ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/375889811

A MTIS method using a combined of whale and moth-flame optimization algorithms

Chapter · November 2023

DOI: 10.1016/B978-0-32-395365-8.00051-8

CITATION
1

READS
122

7 authors, including:

Taybeh Salehnia
Razi University
9 PUBLICATIONS **14** CITATIONS

SEE PROFILE

Saadat Izadi
Razi University
8 PUBLICATIONS **47** CITATIONS

SEE PROFILE

Ahmadreza Montazerolghaem
University of Isfahan
60 PUBLICATIONS **313** CITATIONS

SEE PROFILE

Laith Abualigah
Al al-Bayt University
471 PUBLICATIONS **16,507** CITATIONS

SEE PROFILE

All content following this page was uploaded by Ahmadreza Montazerolghaem on 24 November 2023.

The user has requested enhancement of the downloaded file.

Chapter 45

A MTIS method using a combined of whale and moth-flame optimization algorithms

Taybeh Salehnia[a], Farid MiarNaeimi[b], Saadat Izadi[a], Mahmood Ahmadi[a], Ahmadreza Montazerolghaem[c], Seyedali Mirjalili[d,e], and Laith Abualigah[f,g,h,i,j,k]

[a]Department of Computer Engineering and Information Technology, Razi University, Kermanshah, Iran, [b]Faculty of Engineering, Department of Civil Engineering, University of Sistan and Baluchestan, Zahedan, Iran, [c]Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran, [d]Centre for Artificial Intelligence Research and Optimisation, Torrens University Australia, Brisbane, QLD, Australia, [e]Yonsei Frontier Lab, Yonsei University, Seoul, South Korea, [f]Computer Science Department, Al al-Bayt University, Mafraq, Jordan, [g]Hourani Center for Applied Scientific Research, Al-Ahliyya Amman University, Amman, Jordan, [h]MEU Research Unit, Middle East University, Amman, Jordan, [i]Department of Electrical and Computer Engineering, Lebanese American University, Byblos, Lebanon, [j]School of Computer Sciences, Universiti Sains Malaysia, Pulau Pinang, Malaysia, [k]School of Engineering and Technology, Sunway University Malaysia, Petaling Jaya, Malaysia

45.1 Introduction

Segmentation is considered as an essential step in image processing. This process divides different parts of the image into several categories. Multi-level Thresholding is a method that facilitates this process. The problem is to correctly segment each image to find the best set of thresholds [1]. Thresholding usually uses image processing methods due to its consistency and low Computational Complexity (CC). Two main methods are Otsu's method [2–4] and Kapur's method [5,6]. However, such approaches have high CC for Multi-level Thresholding [7]. Thresholds help each other to separate interesting objects from their background. The higher splitting quality depends on the selected thresholds [8]. Recently, Meta-Heuristic (MH) algorithms like Particle Swarm Optimization (PSO) [9], Whale Optimization Algorithm (WOA) [10], Moth-Flame Optimization (MFO) [11] have been successfully applied for Thresholding problems [3,8,12], and ABC [13,14] and, Harris Hawks Optimizer (HHO) [15] are used in other problems.

MH algorithms have attracted the attention of researchers due to their excellent performance in finding threshold vectors in Multi-level Thresholding Image Segmentation (MTIS) systems. MH algorithms are either used separately in these problems, or been used in a combined version to solve the MTIS. Most MH algorithms are population-based and initially find a plausible answer by randomly moving through the search space. Such algorithms also include two phases of exploration and exploitation to search for the desired solution on the search space, through which the two phases search globally and locally, respectively. Therefore, several attempts have been made in the literature to achieve a better balance between exploration and exploitation phases to ensure maximum performance on a given optimization problem. In this chapter, our contribution is the design and implementation of an MTIS system using a combination of WOA, MFO, and the Inverse Otsu (IO) Function. This modification is developed using the operators of the MFO algorithm in an attempt to enhance the exploitation phase of WOA during the process of finding the optimal solution for a given optimization problem. It is used to increase the system's performance so that the combined MFWOA algorithm performs better than WOA and MFO and provides better solutions. Therefore, the optimal exploration and exploitation properties of MFO and WOA are used in the search space to find the best thresholds. The rest of our chapter is organized as follows: Section 45.2 presents an overview of related work. In Section 45.3, we describe the prerequisites used in the proposed method. Section 45.4 offers the proposed method. Section 45.5 describes the performance analysis and test results. Finally, Section 45.6 presents the conclusions.

45.2 Related work

The works that have been done so far in the field of MTIS using MH algorithms are single MH and Hybrid MH, which are briefly described in the following.

Handbook of Whale Optimization Algorithm. https://doi.org/10.1016/B978-0-32-395365-8.00051-8
Copyright © 2024 Elsevier Inc. All rights reserved, including those for text and data mining, AI training, and similar technologies.

45.2.1 Image segmentation using single meta-heuristics

This section briefly reviews the latest related work on image segmentation. Rodríguez-Esparza et al. proposed the HHO-based solver for image segmentation based on K-means and the Fuzzy IterAg machine learning algorithms [16]. The experimental results show that the proposed method improves accuracy, consistency, and quality compared to the other methods. Anitha et al. [17] presented a Modified WOA (MWOA) to optimize the Multi-level color image Thresholding. The experiments show that the proposed method using MWOA performs better with less CPU computing time, image quality, and feature protection than other state-of-the-art algorithms. Abd El Aziz et al. [3] tested the ability of each of the WOA and MFO algorithms separately. During the optimization, they used Otsu's Fitness Function. Their proposed method was performed on different benchmark images compared with five algorithms. Also, the proposed method is provided a good balance between exploration and exploitation and works better than other algorithms.

Doun et al. [18] provided an Improved Cuckoo Search (ICS) for the optimal Multi-level Thresholding. Two modifications were used to improve the cuckoo search algorithm. In the experiments, six benchmark test images and a series of measures were performed, including Fitness Function value and standard deviation, Peak Signal to Noise Ratio (PSNR), FSIM, and Structure Similarity Index (SSIM). The result shows that the ICS algorithm is superior to other MH algorithms. Salehnia et al. [19] performed three MFO, WOA, and Grasshopper Optimization Algorithm (GOA) for utilizing Multi-level Thresholds, which use a mathematical equation using the corresponding image features as a Fitness Function. The results show that these algorithms are better than other algorithms for the Fitness Function, and GOA achieves a higher performance.

45.2.2 Hybrid meta-heuristics

Abd Elaziz et al. [12] developed a method for determining the optimal threshold for image segmentation. Their proposed method is an enhanced HHO by considering the Salp Swarm Algorithm (SSA), which is called HHOSSA, to improve HHO. The evaluation results show that the proposed method compared to HHO, SSA, and other methods obtained excellent results and performance. Samantaray et al. [20] present a new algorithm, the Harris Hawks-Cuckoo Search (HH-CS) algorithm, based on Multi-level Thresholding. This paper uses eight different images for the breast cancer thermogram image analysis, and some metrics such as PSNR, Feature Similarity Index (FSIM), SSIM are used. HHO-CS algorithm is beneficial for analysis of image and Function optimization. Hosseinzade and Mozafari [5] provided a hybrid algorithm based on Genetic Algorithm (GA) and Simulated Annealing (SA) algorithm for MTIS. The advantage of GA is that it is precise, and the disadvantage is that it is time consuming. The advantage of the SA is that it is fast and has a simple search space, and the disadvantage is that it may stuck in local minima. They used Otsu and Kapur methods as Fitness Functions and obtained their results based on four benchmarks. Their results showed that their proposed algorithm outperforms other algorithms.

45.2.3 Weakness of single and combined algorithms used to solve MTIS problem

In all the papers reviewed in the literature section, MH algorithms have been used individually, improved, and combined to solve the MTIS problem. These algorithms have been trying to obtain relatively optimal thresholds or solutions to the MTIS problem. But according to the results of the algorithms seen in the papers and according to their evaluation, not all of them are able to find the best global answer. In other words, according to the numbers observed for PSNR, SSIM and processing time in these papers, the accuracy of the segmented image using the thresholds obtained from these methods is low and most of them have high computation time. This indicates that the algorithms used in the existing papers have not been able to obtain the optimal global answer and the thresholds obtained by the respective algorithms can not segment the image pixels more accurately. Therefore, in this chapter, in order to improve MFO, a combined MFWOA algorithm is proposed in which the operators used in WOA help to increase the power of MFO in finding the optimal answer.

45.3 Preliminaries

This section will discuss the Otsu method, WOA, and MFO.

45.3.1 Fitness function

In this chapter, the Otsu Thresholding method (IO Function) is used as a Fitness Function in the corresponding MH algorithms to determine the optimal threshold vector for classifying and boundaring image pixels (Eq. (45.4)). Otsu Thresholding method [21] is a popular method used as a Fitness Function in most MTIS methods that use MH algorithms. The Otsu method is an automatic Thresholding method obtained according to the image histogram. Using the Otsu method, the boundaries of objects in the desired image can be specified. The Otsu function is computed using Eq. (45.1).

$$F = \sum_{i=0}^{k} SUM_i (\mu_i - \mu_1)^2 \tag{45.1}$$

$$SUM_i = \sum_{j=T_i}^{T_{i+1}-1} P_j \tag{45.2}$$

$$\mu_i = \sum_{j=T_i}^{T_{i+1}-1} i \frac{P_j}{SUM_i} \quad \text{where } P_j = f(j)/NUM_p \tag{45.3}$$

$$Fit = 1/F \tag{45.4}$$

In Eq. (45.1), μ_1 is the image density average for $T_1 = 0$ and $T_2 = I$ (where I is the maximum pixel density of the image, which is 255 for gray images), μ_i is the density average of the C_i class for T_i and $T_{i+1} - 1$, k is the number of searched thresholds, and SUM_i is the sum of probabilities. In Eq. (45.2) and Eq. (45.3), P_j indicates the probability of the gray level j, $f(j)$ is the frequency of the gray level j, and NUM_p is the total number of pixels in the image. Eq. (45.4) is the same Fitness Function used in the algorithms in this chapter.

45.3.2 Whale optimization algorithm

In WOA [10], as with most optimization algorithms, the optimization process begins with a randomly generated set of candidate solutions ($\overrightarrow{X_i}$ vector) [10]. It should be noted that for the MTIS problem in this chapter, the position of each threshold value or the position of each solution ($\overrightarrow{X_i}$) is between the minimum pixel brightness and the maximum pixel brightness in the image. Each solution is represented as a vector according to Eq. (45.5). The solutions produced using Eq. (45.6) and evaluated using Eq. (45.4).

$$\overrightarrow{X_i} = \left(x_{i,1}, x_{i,2}, \ldots, x_{i,k}\right) \quad \text{where } 0 \le x_{i,1}, x_{i,2}, \ldots, x_{i,k} \le H \tag{45.5}$$

$$x_{i,j} = lb + rand(0,1) \times (ub - lb), \quad x_{i,j} \in \overrightarrow{X_i}, \quad j = 1, 2, \ldots, k \tag{45.6}$$

where in MTIS problem, lb and ub are the lower bound and the upper bound, respectively, $x_{i,k}$ represents each threshold of the threshold vector, $rand(0,1)$ is a random number between 0 and 1, and H represents the maximum brightness of the pixels in the image. The input and output of WOA are the image histogram and the threshold vector, respectively. This algorithm is inspired by humpback whales' bubble-net hunting method. The WOA is performed in three phases as follows [10]:

- Siege hunting phase.
- Exploitation phase: The bubble net attacking method.
- Exploration phase: Hunting search.

Once the best search agent is identified, other search agents try to update their location to the best search agent. As:

$$\overrightarrow{D} = \left| \overrightarrow{C} . \overrightarrow{X}^*(t) - \overrightarrow{X}(t) \right| \tag{45.7}$$

$$\overrightarrow{X}(t+1) = \overrightarrow{X}^*(t) - \overrightarrow{A} . \overrightarrow{D} \tag{45.8}$$

where t denotes the current iteration, $\overrightarrow{A}$ and $\overrightarrow{C}$ are the coefficient vectors, $\overrightarrow{D}$ is the distance between the position of $\overrightarrow{X}^*(t)$ and $\overrightarrow{X}(t)$, $X^*(t)$ the location vector is the best solution obtained at present, and $\overrightarrow{X}(t)$ is the location vector. Vectors $\overrightarrow{A}$

and $\overrightarrow{C}$ are calculated as follows:

$$\overrightarrow{A} = 2\overrightarrow{a}.\overrightarrow{r} - \overrightarrow{a} \quad (45.9)$$

$$\overrightarrow{C} = 2\overrightarrow{r} \quad (45.10)$$

where $\overrightarrow{a}$ decreases linearly from 2 to 0 during iterations (in both exploration and exploitation phases), and $\overrightarrow{r}$ is considered a random vector between 0 and 1. Two methods have been designed to model the bubble net behavior of whales mathematically:

a. *Contractile blocking mechanism*

This behavior is achieved by increasing a value in Eq. (45.9). The oscillation range of $\overrightarrow{A}$ is reduced by a. In other words, $\overrightarrow{A}$ is a random value in the distance $[-a, a]$, and is decrease from 2 to 0 during iterations. The new location of the search agent can be defined by selecting random values of a in the range −1 to 1 anywhere between the primary area of the agent and the location of the current best agent.

b. *Spiral Updating Location*

This method first calculates the distance between the whale located in the bait's $\overrightarrow{X}$ and $\overrightarrow{Y}$ coordinates in $\overrightarrow{X}^*(t)$ and $\overrightarrow{Y}^*(t)$. A spiral equation is created between the whale's position and the bait to mimic the spiral-shaped movement of the humpback whale:

$$\overrightarrow{X_i}(t+1) = \begin{cases} \overrightarrow{X}^*(t) - \overrightarrow{A}.\overrightarrow{D}, & p < 0.5 \\ \overrightarrow{D}'(t).e^{bl}.cos(2\pi l) + \overrightarrow{X}^*(t), & p \geq 0.5 \end{cases} \quad (45.11)$$

where $\overrightarrow{D}'$ refers to the distance from the 1st whale to the bait (the best solution obtained so far), b is a constant for defining the shape of the logarithmic spiral, and l is a random number between −1 and 1. It is assumed that the whale to model this simultaneous behavior with a 50% probability chooses one of the contractile siege mechanism or spiral models to update the whales' position during optimization. Also:

$$\overrightarrow{D} = \left|\overrightarrow{C}.\overrightarrow{X_{rand}} - \overrightarrow{X}\right| \quad (45.12)$$

$$\overrightarrow{X}(t+1) = \overrightarrow{X_{rand}} - \overrightarrow{A}.\overrightarrow{D} \quad (45.13)$$

where $\overrightarrow{X_{rand}}$ is the current population's randomly selected position vector (random whale). A random search agent is selected in $|\overrightarrow{A}| > 1$ mode, while the best solution is selected when $|\overrightarrow{A}| < 1$ to update the position of the search agents. Finally, WOA stops by reaching the stop condition, and the best solution in the MTIS is the same threshold vector as the final answer or output of the algorithm. Fig. 45.1 shows the process of producing optimal thresholds in the MTIS using WOA.

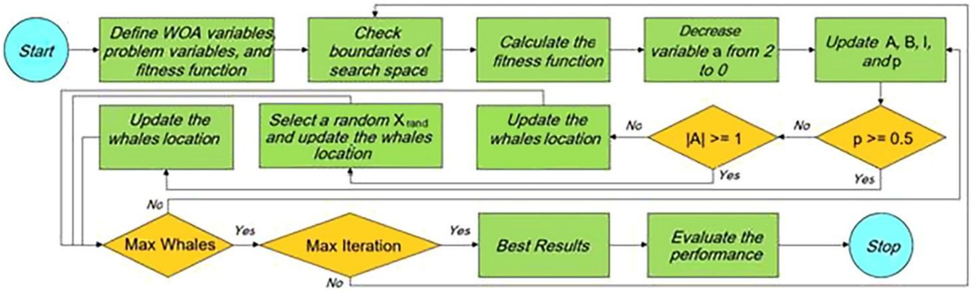


FIGURE 45.1 The structure of the WOA in the MTIS problem.

45.3.3 Moth-flame optimization algorithm

MFO Algorithm is another nature-inspired MH for solving optimization problems designed in the year 2016 [11]. Like other MH algorithms, the MFO starts the optimization process with an initial population $\overrightarrow{X_l}$ $(i = 1, 2, \ldots, N)$ of N moths that are randomly located in different locations. It should be noted here that moths and flames are both solutions. The difference between them is the way we treat and update them in each iteration. The moths are actual search agents that move around the search space, whereas flames are the best position of moths that obtains so far. In other words, flames can be considered as flags or pins that are dropped by moths when searching the search space. Therefore, each moth searches around a flag (flame) and updates it in case of finding a better solution. With this mechanism, a moth never lose its best solution. Each moth or solution ($\overrightarrow{X_i}$) is shown as Eq. (45.5). The position of each moth is initialized using Eq. (45.6) and evaluated using Eq. (45.4).

$$\overrightarrow{X_i} = \overrightarrow{D_l}.e^{bl}.cos(2\pi l) + \overrightarrow{F_u} \tag{45.14}$$

where $\overrightarrow{F_u}$ is the u^{th} flame, b is a constant for defining the shape of the logarithmic spiral, $\overrightarrow{D_l}$ defines the distance between the i^{th} moth $\overrightarrow{X_l}$ and the u^{th} flame $\overrightarrow{F_u}$ ($\overrightarrow{D_l} = |\overrightarrow{F_u} - \overrightarrow{X_l}|$), and $l \in [-1, 1]$ is a random number. The Fitness Function is then calculated for each search agent [11]. Here, the locations update is repeated until the stop conditions are met. In MFO, the exploitation of the best solutions may degrade because of the updating of moths' position regarding to N different locations in the search space. So, a technique is used using Eq. (45.15) [11].

$$F_{num} = round\left(N - z \times \frac{N-1}{iter}\right) \tag{45.15}$$

where F_{num} is number of flames, z is the current number of iterations, and iter indicates the maximum number of iterations. The location and Fitness of the best target are ultimately given to the output as the best approximation of the global optimum. Fig. 45.2 shows the process of generating optimal thresholds in the MTIS problem using the MFO algorithm [11].

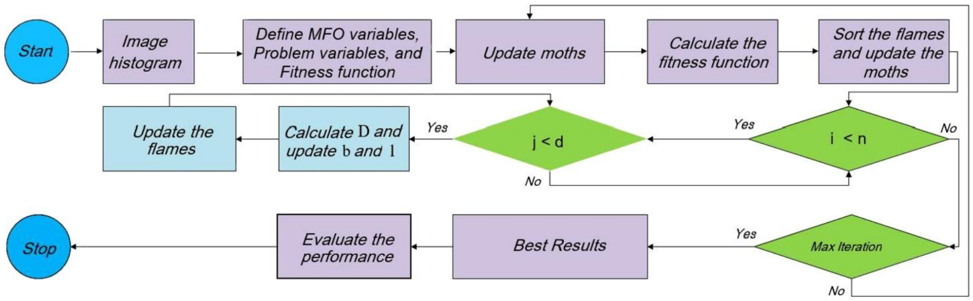


FIGURE 45.2 The structure of the MFO.

45.4 Proposed method

This chapter, uses a combination of two MH algorithms, i.e., WOA and MFO, to improvise the MFO and solve the MTIS problem. In most optimization algorithms, the process consists of two main stages: exploration and exploitation.

Exploration refers to the ability of the algorithm to search the search space globally, in which case the algorithm does not get stuck in the local optimization. Exploitation refers to the ability to discover solutions to improve their quality locally. The better the balance between these two phases of exploration and exploitation, the better the algorithm's performance. WOA is more concentrated in the exploration phase, and MFO is more concentrated in the exploitation phase. If WOA is combined with MFO, it can achieve much better performance. Therefore, in this chapter, we combined the exploitation phase of WOA with exploration phase of MFO and solve the MTIS problem. In MFWOA, the solutions during the exploitation phase are updated using the operators of WOA, and in the exploration phase, only the operators of MFO are used. Then, it computes the quality of each solution according to its Fitness Function value (Eq. (45.4)). Finally, MFWOA stops by reaching the stop condition, and the best solution in the MTIS is the same threshold vector as the final answer or output of the MFWOA. In this chapter, to determine the best threshold vector, the MFWOA algorithm is repeated 100 times on the search space (image histogram). Therefore, at the beginning of optimization, an initial population of solutions is randomly generated using Eq. (45.6). Solutions are distributed over the search space, and then the Fitness Function for the all solution is calculated according to Eq. (45.4). Then, in the search space exploration step, the position of the other solutions is updated based on the metric solution and according to Eq. (45.14) and Eq. (45.15). In the exploitation phase, the solutions are updated using high-powered WOA algorithm operators. In this case, the MFWOA does not get stuck in the local optimization at the beginning of the optimization and achieves a high improvement with the help of WOA algorithm operators in the exploitation phase. Therefore, in this stage, the solutions are updated using Eq. (45.11). The MFWOA output is the same as the optimal threshold vector. Fig. 45.3 shows the flowchart of the MFWOA algorithm for determining the threshold vector in the MTIS process.

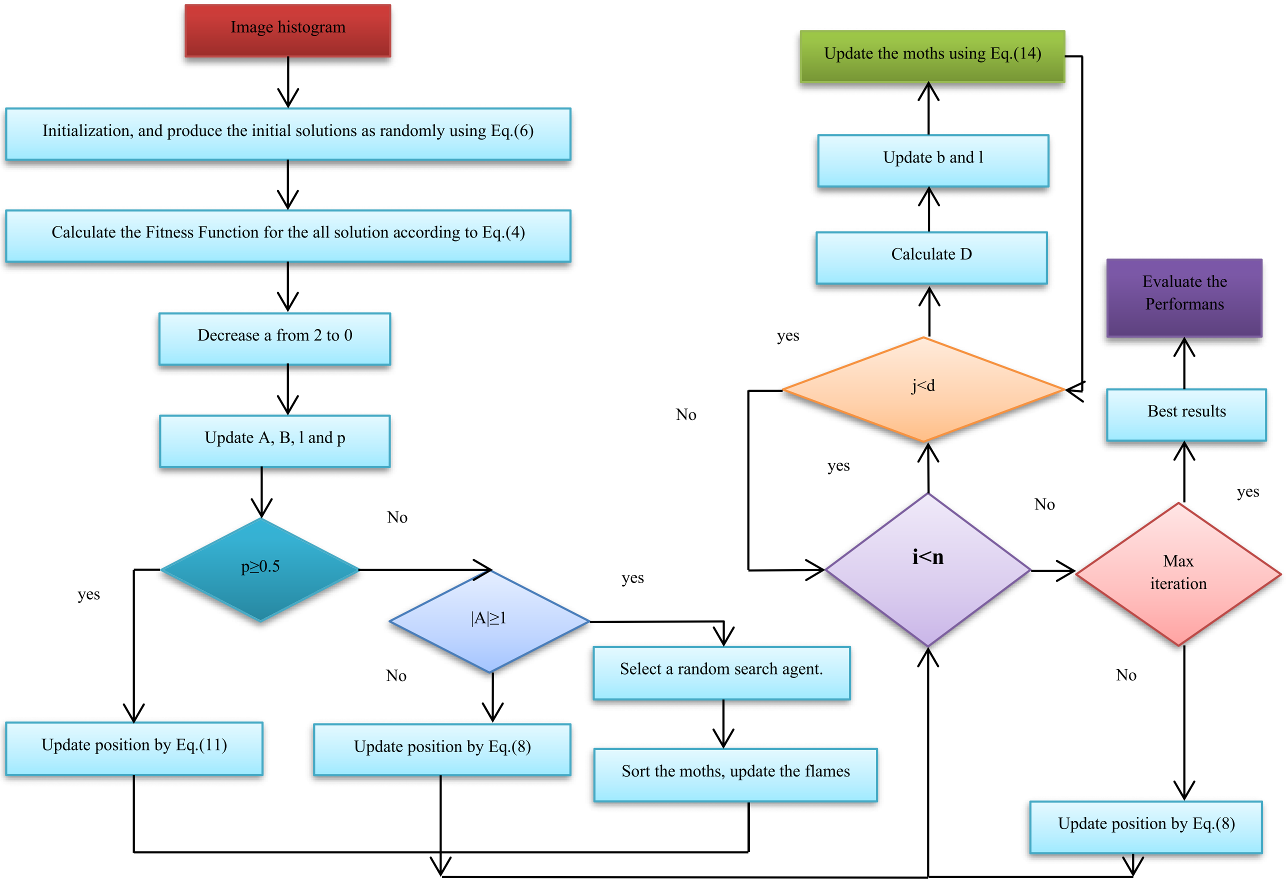


FIGURE 45.3 The proposed MFWOA structure.

45.4.1 Computational complexity of MFWOA

The CC is a field of computational theory that examines the cost of problem-solving process. The CC of MH algorithms is estimated based on the number of search agents, number of problem dimensions, and the maximum number of iterations [22]. The CC of the sorting process for N search agents at the best and worst state is equal to $CC(N \times logN)$ and $CC(N^2)$, respectively [10,11]. The CC of the position updating process in a D-dimensional space is also equal to $CC(N \times D)$. Assuming $It_{max}^{WOA} = It_{max}^{MFO} = T$, and applying an equal number of search agents for the WOA and MFO $\left(N^{WOA} = N^{MFO} = N\right)$. Therefore, the CC of the MFWOA during the first phase $\left(CC^{WOA}\right)$ and the second phase $\left(CC^{MFO}\right)$ optimization process can be defined as [11]:

$$\begin{aligned} O^{WOA} &= CC\left(T \times [CC\left(Sorting\right) + CC\left(position\ update\right)]\right) \\ &= O\left(T \times \left[N^2 + N \times D\right]\right) = O\left(T \times N^2 + T \times N \times D\right), \\ O^{MFO} &= O\left(T \times \left[N^2 + N \times D\right]\right) = O\left(T \times N^2 + T \times N \times D\right) \end{aligned} \tag{45.16}$$

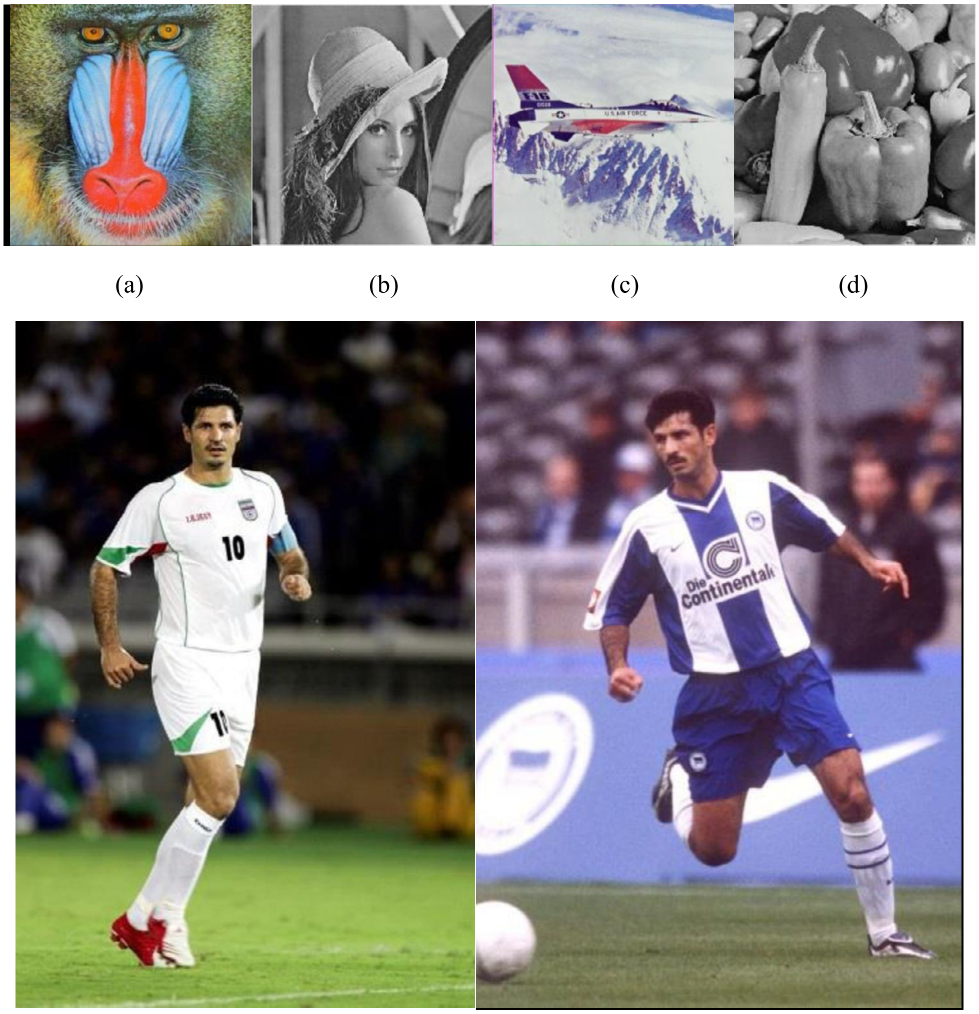
The overall CC of the proposed MFWOA is obtained as;

$$O^{WOA-MFO} = O^{WOA} + O^{MFO} = O\left(T \times N^2 + T \times N \times D\right) \tag{45.17}$$

The CC of all three algorithms (WOA, MFO, MFWOA) is the same. Because all three algorithms (WOA, MFO, MFWOA) have almost the same structure.

45.5 Performance analysis and test results

In this section, various experiments that have been performed on the test images. We took these images from the Berkeley Segmentation Dataset and Benchmark (Fig. 45.4(a)-(d)), and **Ali Daei** images which is Iranian sports legend in the field of football (Fig. 45.4(e), (f)). For the corresponding MH algorithms, 100 search agents look up for the best threshold vector (the best solution) on the search space during 100 iterations. The reason why we have chosen 100 search agents and 100 iterations for the MH algorithms used in this chapter is that the higher the number of population members (search agents) and the number of iterations in the algorithms, these algorithms will achieve more accurate and better answers. The stop condition for any algorithm is to reach the iteration 100th (same conditions for all algorithms: the same Fitness Function, 100 search agents, and 100 iterations). In this case, it is better to calculate the statistical results using ANalysis Of VAriance (ANOVA) or P-Value for the corresponding algorithms and identify the best algorithm in terms of performance.



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

(e) (f)

FIGURE 45.4 Test images in proposed methods. (a) Test 1 (b) Test 2 (c) Test 3 (d) Test 4 (e) Test 5 (f) Test 6.

45.5.1 Evaluation metrics

Like other researches in this field, this chapter uses SSIM, PSNR, processing time, CC, Fitness Function value, threshold values, and statistical test evaluation metrics to evaluate the algorithms and compare the proposed method (MFWOA technique) to similar algorithms. The proposed algorithm and the comparative mechanisms are programmed in "MATLAB® 2018b" and run in a "Windows 7-64bit" environment on a laptop with an Intel Core i4 GHz processor and "6 GB" memory.

45.5.1.1 Peak signal-to-noise ratio (PSNR)

The PSNR evaluation metric is a famous metric used to measure the similarity between the segmented and original images. The amount of PSNR for the image is obtained using Eq. (45.19) and depends on the Mean Squared Error (MSE) value [20].

$$MSE = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} (I_O(m,n) - I_S(m,n)) \tag{45.18}$$

$$PSNR(I_O, I_S) = 10 Log_{10}(\frac{255^2}{MSE}) \tag{45.19}$$

For the image with a size of $m \times n$, $I_O(m,n)$ represents the original image pixels and $I_S(m,n)$ represents the segmented image pixels.

45.5.1.2 Structural similarity index measure (SSIM)

The SSIM is a famous metric in the image segmentation methods that is used to measure the amount of structural similarity between the original image (I_O) and segmented image (I_S). This metric is obtained using Eq. (45.20) [3].

$$SSIM(I_O, I_S) = \frac{(2\mu_1\mu_S + c_1)(2\sigma_{1,S} + c_2)}{(\mu_1^2 + \mu_S^2 + c_1)(\sigma_1^1 + \sigma_S^2 + c_2)} \tag{45.20}$$

Here, μ_1 and μ_S are the mean brightness intensity of the I_O and I_S, respectively. The σ_1 and σ_s, represent the standard deviation of images I_O and I_S images, respectively. The $\sigma_{1,S}$ represents covariance between I_O and I_S images. c_1 and c_2 are two constant values that are 6.50 and 58.52, respectively [3]. The higher the SSIM value in image segmentation methods and the closer it is to 1, the corresponding method is more effective.

45.5.1.3 Processing time

In image segmentation methods, processing time (second) is also one of the essential metrics for evaluating algorithms. In this chapter, we have calculated the processing time of each MH algorithm in 100 iterations.

45.5.1.4 Computational complexity

Computational Complexity (CC) is one of the metrics for evaluating MH algorithms, which can be used to compare algorithms with more accuracy and certainty. Table 45.1 shows the CC for the proposed algorithm and the comparable algorithms.

TABLE 45.1 The Computational complexity.

Algorithm	Computational complexity
HHO [24]	$O(N + T \times N \times D + T \times N)$
EO [25]	$O(T \times N \times D + T \times C \times N)$
MPA [26]	$O(T \times N + T \times N \times D + T \times C)$
WOA [10]	$O(T \times N^2 + T \times N \times D)$
MFO [11]	$O(T \times N^2 + T \times N \times D)$
MFWOA	$O(T \times N^2 + T \times N \times D)$

According to Table 45.1, T, N, C, and D represent the number of iterations, the number of population members, cost of Fitness Function, and the Dimensions size of each population member, respectively. As can be seen from Table 45.2, in this chapter, the CC of the WOA, MFO, EO and MFWOA is the same. MPA and HHO algorithms are also more CC. But in general, the order of CC in all of them is $T \times N^2$.

45.5.1.5 Fitness function value

The Fitness Function, which is essential in all MH algorithms, and its selection is a fundamental principle, is necessary for MTIS methods performed using MH algorithms. In this chapter, the optimal thresholds are obtained using the minimization of the Fitness Function that the exact inverse Otsu Function (Eq. (45.4)).

45.5.1.6 Threshold values

The values of the threshold vector are an essential metric in MTIS. The ultimate image segmentation is done in MTIS methods using the threshold vector. In this chapter, the obtained threshold vector by each algorithm is presented in Table 45.7.

45.5.1.7 Statistical test (P-Value)

As with previous papers in the MTIS field [4,19], in this chapter, to compare the proposed MFWOA method with other MH algorithms, we use the ANOVA or P-value with a significant level of 0.05 [23]. The P-Value for PSNR, SSIM, processing time and Fitness Function is calculated, and then the proposed method is compared with other algorithms. Similar to previous papers [4,19], there are two hypotheses of zero and alternatives. According to the hypothesis of zero, there should be no significant difference between the mean values of the compared algorithms and the MFWOA algorithm (P-Value should be more than 0.05). However, according to the alternative hypothesis, there should be a significant difference between the proposed MFWOA method and comparative algorithms (i.e., P-Value should be less than 0.05).

45.5.2 The results and discussions

In this section, the results of the proposed MFWOA method and comparable algorithms are thoroughly examined using evaluation metrics. In Table 45.2, the list of constant parameters used in each MH algorithm with their numerical values is recorded.

TABLE 45.2 The constant parameters of each algorithm and their values.

Algorithm	Parameters	Value
WOA [10]	A	[0, 2]
	B	1
	L	[−1, 1]
MFO [11]	B	1
	L	[−1, 1]
EO [25]	V	1
	a_1	2
	a_2	1
	GP	0.5
HHO [24] MPA [26]	E_0	[−1, 1]
	FADs	0.2
	P	0.5

We compare our proposed algorithm with WOA [10], MFO [11], HHO [24], Equilibrium Optimizer (EO) [25], and Marine Predators Algorithm (MPA) [26] algorithms. The reason for choosing the EO, MPA, and HHO algorithms to compare with our work is that these three algorithms are strong and new algorithms. So we chose them to compare with our work to show the superiority of our proposed algorithm over them. We tested our proposed MFWOA algorithm and others for different threshold levels of k ($k = 2, 3, 4, 5, 6, 7, 8, 9, 10, 16, 32$) on the eight images, as it can be seen in Fig. 45.4, to be able to make more accurate evaluations and comparisons using the relevant evaluation metrics. Table 45.3 shows the value of the Fitness Function obtained from the proposed MFWOA algorithm and other algorithms, for different thresholds for all test images, during 100 iterations of the algorithms. We introduce the maximum and minimum values of the Fitness Function by each algorithm at each threshold level for some images, which can be seen in Table 45.3. This chapter obtains

TABLE 45.3 Value of Fitness Function for different threshold levels during 100 runs for different images.

	k	Optimization Algorithms							k	Optimization Algorithms					
		HHO	EO	WOA	MPA	MFO	MFWOA			HHO	EO	WOA	MPA	MFO	MFWOA
Test 1	2	1559.8	1559.8	1559.9	1559.8	1559.8	1559.8	Test 4	2	2399.9	2399.9	2399.1	2399.9	2399.9	2297.8
	3	1990.8	2021	2014.3	2021	2021	**1990.5**		3	3154.6	3211	3209.9	3211	3211	3010.9
	4	2313.6	2443.3	2315	2455.5	2455.5	**2313**		4	3569.2	4008.9	3565	4044.1	4044.1	3443.3
	5	2784	2794.5	2771.5	2810	2784.1	2773.4		5	4295.5	4406.7	4422.1	4426.1	4406.4	4113.1
	6	3268.1	3239.1	3207	3244.5	3244.5	3053.7		6	4362.1	5190.6	4722.7	5185.2	5217.4	4119.1
	7	3515.7	3624.7	3531.6	3677	3599	3488.1		7	4526.3	5768.5	6040	5543.1	6050.7	4489.7
	8	3947.8	3992	3938.6	4033.5	4005.6	3930.1		8	5616.9	6392.3	6339.1	6326.3	6390.7	5545.4
	9	4373.6	4396.7	4339	4465.9	4433.7	4264.8		9	6179	7031	6355	7191.8	7224.1	6372.3
	10	3669.3	4813.7	4279.5	4898.4	4334.1	3576.6		10	6384.5	7569.8	6848.3	7443.4	6507.8	6355.1
	16	5915.4	7155.2	5609.6	7265.4	7263.4	5630.1		16	9344	11339	8746.1	11533	10343	8565.9
	32	10770	13326	10324	12680	12036	8482.3		32	15213	21019	13420	20175	19148	12483
Test 2	2	1952.3	1952.3	1951.8	1952.3	1952.3	1950.4	Test 5	2	2510	2510	2510	2510	2510	2510
	3	2600.6	2600.6	2520.7	2600.6	2600.6	2522.5		3	3518.9	3518.3	3518.3	3518.3	3518.3	3518.3
	4	2947.2	3158.1	3173	3174	3174	2929.7		4	3538.4	4060.8	3678.5	4102.3	3552.3	3578.1
	5	3565.1	3579.8	3563.4	3600.7	3600.7	3431.2		5	4910.8	4895.8	4967	4987.1	4555.6	4533.4
	6	3792.8	4122.4	3742.3	4174.2	4170.1	3749.2		6	4199.8	5547	5398.4	5561.5	5571.2	40118
	7	4413.1	4609	4061	4600.8	4519.8	4431.4		7	5254.1	6196.1	6456	6145.6	6024.5	5079.4
	8	4583.4	5094.2	4605.6	5174.3	5091	4454.8		8	4648.8	6841.5	6763	7030.4	7030.4	4111.3
	9	4926	5603.4	5596.9	5601	5358.9	4871		9	7560.4	7527.5	7331.9	7924.9	7604.7	6111.3
	10	4811	6048.9	5349.1	6093.4	6091.2	4450.7		10	8176	8285.6	8024.5	8387	8066.1	8050.4
	16	6669.4	9040.4	7514.6	9170.8	8924.3	6385.9		16	7306.4	12145	10091	10353	11900	6716
	32	11729	16658	12333	15563	14832	10257		32	16636	21332	17154	20705	17171	16237
Test 3	2	1962.1	1962.1	1961.7	1962.1	1962.1	1961.1	Test 6	2	5686.4	5686.4	5686.2	5686.4	5686.4	5685.8
	3	3176.1	3176.1	3176.1	3176.1	3176.1	3174.6		3	8068.4	8117.4	8120	8121	8121	8120.1
	4	3331.2	3549.2	3330.6	3607.8	3607.8	3207.8		4	8522.9	10507	8522.4	10507	85507	8522.2
	5	3758.2	4537.6	3757.4	4537.6	4537.6	3600.3		5	10859	10936	10884	10961	10902	10907
	6	4936.6	4940.3	4953.4	4969.3	4969.2	4944.8		6	11263	13251	11350	13347	13347	10328
	7	5897.3	5851.6	5104	5899	5400.9	5023.1		7	13655	15006	13725	15734	13708	12784
	8	5897.4	6264.3	6244.9	6055.6	6330.7	5706.9		8	15567	16145	15087	16187	15187	15167
	9	6091.4	7005	6403.1	7260.5	7260.5	5818.6		9	16435	17983	16540	18573	16552	16487
	10	7338.1	7654.3	7247.4	7692.2	7692.1	6521		10	14284	15385	14442	16960	14974	13875
	16	10138	11471	9951.5	11003	11003	8578.7		16	25440	28772	25925	31412	25174	25191
	32	16967	21469	16748	19655	21819	13493		32	39047	53566	36612	50999	39423	36698

the proposed MFWOA and other algorithms by minimizing the Fitness Function. Therefore, according to Table 45.3, if each algorithm's obtained Fitness Function value is lower, the corresponding algorithm performs better.

Also, as shown in Table 45.3, by increasing the value of k, the value of the obtained Fitness Function by all algorithms for Test3 and Test5 increases. For the Test1 image, the value of the obtained Fitness Function by HHO, WOA, and MFWOA decreases at $k = 10$ and increases with an increasing value of k. For the Test2 image, the value of the obtained Fitness Function by MFWOA decreases at $k = 10$ and increases with the value of k. For the Test4 image, the value of the obtained Fitness Function by the MFO decreases at $k = 10$ and then increases as the value of k increases. For the Test6 image, the value of the obtained Fitness Function by all algorithms decreases at $k = 10$ and then increases as the value of k increases. As per the results in Table 45.3, in most cases, the proposed MFWOA algorithm has a lower Fitness Function value than other algorithms. If it is higher, it does not differ much from different algorithms. It does not reduce the PSNR and SSIM values. Therefore, the proposed MFWOA algorithm has the necessary efficiency. It can be said that the proposed algorithm has better performance than other algorithms and can achieve the most suitable thresholds.

For Test1 image:

At the level $k = 2$, all algorithms have the same value. When $k = 3$, the lowest Fitness Function values are related to MFWOA, HHO, WOA, and other algorithms with the same value. Considering $k = 4$, the lowest Fitness Function values are related to HHO, WOA, MFWOA, EO, and MPA=MFO, respectively. At the $k = 5$, the lowest values of the Fitness Function are related to WOA, WOA=MFWOA, HHO=MFO, MPA, and EO, respectively. At the $k = 6$, the lowest values of the Fitness Function are associated with MFWOA, WOA, EO, MFO=MPA, and HHO, respectively. In the case of $k = 7$, the lowest values of the Fitness Function are related to MFWOA, WOA, HHO, MFO, EO, MFO, and MPA, respectively. Assuming $k = 8$, the lowest values of the Fitness Function are related to MFWOA, WOA, HHO, EO, MFO, and MPA, respectively. At the $k = 9$, the lowest values of the Fitness Function are related to MFWOA, WOA, HHO, EO, and MPA=MFO, respectively. At the $k = 10$, the lowest values of the Fitness Function are related to MFWOA, WOA, HHO, EO, MFO, and MPA, respectively. At the $k = 16$, the lowest values of the Fitness Function are related to MFWOA, WOA, HHO, EO, MFO, and MPA, respectively. At the $k = 32$, the lowest values of the Fitness Function are related to MFWOA, WOA, HHO, MFO, MPA, and EO, respectively.

For Test2 image:

When $k = 2$, the lowest values of the Fitness Function are related to MFWOA, WOA, and other algorithms that have the same value. At the $k = 3$, the lowest Fitness Function values are associated with MFWOA, WOA, and other algorithms with the same value. At the $k = 4$, the lowest Fitness Function values are related to MFWOA, HHO, EO, WOA, MPA=MFO, respectively. When $k = 5$, the lowest values of the Fitness Function are associated with MFWOA, WOA, HHO, EO, and MPA=MFO, respectively. At the $k = 6$, the lowest values of the Fitness Function are related to MFWOA, WOA=HHO, EO, MPA, and MFO, respectively. At the $k = 7$, the lowest values of the Fitness Function are associated with MFWOA, WOA, HHO, EO, MFO, and MPA, respectively. At the $k = 8$, the lowest values of the Fitness Function are related to MFWOA, HHO, WOA, MFO, EO, and MPA, respectively. At the $k = 9$, the lowest values of the Fitness Function are related to MFWOA, HHO, MFO, WOA, MPA, and EO, respectively. At the $k = 10$, the lowest values of the Fitness Function are related to MFWOA, HHO, WOA, EO, and MPA=MFO, respectively. At the $k = 16$, the lowest values of the Fitness Function are related to MFWOA, HHO, WOA, EO, MFO, and MPA, respectively. At the $k = 32$, the lowest values of the Fitness Function are related to MFWOA, HHO, WOA, MFO, MPA, and EO, respectively.

For Test3 image:

When $k = 2$, the lowest values of the Fitness Function are related to MFWOA, WOA, and other algorithms that have the same value. At the $k = 3$, all algorithms have the same value. At the $k = 4$, the lowest Fitness Function values are related to MFWOA, WOA, HHO, EO, MPA=MFO, respectively. When $k = 5$, the lowest values of the Fitness Function are associated with MFWOA, WOA, HHO, and MFO=MPA=EO, respectively. At the $k = 6$, the lowest values of the Fitness Function are related to HHO, EO, WOA, MFWOA, and MPA=MFO, respectively. At the $k = 7$, the lowest values of the Fitness Function are associated with MFWOA, WOA, MFO, EO, HHO, and MPA, respectively. At the $k = 8$, the lowest values of the Fitness Function are related to MFWOA, HHO, MPA, WOA, EO, and MFO, respectively. At the $k = 9$, the lowest values of the Fitness Function are related to MFWOA, HHO, WOA, EO, and MPA=MFO, respectively. At the $k = 10$, the lowest values of the Fitness Function are related to MFWOA, WOA, HHO, EO, and MPA=MFO, respectively. At the $k = 16$, the lowest values of the Fitness Function are related to MFWOA, WOA, HHO, MFO=MPA, and EO, respectively. At the $k = 32$, the lowest values of the Fitness Function are associated with MFWOA, WOA, HHO, MPA, EO, and MFO, respectively.

For Test4 image:

When $k = 2$, the lowest values of the Fitness Function are related to MFWOA, and other algorithms have the same value. At the $k = 3$, the lowest Fitness Function values are associated with MFWOA, HHO, WOA, and other algorithms with the same value. At the $k = 4$, the lowest Fitness Function values are related to MFWOA, WOA, HHO, EO, MPA=MFO, respectively. At the $k = 5$, the lowest values of the Fitness Function are associated with MFWOA, HHO, EO=MFO, WOA, and MPA, respectively. At the $k = 6$, the lowest values of the Fitness Function are related to MFWOA, WOA=HHO, EO, MPA, and MFO, respectively. At the $k = 7$, the lowest values of the Fitness Function are associated with MFWOA, HHO, MPA, EO, WOA, and MFO, respectively. At the $k = 8$, the lowest values of the Fitness Function are related to MFWOA, HHO, MPA, WOA, MFO, and EO, respectively. At the $k = 9$, the lowest values of the Fitness Function are related to HHO, WOA, MFWOA, EO, MPA, and MFO, respectively. At the $k = 10$, the lowest values of the Fitness Function are related to MFWOA, HHO, MFO, WOA, MPA, and EO, respectively. At the $k = 16$, the lowest values of the Fitness Function

TABLE 45.4 Value of PSNR for different threshold levels during 100 runs for different images.

	k	Optimization Algorithms							k	Optimization Algorithms					
		HHO	EO	WOA	MPA	MFO	MFWOA			HHO	EO	WOA	MPA	MFO	MFWOA
Test 1	2	24.3	24.3	24.298	24.3	24.3	24.304	Test 4	2	22.824	22.824	22.814	22.824	22.824	22.838
	3	21.773	22.9	22.401	22.9	22.9	22.948		3	22.049	20.55	20.317	20.55	20.55	22.582
	4	22.947	24.512	24.712	20.658	20.658	24.966		4	23.304	19.604	23.259	19.604	19.604	23.655
	5	24.339	22.9	25.988	22.9	25.423	25.989		5	22.096	23.578	23.337	23.579	23.524	23.972
	6	26.711	23.036	22.524	23.036	23.036	26.989		6	20.332	21.946	24.364	21.946	20.557	24.365
	7	27.306	20.658	26.151	20.658	22.9	27.932		7	23.99	19.604	19.981	23.887	19.605	23.998
	8	27.523	25.471	22.624	23.036	25.463	27.705		8	23.374	23.906	20.847	21.947	20.558	24.447
	9	22.253	23.036	24.003	23.036	21.897	24.585		9	22.289	20.55	21.128	21.947	20.558	24.925
	10	28.636	22.901	26.547	20.658	25.501	28.66		10	21.129	24.712	25.983	22.087	23.906	25.991
	16	22.267	26.662	25.341	23.036	22.901	26.677		16	24.034	23.955	21.198	23.398	21.168	26.176
	32	30.528	25.834	20.282	26.114	30.213	31.533		32	26.663	25.203	26.978	25.516	27.964	27.973
Test 2	2	22.973	22.973	22.966	22.973	22.973	22.986	Test 5	2	21.47	21.47	21.47	21.466	21.47	21.476
	3	21.229	21.229	22.373	21.229	21.229	22.483		3	19.323	19.636	19.636	19.636	19.756	19.836
	4	23.323	19.917	19.93	19.917	19.917	22.937		4	21.894	18.648	22.189	18.648	18.648	22.264
	5	20.996	21.233	21.816	21.254	21.233	22.663		5	18.668	19.636	20.313	19.636	22.753	22.805
	6	21.277	23.259	24.825	21.244	21.234	24.833		6	20.549	19.87	22.506	19.853	19.853	22.589
	7	23.423	24.632	22.37	21.233	24.632	25.097		7	22.186	22.791	19.636	18.899	22.791	22.861
	8	21.343	22.11	23.158	21.244	24.373	25.383		8	22.919	22.753	21.798	19.853	19.87	24.326
	9	22.848	24.678	23.35	21.233	24.802	24.883		9	22.445	22.791	22.477	19.636	19.979	22.847
	10	23.278	24.538	27.019	24.64	25.241	27.934		10	21.395	21.705	21.717	21.892	22.667	22.728
	16	20.895	23.298	20.019	21.26	25.94	25.984		16	20.27	23.339	25.026	25.163	22.574	25.288
	32	21.653	25.983	20.547	27.303	25.748	27.759		32	19.132	23.339	18.182	26.08	26.105	27.856
Test 3	2	24.985	24.985	24.972	24.985	24.985	24.985	Test 6	2	22.16	22.17	22.167	22.16	22.16	22.167
	3	23.304	23.304	23.304	23.304	23.304	23.304		3	20.436	20.426	20.265	20.426	20.426	20.45
	4	25.851	22.544	25.799	22.544	22.544	25.801		4	22.278	19.17	22.218	19.17	19.17	22.343
	5	25.332	23.304	26.124	23.304	23.304	26.235		5	21.02	23.072	23.564	20.426	23.123	23.583
	6	25.033	23.304	23.271	23.304	23.304	24.143		6	23.053	20.894	22.107	20.508	20.508	23.543
	7	23.33	23.304	25.431	23.304	22.544	25.485		7	20.933	19.17	23.478	19.17	23.276	23.695
	8	24.459	25.831	24.406	25.831	23.304	25.989		8	23.61	23.178	21.927	20.508	20.508	23.705
	9	25.536	23.304	25.388	23.304	23.304	25.706		9	21.055	20.508	20.06	20.508	23.322	23.399
	10	24.659	23.304	25.513	23.304	23.304	25.64		10	24.453	20.508	23.306	19.17	23.072	24.799
	16	25.468	23.304	23.254	25.978	25.975	28.053		16	22.073	20.932	22.785	19.17	22.615	22.712
	32	25.194	25.975	26.725	27.197	25.011	30.676		32	21.579	23.136	24.572	25.315	25.112	25.337

are related to MFWOA, WOA, HHO, MFO, EO, and MPA, respectively. At the $k = 32$, the lowest values of the Fitness Function are related to MFWOA, WOA, HHO, MFO, MPA, and EO, respectively.

For Test5 image:

When $k = 2$ and 3, all algorithms have the same value. At the $k = 4$, the lowest Fitness Function values are related to HHO, MFO, MFWOA, WOA, EO, and MPA, respectively. At the $k = 5$, the lowest values of the Fitness Function are related to MFWOA, MFO, EO, HHO, MPA, and WOA, respectively. At the $k = 6$, the lowest values of the Fitness Function are related to MFWOA, HHO, WOA, EO, MPA, and MFO, respectively. At the $k = 7$, the lowest values of the Fitness Function are related to MFWOA, HHO, MFO, MPA, WOA, and EO, respectively. At the $k = 8$, the lowest values of the Fitness Function are related to MFWOA, HHO, WOA, EO, and MFO=MPA, respectively. At the $k = 9$, the lowest values of the Fitness Function are associated with MFWOA, WOA, EO, HHO, MFO, and MPA, respectively. At the $k = 10$, the lowest values of the Fitness Function are related to WOA, MFWOA, MFO, HHO, EO, and MPA, respectively. At the $k = 16$,

TABLE 45.5 Value of SSIM for different threshold levels during 100 runs for different images.

	k	Optimization Algorithms							k	Optimization Algorithms					
		HHO	EO	WOA	MPA	MFO	MFWOA			HHO	EO	WOA	MPA	MFO	MFWOA
Test 1	2	0.75532	0.75532	0.75529	0.75532	0.75532	0.75602	Test 4	2	0.70335	0.70335	0.70209	0.70335	0.70335	0.70713
	3	0.6952	0.7155	0.69704	0.7155	0.7155	0.71644		3	0.6951	0.6794	0.67302	0.6794	0.6794	0.69895
	4	0.71683	0.76391	0.77424	0.62461	0.62461	0.7752		4	0.71472	0.64733	0.71068	0.64733	0.64733	0.72004
	5	0.75717	0.7155	0.81393	0.7155	0.80591	0.8173		5	0.70014	0.73655	0.72717	0.73654	0.73519	0.73579
	6	0.83815	0.72179	0.7233	0.72179	0.72179	0.83815		6	0.68101	0.69013	0.75031	0.69013	0.68094	0.75339
	7	0.83708	0.62461	0.83132	0.62461	0.7155	0.83986		7	0.77418	0.64733	0.66918	0.7516	0.64733	0.77696
	8	0.85276	0.81067	0.73891	0.72179	0.80813	0.85832		8	0.75212	0.7353	0.7021	0.69013	0.68095	0.75525
	9	0.69315	0.72179	0.77298	0.72179	0.70013	0.7746		9	0.71801	0.6794	0.70808	0.69013	0.68095	0.7769
	10	0.87498	0.7155	0.84179	0.62461	0.81031	0.87866		10	0.72621	0.76047	0.79503	0.69895	0.7353	0.79891
	16	0.69532	0.85805	0.82013	0.72179	0.7155	0.85611		16	0.79158	0.75352	0.72806	0.73423	0.72726	0.79323
	32	0.90707	0.82056	0.60426	0.82678	0.90384	0.93129		32	0.85374	0.77731	0.83593	0.79718	0.82723	0.85744
Test 2	2	0.68017	0.68017	0.67086	0.68017	0.68017	0.68151	Test 5	2	0.63566	0.63566	0.63566	0.6317	0.63566	0.6357
	3	0.69931	0.69931	0.68173	0.69931	0.69931	0.69942		3	0.51512	0.5289	0.5289	0.5289	0.53344	0.5389
	4	0.72248	0.64058	0.64053	0.64058	0.64058	0.72825		4	0.65709	0.48206	0.66658	0.48206	0.48206	0.66762
	5	0.70998	0.70009	0.71525	0.70132	0.70009	0.71595		5	0.4673	0.5289	0.56448	0.5289	0.7079	0.70855
	6	0.70653	0.70328	0.73991	0.70207	0.69942	0.73993		6	0.70239	0.53843	0.67887	0.53565	0.53565	0.67204
	7	0.70042	0.75025	0.69395	0.70009	0.75025	0.77269		7	0.67159	0.71211	0.5289	0.49342	0.71211	0.71629
	8	0.72172	0.67525	0.72049	0.70207	0.72391	0.75438		8	0.76008	0.7079	0.64822	0.53565	0.53843	0.76131
	9	0.74742	0.75157	0.70676	0.70009	0.74343	0.75839		9	0.71149	0.71211	0.66499	0.5289	0.54019	0.71369
	10	0.76407	0.72984	0.79361	0.75254	0.75629	0.79414		10	0.72801	0.64803	0.63847	0.65468	0.70859	0.7287
	16	0.70873	0.70452	0.67308	0.70143	0.78095	0.78137		16	0.7162	0.7357	0.78329	0.78231	0.70222	0.78444
	32	0.76002	0.82033	0.69726	0.80164	0.80385	0.82164		32	0.49859	0.7357	0.43112	0.82113	0.81574	0.8539
Test 3	2	0.82627	0.82627	0.82462	0.82627	0.82627	0.82628	Test 6	2	0.6727	0.6727	0.6723	0.6727	0.6727	0.67288
	3	0.80202	0.80202	0.80202	0.80202	0.80202	0.80206		3	0.65881	0.60842	0.60297	0.60842	0.60842	0.65907
	4	0.85073	0.78395	0.85124	0.78395	0.78395	0.85575		4	0.67564	0.57719	0.67188	0.57719	0.57719	0.67628
	5	0.83639	0.80202	0.85251	0.80202	0.80202	0.85577		5	0.66585	0.70198	0.71132	0.60842	0.70633	0.71231
	6	0.83252	0.80202	0.80241	0.80202	0.80202	0.83961		6	0.71637	0.6588	0.67332	0.61308	0.61308	0.71783
	7	0.80306	0.80202	0.83654	0.80202	0.78395	0.83781		7	0.66091	0.57719	0.71563	0.57719	0.71419	0.71651
	8	0.81083	0.85136	0.82185	0.85136	0.82202	0.85823		8	0.71285	0.71004	0.66783	0.61308	0.61308	0.71508
	9	0.84288	0.80202	0.82262	0.80202	0.80202	0.84371		9	0.67035	0.61308	0.60412	0.61308	0.71404	0.71547
	10	0.81873	0.80202	0.85379	0.80202	0.80202	0.85471		10	0.78293	0.61308	0.6754	0.57719	0.70198	0.78997
	16	0.86899	0.80202	0.80049	0.85863	0.85837	0.8727		16	0.74956	0.6608	0.69741	0.57719	0.76925	0.76984
	32	0.83703	0.85837	0.83454	0.87201	0.83475	0.9134		32	0.69714	0.70413	0.72049	0.72909	0.72309	0.75979

the lowest values of the Fitness Function are related to MFWOA, HHO, WOA, MPA, MFO, and EO, respectively. At the $k = 32$, the lowest values of the Fitness Function are related to MFWOA, HHO, WOA, MFO, MPA, and EO, respectively.

For Test6 image:

At the level $k = 2$, the lowest values of the Fitness Function are related to MFWOA, and other algorithms have the same value. At the $k = 3$, the lowest Fitness Function values are associated with HHO, WOA=MFWOA, EO, and other algorithms have the same value. At the $k = 4$, the lowest Fitness Function values are related to HHO=WOA=MFWOA and MPA=EO=MFO, respectively. At the $k = 5$, the lowest values of the Fitness Function are associated with HHO, WOA, MFO, MFWOA, EO, and MPA, respectively. At the $k = 6$, the lowest values of the Fitness Function are related to HHO, WOA, MFWOA, EO, and MFO=MPA, respectively. At the $k = 7$, the lowest values of the Fitness Function are associated with MFWOA, HHO, MFO, WOA, EO, and MPA, respectively. When $k = 8$, the lowest values of the Fitness Function are related to WOA, MFWOA, MFO, HHO, EO, and MPA, respectively. At the $k = 9$, the lowest values of the Fitness Function

TABLE 45.6 Value of execution time for different threshold levels during 100 runs for different image.

	k	HHO	EO	WOA	MPA	MFO	MFWOA		k	HHO	EO	WOA	MPA	MFO	MFWOA
		Optimization Algorithm								Optimization Algorithm					
Test 1	2	2.5885	35.545	1.482	2.3924	3.0818	2.3628	Test 4	2	2.5304	35.087	1.3939	2.5094	2.986	2.2661
	3	5.8835	49.898	4.497	5.9809	6.4076	5.3508		3	12.838	151.02	8.1752	8.4191	14.121	9.54
	4	6.7822	66.293	5.2668	7.0125	7.3707	6.2322		4	14.488	189.58	12.999	14.736	19.804	13.94
	5	7.4725	65.063	5.9615	7.9977	8.1565	6.9915		5	12.578	224.55	10.74	12.444	12.82	11.756
	6	8.3069	73.484	6.5406	15.31	16.713	7.8798		6	8.5955	71.098	6.4763	8.5368	8.9628	7.6614
	7	17.237	246.06	15.169	26.738	28.324	16.516		7	9.5911	76.536	7.4286	20.116	13.905	8.7317
	8	20.89	209.33	18.593	33.58	36.424	27.621		8	14.203	88.523	11.925	14.429	14.888	13.436
	9	26.945	113.04	23.851	26.723	27.196	25.491		9	11.537	91.224	8.93	11.604	11.987	10.517
	10	12.246	102.65	9.697	12.676	13.144	11.475		10	12.783	97.781	9.7056	12.727	13.228	11.422
	16	16.357	143.99	11.514	15.888	16.942	14.11		16	15.583	142.23	12.028	16.086	16.847	14.634
	32	26.21	323.04	19.098	26.871	28.344	23.974		32	25.36	251.9	19.156	27.068	27.055	24.158
Test 2	2	2.549	128.42	1.451	2.4985	6.6195	2.1516	Test 5	2	2.6203	57.697	1.391	2.4971	3.2826	2.228
	3	9.6909	186.56	8.2486	9.5985	9.9906	9.0861		3	6.2835	50.61	4.8264	6.2362	6.6442	5.7335
	4	6.543	57.122	5.0665	6.7625	7.3037	6.0152		4	6.7514	56.324	5.1596	13.529	17.062	6.1574
	5	7.9833	66.109	6.0976	7.9914	8.3577	7.1817		5	17.838	207.19	15.94	17.85	18.19	17.125
	6	8.8229	72.188	6.6915	8.6627	9.2596	7.903		6	8.9051	72.104	6.6849	8.8714	9.3535	7.9926
	7	9.8139	79.02	7.5362	19.877	14.386	8.8675		7	9.9042	101.6	7.671	10.287	10.415	9.1771
	8	14.869	88.98	12.273	14.719	15.003	13.77		8	10.791	127.49	8.2182	10.931	11.497	9.8845
	9	26.28	92.145	17.342	27.478	28.636	22.641		9	12.317	95.167	9.2849	12.138	12.741	11.044
	10	29.133	267.31	26.483	29.127	29.663	28.062		10	13.167	259.31	10.039	12.981	13.848	11.913
	16	15.299	136.88	11.539	15.586	15.958	13.82		16	17.257	347.67	12.222	16.544	17.589	14.839
	32	25.04	246.95	18.619	25.772	75.463	22.633		32	27.963	261.04	19.505	30.369	28.826	25.996
Test 3	2	3.6559	71.727	3.3928	8.0961	9.1527	6.1958	Test 6	2	2.6428	62.506	1.4912	2.3882	3.1865	2.2949
	3	12.349	122.93	10.851	18.88	15.85	14.015		3	6.3289	49.805	4.6819	6.0926	6.6972	5.5972
	4	13.928	60.724	8.8811	10.274	13.509	9.4412		4	7.1452	59.161	5.6421	7.4866	8.4836	6.6062
	5	10.871	89.487	9.2188	11.257	11.626	10.337		5	21.886	67.944	9.7285	11.672	17.889	11.621
	6	8.8765	75.884	6.9928	9.0694	9.3399	8.1975		6	28.705	98.079	22.445	27.226	29.218	26.46
	7	9.5359	80.774	7.3131	9.7526	9.8271	8.5936		7	21.654	134.25	18.968	21.65	21.815	20.477
	8	10.41	85.279	8.0442	10.447	10.864	9.361		8	11.129	118.45	8.424	11.156	11.659	10.145
	9	11.26	94.081	8.8589	11.642	25.777	10.467		9	12.908	101.33	9.5723	12.67	13.173	11.612
	10	26.129	268.21	23.428	26.423	37.202	25.088		10	13.316	126.21	10.485	13.549	14.324	12.261
	16	26.046	322.93	22.119	26.311	26.775	24.621		16	23.844	164.18	12.349	21.124	37.755	15.169
	32	26.014	477.14	18.658	26.333	27.581	23.482		32	48.453	316.88	39.833	48.312	49.638	45.08

are related to HHO, MFWOA, WOA, MFO, EO, and MPA, respectively. At the $k = 10$, the lowest values of the Fitness Function are related to MFWOA, HHO, WOA, MFO, EO, and MPA, respectively. Considering $k = 16$, the lowest values of the Fitness Function are related to MFO, MFWOA, HHO, WOA, EO, and MPA, respectively. At the $k = 32$, the lowest values of the Fitness Function are related to WOA, MFO, MFWOA, HHO, MPA, and EO, respectively.

Table 45.4 shows the PSNR values of the segmented image from the obtained optimal thresholds by each MH algorithm and the proposed MFWOA algorithm for different photos at different threshold levels. Inspecting the results in Table 45.4, the proposed MFWOA method has achieved the desired quality for the segmented images at different threshold levels. The proposed MFWOA method has a higher PSNR value than the MFO, WOA, HHO, EO, and MPA algorithms for all threshold levels and relevant images. According to Table 45.4, the value of the PSNR decreases by each algorithm, with increasing k and increasing at other levels.

Table 45.5 shows the value of SSIM for different images and different threshold levels after 100 runs. According to Table 45.5, the proposed method has a higher SSIM value for all images than the existing algorithms compared in

TABLE 45.7 The value of thresholds during 100 runs for different images.

Image	k	EO	WOA	HHO	MPA	MFO	MFWOA
Test 1	2	99, 159	86, 149	99, 156	98, 159	98, 159	94, 155
	3	29, 90, 117	29, 57, 117	29, 90, 116	29, 90, 117	29, 90, 116	29, 79, 117
	4	29, 116, 117, 236	29, 85, 119, 236	29, 90, 102, 159	29, 29, 90, 117	29, 109, 119, 218	29, 68, 103, 170
	5	29, 29, 90, 92, 117	29, 29, 59, 59, 133	29, 29, 92, 107, 109	29, 90, 111, 151, 231	29, 29, 69, 81, 104	29, 49, 90, 108, 150
Test 2	2	76, 140	76, 140	76, 140	76, 140	76, 140	76, 140
	3	125, 153, 253	118, 153, 254	4, 85, 127	4, 88, 126	127, 153, 254	42, 108, 169
	4	4, 125, 126, 254	24, 132, 149, 226	4, 125, 127, 252	4, 84, 85, 142	85, 147, 160, 254	37, 121, 131, 207
	5	4, 75, 125, 140, 253	4, 46, 116, 128, 254	4, 116, 130, 167, 225	4, 4, 94, 100, 118	109, 142, 144, 207, 222	4, 79, 123, 145, 244
Test 3	2	93, 158	93, 159	92, 157	91, 159	92, 156	92, 157
	3	129, 153, 255	95, 147, 249	1, 89, 125	1, 88, 128	127, 154, 255	32, 108, 167
	4	1, 129, 130, 255	1, 112, 138, 255	1, 51, 98, 128	1, 87, 94, 153	90, 137, 141, 254	30, 91, 111, 178
	5	128, 153, 155, 255, 255	129, 165, 166, 255, 255	1, 51, 87, 109, 167	1, 13, 87, 87, 130	138, 156, 170, 255, 255	43, 76, 113, 150, 184
Test 4	2	48, 125	46, 126	48, 125	48, 125	48, 125	47, 125
	3	104, 132, 255	74, 138, 255	1, 56, 100	1, 56, 100	107, 135, 255	25, 83, 151
	4	1, 100, 104, 255	1, 59, 108, 253	98, 141, 255, 255	107, 112, 194, 255	51, 105, 118, 255	53, 92, 140, 254
	5	104, 132, 137, 255, 255	1, 17, 107, 128, 255	1, 1, 53, 57, 100	1, 4, 7, 61, 99	82, 103, 130, 251, 254	1, 7, 55, 82, 151
Test 5	2	90, 168	82, 164	90, 167	89, 169	90, 168	87, 166
	3	1, 68, 97	1, 43, 99	1, 68, 98	1, 72, 94	90, 128, 255	30, 79, 150
	4	1, 97, 98, 255	1, 96, 98, 255	1, 38, 79, 97	1, 43, 82, 97	1, 91, 94, 196	1, 57, 85, 130
	5	1, 1, 68, 68, 97	1, 5, 97, 100, 253	1, 64, 101, 116, 255	1, 1, 70, 70, 95	1, 54, 105, 107, 232	1, 39, 91, 97, 194
Test 6	2	0.59534	3.7494	1.3487	0.80151	1.247	119, 177
	3	16, 139, 154	16, 141, 156	16, 139, 154	16, 139, 154	16, 139, 154	16, 139, 154
	4	16, 154, 154, 231	16, 133, 136, 174	16, 150, 160, 230	16, 16, 139, 155	17, 114, 139, 173	16, 87, 138, 167
	5	16, 17, 139, 139, 154	16, 16, 90, 141, 143	16, 16, 141, 141, 153	16, 16, 138, 141, 155	22, 22, 115, 120, 129	16, 16, 122, 140, 150

TABLE 45.8 The P-Value and Mean difference of the PSNR, SSIM, execution time, and Fitness values for the proposed method.

Mean difference	P-Value	Algorithms	Proposed Method	Metric	Mean difference	P-Value	Algorithms	Proposed Method	Metric
-0.2620	0.39 (*)	HHO	MFWOA	execution time	-117.3716	0.032 (*)	HHO	MFWOA	Fitness values
-181.8310	0.023 (*)	EO			-149.1045	0.048 (*)	EO		
11.6823	0.031 (*)	WOA			-135.0909	0.045 (*)	WOA		
0.7902	0.039 (*)	MPA			-194.0841	0.05	MPA		
2.5068	0.046 (*)	MFO			-193.6614	0.05	MFO		
0.0020	0.045 (*)	HHO	MFWOA	SSIM	0.0585	0.044 (*)	HHO	MFWOA	PSNR
0.0009	0.029 (*)	EO			0.0306	0.038 (*)	EO		
0.0075	0.030 (*)	WOA			0.3249	0.035 (*)	WOA		
0.0033	0.036 (*)	MPA			0.2165	0.034 (*)	MPA		
0.0049	0.028 (*)	MFO			0.3168	0.031 (*)	MFO		

Table 45.5. As the value of k increases, the obtained SSIM for all photos by all algorithms has an ascending/descending trend. The value of SSIM does not increase with the increasing value of k. Instead, at some levels, the threshold decreases and then rises again. In general, the value of SSIM and PSNR at higher threshold levels is much higher than at lower threshold levels, but as the threshold levels increase, the values of PSNR and SSIM often fluctuate. For example, in this chapter, the value of SSIM at the threshold level $k = 32$ is higher than the lower threshold levels. However, it can be seen from Table 45.4 and Table 45.5 that the proposed MFWOA algorithm has better results than all other algorithms at all levels of the lower, middle, and upper levels, and this is because according to the combination of the WOA and MFO algorithms.

This chapter considers each image separately as an optimization problem and a search space for each MH algorithm and the proposed MFWOA method. The results in some cases may be different for each parameter, depending on the structure of the respective MH algorithm. Table 45.6 shows the execution time for different algorithms over 100 runs. For all images, WOA and EO have the minimum and maximum execution time at all threshold levels, respectively. Also, the execution time of MFWOA is longer than WOA and less than MFO because the combination of WOA and MFO is used in the proposed MFWOA method.

Inspecting the results in Table 45.6, EO is slower than other algorithms, and WOA is faster than other algorithms. MFWOA is then faster than HHO, EO, MFO, and MPA. HHO is also faster than EO, MFO, and MPA. MPA is also quicker than EO and MFO. MFO is also faster than EO. This difference is in the speed of operation of algorithms is the difference in their structure and the use of special operators that each algorithm has used to achieve the final answer. For example, the HHO has fewer parameters, low complexity, and high speed. MFO is also very accurate and is one of the efficient algorithms. The parameters and operators in any MH algorithm will determine the degree of convergence and efficiency. Early convergence may occur if these parameters and operators are not appropriately selected. WOA has a straightforward structure, and fewer parameters are used in its design. WOA uses more straightforward operators. Therefore, it has less CC and is faster. The EO algorithm is slow because it examines the various conditions to increase its performance and search the search space locally and globally. MPA also has a relatively high CC due to its structure.

HHO, MFO, EO, MPA, and WOA have also made the best use of parameters and operators to achieve the best answer. In any case, the proposed MFWOA algorithm has better performance in terms of SSIM and PSNR than the other five compared algorithms, and this is because it uses a combination, which in this case also has the high capability. Some algorithms take advantage of the exploration phase and the capacity of algorithms in the exploitation phase. Therefore, using the combination of features of WOA and MFO algorithms in the proposed MFWOA method is the main factor of MFWOA superiority over the compared algorithms. Table 45.8 shows the values of the obtained thresholds for the various images obtained by each algorithm after 100 runs. Figs. 45.5 to 45.10 show the Thresholding images of “Test1”, “Test2”, “Test3”, “Test4”, “Test5”, and “Test6” that obtained by all algorithms, respectively.

Table 45.8 shows the P-Value and mean value for the metrics of the Fitness Function, SSIM, PSNR, and execution time. In particular, the P-Value indicates the probability of error in accepting the validity of the observed results, valid in the sense that the experimental result well represents the community. For example, a P-Value of 0.05 indicates a 5% probability that the relationship we observed in the sample is “accidental”. The lower the P-Value, the higher the accuracy of our work and the lower the error rate. In this chapter, Hypothesis Zero assumes that there is no significant difference between the mean values of the algorithms. However, the alternative hypothesis considers a considerable difference between them. The

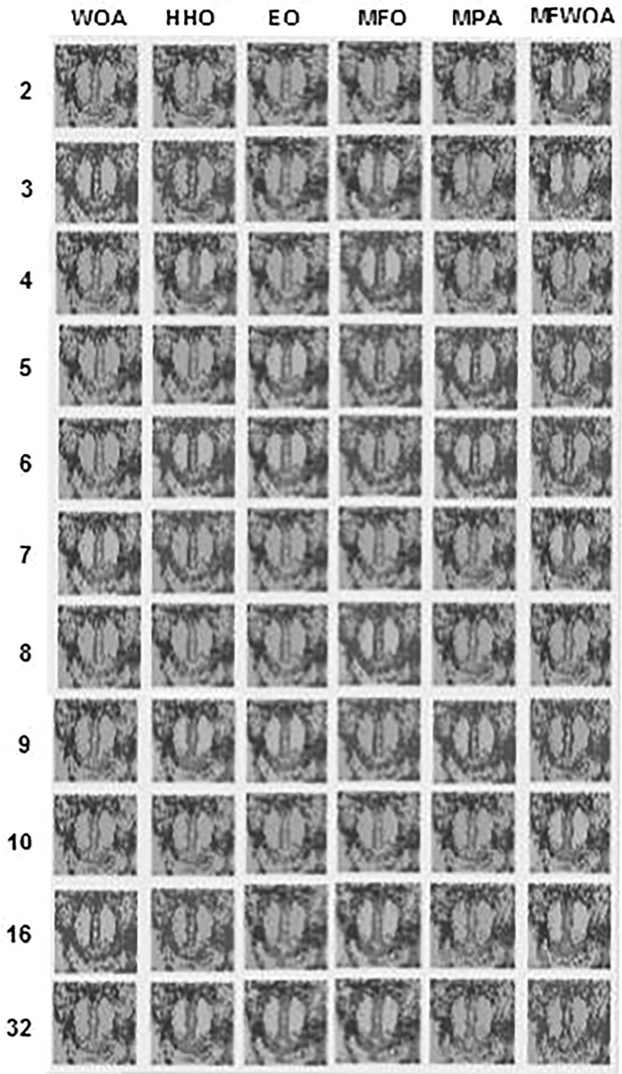


FIGURE 45.5 The Thresholding images of the "Test1" image obtained by all algorithms.

negative value difference in Table 45.8 indicates that the proposed algorithm performs worse than the compared algorithms in terms of the relevant evaluation metric. However, considering that in our proposed method, the threshold values are obtained by minimizing the Fitness Function, so the lower the value of the Fitness Function, the better the corresponding algorithm. So, if the value difference value for the Fitness Function metric is negative, the relevant algorithm has better performance than other algorithms. Also, the higher the mean difference per execution time for a negative algorithm, the faster the corresponding algorithm and the shorter the execution time than other algorithms.

Considering the results in Table 45.8, considering that the proposed algorithm has a negative value for Fitness Function values compared to different algorithms, the proposed algorithm is better than these algorithms in terms of evaluation metric values of the Fitness Function. Wherever the P-Value difference of the proposed algorithm with other algorithms is less than 0.05, it means that the performance of the proposed algorithm is generally better than the corresponding algorithms. In 18 cases, the P-Value difference of the proposed method with other algorithms is less than 0.05 and is indicated by (*). It shows a significant difference between the presented and compared algorithms, and the null hypothesis is incorrect.

Therefore, according to Table 45.8, the null hypothesis is rejected for 18 cases, and there is a significant difference between the proposed algorithm and other algorithms. In all three cases, the P-Value is 0.05, meaning there is a 5% chance that the relationship we observed in the sample is "accidental." In this chapter, based on Table 45.8, the P-Value difference of the proposed algorithm with other algorithms for SSIM is less than 0.05. Therefore, the null hypothesis for this metric is not accepted, and this shows a significant difference between the proposed algorithm and other comparative algorithms

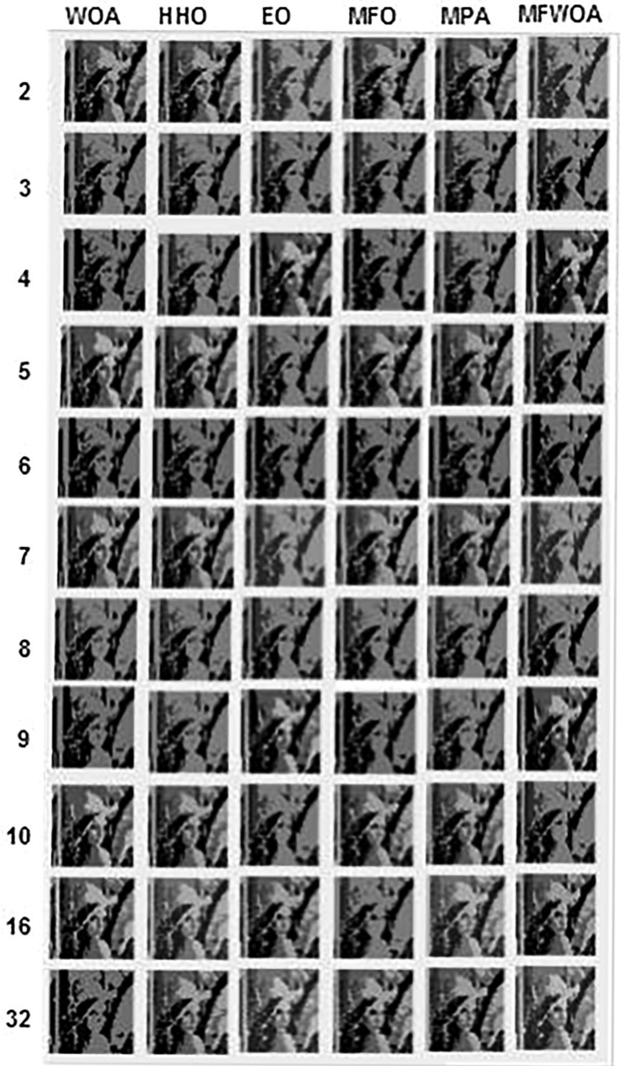


FIGURE 45.6 The Thresholding images of the "Test2" image obtained by all algorithms.

for this metric (SSIM). Also, the difference between the P-Value of the proposed algorithm and the HHO, EO, and WOA algorithms per Fitness values is less than 0.05. Therefore, hypothesis zero for this metric is not accepted. It shows a significant difference between the proposed algorithm and the corresponding algorithms for this metric (Fitness values). Also, for this metric, the value of the difference between the proposed method, MPA, and MFO algorithms is equal to 0.05. It shows that 5% may be "accidental" in the sample for the proposed algorithm and MPA and MFO algorithms.

Table 45.8 also revealed the difference between the P-Value of the proposed algorithm and the EO, WOA, MFO, and MPA algorithms per execution time is less than 0.05. Therefore, the zero hypotheses are rejected. It shows a significant difference between the proposed algorithm and these algorithms for this metric. Also, the difference between the P-Value of the proposed algorithm and the HHO algorithm per execution time metric is more than 0.05. Therefore, the alternative hypothesis is rejected, which means a significant difference between the proposed algorithm and this algorithm. The difference in the P-Value for the proposed algorithm and other algorithms for PSNR is less than 0.05. Therefore, the zero hypotheses are rejected. It shows a significant difference between the proposed algorithm and these algorithms for this metric. It should be noted that the alternative hypothesis is accepted in this chapter, and it is argued that there is a significant difference between the proposed algorithm and other algorithms. Because in most cases, the P-Value is less than 0.05. According to Table 45.9 and Table 45.10, the Mean value of the proposed MFWOA algorithm is positive with the other algorithms against the PSNR, Fitness Function, execution time and SSIM evaluation metric. The MFWOA algorithm performs better than the different algorithms. The mean difference value for both PSNR and SSIM evaluation metrics is

FIGURE 45.7 The Thresholding images of the "Test3" image obtained by all algorithms.

negative, meaning that the MFWOA algorithm performs worse than other algorithms. For the Fitness Function evaluation metric, given that we have used the minimization of this metric to obtain solutions, the lower the Fitness Function value for each algorithm, it can be said that the relevant algorithm is more efficient than other algorithms. Of course, it should be noted that in some cases, the value of the Fitness Function for MFWOA is slightly higher than different algorithms, but the results of PSNR and SSIM for the proposed MFWOA algorithm are better than other algorithms. So, anywhere in Table 45.9, the Mean difference is negative for the Fitness Function evaluation metric, meaning that the MFWOA algorithm performs better. Also, considering that the lower the value of execution time, the higher the speed, so for this metric (execution time), the difference between the proposed algorithm and other algorithms is negative, i.e., the proposed algorithm is faster than the algorithm has the desired.

As shown from Table 45.9, for the Fitness Function, for different images at different threshold levels, it has a positive Mean difference value in 35 cases, indicating that the Fitness Function value for MFWOA is higher than other algorithms. In 19 points, the value difference is zero, meaning that the value of the Fitness Function is the same for MFWOA and the corresponding algorithm. In other cases, the value difference is negative. As shown from Table 45.9, for the PSNR evaluation metric, the Mean difference value is negative in 5 cases, indicating that the PSNR value from the MFWOA is lower than the other algorithms. In 16 points, the Mean difference value is equal to zero, indicating that the PSNR value of the MFWOA is the same as the PSNR value of the other algorithms. In other cases, the mean difference value is positive,

FIGURE 45.8 The Thresholding images of the "Test4" image obtained by all algorithms.

indicating the superiority of the proposed MFWOA algorithm over this metric (PSNR) is different from other algorithms. In the SSIM evaluation metric, Table 45.10 evidently shows that the Mean difference value is negative in one case only, indicating that the SSIM value obtained from MFWOA is less than the other algorithms. In one case, the Mean difference value is equal to zero, indicating that the SSIM value from the MFWOA is the same as the SSIM value from the other algorithms.

In other cases, the value difference is positive, indicating the superiority of the proposed MFWOA algorithm over this metric (SSIM) is different from other algorithms. For the execution time evaluation metric, for all cases, the Mean difference value of the two algorithms MFWOA and HHO for the Test1 and Test6 images is negative, indicating that the execution time value obtained from the MFWOA is lower than the HHO. The speedup in the MFWOA is higher than the HHO for the two corresponding images. The MFO is lower than other algorithms at those levels. In other cases, the mean difference is positive, indicating that the proposed MFWOA algorithm for this metric (execution time) is slower than different algorithms. In summary, the results of our experiments show that the use of MFWOA for MTIS is more efficient than other algorithms. However, WOA, compared to MFO, showed promising results for a few thresholds, while its performance is in most cases weaker than MFO (according to SSIM and PSNR results). It could be because the MFO can switch between the exploration and operation phases, which are the two main phases in any MH algorithm. MFOs better escape local optimization and early convergence and find more accurate answers to the problem. At the same time, WOA is trapped in the local optimization in the early stages of optimization and cannot find optimal global solutions in the

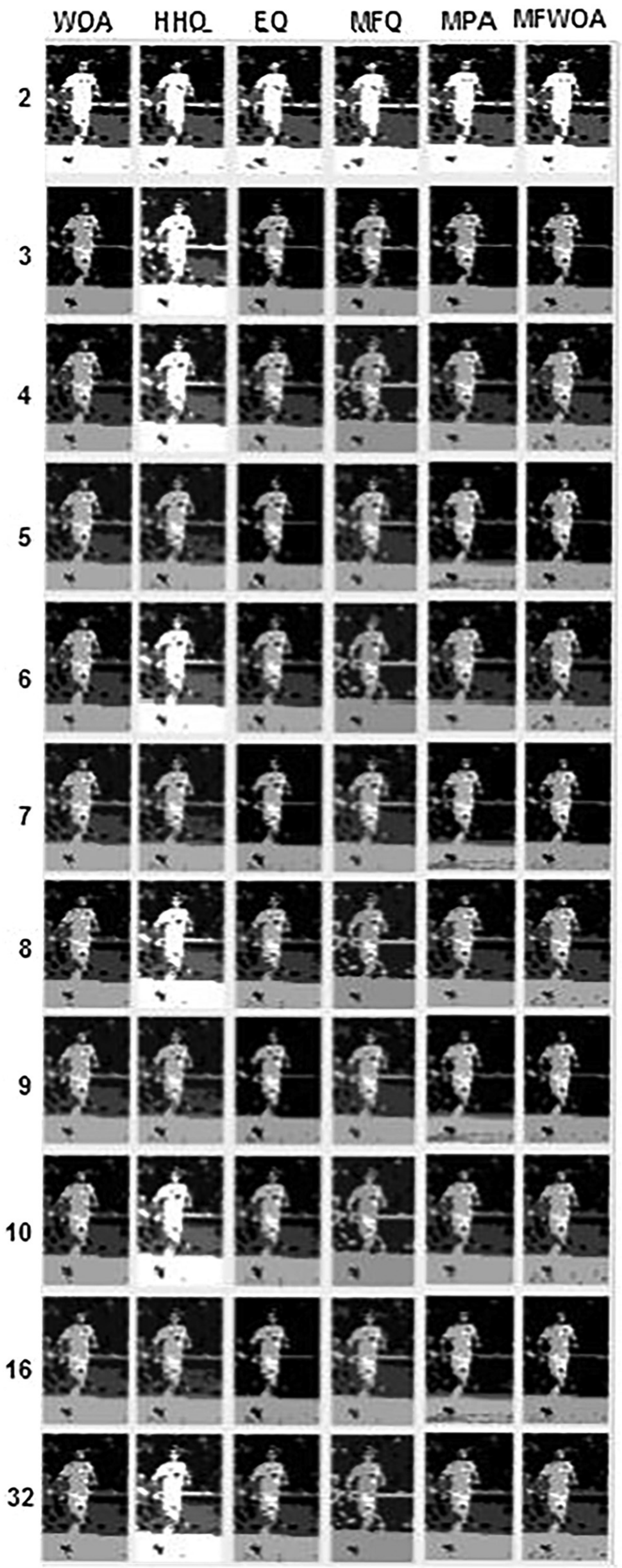


FIGURE 45.9 The Thresholding images of the "Test5" image obtained by all algorithms.

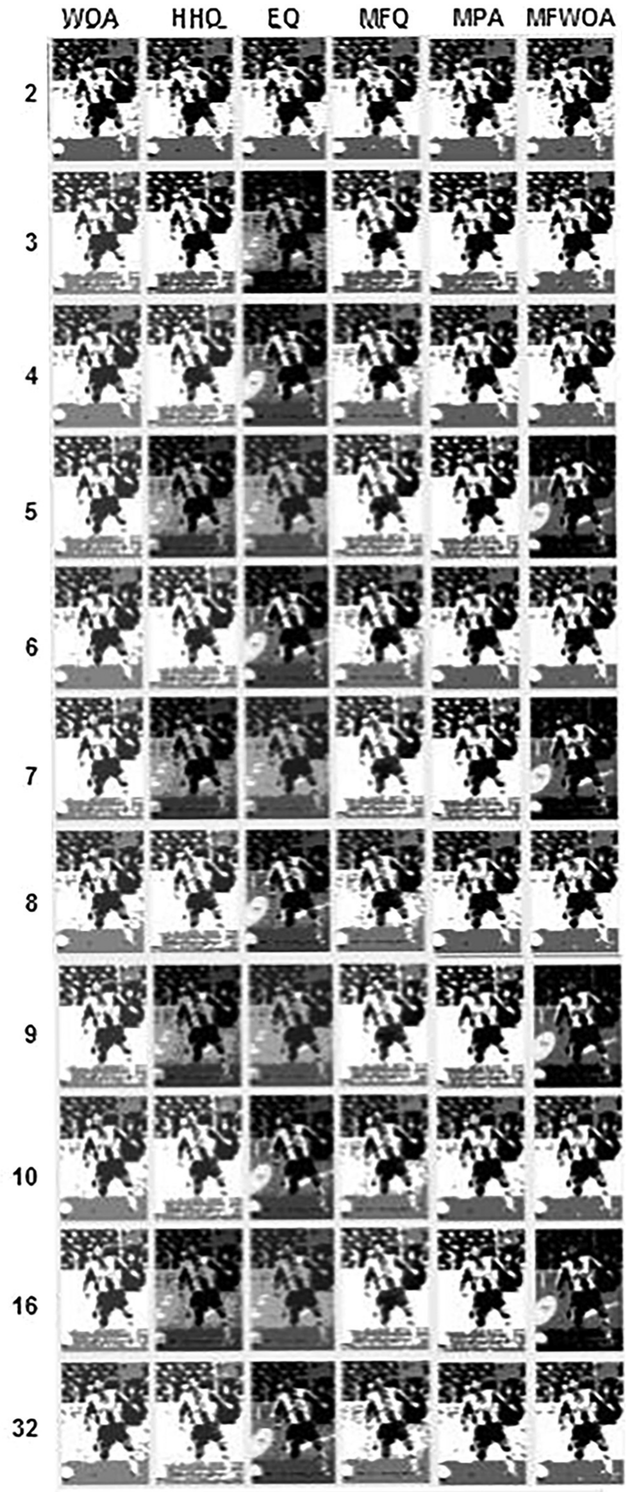


FIGURE 45.10 The Thresholding images of the "Test6" image obtained by all algorithms.

search space. Thus, combining MFO with WOA helps improve WOA and, after merging with MFO, makes WOA more capable of switching between exploration and operation phases and achieves better outputs. It should be noted that for all values obtained in Tables 45.9 and 45.10, significant difference at level P-Value <0.05.

TABLE 45.9 The Mean difference of the Fitness Function and PSNR values for the proposed method.

	k	PSNR					Fitness Function				
		HHO	EO	WOA	MPA	MFO	HHO	EO	WOA	MPA	MFO
Test 1	2	0.0040	0.0040	0.0060	0.0040	0.0040	0	0	-0.0001	0	0
	3	1.1750	0.0480	0.5470	0.0480	0.0480	-0.0003	-0.0305	-0.0238	-0.0305	-0.0305
	4	2.0190	0.4540	0.2540	4.3080	4.3080	0.0164	-0.1133	0.0150	-0.1255	-0.1255
	5	1.6500	3.0890	0.0010	3.0890	0.5660	-0.0106	-0.0211	0.0019	-0.0366	-0.0107
	6	0.2780	3.9530	4.4650	3.9530	3.9530	-0.2144	-0.1854	-0.1533	-0.1908	-0.1908
	7	0.6260	7.2740	1.7810	7.2740	5.0320	-0.0276	-0.1366	-0.0435	-0.1889	-0.1109
	8	0.1820	2.2340	5.0810	4.6690	2.2420	-0.0177	-0.0619	-0.0085	-0.1034	-0.0755
	9	2.3320	1.5490	0.5820	1.5490	2.6880	-0.1088	-0.1319	-0.0742	-0.2011	-0.1689
	10	0.0240	5.7590	2.1130	8.0020	3.1590	-0.0927	-1.2371	-0.7029	-1.3218	-0.7575
	16	4.4100	0.0150	1.3360	3.6410	3.7760	-0.2853	-1.5251	0.0205	-1.6353	-1.6333
	32	1.005	5.6990	11.2510	5.4190	1.3200	-2.2877	-4.8427	-1.8417	-4.1977	-3.5537
Test 2	2	0.0130	0.0130	0.0200	0.0130	0.0130	-0.0019	-0.0019	-0.0014	-0.0019	-0.0019
	3	1.2540	1.2540	0.1100	1.2540	1.2540	-0.0781	-0.0781	0.0018	-0.0781	-0.0781
	4	-0.3860	3.0200	3.0070	3.0200	3.0200	-0.0175	-0.2284	-0.2433	-0.2443	-0.2443
	5	1.6670	1.4300	0.8470	1.4090	1.4300	-0.1339	-0.1486	-0.1322	-0.1695	-0.1695
	6	3.5560	1.5740	0.0080	3.5890	3.5990	-0.0436	-0.3732	0.0069	-0.4250	-0.4209
	7	1.6740	0.4650	2.7270	3.8640	0.4650	0.0183	-0.1776	0.3704	-0.1694	-0.0884
	8	4.0400	3.2730	2.2250	4.1390	1.0100	-0.1286	-0.6394	-0.1508	-0.7195	-0.6362
	9	2.0350	0.2050	1.5330	3.6500	0.0810	-0.0550	-0.7324	-0.7259	-0.7300	-0.4879
	10	4.6560	3.3960	0.9150	3.2940	2.6930	-0.3603	-1.5982	-0.8984	-1.6427	-1.6405
	16	5.0890	2.6860	5.9650	4.7240	0.0440	-0.2835	-2.6545	-1.1287	-2.7849	-2.5384
	32	6.1060	1.7760	7.2120	0.4560	2.0110	-1.4720	-6.401	-2.076	-5.306	-4.475
Test 3	2	0	0	0.0130	0	0	-0.0010	-0.0010	-0.0006	-0.0010	-0.0010
	3	0	0	0	0	0	-0.0015	-0.0015	-0.0015	-0.0015	-0.0015
	4	-0.0500	3.2570	0.0020	3.2570	3.2570	-0.1234	-0.3414	-0.1228	-0.4000	-0.4000
	5	0.9030	2.9310	0.1110	2.9310	2.9310	-0.1579	-0.9373	-0.1571	-0.9373	-0.9373
	6	-0.8900	0.8390	0.8720	0.8390	0.8390	0.0082	0.0045	-0.0086	-0.0245	-0.0244
	7	2.1550	2.1810	0.0540	2.1810	2.9410	-0.8742	-0.8285	-0.0809	-0.8759	-0.3778
	8	1.5300	0.1580	1.5830	0.1580	2.6850	-0.1905	-0.5574	-0.5380	-0.3487	-0.6238
	9	0.1700	2.4020	0.3180	2.4020	2.4020	-0.2728	-1.1864	-0.5845	-1.4419	-1.4419
	10	0.9810	2.3360	0.1270	2.3360	2.3360	-0.8171	-1.1333	-0.7264	-1.1712	-1.1711
	16	1.5850	4.7490	4.7990	2.0750	2.0780	-1.5593	-2.8923	-1.3728	-2.4243	-2.4243
	32	5.4820	4.7010	3.9510	3.4790	5.6650	-3.474	-7.9760	-3.255	-6.162	-8.326
Test 4	2	0.0140	0.0140	0.0240	0.0140	0.0140	-0.1021	-0.1021	-0.1013	-0.1021	-0.1021
	3	0.5330	2.0320	2.2650	2.0320	2.0320	-0.1437	-0.2001	-0.1990	-0.2001	-0.2001
	4	0.3510	4.0510	0.3960	4.0510	4.0510	-0.1259	-0.5656	-0.1217	-0.6008	-0.6008
	5	1.8760	0.3940	0.6350	0.3930	0.4480	-0.1824	-0.2936	-0.3090	-0.3130	-0.2933
	6	4.0330	2.4190	0.0010	2.4190	3.8080	-0.2430	-1.0715	-0.6036	-1.0661	-1.0983
	7	0.0080	4.3940	4.0170	0.1110	4.3930	-0.0366	-1.2788	-1.5503	-1.0534	-1.5610
	8	1.0730	0.5410	3.6000	2.5000	3.8890	-0.0715	-0.8469	-0.7937	-0.7809	-0.8453
	9	2.6360	4.3750	3.7970	2.9780	4.3670	0.1933	-0.6587	0.0173	-0.8195	-0.8518
	10	4.8620	1.2790	0.0080	3.9040	2.0850	-0.0294	-1.2147	-0.4932	-1.0883	-0.1527
	16	2.1420	2.2210	4.9780	2.7780	5.0080	-0.7781	-2.7731	-0.1802	-2.9671	-1.7771
	32	1.3100	2.7700	0.9950	2.4570	0.0090	-2.7300	-8.5360	-0.937	-7.692	-6.665

continued on next page

TABLE 45.9 (*continued*)

	k	PSNR					Fitness Function				
		HHO	EO	WOA	MPA	MFO	HHO	EO	WOA	MPA	MFO
Test 5	2	0.0060	0.0060	0.0060	0.0100	0.0060	0	0	0	0	0
	3	0.5130	0.2000	0.2000	0.2000	0.0800	-0.0001	0	0	0	0
	4	0.3700	3.6160	0.0750	3.6160	3.6160	0.0040	-0.0483	-0.0100	-0.0524	0.0026
	5	4.1370	3.1690	2.4920	3.1690	0.0520	-0.0377	-0.0362	-0.0434	-0.0454	-0.0022
	6	2.0400	2.7190	0.0830	2.7360	2.7360	3.5918	3.4571	3.4720	3.4556	3.4547
	7	0.6750	0.0700	3.2250	3.9620	0.0700	-0.0175	-0.1117	-0.1377	-0.1066	-0.0945
	8	1.4070	1.5730	2.5280	4.4730	4.4560	-0.0537	-0.2730	-0.2652	-0.2919	-0.2919
	9	0.4020	0.0560	0.3700	3.2110	2.8680	-0.1449	-0.1416	-0.1221	-0.1814	-0.1493
	10	1.3330	1.0230	1.0110	0.8360	0.0610	-0.0126	-0.0235	0.0026	-0.0337	-0.0016
	16	5.0180	1.9490	0.2620	0.1250	2.7140	-0.0590	-0.5429	-0.3375	-0.3637	-0.5184
	32	8.7240	4.5170	9.6740	1.7760	1.7510	-0.0399	-0.5095	-0.0917	-0.4468	-0.0934
Test 6	2	0.0070	-0.0030	0	0.0070	0.0070	-0.0001	-0.0001	-0.0000	-0.0001	-0.0001
	3	0.0140	0.0240	0.1850	0.0240	0.0240	0.0052	0.0003	0.0000	-0.0001	-0.0001
	4	0.0650	3.1730	0.1250	3.1730	3.1730	-0.0001	-0.1985	-0.0000	-0.1985	-7.6985
	5	2.5630	0.5110	0.0190	3.1570	0.4600	0.0048	-0.0029	0.0023	-0.0054	0.0005
	6	0.4900	2.6490	1.4360	3.0350	3.0350	-0.0935	-0.2923	-0.1022	-0.3019	-0.3019
	7	2.7620	4.5250	0.2170	4.5250	0.4190	-0.0871	-0.2222	-0.0941	-0.2950	-0.0924
	8	0.0950	0.5270	1.7780	3.1970	3.1970	-0.0400	-0.0978	0.0080	-0.1020	-0.0020
	9	2.3440	2.8910	3.3390	2.8910	0.0770	0.0052	-0.1496	-0.0053	-0.2086	-0.0065
	10	0.3460	4.2910	1.4930	5.6290	1.7270	-0.0409	-0.1510	-0.0567	-0.3085	-0.1099
	16	0.6390	1.7800	-0.0730	3.5420	0.0970	-0.0249	-0.3581	-0.0734	-0.6221	0.0017
	32	3.758	2.2010	0.7650	0.0220	0.2250	-0.2349	-1.6868	0.0086	-1.4301	-0.2725

TABLE 45.10 The Mean difference of the SSIM and execution time for the proposed method.

	k	SSIM					Fitness Function				
		HHO	EO	WOA	MPA	MFO	HHO	EO	WOA	MPA	MFO
Test 3	2	0.0007	0.0007	0.0007	0.0007	0.0007	-1.8325	0.0007	0.0007	0.0007	0.0007
	3	0.0212	0.0009	0.0194	0.0009	0.0009	-5.1671	0.0009	0.0194	0.0009	0.0009
	4	0.0584	0.0113	0.0010	0.1506	0.1506	-6.0070	0.0113	0.0010	0.1506	0.1506
	5	0.0601	0.1018	0.0034	0.1018	0.0114	-6.6552	0.1018	0.0034	0.1018	0.0114
	6	0	0.1164	0.1148	0.1164	0.1164	-7.4688	0.1164	0.1148	0.1164	0.1164
	7	0.0028	0.2153	0.0085	0.2153	0.1244	-16.3971	0.2153	0.0085	0.2153	0.1244
	8	0.0056	0.0476	0.1194	0.1365	0.0502	-20.0317	0.0476	0.1194	0.1365	0.0502
	9	0.0814	0.0528	0.0016	0.0528	0.0745	-26.1704	0.0528	0.0016	0.0528	0.0745
	10	0.0037	0.1632	0.0369	0.2540	0.0684	-11.3673	0.1632	0.0369	0.2540	0.0684
	16	0.1608	-0.0019	0.0360	0.1343	0.1406	-15.5009	-0.0019	0.0360	0.1343	0.1406
	32	0.0242	0.1107	0.3270	0.1045	0.0274	-25.2787	0.1107	0.3270	0.1045	0.0274
Test 4	2	0.0013	0.0013	0.0106	0.0013	0.0013	0.0013	0.0013	0.0106	0.0013	0.0013
	3	0.0001	0.0001	0.0177	0.0001	0.0001	0.0001	0.0001	0.0177	0.0001	0.0001
	4	0.0058	0.0877	0.0877	0.0877	0.0877	0.0058	0.0877	0.0877	0.0877	0.0877
	5	0.0060	0.0159	0.0007	0.0146	0.0159	0.0060	0.0159	0.0007	0.0146	0.0159
	6	0.0334	0.0366	0.0000	0.0379	0.0405	0.0334	0.0366	0.0000	0.0379	0.0405
	7	0.0723	0.0224	0.0787	0.0726	0.0224	0.0723	0.0224	0.0787	0.0726	0.0224
	8	0.0327	0.0791	0.0339	0.0523	0.0305	0.0327	0.0791	0.0339	0.0523	0.0305
	9	0.0110	0.0068	0.0516	0.0583	0.0150	0.0110	0.0068	0.0516	0.0583	0.0150
	10	0.0301	0.0643	0.0005	0.0416	0.0378	0.0301	0.0643	0.0005	0.0416	0.0378
	16	0.0726	0.0768	0.1083	0.0799	0.0004	0.0726	0.0768	0.1083	0.0799	0.0004
	32	0.0616	0.0014	0.1244	0.0200	0.0178	0.0616	0.0013	0.1244	0.0200	0.0178

continued on next page

TABLE 45.10 (*continued*)

	k	SSIM					Fitness Function				
		HHO	EO	WOA	MPA	MFO	HHO	EO	WOA	MPA	MFO
Test 5	2	0.0000	0.0000	0.0017	0.0000	0.0000	0.0000	0.0000	0.0017	0.0000	0.0000
	3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	4	0.0050	0.0718	0.0045	0.0718	0.0718	0.0050	0.0718	0.0045	0.0718	0.0718
	5	0.0194	0.0538	0.0033	0.0538	0.0538	0.0194	0.0538	0.0033	0.0538	0.0538
	6	0.0071	0.0376	0.0372	0.0376	0.0376	0.0071	0.0376	0.0372	0.0376	0.0376
	7	0.0348	0.0358	0.0013	0.0358	0.0539	0.0348	0.0358	0.0013	0.0358	0.0539
	8	0.0474	0.0069	0.0364	0.0069	0.0362	0.0474	0.0069	0.0364	0.0069	0.0362
	9	0.0008	0.0417	0.0211	0.0417	0.0417	0.0008	0.0417	0.0211	0.0417	0.0417
	10	0.0360	0.0527	0.0009	0.0527	0.0527	0.0360	0.0527	0.0009	0.0527	0.0527
	16	0.0037	0.0707	0.0722	0.0141	0.0143	0.0037	0.0707	0.0722	0.0141	0.0143
	32	0.0764	0.0550	0.0789	0.0414	0.0786	0.0764	0.0550	0.0789	0.0414	0.0786
Test 6	2	0.0038	0.0038	0.0050	0.0038	0.0038	0.0038	0.0038	0.0050	0.0038	0.0038
	3	0.0038	0.0195	0.0259	0.0195	0.0195	0.0038	0.0195	0.0259	0.0195	0.0195
	4	0.0053	0.0727	0.0094	0.0727	0.0727	0.0053	0.0727	0.0094	0.0727	0.0727
	5	0.0357	-0.0008	0.0086	-0.0007	0.0006	0.0357	-0.0008	0.0086	-0.0007	0.0006
	6	0.0724	0.0633	0.0031	0.0633	0.0725	0.0724	0.0633	0.0031	0.0633	0.0725
	7	0.0028	0.1296	0.1078	0.0254	0.1296	0.0028	0.1296	0.1078	0.0254	0.1296
	8	0.0031	0.0200	0.0532	0.0651	0.0743	0.0031	0.0200	0.0532	0.0651	0.0743
	9	0.0589	0.0975	0.0688	0.0868	0.0959	0.0589	0.0975	0.0688	0.0868	0.0959
	10	0.0727	0.0384	0.0039	0.1000	0.0636	0.0727	0.0384	0.0039	0.1000	0.0636
	16	0.0017	0.0397	0.0652	0.0590	0.0660	0.0017	0.0397	0.0652	0.0590	0.0660
	32	0.0037	0.0801	0.0215	0.0603	0.0302	0.0027	0.0801	0.0215	0.0603	0.0302
Test 7	2	0.0000	0.0000	0.0000	0.0040	0.0000	0.0000	0.0000	0.0000	0.0040	0.0000
	3	0.0238	0.0100	0.0100	0.0100	0.0055	0.0238	0.0100	0.0100	0.0100	0.0055
	4	0.0105	0.1856	0.0010	0.1856	0.1856	0.0105	0.1856	0.0010	0.1856	0.1856
	5	0.2413	0.1796	0.1441	0.1796	0.0007	0.2413	0.1796	0.1441	0.1796	0.0007
	6	-0.0303	0.1336	-0.0068	0.1364	0.1364	-0.0303	0.1336	-0.0068	0.1364	0.1364
	7	0.0447	0.0042	0.1874	0.2229	0.0042	0.0447	0.0042	0.1874	0.2229	0.0042
	8	0.0012	0.0534	0.1131	0.2257	0.2229	0.0012	0.0534	0.1131	0.2257	0.2229
	9	0.0022	0.0016	0.0487	0.1848	0.1735	0.0022	0.0016	0.0487	0.1848	0.1735
	10	0.0007	0.0807	0.0902	0.0740	0.0201	0.0007	0.0807	0.0902	0.0740	0.0201
	16	0.0682	0.0487	0.0011	0.0021	0.0822	0.0682	0.0487	0.0011	0.0021	0.0822
	32	0.3552	0.1182	0.4228	0.0328	0.0382	0.3553	0.1182	0.4228	0.0328	0.0382
Test 8	2	0.0002	0.0002	0.0006	0.0002	0.0002	-1.9699	0.0002	0.0006	0.0002	0.0002
	3	0.0003	0.0507	0.0561	0.0507	0.0507	-5.6698	0.0507	0.0561	0.0507	0.0507
	4	0.0006	0.0991	0.0044	0.0991	0.0991	-6.4689	0.0991	0.0044	0.0991	0.0991
	5	0.0465	0.0103	0.0010	0.1039	0.0060	-21.1737	0.0103	0.0010	0.1039	0.0060
	6	0.0015	0.0590	0.0445	0.1048	0.1048	-27.9872	0.0590	0.0445	0.1048	0.1048
	7	0.0556	0.1393	0.0009	0.1393	0.0023	-20.9375	0.1393	0.0009	0.1393	0.0023
	8	0.0022	0.0050	0.0473	0.1020	0.1020	-10.4139	0.0050	0.0473	0.1020	0.1020
	9	0.0451	0.1024	0.1114	0.1024	0.0014	-12.1925	0.1024	0.1114	0.1024	0.0014
	10	0.0070	0.1769	0.1146	0.2128	0.0880	-12.5260	0.1769	0.1146	0.2128	0.0880
	16	0.0203	0.1090	0.0724	0.1926	0.0006	-23.0742	0.1090	0.0724	0.1926	0.0006
	32	0.0626	0.0557	0.0393	0.0307	0.0367	-47.6932	0.0557	0.0393	0.0307	0.0267

45.6 Conclusions

In this chapter, the problem of determining the optimal thresholds in a MTIS was considered as an optimization problem. So, a combination of WOA and MFO was used to improve the performance of WOA to solve the problem of MTIS that uses the Fitness Function minimization. Inverse Otsu was also employed as a Fitness Function in the MFWOA algorithm and other MH algorithms. The experimental results of the proposed MFWOA algorithm were compared with MPA, WOA, HHO, MFO, and EO algorithm on the eight different images using PSNR, SSIM, execution time, and Fitness Function evaluation metric. The results demonstrated that the MFWOA algorithm is better for all images regarding PSNR and SSIM than other algorithms. However, in terms of execution time, MFWOA seemed a little slower. Therefore, our proposed MFWOA algorithm performed better than other algorithms regarding PSNR, SSIM, segmentation time, and segmentation accuracy on the tested images. But in terms of execution time evaluation metric, the MFWOA algorithm is faster than WOA and slower than MFO. In some cases, the proposed algorithm was faster than other algorithms. Our next work is to use a combination of the WOA and the Artificial Neural Network (ANN) to improve the WOA as well as the MTIS problem as an optimization problem.

References

[1] E.H. Houssein, et al., A novel Black Widow Optimization algorithm for multilevel thresholding image segmentation, Expert Systems with Applications 167 (2021) 114159, https://doi.org/10.1016/j.eswa.2020.114159.

[2] R. Srikanth, K. Bikshalu, Multilevel thresholding image segmentation based on energy curve with harmony Search Algorithm, Ain Shams Engineering Journal 12 (1) (2021) 1–20, https://doi.org/10.1016/j.asej.2020.09.003.

[3] M. Abd El Aziz, et al., Whale optimization algorithm and moth-flame optimization for multilevel thresholding image segmentation, Expert Systems with Applications 83 (2017) 242–256, https://doi.org/10.1016/j.eswa.2017.04.023.

[4] L. Li, et al., Fuzzy multilevel image thresholding based on improved coyote optimization algorithm, IEEE Access 9 (2021) 33595–33607, https://doi.org/10.1109/ACCESS.2021.3060749.

[5] S. Mozafari, Provide a hybrid method to improve the performance of multilevel thresholding for image segmentation using GA and SA algorithms, in: IEEE-7th Conference on Information and Knowledge Technology (IKT), 2015, pp. 1–6, https://doi.org/10.1109/IKT.2015.7288751.

[6] G. Sun, et al., A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding, Applied Soft Computing 46 (2016) 703–730, https://doi.org/10.1016/j.asoc.2016.01.054.

[7] B. Küçükuğurlu, E. Gedikli, Symbiotic organisms search algorithm for multilevel thresholding of images, Expert Systems with Applications 147 (2020) 113210, https://doi.org/10.1016/j.eswa.2020.113210.

[8] V.K. Bohat, K. Arya, A new heuristic for multilevel thresholding of images, Expert Systems with Applications 117 (2019) 176–203, https://doi.org/10.1016/j.eswa.2018.08.045.

[9] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95-International Conference on Neural Networks, vol. 4, IEEE, 1995, pp. 1942–1948, https://doi.org/10.1109/ICNN.1995.488968.

[10] S. Mirjalili, A. Lewis, The whale optimization algorithm, Advances in Engineering Software 95 (2016) 51–67, https://doi.org/10.1016/j.advengsoft.2016.01.008.

[11] S. Mirjalili, Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm, Knowledge-Based Systems 89 (2015) 228–249, https://doi.org/10.1016/j.knosys.2015.07.006.

[12] M. Abd Elaziz, et al., A competitive chain-based Harris Hawks Optimizer for global optimization and multi-level image thresholding problems, Applied Soft Computing 95 (2020) 106347, https://doi.org/10.1016/j.asoc.2020.106347.

[13] T. Salehnia, A. Fathi, Fault tolerance in LWT-SVD based image watermarking systems using three module redundancy technique, Expert Systems with Applications 179 (2021) 115058, https://doi.org/10.1016/j.eswa.2021.115058.

[14] H. Liu, Research on cloud computing adaptive task scheduling based on ant colony algorithm, Optik 258 (2022) 168677, https://doi.org/10.1016/j.ijleo.2022.168677.

[15] S. Raziani, et al., Selecting of the best features for the knn classification method by Harris Hawk algorithm, in: Conference: 8th International Conference on New Solutions in Engineering, Information Science and Technology of the Century ahead, 2021, https://civilica.com/doc/1196573/.

[16] E. Rodríguez-Esparza, et al., An efficient Harris hawks-inspired image segmentation method, Expert Systems with Applications 155 (2020) 113428, https://doi.org/10.1016/j.eswa.2020.113428.

[17] J. Anitha, et al., An efficient multilevel color image thresholding based on modified whale optimization algorithm, Expert Systems with Applications 178 (2021) 115003, https://doi.org/10.1016/j.eswa.2021.115003.

[18] L. Duan, et al., Multilevel thresholding using an improved cuckoo search algorithm for image segmentation, Journal of Supercomputing 77 (2021) 6734–6753, https://doi.org/10.1007/s11227-020-03566-7.

[19] T. Salehnia, et al., Multi-level image thresholding using GOA, WOA and MFO for image segmentation, in: 8th International Conference on New Strategies in Engineering, Information Science and Technology in the Next Century, Dubai, United Arab Emirates (UAE), Civilica, 2021, https://civilica.com/doc/1196572/.

[20] L. Samantaray, et al., A new Harris hawks-Cuckoo search optimizer for multilevel thresholding of thermogram images, Revue d'Intelligence Artificielle 34 (5) (2020) 541–551, https://doi.org/10.18280/ria.340503.

[21] Nobuyuki Otsu, A threshold selection method from gray-level histograms, IEEE Transactions on Systems, Man and Cybernetics 9 (1) (1979) 62–66, https://doi.org/10.1109/TSMC.1979.4310076.

[22] X.-S. Yang, Nature-Inspired Metaheuristic Algorithms, Luniver Press, 2008.

[23] Z. Xing, An improved emperor penguin optimization based multilevel thresholding for color image segmentation, Knowledge-Based Systems 194 (2020) 105570, https://doi.org/10.1016/j.knosys.2020.105570.

[24] A.A. Heidari, et al., Harris hawks optimization: algorithm and applications, Future Generation Computer Systems 97 (2019) 849–872, https://doi.org/10.1016/j.future.2019.02.028.

[25] A. Faramarzi, et al., Equilibrium optimizer: a novel optimization algorithm, Knowledge-Based Systems 191 (2020) 105190, https://doi.org/10.1016/j.knosys.2019.105190.

[26] A. Faramarzi, et al., Marine predators algorithm: a nature-inspired metaheuristic, Expert Systems with Applications 152 (2020) 113377, https://doi.org/10.1016/j.eswa.2020.113377.