficient and reliable because the possibility of completing the next processing steps depend on it.

On the other hand, it is very difficult to implement an edge detection algorithm that works correctly in any situation. Therefore, until now, the various types of the edge detectors have been introduced, each of which is designed in such a way that they are sensitive to the certain types of the edges.

The conventional edge detectors are often according to the calculation of a first-order or second-order local derivative operator, which is also called a filter or a mask. The mask is applied to each image pixel, and then by applying a threshold, the pixels that are the part of an edge are revealed. The lower the threshold, the falser edges are revealed, and the higher the threshold, the parts of the edge may be lost. Thus, the derivative-based edge detection operator is sensitive to the image large gradients, while its output is zero in the areas with uniform intensity. Some of the conventional edge detectors are the Canny, Sobel, LOG, and Kirsch, each of which, while frequently used in the past research, has shortcomings such as the false edge detection or failure to recognize the real edges in the presence of noise [1]. In an image, noise can be caused by various factors such as the unfavorable lighting conditions, changes in the temperature and humidity, the hardware defects of the image-generating devices, or errors in the image transfer process from one device to another and the performance of each to affect the edge detection algorithms. The phenomenon of noise makes it difficult to detect the edges in the images, because both the noise, and edges contain the high-frequency content and on the other hand trying to reduce noise leads to the blurring, and distortion of the edges. Of course, in the different edge detectors, there are solutions such as smoothing to deal with the noise, but this smoothing also affects the edges.

So far, to find the image edges in the presence of noise, various techniques have been used based on the neural networks, machine learning, fuzzy logic, wavelet transform, and meta-heuristic optimization algorithms [2]. The advantages of using the meta-heuristic optimization algorithms based on the reducing noise are that these algorithms can effectively search the space of possible solutions and easily work in the complex and unconditional problems [3]. Some meta-heuristic optimization algorithms include the particle swarm optimization (PSO), ant colony optimization (ACO),

whale optimization (WO), and grey wolf optimization (GWO) [3].

In this paper, a method is proposed based on the GWO algorithm with a new objective function to detect edges on the noisy images with the higher accuracy than the similar methods. The simulation results respectively show 1.94%, and 0.87% increment in the proposed method average accuracy compared to the similar methods for the edge detection on the BSDS500 database images in the presence of the salt-and-pepper noise with density of 0.01, and the Gaussian noise with zero mean, and variance of 0.01.

In the following, the related work is presented in section 2. The proposed method is introduced in section 3. The results and evaluation are reported in section 4. The paper concluded in section 5.

2. Related Work

The binary PSO (BPSO) based edge detection algorithm was presented based on the weighted sum of five cost factors in its objective function. The presented method was simulated on the BSDS500 database. The results were compared to the canny, and Prewitt edge detectors as well as the ACO, and genetic algorithms using the F-score criterion. The mean of the F-score was reported at 0.29 [4]. An algorithm was introduced based on the optimizing cellular automata rules using the PSO. The algorithm was also extended from the greyscale to the color images. The algorithm was comparable to the Canny, and Sobel edge detectors [5].

A method was presented according to the PSO for enhancing the quality of the detected edges at the multiresolution level. In the pre-processing phase, the discrete wavelet transform was used. Then, the PSO was applied to each of the sub-band images. The output image was reconstructed from the processed sub-band images using the inverse transform. The method was simulated on the Kodak database and evaluated using the root mean square error (RMSE), and peak signal-to-noise ratio (PSNR) criteria [6].

A method was introduced to solve the problem of the edge discontinuity based on the canny, and ACO algorithms. First, the canny edge detector obtained the image edges. Then, the edge endpoint was considered as the ant initial position. The fuzzy triangle membership function is introduced by the neighborhood grey-value. Each pixel fuzzy membership value between the edge endpoints was computed as the ant colony heuristic matrix. The heuristic matrix promoted the ants to search along the real edge to detect the continuous, and complete edge lines [7].

The max-min ACO algorithm was used to identify the image edges. In addition, a heuristic information function was introduced to obtain the nodes that the ants visited around their place. The algorithm exploited the difference between the intensity of two groups of the nodes instead of the two single ones [8].

A multi-population technique was applied to improve the accuracy of the ACO-based edge detector. The ants were divided into the two groups to perform both the advantages of the global and local searches. In the local search, one group of ants moved around the optima. The ants in the other group moved from the current position to jump out of the local optima and explored the global optima [9].

An ant-based edge detector was presented to provide the thinner edges on the noisy images. The method applied the pheromone matrix that represented the edge at each image pixel, and updated the pheromone matrix using the initial pheromone value [10].

An ACO-based technique for the edge detection was discussed in [11] that the quality of its detected edges depended on the choice of the constants, pheromone evaporation rate, number of iterations, and ant movement steps.

A WO algorithm was introduced for the edge detection in the Gaussian noisy images, with the weighted objective function including, the uniformity factor, and the mean gradient magnitude [12].

An edge detection algorithm was presented for the UAV-captured color images according to the modified WO. The color image pixels were represented by the quaternions. The global random position variables, and the information exchange mechanism were also presented into the WO random walk foraging equation. Further, a random disturbance factor was applied into the spiral bubble net predator-prey equation. Then, the method obtained the color image preliminary edge. To enhance the preliminary edge, an edge-point classification method by the radius of the shortest distance between the whale and the current global optimum in each iteration was used [13].

The canny grey wolf optimization (CGWO) method was introduced for the edge detection on the OCT images. First, to improve each image's contrast, the adaptive histogram equalization method was used. Then, the improved image was carried out in the WO algorithm and produced the required threshold level for the canny edge detector. Finally, the output image was obtained in an edge-structured form [14].

3. Proposed Method

The proposed method block diagram is illustrated in Fig. 1, and its steps are described as follows.

3.1. Kirsch Edge Detector

The Kirsch edge detector uses the eight directional masks shown in (1) [1], which respectively have an angle difference of 45°.

$$\begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix}$$
$$\begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix} \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$$
$$\begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} \quad (1)$$

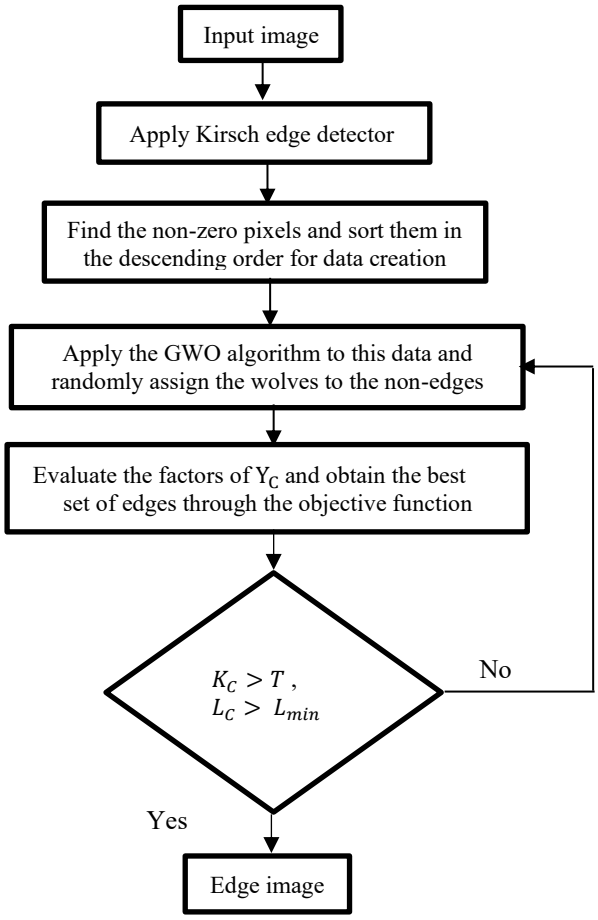

Fig. 1. The block diagram of the proposed method

3.2. GWO Algorithm

The Block diagram of the GWO algorithm is shown in Fig. 2 [15]. The positions of the wolves are computed, and updated by (2), and (3) [15].

$$\vec{x}(t+1) = \vec{x}_p(t) - \vec{A}.\vec{D} \tag{2}$$

$$\vec{D} = \left|\vec{C}.\vec{x}_p(t) - \vec{x}(t)\right| \tag{3}$$

where t indicates the current iteration. $\vec{x}_p$ is the position vector of the prey, and x is the position vector of a grey wolf. $\vec{A}$, and $\vec{C}$ are coefficient vectors that are calculated and updated by (4), and (5).

$$\vec{A} = \vec{a}(2\vec{r}_1 - 1) \tag{4}$$

$$\vec{C} = 2\vec{r}_2 \tag{5}$$

where the components of $\vec{a}$ are linearly decreased from 2 to 0 over each iteration. The $\vec{r}_1$, and $\vec{r}_2$ parameters are random vectors in [0,1]. Note that $\vec{A}$ is decreased by $\vec{a}$ and it is a random value in $[-2\vec{a}, 2\vec{a}]$. When $\vec{A} \in [-1,1]$, the next position of a search agent can be in any position between its current position and the prey position. The $\vec{C}$ vector contains random values in $[0,2]$ which provide random weights for the prey to emphasize ($C > 1$) or deemphasize ($C < 1$) the effect of the prey in defining the distance in (3). The first three best solutions are obtained, and saved as α , β , and δ wolves respectively. These wolves guide the optimization. The positions of α , β , and δ wolves are updated according to (6), (7), and (8). Overall, the GWO algorithm begins with creating a random population of the grey wolves. Over each iteration, the α , β , and δ wolves estimate the probable position of the prey. Each candidate solution updates its distance from the prey. This process continues until the specified number of the iteration reaches and provides candidate pixels. The final decision is based on the comparison between the average of the Kirsch output, K_c , and the threshold value of T . If $K_c > T$, the candidate pixels are considered as the edge pixels. In the proposed method, the value of T is selected 0.477 by performing many experiments on all the BSDS500 database images.

$$\vec{D}_\alpha(t) = \left|\vec{C}_1.\vec{x}_\alpha(t) - \vec{x}(t)\right|, \vec{D}_\beta(t) = \left|\vec{C}_2.\vec{x}_\beta(t) - \vec{x}(t)\right|,$$
$$\vec{D}_\delta(t) = \left|\vec{C}_3.\vec{x}_\delta(t) - \vec{x}(t)\right| \tag{6}$$

$$\vec{x}_1(t) = \vec{x}_\alpha(t) - \vec{A}_1.\vec{D}_\alpha(t), \vec{x}_2(t) = \vec{x}_\beta(t) - \vec{A}_2.\vec{D}_\beta(t),$$
$$\vec{x}_3(t) = \vec{x}_\delta(t) - \vec{A}_3.\vec{D}_\delta(t) \tag{7}$$

$$\vec{x}(t+1) = \frac{\vec{x}_1(t) + \vec{x}_2(t) + \vec{x}_3(t)}{3} \tag{8}$$

Where t indicates the current iteration. $\vec{A}_1, \vec{A}_2, \vec{A}_3, \vec{C}_1, \vec{C}_2,$ and $\vec{C}_3$ are calculated by (4), and (5) respectively.

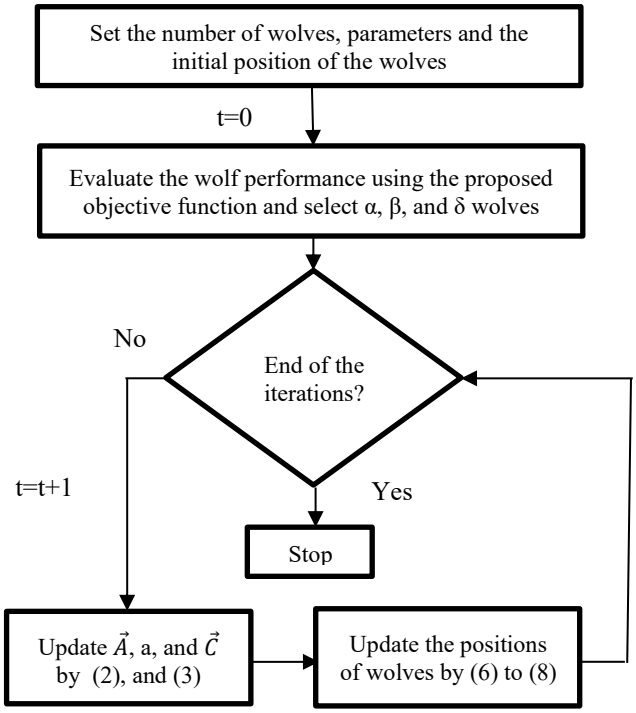

Fig. 2. Block diagram of the GWO algorithm [15]

3.2.1 Proposed Objective Function

The objective function of the proposed method is defined by Y_c according to (9) for the edge detection of the noisy images captured by a conventional camera. Y_c has to be maximized through the GWO algorithm.

$$Y_c = \begin{cases} [(H_c + K_c) - U_c], L_c & K_c > T \\ -\infty & K_c < T \end{cases} \quad (9)$$

Where H_c, U_c, and K_c indicate the homogeneity, uniformity, and average of the Kirsch edge detector outputs. L_c is a desirable length of an edge curve. In the proposed method, L_c is considered 5, means the edge curve including less than 5 pixels has to be omitted. The homogeneity, uniformity, and average of the Kirsch edge detector outputs are defined as follow.

3.2.1.1 Homogeneity Factor

The homogeneity factor, H_c is computed by applying a 3*3 mask to an input image, and using (10), (11), and (12) [13].

$$H_p = \begin{cases} \max(|I_p - I_{n_i}|) > T & ,i = 0,1,2,...,8 \\ 0 & else \end{cases} \quad (10)$$

$$H_c = \frac{1}{L_c} \sum_{p_i \in c} H_{p_i} \quad (11)$$

$$L_c = \sum_{p_i \in c} \begin{cases} 1 & for \quad each \quad an \quad edge \quad curve \quad pixel \\ 0 & else \end{cases} \quad (12)$$

where I_p is an edge curve pixel intensity corresponding to the central position of the mask. I_{n_i} is an edge curve pixel intensity corresponding to the neighboring position of the mask. T is chosen 25 by performing many experiments. c indicates an edge curve, and p_i is a pixel on c. If the examined pixel of p_i is an edge pixel, L_c becomes 1, otherwise 0, so it shows the effect of the noise removal on an edge image.

3.2.1.2 Uniformity Factor

The uniformity factor shows the intensity similarity between an edge curve pixels, and it is calculated by (13) [13].

$$U_c = \frac{1}{L_c} \sum_{i=1}^{L_c - 1} |p_{i+1} - p_i| \quad (13)$$

where I_{p_i}, and $I_{p_{i+1}}$ are the intensities of two pixels on an edge curve at positions i, and $i+1$. The edge curve length is L_c.

3.2.1.3 Kirsch Average

The average of the Kirsch edge detector outputs, K_c is computed by (14) [1] for all an edge curve pixels.

$$K_c = \frac{1}{L_c} \sum_{i=1}^{L_c} K_i \quad (14)$$

Where K_i is the Kirsch output of the i^{th} pixel on an edge curve of c with the length of L_c.

4. Results, and Evaluation

The simulation of the proposed method has been done using MATLAB R2020a software and a computer with a Core i7-2.8GHz processor, and 12GB RAM on the BSDS500 database images [16].

4.1. BSDS500 Database

The BSD database was prepared in 2001 by the University of Berkeley in California, whose images were collected from the Corel database, which is a multipurpose database. The BSD database is used in the fields of image processing, object recognition and deep learning. This database is also one of the most famous databases in the field of edge detection and image segmentation research. The BSD database contains high-quality images, and for each image, the result of edge detection by an experienced human is placed as a reference, which is called the Ground Truth image. The BSD database images, which were captured with an ordinary camera, include various subjects such as nature, city, objects and people. In our research, the developed version of BSD database named the BSDS500 database including 500 images along with their Ground Truth is used. The dimensions of these images are 481*321 and in jpg format.

4.2. Evaluation Criteria

To evaluate the output image of an edge detection algorithm, and compare it with its corresponding Ground Truth (GT) image, so far, various criteria have been used. The most frequent of these criteria are the mean square error (MSE), signal-to-noise ratio (PSNR), precision, recall, F-score, and accuracy, which are defined based on the confusion matrix members shown in Fig. 3 by (15) to (20) [17]. The confusion matrix shows the statistical experiments results based on the real data.

		Edge detector output image	
		Pixels with value of one	pixels with value of zero
Ground Truth (GT) image	Pixels with value of one	TP	FN
	pixels with value of zero	FP	TN

Fig. 3. Confusion matrix members

Where, TP means a pixel in the GT image is one and this pixel is one in the output image of the edge detector. FN

means a pixel in the GT image is one and this pixel is zero in the output image of the edge detector. TN means a pixel in the GT image is zero and this pixel is zero in the output image of the edge detector. FP means a pixel in the GT image is zero and this pixel is one in the output image of the edge detector.

$$MSE = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}[(GT_{(x,y)} - O_{(X<Y)})]^2 \quad (15)$$

$$PSNR = 10\log(\frac{peakval^2}{MSE}) \quad (16)$$

$$precision = \frac{TP}{TP + FP} \quad (17)$$

$$recall = \frac{TP}{IP + FN} \quad (18)$$

$$F - score = \frac{2TP}{2TP + FP + FN} \quad (19)$$

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (20)$$

Where $GT_{(x,y)}$ is an image of Ground Truth. $O_{(x,y)}$ is an edge image. M , and N are respectively the rows, and columns of an image matrix. The value of $peakval$ is one, because an edge image is a binary image.

The lower is the MSE criterion and the higher are the PSNR, Precision, recall, F-score, and accuracy criteria, the edge detector efficiency is higher, which means the edge detector recognizes the image edges more accurately in their actual locations in the image.

4.3. Simulation Results

The simulation results of the proposed method on the BSDS500 database images without noise, and in the presence of the Gaussian noise with zero mean, and variance of 0.01, and the salt-and-pepper noise with density of 0.01 are reported in Table I. The settings of the proposed method parameters are the same in the noisy, and non-noise conditions.

Table I. Simulation results of the proposed method on the BSDS500 database images without noise, and in the presence of the Gaussian, and salt-and-pepper noises

Proposed method	MSE	PSNR	precision
Non-noise	0.0138	42.753	0.8824
	recall	F-score	accuracy
	0.6865	0.7683	0.9372
Salt-and-pepper noise	MSE	PSNR	precision
	0.0194	39.362	0.8912
	recall	F-score	accuracy
	0.6583	0.7505	0.9146
Gaussian noise	MSE	PSNR	precision
	0.0209	38.413	0.8974
	recall	F-score	accuracy
	0.6406	0.7416	0.8983

Table II. Performance evaluation of the proposed method on the BSDS500 database images in the presence of the Gaussian and salt-and-pepper noises with different density

Proposed method		Evaluation Criterion		
Noise type	Noise Density	MSE	PSNR	precision
Salt-and-pepper	0.015	0.026	36.147	0.9073
	0.02	0.042	31.537	0.9146
	0.025	0.076	25.821	0.9260
	0.03	0.125	19.934	0.9386
Noise type	Noise Density	recall	F-score	accuracy
Salt-and-pepper	0.015	0.5996	0.7148	0.8839
	0.02	0.5186	0.6589	0.8534
	0.025	0.3864	0.5363	0.8113
	0.03	0.3864	0.3726	0.7469
Noise type	Noise Density	MSE	PSNR	precision
Gaussian	0.015	0.031	34.569	0.9125
	0.02	0.051	29.864	0.9301
	0.025	0.091	23.173	0.9359
	0.03	0.181	17.173	0.9486
Noise type	Noise Density	recall	F-score	accuracy
Gaussian	0.015	0.5874	0.7104	0.8651
	0.02	0.4923	0.6384	0.8279
	0.025	0.3518	0.5131	0.7863
	0.03	0.2067	0.3364	0.7046

To evaluate the proposed method performance in the presence of noises with different density, the amount of the applied noise was increased from 0.015 to 0.03 with steps of 0.005, and the results are shown in Table II. As it is seen, the proposed method still performs well in the presence of the salt-and-pepper noise and Gaussian noise with increasing their density to 0.03, which are the strong noises. However, these noises with density of more than 0.03, which are rare, destroy the image so much that the proposed method is not able to recognize all the image edges correctly. To show applying the GWO algorithm is the best choice among the PSO, ACO, and WO algorithms for the noisy images edge detection, we used the GWO, PSO, ACO, and WO algorithms with our proposed objective function in the proposed method, and the simulation results are shown in Table III. As it is seen, the average accuracy of the proposed method based on the GWO algorithm is increased by 1.31% compared to the PSO algorithm, by 2.41% compared to the ACO algorithm, and by 0.8% compared to the WO algorithm in the salt-and-pepper noise with density of 0.01. The average accuracy of the proposed method based on the GWO algorithm is also increased by 1.71% compared to the PSO algorithm, by 1.88% compared to the ACO algorithm, and by 0.58% compared to the WO algorithm in the Gaussian noise with zero mean and variance of 0.01. In Table IV, the Performance comparison of the proposed method with the well-known edge detectors is shown in the presence of the Gaussian noise with zero mean, and variance of 0.01, salt-and-pepper noise with density of 0.01, and non-noise on the BSDS500 database images.

Table III. Performance comparison of the proposed method based on the GWO, PSO, ACO, and WO algorithms with our proposed objective function on the BSDS500 database images in the noise presence

Algorithm-Noise	Evaluation Criterion		
	MSE	PSNR	precision
GWO-salt-&-pepper	0.0194	39.362	0.8912
GWO-Gaussian	0.0209	88.413	0.8947
	recall	F-score	accuracy
GWO-salt-&-pepper	0.6583	0.7505	0.9146
GWO-Gaussian	0.6406	0.7416	0.8983
	MSE	PSNR	precision
PSO-salt-&-pepper	0.0261	39.649	0.8452
PSO-Gaussian	0.0294	35.372	0.8517
	recall	F-score	accuracy
PSO-salt-&-pepper	0.6153	0.7132	0.9028
PSO-Gaussian	0.6078	0.7069	0.8832
	MSE	PSNR	precision
ACO-salt-&-pepper	0.0272	36.014	0.8376
ACO-Gaussian	0.0297	35.148	0.8542
	recall	F-score	accuracy
ACO-salt-&-pepper	0.6094	0.7076	0.8931
ACO-Gaussian	0.6014	0.6989	0.8817
	MSE	PSNR	precision
WO-salt-and-pepper	0.0221	38.152	0.8671
WO-Gaussian	0.0243	37.257	0.8738
	recall	F-score	accuracy
WO-salt-and-pepper	0.6274	0.7259	0.9037
WO-Gaussian noise	0.6152	0.7214	0.8931

Table IV. Performance comparison of the proposed method with the well-known edge detectors in the presence of the Gaussian noise with zero mean, and variance of 0.01, in the presence of the salt-and-pepper noise with density of 0.01, and non-noise on the BSDS500 database images

PSNR	MSE	Noise type	Edge detector
25.821	0.0758	Non-noise	Canny
٢٢.634	0.102	Salt-and-pepper	
20.471	0.128	Gaussian	
26.796	0.0691	Non-noise	Sobel
22.943	0.0989	Salt-and-pepper	
21.125	0.119	Gaussian	
24.698	0.0863	Non-noise	LOG
22.243	0.108	Salt-and-pepper	
20.261	0.129	Gaussian	
28.382	0.0576	Non-noise	Kirsch
23.465	0.0948	Salt-and-pepper	
22.063	0.1117	Gaussian	
42.753	0.0138	Non-noise	Proposed
39.362	0.0194	Salt-and-pepper	
38.413	0.0209	Gaussian	

As it is seen in Table IV, for the proposed method in the presence of the salt-and-pepper noise with density of 0.01, the MSE criterion is at least decreased by 76%, and the PSNR criterion is at least increased by 79.5%. In addition, for the proposed method in the presence of the Gaussian noise with a zero mean and a variance of 0.01 is at least increased by 81.29% in comparison with the Canny, Sobel, LOG, Kirsch edge detectors that are the well-known edge detectors. In addition, the proposed method performance is compared with the methods in [4], [6], [10], and [12] in which the edge detection of the noisy images is respectively based on the binary PSO (BPSO), PSO, ACO, and WO algorithms. These performance comparison results in the presence of the salt-and-pepper noise with density of 0.01 are shown in Fig. 4. Similarly, these performance comparison results in the presence of the Gaussian noise with zero mean, and variance of 0.01 are also shown in Fig. 5. The implementation of all these methods has been done in our system using the BSDS500 database images. As it is seen in Fig. 4, the average accuracy of the proposed method compared to the method in [4], [6], [10], and [12] have increased by 2.6%, 2%, 1.22%, and 1.19% respectively on the edge detection of the BSDS500 database images in the presence of salt-and-pepper noise with density of 0.01. As it is seen in Fig. 5, the average accuracy of the proposed method compared to the method in [4], [6], [10], and [12] have increased respectively by 1.76%, 0.65%, 0.2%, and 0.9% on the edge detection of the BSDS500 database images in the presence of the Gaussian noise with zero mean, and variance of 0.01.

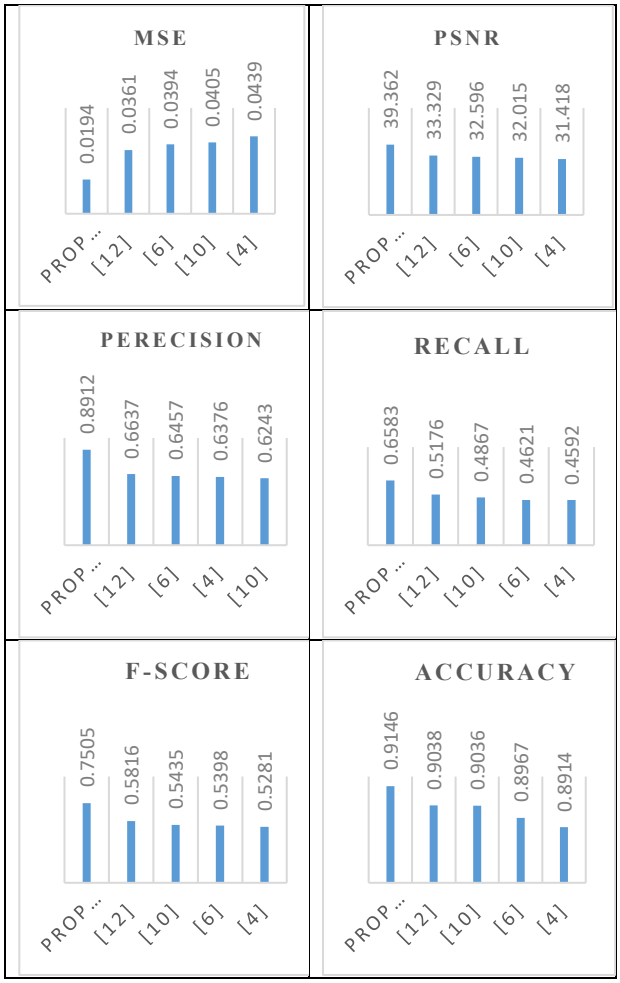

Fig. 4. Performance comparison of the proposed method with methods in [4], [6], [10], and [12] on the BSDS500 database images in the presence of the salt-and-pepper noise with density of 0.01 based on the evaluation criteria of the MSE, PSNR, precision, recall, F-score, and accuracy

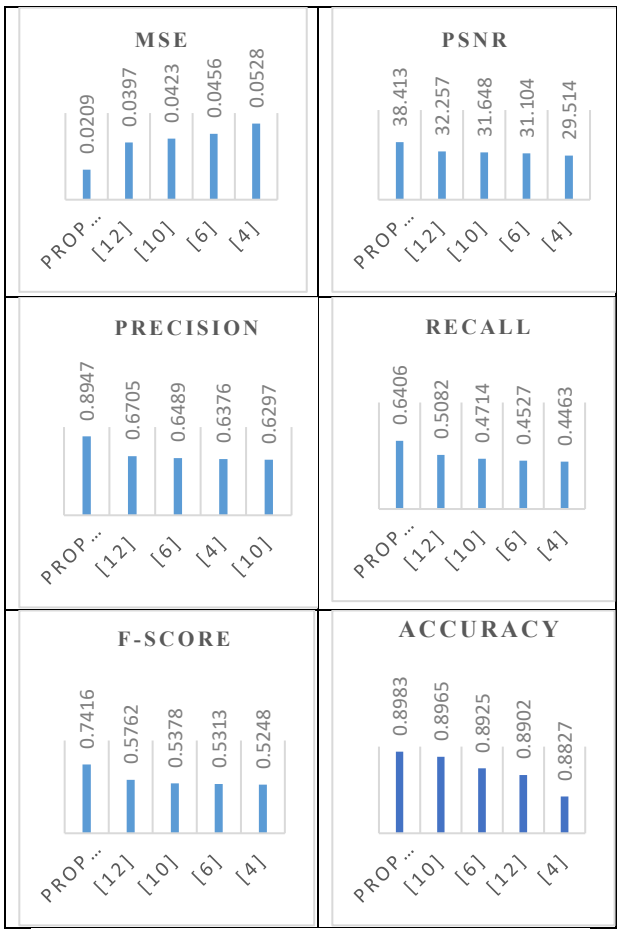

Fig. 5. Performance comparison of the proposed method with methods in [4], [6], [10], and [12] on the BSDS500 database images in the presence of the Gaussian noise with zero mean and variance of 0.01.

Furthermore, the average execution time of the proposed method is also compared to the methods in [6], [10], and [12] on the BSDS500 database images in the presence of the salt-and-pepper noise with density of 0.01, and the Gaussian noise with zero mean, and variance of 0.01. These results are shown in Table V. It has to be noticed that all the methods in Table IV have been implemented with 80 runs for each image of the BSDS500 database on our system with a Core i7-2.8GHz processor, and 12GB RAM to have fair comparisons.

Table V. Comparison of the average of execution time of the proposed method with the similar methods with 80 runs for each image of the BSDS500 database in the presence of noise.

Average Execution Time (sec.)	Method- Noise
50.01	Proposed- salt-and-pepper
50.02	Proposed- Gaussian
53.17	In [6]- salt-and-pepper
53.26	In [6]- Gaussian
49.14	In [10]- salt-and-pepper
48.92	In [10]- Gaussian
50.07	In [12]- salt-and-pepper
50.19	In [12]- Gaussian

A sample visual result of the proposed method on the image #61060 from the BSDS500 database compared to the results of methods in [6], [10], and [12] for this image is shown in Fig. 6 in the presence of the salt-and-pepper noise with density of 0.01. Similarly, the visual results for this image in the presence of the Gaussian noise with zero mean, and variance of 0.01 is shown in Fig. 7. Considering all the visual results of the proposed method on the BSDS500 database images in the presence of the mentioned noises, the higher efficiency of the proposed method is also confirmed in comparison with the similar methods.

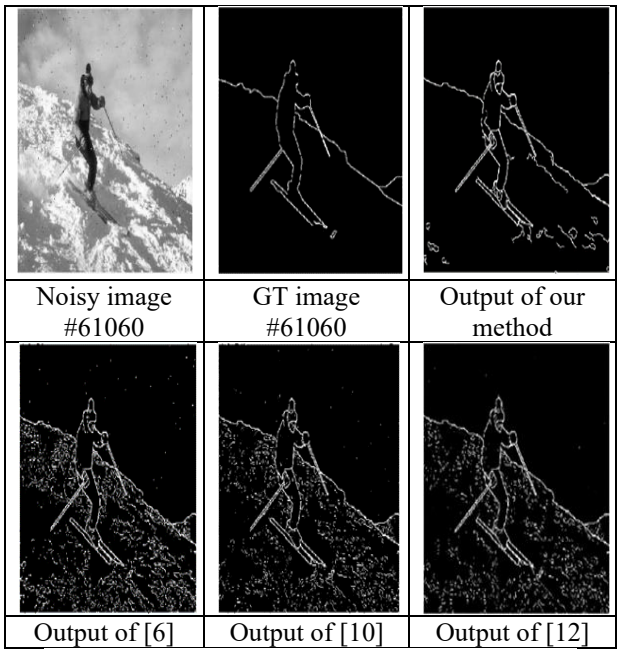

Fig. 6. The extracted edges of noisy image #61060 from the BSDS500 database by the proposed method and methods in [6], [10], and [12] in the presence of the salt-and-pepper noise with density of 0.01.

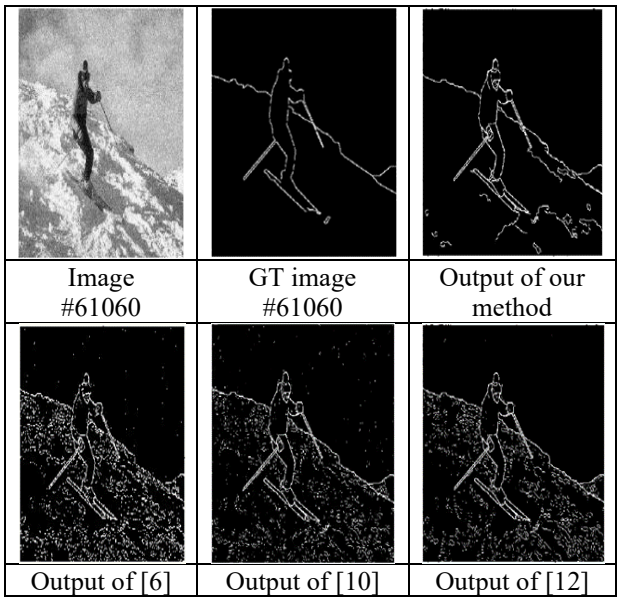

Fig. 7. The extracted edges of noisy image #61060 from the BSDS500 database by using the proposed method and methods in [6], [10], and [12] in the presence of the Gaussian noise with zero mean, and variance of 0.01.

5. Conclusion

To detect edges of a noisy image, the GWO-based edge detector was presented. For this purpose, a new objective

function was proposed for the GWO algorithm in the proposed edge detector. The proposed objective function was combining of the homogeneity, uniformity, and average of the Kirsch masks outputs. The homogeneity was used to find clues from an image edges. The uniformity helped to find the intensity similarity between the edge pixels. The Kirsch edge detector masks were applied as an effective factor to identify the vertical, horizontal, diagonal, and curved edges on an image.

The simulation results of the proposed method were respectively reported at 0.0194, 39.362, 0.8912, 0.6583, 0.7505, and 0.9146 based on the MSE, PSNR, precision, recall, F-Score, and accuracy using the BSDS500 database images in the presence of the salt-and-pepper noise with density of 0.01. These results were also reported respectively at 0.0209, 38.413, 0.8947, 0.6406, 0.7416, and 0.898 in the presence of the Gaussian noise with zero mean, and variance of 0.01.

In addition, increasing the density of these noises from 0.015 to 0.03 with steps of 0.005 showed the proposed method performances were decreased, but these performances were still well in comparison with the other similar methods.

Furthermore, the average precision, recall, and F-score of the proposed method in comparison with the similar methods were respectively increased by 39%, 36.75%, and 37% on the BSDS500 database images in the presence of the salt-and-pepper noise with density of 0.01. These increment rates were also reported at 38.25%, 36.5%, and 36.75% respectively in the presence of the Gaussian noise with zero mean, and variance of 0.01.

Also, the average execution time of the proposed method with 80 runs for each image of the BSDS500 database has respectively obtained at 50.01, and 50.02 seconds in the presence of the salt-and-pepper noise with density of 0.01, and the Gaussian noise with zero mean, and variance of 0.01.

References

[1] R. Muthukrishnan, M. Radha, “Edge detection techniques for image segmentation”, *International Journal of Computer Science & Information Technology*, vol.3, no. 6, pp. 256-267, 2011.

[2] N. A. Golilarz, H. Gao, H. Demirel, “Satellite image de-noising with harris hawks meta heuristic optimization algorithm and improved adaptive generalized Gaussian distribution threshold function”, *IEEE Access*, vol. 7, pp. 57459-57468, 2019.

[3] S. Mirjalili, A. Lewis, “The whale optimization algorithm”, Advances in Engineering Software, vol. 95, pp. 51-67, 2016.

[4] N. S. Dagar, P. K. Dahiya, “Edge detection technique using binary particle swarm optimization”, *Procedia Computer Science*, vol. 167, pp.1421-143, 2020.

[5] D. Dumitru, A, Andreica, L. Diosan, Z. Baliot, “Particle swarm optimization of cellular automata rules for edge detection”, In 21st International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), September 2019, Timisoara, Romania. DOI: 10.1109/SYNASC49474.2019.00052

[6] A. Eleyan, M. Anwar, “Multiresolution edge detection using particle swarm optimization”, *IEEE International Journal of Engineering Science and Application*, vol. 1, no. 1, pp. 11-17, 2017.

[7] Q. Shi, J. An, K. K. Gagnon, R. Cao, H. Xie, “Image edge detection based on the canny edge and the ant colony optimization algorithm”, In 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), October 2019. DOI: 10.1109/CISP-BMEI48845.2019.8965950

[8] S. Kheirinejad, S. M. H. Hasheminejad, N. Riahi, “Max-min ant colony optimization method for edge detection exploiting a new heuristic information function”, In 8th International Conference on Computer and Knowledge Engineering (ICCKE), October 2018. DOI: 10.1109/ICCKE.2018.8566516

[9] S. Wang, “A Novel Image Edge Detection Method Based on Multi-Population Ant Colony Optimization”, In 6th International Conference on Information Science and Control Engineering (ICISCE), June 2019, Shanghai, China. DOI: 10.1109/ICISCE48695.2019.00028

[10] M. Rafsanjani, Z. Varzaneh, “Edge detection in digital images using Ant Colony Optimization”, *Computer Science Journal of Moldova*, vol. 69, no. 3, pp. 343-359, 2015.

[11] A. Srivastava, R. Singh, S. Juneja, G. Verma, “Ant colony optimization based edge detection in digital images”, In 5th International Conference on Computational Intelligence and Communication Technologies (CCICT), July 2022, pp.107-113, Sonepat, India. DOI: 10.1109/CCiCT56684.2022.00031

[12] A. Gautam, M. Biswas, “Whale optimization algorithm based edge detection for noisy image”, In Second International Conference on Intelligent Computing and Control Systems (ICICCS), June 2018, Madurai, India. DOI: 10.1109/ICCONS.2018.8663022

[13] D. Liu, S. Zhou, R. Shen, X. Lu, “Color image edge detection method based on the improved whale optimization algorithm”, *IEEE Access*, vol. 11, pp. 5981-5989, 2023.

[14] M. S. N. Devi, S. Santhi, “Improved edge detection methods in OCT images using a hybrid framework based on CGWO algorithm”, In International Conference on Communication and Signal Processing (ICCSP), April 2019, Chennai, India.
DOI: 10.1109/ICCSP.2019.8698096

[15] S. Mirjalili, S. M. Mirjalili, A. Lewis, “Grey wolf optimizer”, *Advances in Engineeing Software*, vol. 69, pp. 46-6, 2014.

[16] Kaggle.com/datasets/balraj98/berkeley-segmentation-dataset-500-bsds500/

[17] M. Hossin, M. N. Sulaiman, “A review on evaluation metrics for data classification evaluations”, *International journal of data mining & knowledge management process*, vol. 5, no. 2, pp. 1-11, 2015.