ORIGINAL ARTICLE

Improved manta ray foraging optimization for multi-level thresholding using COVID-19 CT images

Essam H. Houssein[1] · Marwa M. Emam[1] · Abdelmgeid A. Ali[1]

Abstract

Coronavirus disease 2019 (COVID-19) is pervasive worldwide, posing a high risk to people's safety and health. Many algorithms were developed to identify COVID-19. One way of identifying COVID-19 is by computed tomography (CT) images. Some segmentation methods are proposed to extract regions of interest from COVID-19 CT images to improve the classification. In this paper, an efficient version of the recent manta ray foraging optimization (MRFO) algorithm is proposed based on the oppositionbased learning called the MRFO-OBL algorithm. The original MRFO algorithm can stagnate in local optima and requires further exploration with adequate exploitation. Thus, to improve the population variety in the search space, we applied Opposition-based learning (OBL) in the MRFO's initialization step. MRFO-OBL algorithm can solve the image segmentation problem using multilevel thresholding. The proposed MRFO-OBL is evaluated using Otsu's method over the COVID-19 CT images and compared with six meta-heuristic algorithms: sine-cosine algorithm, moth flame optimization, equilibrium optimization, whale optimization algorithm, slap swarm algorithm, and original MRFO algorithm. MRFO-OBL obtained useful and accurate results in quality, consistency, and evaluation matrices, such as peak signal-to-noise ratio and structural similarity index. Eventually, MRFO-OBL obtained more robustness for the segmentation than all other algorithms compared. The experimental results demonstrate that the proposed method outperforms the original MRFO and the other compared algorithms under Otsu's method for all the used metrics.

Keywords COVID-19 CT images · Otsu's method · Multilevel thresholding image segmentation · Manta ray foraging optimization · Meta-heuristics algorithms

1 Introduction

The coronavirus disease 2019 (COVID-19) is one of the newest viruses [1, 2]. It had arisen from a virus named SARSCoV-2. The first cases of COVID-19 were recorded in Wuhan, China, in December 2019 [3]. COVID-19 is a deadly pandemic that the world is currently facing. COVID-19 has impacted approximately 11 million globally, including over 500,000 people in Europe and the USA [4]. COVID-19 has become a worldwide pandemic because of an exponential growth rate and poorly understood transmission mechanism [5, 6]. Cough, fever, sore throat, headache, exhaustion, and discomfort in the muscles are the previously reported symptoms of COVID-19 [1]. This virus also contributes to pneumonia, an infection that inflames the lungs' air sacs. COVID-19 influences countries' health, and its effects (e.g., economic and psychological) are crucial. Early detection may be mirrored in early therapy. Accordingly, COVID-19 is an epidemic with a wide range of challenges to tackle. Based on the knowledge mentioned above, there is a need for tools to detect this fatal disease on time. One of the critical problems is using multiple tests to diagnose COVID-19. The real-time polymerase chain reaction is the most commonly used test. It is invasive and time-consuming, and the

✉ Essam H. Houssein
  essam.halim@mu.edu.eg

  Marwa M. Emam
  marwa.khalef@mu.edu.eg

  Abdelmgeid A. Ali
  a.ali@mu.edu.eg

[1] Faculty of Computers and Information, Minia University, Minia, Egypt

diagnosis is false-negative. Chest computed tomography (CT) is another test that plays an essential role in COVID-19 diagnosis [7].

CT offers a guide to pathophysiology, which might shed light on the diagnosis and development of specific disease stages. It evolves as a useful diagnostic method for the medical practice of lung disease associated with COVID-19 [5]. Preliminary studies suggested that chest CT has a high detection sensitivity for COVID-19 lung pathology. Several groups have demonstrated the capacity for diagnosis based on CAD systems, with accuracy as high as 95% [7]. Medical imaging has been used recently for multiple diagnoses of the disease. Medical imaging methods may be used as a critical diagnostic to guide a possible disease.

Image segmentation is considered a critical process in image processing. It plays a significant role in image research in different applications such as computer vision, pattern recognition, and medical images [8, 9]. The classification system may fail if the segmentation result is incorrect. Segmentation aimed to split the image into several homogeneous regions or segments containing similar features, such as texture, color, brightness, contrast, form, and size, regarding a specific thresholding value(s) [10]. Several methods (e.g., edge-based, regional-based, threshold-based, and feature-based clustering) have been suggested to solve the current problems and improve research quality in the literature.

Thresholding is extremely important in the image segmentation domain. It is categorized into two types: bi-level and multilevel thresholding. The simplest thresholding technique is bi-level thresholding segmentation that divides an image into two classes by searching a single thresholding value. Multilevel thresholding divides the image into several distinct parts by establishing several threshold values [11]. Multilevel thresholding is the most common image segmentation method that attempts to group the pixels that share characteristics into a finite number of pixels and segment an image into more than two regions based on pixel intensity [12].

Most image segmentation based on multilevel thresholding methods uses the image histogram as an input to determine the thresholds by maximizing or minimizing the objective functions [13], for example, fuzzy entropy [14], Kapur's entropy [15], Tsallis entropy [16], and Otsu's method [17].

However, multilevel thresholding segmentation methods have some limitations in computational costs. The computational time will increase for multilevel thresholding problems with increasing thresholding values as they search for the best thresholding values exhaustively to optimize the objective function. There are two types of approaches for finding the appropriate threshold values in multilevel thresholding segmentation: parametric and nonparametric. A parametric approach assumes that each class of the image can identify by using probability density distributions. These classes are all used to represent the pixels in an image. This approach takes a long time to compute. However, the nonparametric methods use discriminated rules to separate the pixels into homogeneous regions. The threshold values are determined by a statistical criterion [18]. Otsu's method [17] presented the maximization of the variance between classes to obtain the optimal threshold values. In this paper, we focused on a parametric approach, in which we measured the specific statistical parameters for each class in the image. To achieve the desired threshold values, we used Otsu's method by maximizing between variances. Otsu suggested a common approach that can determine thresholds to optimize the class variance of the foreground and background intensity levels. It is one of the most suitable threshold selection methods for real-world images [19]. However, there were challenges in these approaches when achieving the desired threshold values for multilevel thresholding. The computational time is also expensive, which grows exponentially with the number of threshold values. The process of searching for optimum thresholding values for multilevel thresholding segmentation can be considered a constrained optimization problem. The optimization algorithms, especially the meta-heuristic methods, are commonly used in related research to solve these problems [20].

Optimization is the process of determining the best possible solution(s) for a particular problem. Trying to find the fastest road to get to a destination or coordinate work activities to minimize the idle time between each task are examples of optimization problems. Surely, everyone uses optimization daily. Optimization intends to reach a satisfying or optimal solution concerning different objectives for a specific problem. Optimization is received a huge increased concentration in the last decade since it has ubiquitous nature with all issues [21]. It appears everywhere and almost in every application like Computer Science, engineering problems [22, 23], and Finance, feature selection [24], fuzzy control systems [25], chemoinformatics, bioinformatics [26], engineering optimization problems [27], improving the collective construction systems [28], and image segmentation problems [29]. So many fields of science need optimization, different optimization methods have been developed. Optimization algorithms can be defined as processes where possible solutions are evaluated to choose the most suitable solution for the problem at hand. The quality of the optimized solution depends on the chosen algorithm to solve the problem. As a result, one needs to select the most appropriate algorithm to solve the problem at hand [30]. An optimization algorithm may start and perform the

optimization process by single or multiple random solutions. In the first case, the optimization process begins with a single random solution, and it is iteratively improved over the iterations. In the second case, a set of solutions is created and enhanced during optimization. Many factors have an impact on the quality of the optimization algorithm, such as the number of features, diversity of training sets, the number of samples, and representativeness [31]. Optimization algorithms can be classified into two categories: deterministic and stochastic methods. Deterministic algorithms may be a good choice when the gradient is accessible. This variety of methods is based on predetermined calculations, implying that the optimization process is replicable. A reproducible process means that the optimization path and the final solution will be identical every time the algorithm is executed with the same initial states. In contrast, stochastic methods have random features that generate different optimization pathways, leading to other optimized solutions, even if the initial conditions are identical at each run. The main advantage of stochastic algorithms is that they do not need gradient information. Gradient information is hard to access due to the problems becoming more and more complex. Researchers have been developing stochastic algorithms in which no gradient information is needed. The development of more advanced stochastic methods has produced the category of meta-heuristics algorithms [30].

Meta-heuristics algorithms have gained enormous interest in multilevel image thresholding [32]. For the past years, researchers have widely demonstrated meta-heuristic algorithms' ability to solve many kinds of difficult optimization problems in engineering, communications, transportation, industry, and social sciences because these algorithms have high performance and are easy to implement [33]. Researchers have proposed meta-heuristic algorithms to solve the multilevel thresholding problems by maximizing some specific criteria for research. Some of the most widespread meta-heuristic algorithms include a genetic algorithm (GA) [34], cuckoo search (CS) [35], grey wolf optimization (GWO) [36], social engineering optimizer (SEO) [37], differential evolution (DE) [38], colony optimization (ACO) [39], sine cosine algorithm (SCA) [40], Harris hawks optimization (HHO) [41], and moth flame optimization (MFO) [42]. Zhao et al. [43] introduced manta ray foraging optimization (MRFO) as one of the most recent additions. MRFO showed more high-quality results than several classical and the latest counterparts.

Most optimization algorithms recorded exemplary achievements in multilevel thresholding, as all can get the optimal threshold result; however, they meet particular challenges because of the trapping in local regions, early conversion, and missing global searchability. These lay grounds for researchers to propose modified and hybrid versions and improve techniques. For the optimization problems, evaluating a candidate and its opposite solution simultaneously can increase the convergence rate toward a globally optimal solution. Opposition-based learning (OBL) [44] is one of the healthiest approaches to enhance the search performance of the meta-heuristic algorithms [45]. In [46], OBL was implemented to increase the efficiency of meta-heuristic optimization methods. OBL has regraded the suggested solutions built through a stochastic repetition system and their "opposite solutions" located in the search space's opposite regions. It has been combined with several bioinspired optimizations to provide smaller predicted distances to the globally optimal relative to randomly experimented solution pairs [47], such as cuckoo optimization algorithm [48], shuffled complex evolution algorithm [49], and fireworks algorithm [50]. In [51], the authors applied the OBL to enhance the Equilibrium Optimizer adopted for image segmentation using Otsu's function to obtain the best threshold values for image segmentation.

A new meta-heuristic swarm optimization algorithm called MRFO algorithm has been applied in structural optimization [43]. Despite its efficacy in search mechanisms, certain areas can be improved to prove its efficiency on challenging optimization problems, like image thresholding, because the algorithm may miss some critical search regions. We integrated the MRFO algorithm with OBL (MRFO-OBL) to generate solutions from potential regions and explore search space more rigorously to address this issue. More importantly, we improved its local searchability with the help of solutions generated from around the promising regions, avoiding traps in local optima. Consequently, the proposed MRFO variant comes with a trade-off equilibrium among exploration and exploitation. We then applied the MRFO-OBL to a multilevel image thresholding problem.

The following are the contributions of this paper:

- Boosted MRFO using OBL to solve image segmentation problems has been proposed.
- We applied the proposed segmentation algorithm to COVID-19 CT images.
- Different segmentation levels are used to assess the stability and the efficiency of the proposed method.
- Several well-known meta-heuristics are compared with the efficiency of the proposed method.
- Validate segmentation efficiency according to peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) values.
- The proposed method can be generalized to different medical imaging diagnoses and applied to different benchmark images.

The rest of the paper is structured as follows: Sect. 2 explains some related works. Section 3 confers an overview of the used methods. Section 4 introduces the proposed method. The experimental results derived from the proposed method to address multilevel thresholding are discussed in Sect. 5. Section 6 concludes the proposed method and discusses future work.

2 Literature review

The image segmentation problem plays a crucial role in image processing and computer vision, which are present in many fields, such as medical diagnosis, object recognition, satellite image processing, and more. There are more methods used to solve the problem of image segmentation. Multilevel thresholding segmentation is considered the best one for solving the image segmentation problem. Due to the time complexity problem with the increased threshold levels, traditional techniques failed to solve the image segmentation problem. Thus, the meta-heuristic algorithms have been more used to solve these problems and overcome the time complexity. This section introduces an overview of the recent works for overcoming the image segmentation problem of medical imaging and standard imaging and shows some recent works of segmentation of COVID-19 medical imaging.

Table 1 summarizes some optimization algorithms used for image segmentation. Moreover, several researchers work on medical imaging segmentation using optimization algorithms; for example, in [52], the authors proposed a method MRI image segmentation using the LASHED optimization algorithm. The statistically verified results demonstrate that the suggested approach improves consistency and segmentation quality. Furthermore, [53, 54] also using the optimization algorithms for medical image segmentation. The researchers in [55] proposed a multilevel thresholding method for medical image segmentation based on a partitioned and cooperative quantum-behaved PSO. The proposed method was tested with four stomach CT images and compared with two modified PSO algorithms. In [56], the authors presented an evolutionary grey gradient algorithm-based multilevel thresholding technique for brain MRI image segmentation (EGGA). To improve the fitness function, they used an adaptive swallow swarm optimization (ASSO) technique. They evaluated the ASSO using twenty-five MR scans, which revealed that it performed better than the original SSO. While in [57], the authors presented an enhanced method of the FPA algorithm to tackle the problem of medical image segmentation. They applied Otsu's method as an objective function and tested the proposed method using eight CT images. This method outperformed several meta-heuristics

algorithms (the original FPA, PSO, GA, and DE). In [58], the authors proposed a segmentation method for MRI brain images using the artificial bee colony algorithm (ABC). They enhanced the ABC algorithm by using the mean best-guided approach. They test the proposed method on 20 MRI images.

Also, the multilevel thresholding has been used for COVID-19 medical imaging segmentation. Therefore, in this paper, some methods for COVID-19 imaging segmentation are presented. In [7], the authors proposed a segmentation method for CT-COVID-19 images based on marine predators algorithm (MPA) and MFO algorithm using the fuzzy entropy fitness function for multilevel thresholding segmentation. They enhanced the MPA algorithm using the MFO algorithm and used 13 CT images to test its performance. In [59], the researchers presented a multilevel thresholding segmentation method for COVID-19 chest X-ray images by hybridized two meta-heuristics algorithms (Slime mould algorithm and WOA algorithm). They applied the Kapur method as an objective function and compared it with five optimization algorithms. Also, in [60], the authors applied the MPA algorithm to diagnose the COVID-19 X-ray images and obtain a good performance. While the authors in [61] proposed a model to predict the COVID-19 using the machine learning techniques with the nature-inspired algorithms. The proposed method was compared against more existing methods and tested on the same datasets.

As conferred from the previous researches mentioned above, image segmentation of medical images, especially in COVID-19 images, is critical to detect the infection with COVID-19.

3 Materials and methods

This section discusses the materials and methods needed for developing the proposed method. The basic MRFO algorithm and its structure are clarified, and some fundamental concepts of the OBL strategy and the objective function are described.

3.1 Image thresholding method: Otsu's method

This subsection explains the standard method's mathematical model used in image thresholding; namely, Otsu's method [17]. The segmentation method is based on the image histogram [75], which carries the pixel's distribution over the image. This method takes the image histogram as the input and then finds the optimal threshold values to its criteria. Otsu's method is one of the segmentation methods used to achieve the image's best threshold values by maximizing the variance between classes and determining

Table 1 List of some algorithms that work for the problem of image segmentation

Reference no.	Year	Algorithm	Technique
[62]	2017	MFO–WOA	Both MFO and WOA algorithms were used for multilevel thresholding segmentation. The proposed method used Otsu's as the fitness function and tested both WOA and MFO using several images
[63]	2020	ABC–SCA	In this method, a hybrid of the ABC algorithm and the SCA algorithm was proposed for multilevel thresholding image segmentation. The SCA is employed as a local search for the ABC to boost its performance. This model obtains good performances compared to several existing meta-heuristics methods
[10]	2020	HHO	The HHO algorithm is used for image segmentation and applied the minimum cross-entropy as a fitness function. The performance of the algorithm has been tested in standard images and digital mammograms. The proposed method is verified based on other comparable optimizers and two machine learning algorithms (K-means and the Fuzzy IterAg)
[64]	2015	FFO	The FFO algorithm has been proposed to maximize Otsu's variance to solve time-consuming and low-accuracy problems in multilevel thresholding image segmentation
[65]	2020	EO	The EO algorithm was used to find the optimal threshold value for a grayscale image and applied the Kapur entropy as a fitness function. The performance of this algorithm is compared with seven other algorithms
[66]	2016	CS	This paper introduced the comparative performance study of different objective functions using cuckoo search and other optimization algorithms to solve the color image segmentation problem using Otsu or Kapur's method
[67]	2018	ABC	This method presented an Otsu segmentation method based on the ABC algorithm
[68]	2020	PSO	This technique was used to segment the color images
[32]	2019	WOA–GWO–PSO	This method used three meta-heuristics algorithms for multilevel thresholding image segmentation to maximize the Otsu method. It tested on 20 benchmark test images using six different thresholds
[69]	2018	Firefly algorithm (FA)	This is a technique for multilevel color image thresholding used the fuzzy entropy as a fitness function and enhanced the FA algorithm by Levy flight (LF) strategy
[70]	2020	PSO	This paper proposed a non-revisiting quantum-behaved PSO (NrQPSO) algorithm to find the optimal multilevel thresholds for gray-level images using Kapur's entropy as an objective function
[71]	2020	Teaching learning based optimization algorithm (TLBO)	In this paper, LebTLBO was applied on ten standard test images and used the Otsu and Kapur's entropy objective functions for image segmentation and compared with the MTEMO, GA, PSO, and BF algorithms for both Otsu and Kapur's entropy methods. The results demonstrated that the LebTLBO outperforms the compared algorithms
[72]	2020	DE	This paper proposed a beta differential evolution (BDE)-based fast color image multilevel thresholding method using two objective functions (Kapur's and Tsallis entropy). The efficiency of the proposed method is examined over existing multilevel thresholding methods such as artificial bee colony, particle swarm optimization, wind-driven optimization, and differential evolution
[73]	2021	Black Widow optimization algorithm (BWO)	The BWO algorithm used to find the best threshold value to solve the multilevel thresholding image segmentation and used both Otsu and Kapur methods as objective functions
[74]	2019	Elephant herding optimization (EHO)	The authors enhance the EHO algorithm by the OBL and dynamic Cauchy mutation (DCM) to solve the multilevel image thresholding problem for image segmentation by maximizes two objective functions: Kapur's entropy and between-class variance

the optimum values for the thresholds that divide the image into many classes. This method identifies the gray image intensity levels L_v and determines the probability distribution in Eq. (1) [8].

$$h_i = \frac{h_i}{PN}, \sum_{i=1}^{PN} Ph_i = 1 \tag{1}$$

where i_l is the intensity level specified in $(0 \le i_l \le L_v - 1)$, PN represents the total number of pixels, h_i denotes the intensity number i_l in the image histogram. The histogram

has standardized in Ph_i. Depending on the probability distribution or thresholding value (th), the classes for bi-level segmentation have determined as:

$$C_1 = \frac{Ph_1}{\omega_0(th)}, \ldots, \frac{Ph_{th}}{\omega_0(th)} \text{ and } C_2 = \frac{Ph_{th+1}^c}{\omega_1(th)}, \ldots, \frac{Ph_L}{\omega_1(th)} \tag{2}$$

where $\omega_0(th)$ and $\omega_1(th)$ are additive probability distributions for C_1 and C_2, respectively, as shown in Eq. (3).

$$\omega_0(th) = \sum_{i=1}^{th} Ph_i \text{ and } \omega_1(th) = \sum_{th+1}^{L} Ph_i \tag{3}$$

It is mandatory to find the average intensity levels μ_0 and μ_1 using Eq. (4). When these values have been calculated, the Otsu-based between-class σ_B^2 is defined in Eq. (5).

$$\mu_0 = \sum_{i=1}^{th} \frac{iPh_i}{\omega_0(th)} \text{ and } \mu_1 = \sum_{i=th+1}^{L} \frac{iPh_i}{\omega_1(th)} \tag{4}$$

$$\sigma_B^2 = \sigma_1 + \sigma_2 \tag{5}$$

In Eq. (5), σ_1 and σ_2 are the C_1 variances specified as follows:

$$\sigma_1 = \omega_0(\mu_0 + \mu_T)^2 \text{ and } \sigma_2 = \omega_1(\mu_1 + \mu_T)^2 \tag{6}$$

where $\mu_T = \omega_0\mu_0 + \omega_1\mu_1$ and $\omega_0 + \omega_1 = 1$ based on σ_1 and σ_2. Equation (7) defines the objective function:

$$F_{otsu}(th) = \max(\sigma_B^2(th)) \text{ where } 0 \le th \le L - 1 \tag{7}$$

where $\sigma_B^2(th)$ is the Otsu's variance for a specified th value. The objective function $F_{otsu}(th)$ in Eq. (7) is updated for multiple thresholds as:

$$F_{otsu}(TH) = Max(\sigma_B^2(th)) \text{ where } 0 \le th_i \le L - 1, i = [1, 2, \ldots, q] \tag{8}$$

where $\mathbf{TH} = [th_1, th_2, \ldots, th_{q-1}]$ is a vector containing multiple thresholds and L denotes maximum gray level. The variances are calculated by Eq. (9).

$$\sigma_B^2 = \sum_{i=1}^{q} \sigma_i = \sum_{i=1}^{q} \omega_1(\mu_1 - \mu_T)^2 \tag{9}$$

where i defines a particular class and ω_i and μ_j are the probability of occurrence and the mean of a level. For multilevel thresholding, such values are obtained as follows:

$$\omega_{q-1}(th) = \sum_{i=th_q+1}^{L} Ph_i \tag{10}$$

for mean values:

$$\mu_{q-1} = \sum_{i=th_q+1}^{L} \frac{iPh_i}{\omega_1(th_q)} \tag{11}$$

3.2 Opposition-based learning (OBL)

OBL is a useful search strategy in bypassing stagnancy in candidate solutions [44]. Tizhoosh [46] proposed the OBL's basic idea, which improves the search mechanism's exploitation worth. Generally, when the initial solutions are near the optimal location in meta-heuristic algorithms, convergence happens instantly; otherwise, late convergence is expected. Here, the OBL strategy obtained better solutions by considering opposite search regions closer to the global optimum. The OBL works by searching both directions in the search space. One of the original solutions is used in these two ways, whereas the opposite solution defines the other direction. The OBL then takes the most suitable solution from all solutions [76].

- *Opposition number* The idea of opposite numbers can be described by explaining OBL. An opposition-based number can be identified as follows: Consider z_0 is a real number on an interval: $z_0 \in [q, w]$. The opposite number of Z_0 is defined by Eq. (12) [76]

$$\overline{z}_0 = q + w - z_0. \tag{12}$$

  Moreover, the opposite number in D-dimensional space is determined by Eqs. (13) and (14)

$$z = z_1, z_2, z_3, \ldots \ldots, z_D \tag{13}$$

$$\overline{z} = [\overline{z}_1, \overline{z}_2, \overline{z}_3, \ldots, \overline{z}_D] \tag{14}$$

  The values of all items in $\overline{z}$ are calculated using Eq. (15)

$$\overline{z}_k = q_k + w_k - z_k \text{ where } k = 1, 2, 3, \ldots, D \tag{15}$$

- *Opposition-based optimization* In the optimization process, the opposite point $\overline{z}_0$ is substituted by the corresponding solution z_0 as regards the fitness function. If $f(z_0)$ is better than $f(\overline{z}_0)$, then z_0 does not alter, oppositely, $z_0 = \overline{z}_0$. So, the solutions are updated as regards the best value of z and $\overline{z}$ [77].

3.3 Manta ray foraging optimization algorithm

Zhao et al. [43] designed the MRFO algorithm, a bioinspired optimizer. MRFO assumes the actions of manta rays in catching the prey. MRFO utilizes three foraging procedures of manta rays: chain foraging, cyclone foraging, and somersault foraging. Chain foraging mimics the process of necessary food searching. Foraging manta rays

systematically grasp up to capture the disappeared or undetected prey in the chain by the last manta ray. Cyclone foraging occurs when the collection amount of the prey is significant. The head is paired with the manta ray's tail, making a spiral to create an edge in a cyclone's eye. In somersault foraging, manta rays perform backward rotation and circle movements throughout as the prey planktons move them into their open lips [78]. Like various meta-heuristic algorithms, MRFO's initialization start is defined randomly to generate random positions for a set of agents X, followed by getting the most desirable agent X_{best} with the best fitness value. The agents are updated according to the aforementioned three strategies. The following are the mathematical models of these strategies.

1. *Chain foraging* In this process, the agent X_k is updated at iteration k using Eq. (16)

$$x_k^{t+1} = \begin{cases} x_k^t + r * (x_{best}^t - x_k^t) + \alpha(x_{best}^t - x_k^t) & k = 1 \\ x_k^t + r * (x_{k-1}^t - x_k^t) + \alpha(x_{best}^t - x_k^t) & k = 2, \ldots, N \end{cases}$$
(16)

$$\alpha = 2 * r * \sqrt{|\log(r)|}$$
(17)

where x_k^t is the k_{th} position in time k, r is a random vector in [0, 1], α is the weight coefficient, and x_{best}^t is the high-concentration plankton [43].

2. *Cyclone foraging* In this process, the agent X_k is updated following Eq. (18)

$$x_k^{t+1} = \begin{cases} x_{best}^t + r * (x_{best}^t - x_k^t) + \beta(x_{best}^t - x_k^t) & k = 1 \\ x_{best}^t + r * (x_{k-1}^t - x_k^t) + \beta(x_{best}^t - x_k^t) & k = 2, \ldots, N \end{cases}$$
(18)

$$\beta = 2 \cdot \exp\left(r \cdot \frac{K_{\max} - k + 1}{K_{\max}}\right) \cdot \sin(2\pi r)$$
(19)

where β is the weight coefficient. Furthermore, based on a random position provided in the search space, the cyclone foraging step agents can change their position to improve the MRFO's exploration. Thus, the current agent location is updated using the following equation:

$$x_k^{t+1} = \begin{cases} x_{rand}^k + r * (x_{rand}^t - x_k^t) + \beta(x_{rand}^t - x_k^t) & k = 1 \\ x_{rand}^t + r * (x_{k-1}^t - x_k^t) + \beta(x_{rand}^t - x_k^t) & k = 2, \ldots, N \end{cases}$$
(20)

$$x_{rand}^t = Low + r * (Up - Low)$$
(21)

where Low and Up are the lower and upper limits of the search space.

3. *Somersault foraging* The mathematical formulation used to update the agent in this phase is defined in the following equation:

$$x_k^{t+1} = x_k^t + s * (r_1.x_{best}^t - r_2.x_k^t)$$
(22)

where s is the somersault factor equal to 2, and r_1 and r_2 are the random numbers in [0, 1].

To conclude, the MRFO algorithm is initialized by generating a random population within specified permissible boundaries. The position-update procedure depends on the individual manta ray at the front of the recent one and the estimated pivot position. Changing from exploration to exploitation phase depends on the ($itr/maxitr$) ratio value. The exploitation phase is determined when ($itr/maxitr < r$), in which the current most suitable position is recognized as a pivot position. The algorithm moves to the exploration phase when ($itr/maxitr > r$). Furthermore, the algorithm can shift between chain foraging and cyclone foraging based on a randomly generated number. Somersault foraging then takes steps to update the individuals' current status through the current best approach. These three distinct foraging mechanisms are conducted interchangeably to simultaneously achieve the optimization problem's optimal global solution and meet the predefined end criterion.

4 The MRFO-OBL algorithm

This section discusses in detail the implementation of the proposed MRFO-OBL algorithm for segmenting the CT chest for COVID-19. The MRFO-OBL model is illustrated in detail in the flowchart presented in Fig. 1. In the proposed method, the OBL is used to boost the convergence of the MRFO toward the global solution. OBL is a local search technique aimed to avoid the random population's disadvantages and increase the algorithm's convergence by enhancing the variety of its solutions. Consequently, the smaller steps enable the complete quest to scan the promising area rigorously. The proposed MRFO-OBL multilevel thresholding image segmentation method has explained with more details in the following phases:

1. *Initialization phase* The first step in the proposed method is to read the input image from the COVID-19 dataset and store it as the gray-scale image. It then obtains the input image histogram and calculates the probability distribution following Eq. (1). The next step initializes the MRFO-OBL parameters, such as maximum iterations it_{max}, population size N, and problem dimensions D. Like many other meta-heuristic algorithms, MRFO-OBL starts with the first population's random initialization x^0 and saving results. The OBL strategy can be applied to the MRFO in the initialization phase to improve the search process as follows:
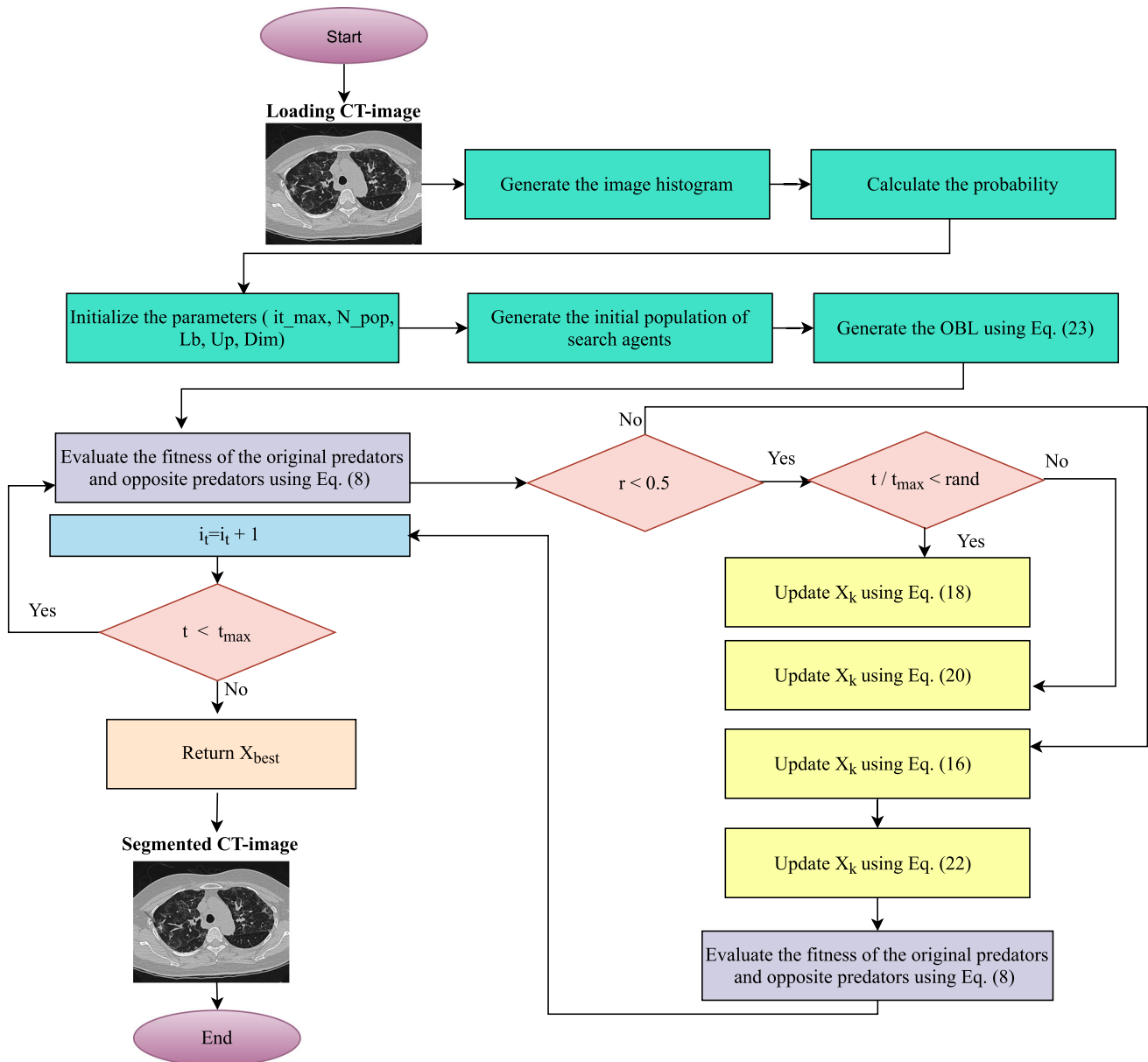
Fig. 1 Flowchart of the proposed method

$$Opp_i = lb_j + ub_j - x_i, i \in 1, 2, \ldots, N \qquad (23)$$

where Opp_i is a vector maintaining solution generated by applying OBL and lb_j and ub_j are the lower and upper bounds of the $j^t h$ component of a vector X.

2. *Apply OBL phase* OBL is used to find the opposite solution of each solution in the previous step. The OBL concept is then utilized to calculate the Opp_i vector of the initial population using Eq. (23).

3. *Optimization phase* This process begins by evaluating the x_i and Opp_i populations using Otsu's method in Eq. (8), followed by comparing the fitness of x_i and Opp_i and saving the best solution with the highest

fitness. The optimization process is split into three main phases of optimization, as discussed in Sect. 3.3. After applying the three phases of the optimization, the proposed method calculates and compares the fitness of x_i and Opp_i after using Otsu's method Eq. (8) and updates the global best solution found so far.

4. *The best solution phase* The steps of the two previous phases are performed again until the stop conditions are reached. The proposed method uses the total number of iterations as stop condition. The proposed method selects the best solution according to the better threshold values. Select the best thresholding values and apply them to the CT image in grayscale.

Algorithm 1 illustrates the pseudo-code of the proposed MRFO-OBL method.

Algorithm 1 The proposed algorithm: MRFO-OBL

Input: $population - size$ N, $Max - iteration$ it_{max}, $current - iteration$ it, lower bound l_b, upper bound u_b, Dimention D.
Output: The best solution.
Initialize search agents (manta ray) populations
while Stop condition not met **do**
    Apply OBL on the initial population using Eq. (23)
    for $k \leq N$ **do**
        Evaluate x_k using the Otsu's method Eq. (8)
        Evaluate Opp_k using the Otsu's method Eq. (8)
        if $Fit_k < FitOpp_k$ **then**
            $x_k = Opp_k$;
        end if
    end for
    for $k = 1$ to N **do**
        if rand< 0.5 **then**   ▷ Cyclone foraging
        if $coef$ < rand **then**
            Using Eq. (18) to update x_k
        else
            Using Eq. (20) to update x_k
        end if
        else   ▷ Chain foraging
            Using Eq. (16) to update x_k
        end if
    end for
    set $S = 2$
    for $k = 1$ to N **do**   ▷ Somersault foraging
        Using Eq. (22) to update x_k
    end for
    for $k = 1$ to N **do**
        Evaluate x_k using the Otsu's method Eq. (8)
        Evaluate Opp_k using the Otsu's method Eq. (8)
        if $Fit_k < FitOpp_k$ **then**
            $x_k = Opp_k$;
        end if
    end for
    for $k = 1$ to N **do**
        Update the fitness of each individual
        if $f(x_k) > f(\boldsymbol{x}_{\text{best}})$ **then**
            $\boldsymbol{x}_{\text{best}} = x_k$
        end if
    end for
end while
Return the best solution, which contains the best thresholds, and apply them to the CT images.

4.1 Evaluation criteria

It is important to check the pixel classification's accuracy for multilevel segmentation. The metrics used in this paper to measure the quality of the segmented images include the *peak signal-to-noise ratio (PSNR)* and the *structure similarity index (SSIM)*. These metrics are defined briefly as:

PSNR It is applied to validate the similarities among the original and segmented images using the Root Mean Square Error (RMSE) [79] of each pixel. PSNR and RMSE are represented as:

$$PSNR = 20 \log_{10}\left(\frac{255}{RMSE}\right), (\text{dB}) \tag{24}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{r_{ro}} \sum_{j=1}^{c_{co}} \left(I_{og}(i,j) - I_{segg}(i,j)\right)^2}{r_{ro} \times c_{co}}} \tag{25}$$

where I_{og} is the original image, I_{segg} is the segmented image, while r_{ro} and c_{co} are the maximum numbers of rows and columns of the images, respectively.

SSIM It is another measure of the similarity among the original and segmented image [80]. It is identifiable as follows:

$$SSIM(I_{og}, I_{segg}) = \frac{(2\mu_1\mu_{segg} + c_1)(2\sigma_{1,segg} + c_2)}{(\mu_I^2 + \mu_{segg}^2 + c_1)(\sigma_I^2 + \sigma_{segg}^2 + c_2)} \tag{26}$$

where μ_I and μ_{segg} are the mean intensities of the original image I_{og} and the segmented image I_{segg}, σ_I and σ_{segg} are the standard deviations of I_{og} and I_{seg}. $\sigma_{I_{og}, I_{segg}}$ is the covariance of I_{og} and I_{segg}, and c_1 and c_2 are constants. Moreover, to validate the quality of threshold values, the fitness value is used.

To statistically illustrate the significance of the proposed method, the *Wilcoxon rank-sum* test has been utilized to determine the difference between MRFO-OBL and all other methods used in the comparison [81]. It is a non-parametric test to compare the outcomes of each pair of methods [82]. This test is based on two assumptions: (i) null assumption, which designates no difference between the ranks of a pair of algorithms extracted from the results, and (ii) alternative assumption, which affirms a discrepancy between the results' ranks produced by a pair of algorithms. It is based on a 5% significant level. According to the P values, if $(P > 0.05)$, then the null assumption is true. If $(P < 0.05)$, then the alternative assumption is true.

5 Experimental results and discussion

In this section, the experiments have conducted to verify the proposed MRFO-OBL method's performance and compare their performance with some of the state-of-the-art algorithms and the original MRFO algorithm to tackle the image segmentation problem. The experiments were conducted on a set of chest CT COVID-19 images, namely CT-image1, CT-image2, CT-image3, CT-image4, CT-image5, CT-image6, CT-image7, CT-image8, CT-image9, and CT-image10 as shown in Fig. 3. The proposed method was evaluated in terms of the fitness function, also to assess the quality of the segmented image, we used a set of performance metrics, including the PSNR and SSIM. The reminder of this section is organized as follows: Sect. 5.1 shows the dataset used in the experiments. Section 5.2 explains the environmental setup. Section 5.3 describes the experimental settings. Section 5.4 shows Otsu's fitness, PSNR, SSIM, and Wilcoxon rank-sum test results.

5.1 Dataset description

In this paper, computed tomography (CT) images for COVID-19 from [83, 84] datasets used to test the proposed method. The CT COVID-19 dataset has 349 CT images containing clinical findings of COVID-19 from 216 patients. The COVID-19 images were collected from patients with ages ranging from 40 to 84 of both genders. Figure 2 shows case study of samples of the CT COVID-19 dataset and metadata for each sample. The proposed method evaluates ten test images from this dataset for different patients to test the proposed method's performance. The test images are labeled as CT-image1, CT-image2, ..., CT-image10. Figure 3 displays the selected test images and their corresponding histograms.

5.2 Experimental setup

The efficiency of the proposed method is compared to six meta-heuristic algorithms which are MFO [42], whale optimization algorithm (WOA) [85], SCA [40], slap swarm algorithm (SSA) [86], equilibrium optimization (EO) [87], and the original MRFO [43]. The COVID-19 CT images are segmented with 7, 8, 9, and 10 thresholds. Also, the proposed method's assessment is achieved according to fitness, PSNR, and SSIM. Both algorithms have been coded and run on MATLAB R2016b on a computer running Windows 10 (64 bit), Intel Core I7, and 8 GB RAM.

5.3 Parameter settings

Table 2 presents the default parameters values of each algorithm that are applied in the experiments. As mentioned in [88], default parameter values is a fair parametrization. Moreover, employing default values reduce comparison bias risks as no algorithm could be advantaged with a better parametrization. To ensure fair benchmarking comparison, the proposed and counterparts algorithms are evaluated over 30 independent runs with a maximum number of 500 iterations for each test image, and the population size is set to 30.

5.4 MRFO-OBL results and discussion

This subsection displays and discusses the proposed method results. The Otsu's method in Eq. (8) has been used as an objective function. Tables 3 and 4 show the segmented COVID-19 CT images obtained from the MRFO-OBL with different numbers of thresholds $nTh = 7, 8, 9, 10$ outlined over histograms.

Table 5 shows the gained threshold values at level 7 outlined over the CT-image3 histogram and the segmented images at the same level acquired by the compared algorithms. Table 6 shows the gained threshold values at level 9 outlined over the CT-image10 histogram and the segmented images at the same level acquired by the compared algorithms.

Tables 7, 8, and 9 include the mean results of fitness, PSNR, and SSIM evaluation matrices, respectively. A higher value means a more reliable and more effective algorithm. *In summary, the following observation from the experiments are worth mentioning:*

- *In terms of fitness values* Table 7 shows the fitness values for each level in the comparison of MRFO-OBL with all other algorithms. The best values are presented in bold. Higher fitness values were achieved using the MRFO-OBL and MRFO algorithms, but one gave better results than the other for a set of test images. For CT-image6, CT-image9, and CT-image10 images, the MRFO-OBL algorithm values are the best for all or at least three levels. Moreover, for CT-image1, CT-image4, CT-image5, and CT-image8, MRFO-OBL has higher values for only two levels, while the CT-image3 image obtained only one higher value for level 9. In MRFO, the CT-image7 image acquired higher values for each level, and the CT-image2 image acquired higher values for three levels (level 8, 9, and 10). Also, for CT-image4, CT-image5, and CT-image8, the MRFO algorithm has higher fitness values at only two levels comparing with other algorithms, while for images CT-image1, CT-image3, and CT-image9 has

Age	Gender	Location	Medical History	Time
41	Male	Wuhan, China	No history of hepatitis, tuberculosis or diabetes.	Day 6 after the onset of illness

Clinical Diagnosis
Chest tightness, unproductive cough, pain and weakness for one week on presentation. Fever with body temperatures ranging from 37.2oC to 40oC. Severe cough with phlegm at onset of illness. Experienced dizziness. respiratory failure, was given high flow noninvasive ventilation. Physical examination of cardiovascular, abdominal and neurologic examination was normal. Mild lymphopenia (less than 900 cells per cubic milli-meter) was observed, but white blood cell and blood platelet count was normal in a complete blood count (CBC) test. Elevated levels of C-reactive protein (CRP, 41.4 mg/L of blood, reference range 0-6 mg/L) was observed and levels of aspartate aminotransferase, lactic dehydrogenase, and creatine kinase were slightly elevated in blood chemistry tests.

(a) Sample 1   **(b)** Metadata of sample 1

Age	Gender	Location	Medical History	Time
50	Male	Beijing, China	No history of hepatitis, tuberculosis or diabetes.	Illness Day 8, Hospital Day 0

Clinical Diagnosis
Early stage: Ground glass shadow in multiple lobes and segments of bilateral lungs, and the lesions were adjacent to the pleura.

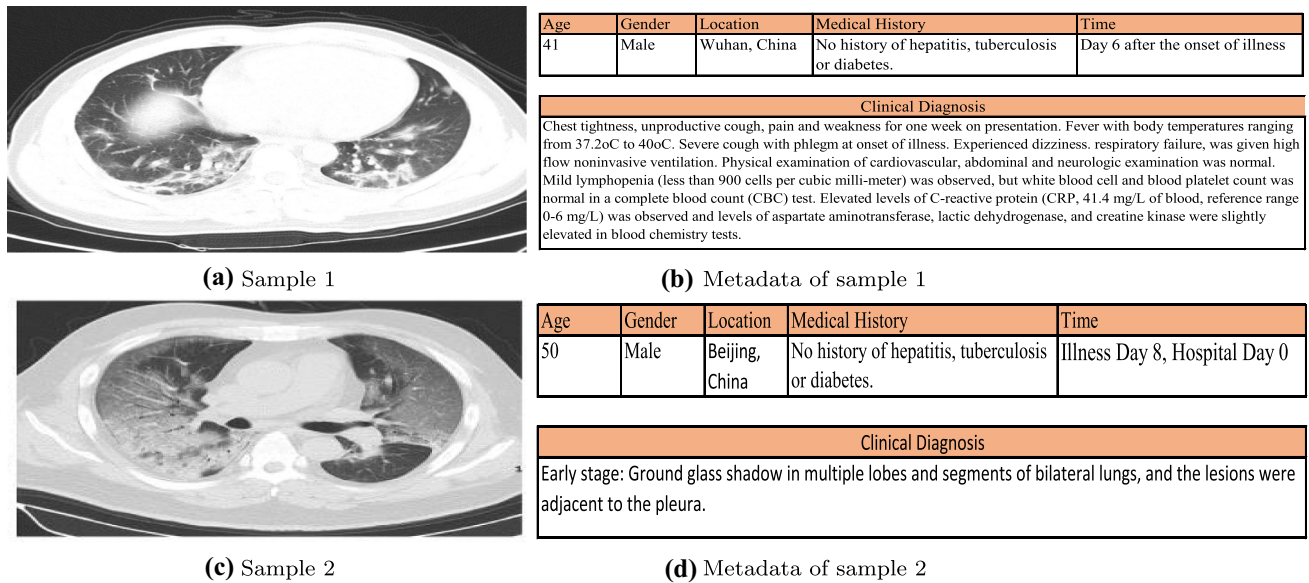**(c)** Sample 2   **(d)** Metadata of sample 2

Fig. 2 Some images of the CT COVID-19 dataset [83] and its metadata

obtained higher values at only one level. The best values were acquired for level 7 and level 8 at only three test images in the SSA algorithm. On the other hand, the EO, WOA, SCA, and MFO algorithms have not produced any best value in either image. In general, the MRFO-OBL and MRFO outperform all algorithms.

- *In terms of PSNR values* Table 8 shows the algorithms' PSNR values. It provides an estimate of the similarity between the segmented image and the original, as mentioned above, where a higher value represents better quality segmentation. Higher PSNR values were generally achieved using the MRFO-OBL and EO algorithms, but one gave improved results than the other for a set of test images. In MRFO-OBL, the PSNR obtained for the CT-image2 image gives higher values for all levels than the compared algorithms, while the CT-image3 and CT-image4 images show the best results compared to the other algorithms for all levels except level 10. Also, in CT-image1 and CT-image10, it has higher values for only two levels (level 9, level 10), and (level 8, level 9), respectively. For CT-image7 and CT-image9, the MRFO-OBL has only one higher value at level 10 and level 9, respectively. In EO, the images CT-image6, CT-image7, and CT-image9 acquired higher values for three levels, while the images CT-image1 and CT-image8 have higher values for only two levels (level 7, level 8) and (level 8, level 10), respectively. In CT-image3, CT-image4, and CT-image10, the EO has higher values than other comparable algorithms at only one level. In WOA, the CT-image5 image acquired for all levels than the compared algorithms, while in CT-image10 has only one higher

value at level 10. The best PSNR values were achieved in the SSA algorithm for two levels only in the CT-image8 image. In the MRFO algorithm, the higher PSNR value is achieved only one at level 8 for image CT-image6. The MFO and SCA algorithms do not give any high PSNR results in any of the images. As a result, the highest PSNR values are generated through the MRFO-OBL and EO algorithms, displaying the most tested images' greatest efficiency, indicating a great affiliation between the original and the segmented image.

- *In terms of SSIM values* According to the SSIM values in Table 9, it was perceived that the obtained SSIM values in the MRFO-OBL are better than those all algorithms compared, implying significance in most images, particularly in the CT-image1, CT-image2, CT-image3, CT-image4, CT-image6, CT-image7, and CT-image10. In the EO algorithm, the SSIM higher values were obtained for image CT-image9, while for image CT-image5, the SSIM higher values were achieved by the WOA algorithm. The MFO algorithm has only one high SSIM value for CT-image4 image at level 10. The MRFO and SCA algorithms do not have any higher SSIM values at any test image.

The following points can be observed from this analysis: Table 10 shows the Wilcoxon rank-sum test results for the best values applied on PSNR between MRFO-OBL and each counterpart. As mentioned above, according to the P values, if $(P > 0.05)$, then the null assumption is true. If $(P < 0.05)$, then the alternative assumption is true. To clarify the discussion of these values, the symbols $++$ and $-$ are involved. The $++$ symbol means a significant

(a) CT-image1   **(b)** Histogram of (a)   **(c)** CT-image2   **(d)** Histogram of (c)

(e) CT-image3   **(f)** Histogram of (e)   **(g)** CT-image4   **(h)** Histogram of (g)

(i) CT-image5   **(j)** Histogram of (i)   **(k)** CT-image6   **(l)** Histogram of (k)

(m) CT-image7   **(n)** Histogram of (m)   **(o)** CT-image8   **(p)** Histogram of (o)

(q) CT-image9   **(r)** Histogram of (q)   **(s)** CT-image10   **(t)** Histogram of (s)
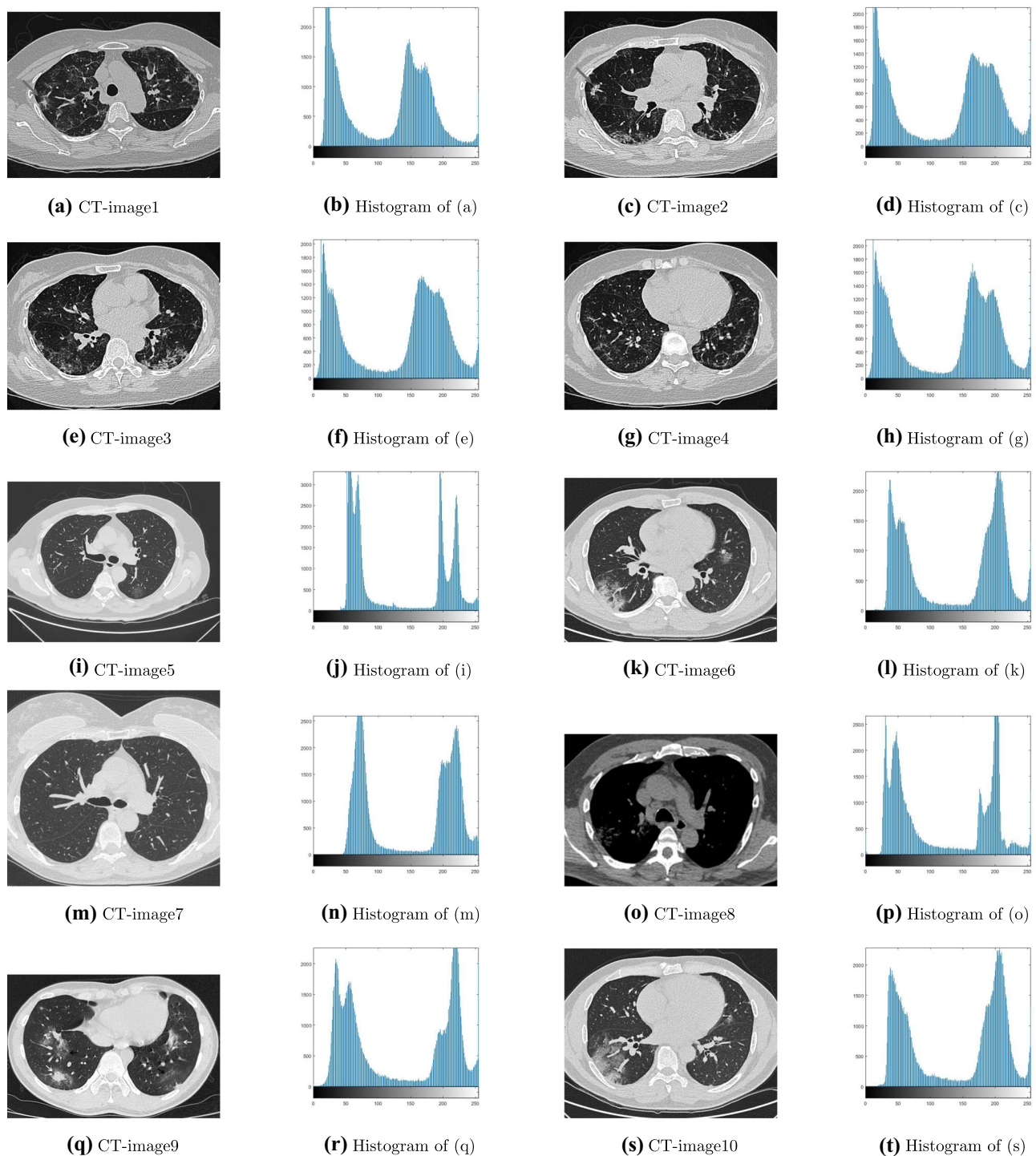
Fig. 3 COVID-19 CT test images and their corresponding histograms

difference at level $(P < 0.05)$, which implies MRFO-OBL performs better than the compared algorithms, while − means MRFO-OBL performance is similar or worse than the compared algorithms. According to this table, the numbers of $(P < 0.05)$ are 17 (MRFO-OBL vs. MFO), 27 (MRFO-OBL vs. WOA), 40 (MRFO-OBL vs. SCA), 31 (MRFO-OBL vs. SSA), 11 (MRFO-OBL vs. EO), and 17 (MRFO-OBL vs. MRFO), respectively. So, it is feasible to recognize that the MRFO-OBL has a significant difference from the compared algorithms.

Table 2 Parameter settings for MRFO-OBL and the compared algorithms	Algorithm name	Parameters setting
	Common parameters	Number of population $N = 30$
		Maximum iterations $It_{max} = 500$
		Number of runs 30
	MFO	$b = 1$ and a decreases linearly from -1 to -2 (default)
	WOA	α decreases linearly from 2 to 0 (default)
		$a2$ decreases linearly from -1 to -2 (default)
	SCA	$A = 2$ (default)
	SSA	$c1$ as defined in the original research
	EO	$a_1 = 2, a_2 = 1$ (default)
	MRFO and MRFO-OBL	$S = 2$ (default)

Table 3 Segmented images and threshold values acquired by the proposed method over the test images' histograms

Image	nTh = 7	nTh = 8	nTh = 9	nTh = 10
CT-image1				
CT-image2				
CT-image3				
CT-image4				
CT-image5				

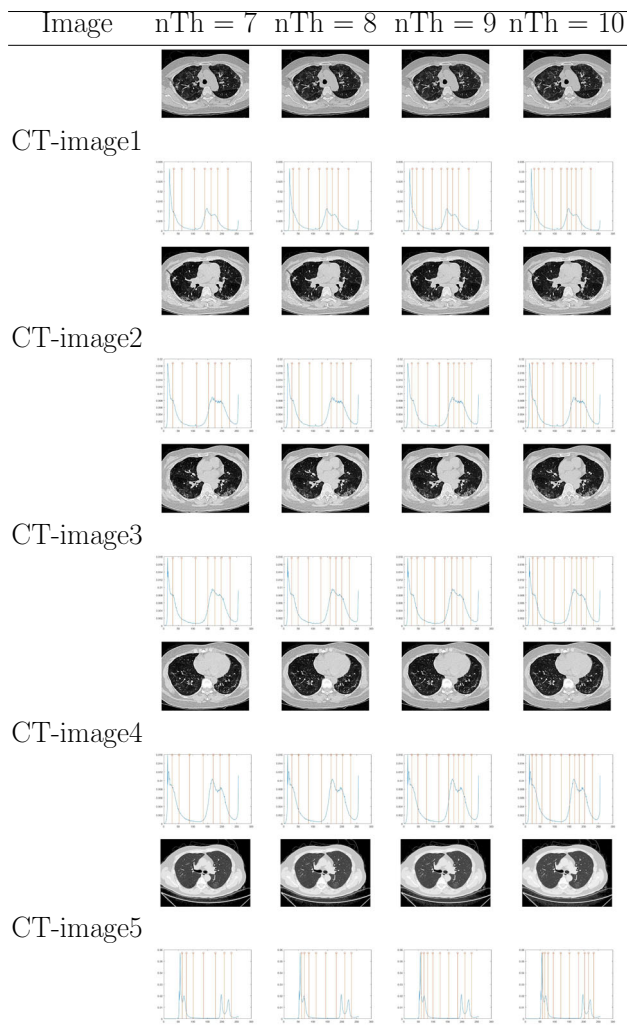**Table 4** Segmented images and threshold values acquired by the proposed method over the test images' histograms

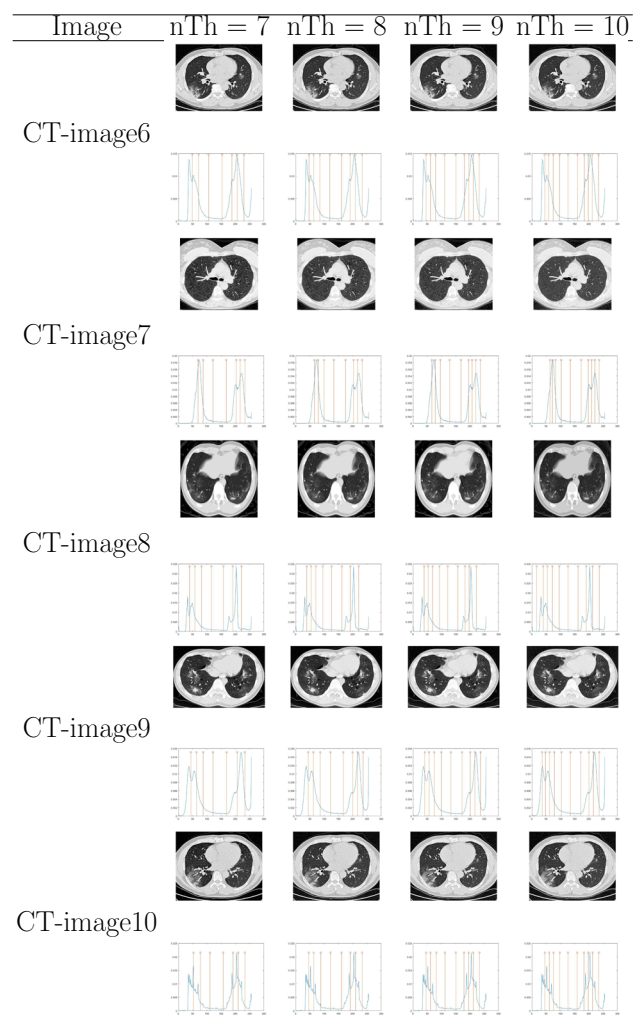| Image	nTh = 7	nTh = 8	nTh = 9	nTh = 10
CT-image6				
CT-image7				
CT-image8				
CT-image9				
CT-image10				

Table 5 Segmented image and its histograms acquired by the compared algorithms over CT-image3
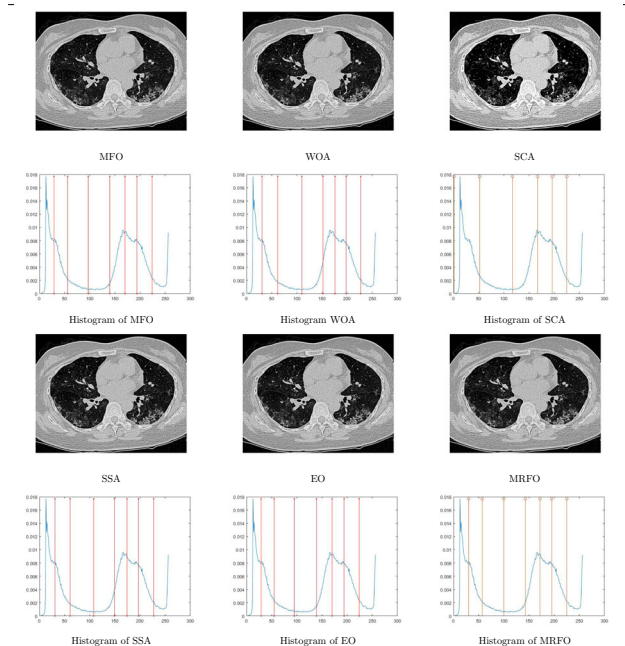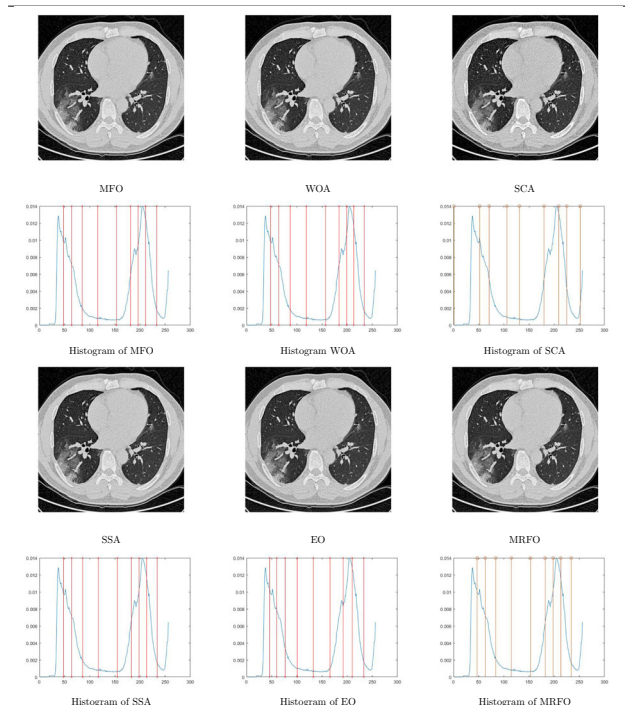

Table 6 Segmented image and its histograms acquired by the compared algorithms over CT-image10

Particularly, the major outcomes of the results reported in Table 10 are summarized as follows: the proposed MRFO-OBL method has a better quality for segmentation.

6 Conclusion and future work

With the spread of COVID-19 worldwide since December 2019, the entire world has moved to find techniques to help distinguish infected persons from normal ones. After many efforts and confirmation by the medical specialists, CT images could significantly identify whether the suspected patients have been infected. Various segmentation methods are applied to extract regions of interest from CT images essential to improve the classification methods. Medical image segmentation considers a crucial step for many medical applications that need to be done correctly for practical image analysis. One of the most primary and essential techniques for image segmentation is thresholding. In this paper, finding the optimum thresholding values in multilevel thresholding image segmentation was considered an optimization problem where Otsu's method has been used as an objective function. So, a new enhanced version of the MRFO algorithm has been introduced to solve this problem. This algorithm aims to determine the best threshold values that maximize Otsu's function. This method depends on the OBL strategy to improve MRFO's ability to reach the optimum threshold value. The proposed method has been used for COVID-19 CT image segmentation. The performance evaluation of MRFO-OBL is evaluated using 10 COVID-19 CT images with threshold numbers *nTh* from 7 to 10 and compared to six metaheuristic algorithms: MFO, WOA, SCA, SSA, EO, and the original MRFO. The proposed method's performance has been assessed based on three measures, the best fitness values, PSNR, and SSIM metrics. The results showed that the proposed method gets good results compared to the other competed algorithms. In the future, we will extend the proposed method to test in other applications such as feature selection and color image segmentation and increase the number of thresholds to attain more reliable results. We will also hybridize the proposed method with different meta-heuristic optimization algorithms to improve the segmentation results and applied more activation functions such as Kapur entropy and Fuzzy entropy.

Table 7 Comparison between MRFO-OBL and all other algorithms according to the fitness mean values

Test image	nTh	MFO	WOA	SCA	SSA	EO	MRFO	MRFO-OBL
CT-image1	7	4590.9223	4590.7594	4550.3620	4591.2335	4590.8824	4591.2189	**4591.2581**
	8	4602.6112	4601.9551	4563.1350	**4603.1168**	4602.6202	4603.1046	4603.1032
	9	4610.4316	4609.9864	4573.4895	4610.5302	4609.8343	4610.8964	**4610.9196**
	10	4615.9072	4615.2665	4580.5554	4616.5550	4615.5992	**4616.7518**	4616.6997
CT-image2	7	6162.5203	6162.7166	6121.6036	**6162.5849**	6162.4394	6162.5764	6162.5092
	8	6176.2519	6176.3151	6140.7819	6176.7048	6176.2249	**6176.7342**	6176.7112
	9	6185.0236	6185.7708	6146.6306	6185.3138	6184.6617	**6185.6179**	6185.4133
	10	6191.3828	6192.1284	6151.4200	6192.1168	6191.0136	**6192.3022**	6192.2280
CT-image3	7	5878.3160	5878.5465	5835.4659	**5878.4737**	5878.3046	5878.4162	5878.4337
	8	5892.6010	5892.4689	5848.8000	**5892.9150**	5892.6781	5892.9090	5892.9051
	9	5901.2130	5900.9025	5862.0889	5901.4772	5901.0282	5901.3900	**5901.4849**
	10	5907.5981	5908.1545	5867.3855	5908.4200	5907.4222	**5908.5157**	5908.3732
CT-image4	7	5889.6072	5889.3875	5857.1493	5889.8926	5889.4591	**5889.7649**	5889.6231
	8	5903.2803	5903.7679	5862.5667	5903.7527	5903.3091	5903.7270	**5903.7288**
	9	5911.4146	5911.2663	5874.8074	5911.7367	5910.8755	5911.6846	**5911.6794**
	10	5917.7558	5917.9608	5882.0751	5918.2202	5917.2108	**5918.5509**	5918.4579
CT-image5	7	5304.3801	5302.4643	5277.2172	5304.0706	5304.1015	5304.4442	**5304.5220**
	8	5307.9534	5306.5537	5284.4521	5307.4206	5307.3813	**5308.2301**	5308.2115
	9	5311.0056	5309.9221	5289.7828	5310.6143	5310.3068	**5311.4731**	5311.3428
	10	5313.6503	5312.2744	5291.5363	5312.9069	5313.0356	5313.8453	**5313.9448**
CT-image6	7	5582.7957	5582.4661	5550.0622	5582.8830	5582.7968	5583.4178	**5583.4676**
	8	5591.8048	5589.5535	5554.1609	5591.6058	5591.8745	5592.3048	**5592.3607**
	9	5599.0451	5597.6480	5566.4085	5598.2920	5598.1596	5599.4109	**5599.4356**
	10	5603.3965	5602.8205	5571.1188	5603.3789	5603.2528	5604.0824	**5604.1412**
CT-image7	7	4995.2435	4991.8540	4970.8321	4994.6152	4994.6073	**4995.6833**	4995.4828
	8	5001.8768	5000.0718	4975.5508	5001.0978	5000.8706	**5002.8458**	5002.8223
	9	5006.5489	5004.4128	4979.8987	5005.9421	5004.9428	**5006.9064**	5006.6069
	10	5010.0312	5007.7376	4986.0513	5009.1688	5008.6692	**5010.3229**	5010.1842
CT-image8	7	4782.6167	4782.4267	4754.9228	4782.7783	4782.2585	4782.8771	**4782.9952**
	8	4790.9260	4790.8360	4765.0191	4791.3610	4790.2853	4791.3374	**4791.3929**
	9	4797.8587	4798.3456	4770.5763	4798.1746	4798.0153	**4798.3585**	4798.3519
	10	4802.7967	4803.2004	4777.8819	4803.2020	4802.9119	**4803.4234**	4803.4026
CT-image9	7	6611.2568	6609.7389	6574.1046	6611.0689	6611.3992	6611.4055	**6611.4420**
	8	6620.7094	6620.1026	6585.8934	6620.9329	6620.1422	**6621.2717**	6621.0636
	9	6628.1612	6627.4954	6593.0302	6628.5247	6627.2801	6629.1017	**6629.1619**
	10	6633.6733	6633.1912	6598.2853	6633.8207	6633.8293	6634.0570	**6634.0668**
CT-image10	7	5541.8543	5541.3134	5508.8022	5541.9334	5541.4640	5542.3078	**5542.3602**
	8	5551.2935	5548.4211	5517.1734	5551.1612	5551.3348	5551.5943	**5551.7720**
	9	5558.2501	5556.1898	5522.4249	5558.6341	5557.7141	5559.2996	**5559.3458**
	10	5563.3573	5563.0640	5530.5983	5563.1279	5563.2524	5564.0836	**5564.3549**

Table 8 Comparison between MRFO-OBL and all other algorithms according to the PSNR mean values

Test image	nTh	MFO	WOA	SCA	SSA	EO	MRFO	MRFO-OBL
CT-image1	7	22.7852	22.7217	20.4506	22.7112	**22.8869**	22.7883	22.8366
	8	23.6051	23.6016	21.4088	23.5316	**23.6355**	23.5724	23.6096
	9	24.3834	24.3157	22.0108	24.3711	24.5068	24.4058	**24.4777**
	10	24.9569	24.9986	22.7393	24.9413	25.2453	25.0747	**25.0750**
CT-image2	7	23.3401	23.2898	21.0569	23.3548	23.3451	23.3543	**23.3691**
	8	24.2526	24.2280	22.2538	24.2437	24.2915	24.2500	**24.2945**
	9	25.1918	25.1816	22.4196	25.1842	25.1998	25.2425	**25.2675**
	10	25.9299	25.9645	22.6362	25.9269	26.0340	25.9906	**26.0447**
CT-image3	7	23.3127	23.3402	20.7045	23.3244	23.3664	23.3779	**23.3835**
	8	24.2557	24.2363	21.5903	24.2289	24.2585	24.2597	**24.2920**
	9	25.2191	25.0960	22.4194	25.1901	25.1978	25.2401	**25.2815**
	10	25.9385	25.9699	23.0266	25.9578	**26.0504**	26.0143	26.0406
CT-image4	7	23.1153	22.9441	21.0348	23.0054	23.0594	23.0455	**23.1315**
	8	24.1451	24.1102	21.4647	24.1141	24.1073	24.1247	**24.1701**
	9	25.2197	25.1061	22.2991	25.2064	25.1943	25.2826	**25.3379**
	10	26.0147	25.9803	22.7990	25.9808	**26.1555**	26.0265	26.0860
CT-image5	7	17.7297	**18.0911**	17.3997	17.8317	17.7346	17.6816	17.6923
	8	18.0694	**18.6874**	17.1534	18.3007	18.1689	18.0973	18.1808
	9	18.4704	**19.4253**	18.0159	19.0553	18.6530	18.4984	18.6158
	10	18.8423	**20.3143**	18.2669	19.1845	19.3208	18.9574	19.0474
CT-image6	7	21.1169	21.0827	19.8469	21.0500	**21.1999**	21.0786	21.1912
	8	22.0606	22.1863	20.3460	22.1110	22.2209	**22.2276**	22.2106
	9	22.9744	22.7802	20.8418	23.0122	**23.0780**	22.9386	23.0090
	10	23.4674	23.4204	22.0136	23.7771	**23.8257**	23.5627	23.6860
CT-image7	7	19.6031	19.3921	18.3899	19.4554	**19.7201**	19.4133	19.6895
	8	20.6718	20.3370	18.5243	20.6694	**20.8207**	20.7551	20.7020
	9	21.5069	21.4917	19.9200	21.5961	**21.8813**	21.4188	21.8512
	10	22.4764	22.9517	20.4139	22.1888	22.8390	22.6882	**22.9905**
CT-image8	7	24.1907	24.1384	22.3768	24.1360	**24.3791**	24.0661	24.1512
	8	25.6295	25.5060	23.0455	25.3830	**25.8864**	25.4969	25.6688
	9	27.0029	27.0489	23.5396	**27.1684**	26.9507	27.0617	27.0500
	10	27.8852	27.8808	24.2742	**27.9904**	27.9053	27.9826	27.9897
CT-image9	7	21.7801	21.6394	20.3303	21.7302	**21.8315**	21.7791	21.7893
	8	22.6114	22.4253	21.0667	22.4741	**22.7819**	22.5405	22.6872
	9	23.6409	23.5144	21.5845	23.4062	24.0351	23.6402	**23.7957**
	10	24.6765	24.1344	21.9324	24.5486	**25.0857**	24.7814	25.0439
CT-image10	7	20.5813	20.3124	19.4163	20.4473	**20.6924**	20.4594	20.6069
	8	21.9369	21.6074	20.4364	21.9044	22.1685	22.0541	**22.1712**
	9	22.9325	22.9381	20.8123	23.0184	23.0016	23.0701	**23.0850**
	10	23.4817	**23.8029**	22.0695	23.6455	23.7193	23.5813	23.6204

Table 9 Comparison between MRFO-OBL and all other algorithms according to the SSIM mean values

Test image	nTh	MFO	WOA	SCA	SSA	EO	MRFO	MRFO-OBL
CT-image1	7	0.8772	0.8747	0.8447	0.8731	0.8819	0.8763	**0.8865**
	8	0.8957	0.8977	0.8613	0.8934	0.8987	0.8955	**0.9035**
	9	0.9063	0.9052	0.8784	0.9056	0.9130	0.9059	**0.9138**
	10	0.9146	0.9171	0.8940	0.9128	**0.9240**	0.9171	0.9230
CT-image2	7	0.9147	0.9092	0.8887	0.9143	0.9185	0.9153	**0.9245**
	8	0.9275	0.9261	0.9109	0.9263	0.9300	0.9274	**0.9348**
	9	0.9378	0.9347	0.9141	0.9374	0.9416	0.9374	**0.9444**
	10	0.9453	0.9446	0.9129	0.9444	0.9511	0.9464	**0.9523**
CT-image3	7	0.9219	0.9199	0.8759	0.9209	0.9270	0.9249	**0.9303**
	8	0.9356	0.9359	0.9015	0.9350	0.9371	0.9364	**0.9443**
	9	0.9463	0.9422	0.9172	0.9446	0.9494	0.9467	**0.9526**
	10	0.9536	0.9532	0.9271	0.9525	0.9578	0.9550	**0.9605**
CT-image4	7	0.9168	0.9092	0.8900	0.9105	0.9215	0.9148	**0.9276**
	8	0.9323	0.9309	0.8977	0.9309	0.9327	0.9317	**0.9402**
	9	0.9450	0.9408	0.9148	0.9437	0.9492	0.9460	**0.9531**
	10	**0.9535**	0.9522	0.9196	0.9522	0.9593	0.9539	0.9595
CT-image5	7	0.7591	**0.7748**	0.7418	0.7626	0.7607	0.7568	0.7629
	8	0.7710	**0.7935**	0.7326	0.7824	0.7769	0.7725	0.7731
	9	0.7849	**0.8163**	0.7696	0.8036	0.7935	0.7864	0.7913
	10	0.7975	**0.8491**	0.7745	0.8107	0.8173	0.8021	0.8057
CT-image6	7	0.8856	0.8856	0.8574	0.8837	**0.8890**	0.8848	0.8872
	8	0.9060	0.9104	0.8695	0.9063	0.9119	0.9108	**0.9250**
	9	0.9207	0.9172	0.8815	0.9232	0.9251	0.9194	**0.9348**
	10	0.9279	0.9272	0.9080	0.9319	0.9357	0.9299	**0.9393**
CT-image7	7	0.7447	0.7407	0.7029	0.7418	0.7487	0.7386	**0.8484**
	8	0.7811	0.7704	0.7062	0.7826	0.7885	0.7856	**0.8799**
	9	0.8101	0.8059	0.7543	0.8127	0.8239	0.8067	**0.9031**
	10	0.8399	0.8441	0.7731	0.8304	0.8521	0.8469	**0.9250**
CT-image8	7	0.9291	0.9281	0.9040	**0.9285**	0.9315	0.9273	0.9263
	8	0.9426	0.9421	0.9129	**0.9396**	0.9496	0.9394	0.9389
	9	0.9575	0.9597	0.9198	0.9588	0.9588	0.9598	**0.9599**
	10	0.9641	0.9648	0.9270	0.9650	**0.9654**	0.9655	**0.9654**
CT-image9	7	0.8585	0.8517	0.8228	0.8542	0.8620	0.8583	**0.8670**
	8	0.8786	0.8719	0.8429	0.8733	**0.8870**	0.8754	0.8813
	9	0.9036	0.9009	0.8623	0.8964	**0.9172**	0.9041	0.9085
	10	0.9240	0.9111	0.8700	0.9206	**0.9346**	0.9267	0.9321
CT-image10	7	0.8711	0.8610	0.8456	0.8655	**0.8752**	0.8665	0.8656
	8	0.9060	0.8969	0.8729	0.9044	0.9150	0.9095	**0.9198**
	9	0.9247	0.9258	0.8818	0.9261	0.9284	0.9266	**0.9357**
	10	0.9322	0.9372	0.9163	0.9348	0.9368	0.9335	**0.9412**

Table 10 Comparison of P values acquired by the Wilcoxon signed-rank test between the pairs of MRFO-OBL versus the counterparts for PSNR results

Test image	nTh	MFO	WOA	SCA	SSA	EO	MRFO
CT-image1	7	6.834E−01 −	3.969E−04 ++	2.781E−11 ++	3.416E−06 ++	1.334E−01 −	6.836E−02 −
	8	6.308E−01 −	9.300E−02 −	2.031E−09 ++	3.748E−05 ++	2.739E−01 −	8.032E−02 −
	9	2.683E−03 ++	1.438E−08 ++	7.327E−11 ++	4.070E−05 ++	8.225E−02 −	1.237E−03 ++
	10	9.876E−03 ++	8.406E−03 ++	3.674E−11 ++	2.486E−06 ++	2.051E−06 ++	5.003E−01 −
CT-image2	7	5.741E−01 −	1.650E−03 ++	2.993E−11 ++	5.640E−01 −	1.710E−01 −	8.591E−01 −
	8	7.006E−02 −	7.296E−02 −	2.992E−11 ++	6.764E−03 ++	3.553E−01 −	1.060E−02 ++
	9	6.972E−03 ++	7.585E−09 ++	3.020E−11 ++	2.277E−05 ++	1.679E−03 ++	8.760E−02 −
	10	1.292E−05 ++	1.404E−04 ++	3.018E−11 ++	1.428E−08 ++	6.520E−01 −	1.701E−04 ++
CT-image3	7	6.794E−03 ++	8.026E−04 ++	3.012E−11 ++	3.767E−02 ++	7.900E−01 −	6.350E−01 −
	8	1.202E−01 −	7.031E−05 ++	2.977E−11 ++	1.551E−05 ++	6.721E−02 −	3.896E−03 ++
	9	4.059E−02 ++	1.044E−09 ++	3.018E−11 ++	2.952E−05 ++	7.007E−01 −	2.736E−03 ++
	10	5.827E−03 ++	7.187E−02 −	3.016E−11 ++	1.805E−05 ++	5.554E−02 −	7.534E−02 −
CT-image4	7	6.149E−01 −	1.132E−04 ++	2.403E−09 ++	1.426E−04 ++	6.650E−02 −	3.135E−02 ++
	8	1.205E−01 −	9.591E−04 ++	3.010E−11 ++	5.303E−04 ++	1.298E−03 ++	5.717E−03 ++
	9	3.709E−05 ++	8.110E−09 ++	3.020E−11 ++	9.188E−06 ++	7.897E−05 ++	2.704E−02 ++
	10	2.458E−01 −	3.572E−02 ++	3.010E−11 ++	4.827E−04 ++	3.323E−02 ++	1.861E−02 ++
CT-image5	7	3.633E−01 −	1.267E−01 −	7.478E−03 ++	2.137E−01 −	4.023E−01 −	4.519E−01 −
	8	9.469E−02 −	6.788E−01 −	7.177E−05 ++	8.187E−01 −	7.616E−01 −	2.661E−01 −
	9	1.690E−01 −	6.340E−02 −	1.365E−03 ++	4.418E−01 −	3.992E−01−	6.407E−02 −
	10	8.499E−02 −	1.154E−01 −	1.406E−04 ++	2.580E−01 −	1.221E−02 ++	1.735E−01 −
CT-image6	7	2.064E−02 ++	2.342E−05 ++	6.503E−07 ++	7.935E−06 ++	9.646E−01 −	2.911E−05 ++
	8	4.318E−03 ++	9.472E−02 −	3.508E−07 ++	3.083E−02 ++	4.732E−01 −	2.156E−01 −
	9	1.958E−01 −	1.698E−04 ++	9.243E−09 ++	1.070E−01 −	1.433E−01 −	2.087E−02 ++
	10	4.426E−03 ++	2.455E−04 ++	1.584E−04 ++	2.253E−04 ++	1.833E−01 −	9.908E−02 −
CT-image7	7	5.150E−01 −	1.134E−02 ++	5.070E−03 ++	3.535E−02 ++	3.946E−01 −	5.417E−02 −
	8	9.882E−01 −	7.945E−03 ++	2.274E−05 ++	3.631E−01 −	5.249E−01 −	6.194E−01 −
	9	7.974E−02 −	1.122E−02 ++	1.056E−03 ++	1.909E−02 ++	9.234E−01 −	1.215E−03 ++
	10	1.857E−03 ++	1.669E−01 −	5.971E−05 ++	1.748E−05 ++	2.398E−01 −	3.330E−04 ++
CT-image8	7	9.528E−01 −	2.573E−03 ++	2.911E−09 ++	1.750E−02 ++	3.152E−03 ++	1.274E−03 ++
	8	8.360E−01 −	1.117E−01 −	3.001E−11 ++	3.310E−03 ++	6.244E−02 −	1.409E−01 −
	9	5.541E−01 −	8.994E−01 −	2.958E−11 ++	7.103E−06 ++	9.166E−03 ++	2.889E−01 −
	10	1.659E−02 ++	1.409E−02 ++	2.999E−11 ++	8.072E−01 −	1.554E−03 ++	5.486E−01 −
CT-image9	7	7.429E−01 −	2.839E−02 ++	5.191E−10 ++	9.215E−02 −	3.057E−02 ++	7.297E−01 −
	8	7.124E−02 −	2.136E−07 ++	8.338E−08 ++	1.131E−04 ++	2.035E−01 −	1.129E−03 ++
	9	1.785E−01 −	2.051E−03 ++	1.359E−07 ++	2.951E−05 ++	6.458E−02 −	6.432E−02 −
	10	1.836E−02 ++	3.962E−09 ++	4.062E−11 ++	3.765E−04 ++	8.534E−01 −	1.293E−02 ++
CT-image10	7	1.082E−01 −	8.764E−08 ++	2.571E−05 ++	2.278E−06 ++	4.460E−01 −	1.750E−04 ++
	8	1.005E−02 ++	7.677E−03 ++	1.976E−06 ++	1.063E−03 ++	3.891E−02 ++	3.125E−01 −
	9	1.051E−02 ++	6.202E−01 −	1.871E−07 ++	4.840E−04 ++	7.726E−02 −	2.798E−01 −
	10	3.262E−02 ++	4.203E−01 −	9.045E−08 ++	1.299E−02 ++	9.705E−01 −	5.448E−02 −

Author contributions EHH was involved in supervision, methodology, conceptualization, formal analysis, writing—review & editing. MME was involved in software, resources, data curation, writing—original draft. AAA was involved in supervision, writing—review & editing. All authors read and approved the final paper.

Declarations

Conflict of interest The authors declare that there is no conflict of interest.

Ethical standard This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Khalifa NEM, Taha MHN, Hassanien AE, Elghamrawy S (2020) Detection of coronavirus (covid-19) associated pneumonia based on generative adversarial networks and a fine-tuned deep transfer learning model using chest X-ray dataset. arXiv preprint arXiv:2004.01184
2. Rahimi I, Chen F, Gandomi AH (2021) A review on covid-19 forecasting models. Neural Comput Appl 1–11
3. Yousri D, Elaziz MA, Abualigah L, Oliva D, Al-qaness MAA, Ewees AA (2020) Covid-19 X-ray images classification based on enhanced fractional-order cuckoo search optimizer using heavy-tailed distributions. Appl Soft Comput 101:107052
4. Devi A, Nayyar A (2021) Perspectives on the definition of data visualization: a mapping study and discussion on coronavirus (covid-19) dataset. In: Emerging technologies for battling Covid-19: applications and innovations, pp 223–240
5. Harmon SA, Sanford TH, Sheng X, Turkbey EB, Roth H, Ziyue X, Yang D, Myronenko A, Anderson V, Amalou A et al (2020) Artificial intelligence for the detection of covid-19 pneumonia on chest CT using multinational datasets. Nat Commun 11(1):1–7
6. Sharma K, Singh H, Sharma DK, Kumar A, Nayyar A, Krishnamurthi R (2021) Dynamic models and control techniques for drone delivery of medications and other healthcare items in covid-19 hotspots. In: Emerging technologies for battling covid-19: applications and innovations, pp 1–34
7. Elaziz MA, Ewees AA, Yousri D, Naji HS, Alwerfali QA, Awad SL, Al-Qaness MAA (2020) An improved marine predators algorithm with fuzzy entropy for multilevel thresholding: real world example of covid-19 CT image segmentation. IEEE Access 8:125306–125330
8. Houssein EH, Helmy BE, Oliva D, Elngar AA, Shaban H (2020) A novel black widow optimization algorithm for multilevel thresholding image segmentation. Expert Syst Appl 167:114159
9. Merzban MH, Elbayoumi M (2019) Efficient solution of otsu multilevel image thresholding: a comparative study. Expert Syst Appl 116:299–309
10. Rodríguez-Esparza E, Zanella-Calzada LA, Oliva D, Heidari AA, Zaldivar D, Pérez-Cisneros M, Foong LK (2020) An efficient Harris Hawks-inspired image segmentation method. Expert Syst Appl 155:113428
11. He L, Huang S (2020) An efficient krill herd algorithm for color image multilevel thresholding segmentation problem. Appl Soft Comput 89:106063
12. Aja-Fernández S, Curiale AH, Vegas-Sánchez-Ferrero G (2015) A local fuzzy thresholding methodology for multiregion image segmentation. Knowl-Based Syst 83:1–12
13. Ayala HVH, dos Santos FM, Mariani C, dos Santos Coelho L (2015) Image thresholding segmentation based on a novel beta differential evolution approach. Expert Syst Appl 42(4):2136–2142
14. Kosko B (1986) Fuzzy entropy and conditioning. Inf Sci 40(2):165–174
15. Kapur JN, Sahoo PK, Wong AKC (1985) A new method for gray-level picture thresholding using the entropy of the histogram. Comput Vis Graph Image Process 29(3):273–285
16. Tsai W-H (1985) Moment-preserving thresolding: a new approach. Comput Vis Graph Image Process 29(3):377–393
17. Otsu N (1979) A threshold selection method from gray-level histograms. IEEE Trans Syst Man Cybern 9(1):62–66
18. Oliva D, Hinojosa S, Osuna-Enciso V, Cuevas E, Pérez-Cisneros M, Sanchez-Ante G (2019) Image segmentation by minimum cross entropy using evolutionary methods. Soft Comput 23(2):431–450
19. Sahoo PK, Soltani SAKC, Wong AKC (1988) A survey of thresholding techniques. Comput Vis Graph Image Process 41(2):233–260
20. Wang M, Chen H, Yang B, Zhao X, Lufeng H, Cai ZN, Huang H, Tong C (2017) Toward an optimal kernel extreme learning machine using a chaotic moth-flame optimization strategy with applications in medical diagnoses. Neurocomputing 267:69–84
21. Hussien AG (2021) An enhanced opposition-based salp swarm algorithm for global optimization and engineering problems. J Ambient Intell Humaniz Comput 1–22
22. Hashim FA, Houssein EH, Mabrouk MS, Al-Atabany W, Mirjalili S (2019) Henry gas solubility optimization: a novel physics-based algorithm. Future Gener Comput Syst 101:646–667
23. Hashim FA, Hussain EH, Houssein K, Mabrouk MS, Al-Atabany W (2020) Archimedes optimization algorithm: a new meta-heuristic algorithm for solving optimization problems. Appl Intell 51(3):1531–1551
24. Houssein EH, Neggaz N, Hosney ME, Mohamed WM, Hassaballah M (2021) Enhanced Harris Hawks optimization with genetic operators for selection chemical descriptors and compounds activities. Neural Comput Appl 1–18
25. Precup R-E, David R-C, Roman R-C, Petriu EM, Szedlak-Stinean A-I (2021) Slime mould algorithm-based tuning of cost-effective fuzzy controllers for servo systems. Int J Comput Intell Syst 14(1):1042–1052
26. Hashim FA, Houssein EH, Hussain K, Mabrouk MS, Al-Atabany W (2020) A modified henry gas solubility optimization for solving motif discovery problem. Neural Comput Appl 32(14):10759–10771
27. Houssein EH, Saad MR, Hashim FA, Shaban H, Hassaballah M (2020) Lévy flight distribution: a new metaheuristic algorithm for solving engineering optimization problems. Eng Appl Artif Intell 94:103731
28. Zapata H, Perozo N, Angulo W, Contreras J (2020) A hybrid swarm algorithm for collective construction of 3d structures. Int J Artif Intell 18(1):1–18
29. Gupta S, Deep K (2020) Hybrid sine cosine artificial bee colony algorithm for global optimization and image segmentation. Neural Comput Appl 32(13):9521–9543
30. Houssein EH, Mahdy MA, Blondin MJ, Shebl D, Mohamed WM (2021) Hybrid slime mould algorithm with adaptive guided differential evolution algorithm for combinatorial and global optimization problems. Expert Syst Appl 174:114689
31. Tharwat A, Hassanien AE, Elnaghi BE (2017) A BA-based algorithm for parameter optimization of support vector machine. Pattern Recognit Lett 93:13–22
32. Bohat VK, Arya KV (2019) A new heuristic for multilevel thresholding of images. Expert Syst Appl 117:176–203
33. Cuevas E, Gálvez J, Avalos O (2020) Introduction to optimization and metaheuristic methods. In: Recent metaheuristics algorithms for parameter identification. Springer, pp 1–8

34. Holland JH (1992) Genetic algorithms. Sci Am 267(1):66–73

35. Yang X-S, Deb S (2009) Cuckoo search via lévy flights. In: 2009 World congress on nature & biologically inspired computing (NaBIC). IEEE, pp 210–214

36. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61

37. Fathollahi-Fard AM, Hajiaghaei-Keshteli M, Tavakkoli-Moghaddam R (2018) The social engineering optimizer (SEO). Eng Appl Artif Intell 72:267–293

38. Storn R, Price K (1997) Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11(4):341–359

39. Dorigo M, Birattari M, Stutzle T (2006) Ant colony optimization. IEEE Comput Intell Mag 1(4):28–39

40. Mirjalili S (2016) SCA: a sine cosine algorithm for solving optimization problems. Knowl-Based Syst 96:120–133

41. Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019) Harris hawks optimization: algorithm and applications. Future Gener Comput Syst 97:849–872

42. Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl-Based Syst 89:228–249

43. Zhao W, Zhang Z, Wang L (2020) Manta Ray foraging optimization: an effective bio-inspired optimizer for engineering applications. Eng Appl Artif Intell 87:103300

44. Aarts E, Aarts EHL, Lenstra JK (2003) Local search in combinatorial optimization. Princeton University Press, Princeton

45. Rojas-Morales N, Rojas M-CR, Ureta EM (2017) A survey and classification of opposition-based metaheuristics. Comput Ind Eng 110:424–435

46. Tizhoosh HR (2005) Opposition-based learning: a new scheme for machine intelligence. In: International conference on computational intelligence for modelling, control and automation, 2005 and international conference on intelligent agents, web technologies and internet commerce, vol 1. IEEE, pp 695–701

47. Hongpei X, Erdbrink CD, Krzhizhanovskaya VV (2015) How to speed up optimization? Opposite-center learning and its application to differential evolution. Procedia Comput Sci 51:805–814

48. Li J, Chen T, Zhang T, Li YX (2016) A cuckoo optimization algorithm using elite opposition-based learning and chaotic disturbance. J Softw Eng 10:16–28

49. Zhao F, Zhang J, Wang J, Zhang C (2015) A shuffled complex evolution algorithm with opposition-based learning for a permutation flow shop scheduling problem. Int J Comput Integr Manuf 28(11):1220–1235

50. Gong C (2016) Opposition-based adaptive fireworks algorithm. Algorithms 9(3):43

51. Dinkar SK, Deep K, Mirjalili S, Thapliyal S (2021) Opposition-based Laplacian equilibrium optimizer with application in image segmentation using multilevel thresholding. Expert Syst Appl 174:114766

52. Aranguren I, Valdivia A, Morales-Castañeda B, Oliva D, Elaziz MA, Perez-Cisneros M (2021) Improving the segmentation of magnetic resonance brain images using the lshade optimization algorithm. Biomed Signal Process Control 64:102259

53. Kim YJ, Jang H, Lee K, Park S, Min S-G, Hong C, Park JH, Lee K, Kim J, Hong W et al (2019) Paip 2019: liver cancer segmentation challenge. Med Image Anal 67(101854):2021

54. Kandhway P, Bhandari AK, Singh A (2020) A novel reformed histogram equalization based medical image contrast enhancement using krill herd optimization. Biomed Signal Process Control 56:101677

55. Li Y, Bai X, Jiao L, Xue Yu (2017) Partitioned-cooperative quantum-behaved particle swarm optimization based on multilevel thresholding applied to medical image segmentation. Appl Soft Comput 56:345–356

56. Panda R, Agrawal S, Samantaray L, Abraham A (2017) An evolutionary gray gradient algorithm for multilevel thresholding of brain MR images using soft computing techniques. Appl Soft Comput 50:94–108

57. Wang R, Zhou Y, Zhao C, Haizhou W (2015) A hybrid flower pollination algorithm based modified randomized location for multi-threshold medical image segmentation. Bio-Med Mater Eng 26(s1):S1345–S1351

58. Alrosan A, Alomoush W, Norwawi N, Alswaitti M, Makhadmeh SN (2021) An improved artificial bee colony algorithm based on mean best-guided approach for continuous optimization problems and real brain MRI images segmentation. Neural Comput Appl 33(5):1671–1697

59. Abdel-Basset M, Chang V, Mohamed R (2020) Hsma_woa: a hybrid novel slime mould algorithm with whale optimization algorithm for tackling the image segmentation problem of chest X-ray images. Appl Soft Comput 95:106642

60. Sahlol AT, Yousri D, Ewees AA, Al-Qaness MAA, Damasevicius R, Elaziz MA (2020) Covid-19 image classification using deep features and fractional-order marine predators algorithm. Sci Rep 10(1):1–15

61. Zivkovic M, Nebojsa Bacanin K, Venkatachalam AN, Djordjevic A, Strumberger I, Al-Turjman F (2021) Covid-19 cases prediction by using hybrid machine learning and beetle antennae search approach. Sustain Cities Soc 66:102669

62. Aziz MAE, Ewees AA, Hassanien AE (2017) Whale optimization algorithm and moth-flame optimization for multilevel thresholding image segmentation. Expert Syst Appl 83:242–256

63. Ewees AA, Elaziz MA, Al-Qaness MAA, Khalil HA, Kim S (2020) Improved artificial bee colony using sine-cosine algorithm for multi-level thresholding image segmentation. IEEE Access 8:26304–26315

64. Zhou C, Tian L, Zhao H, Zhao K (2015) A method of two-dimensional otsu image threshold segmentation based on improved firefly algorithm. In: 2015 IEEE international conference on cyber technology in automation, control, and intelligent systems (CYBER). IEEE, pp 1420–1424

65. Abdel-Basset M, Chang V, Mohamed R (2020) A novel equilibrium optimization algorithm for multi-thresholding image segmentation problems. Neural Comput Appl 1–34

66. Bhandari AK, Kumar A, Chaudhary S, Singh GK (2016) A novel color image multilevel thresholding based segmentation using nature inspired optimization algorithms. Expert Syst Appl 63:112–133

67. Gao H, Zheng F, Pun C-M, Haidong H, Lan R (2018) A multilevel thresholding image segmentation based on an improved artificial bee colony algorithm. Comput Electr Eng 70:931–938

68. Farshi TR, Drake JH, Özcan E (2020) A multimodal particle swarm optimization-based approach for image segmentation. Expert Syst Appl 149:113233

69. Pare S, Bhandari AK, Kumar A, Singh GK (2018) A new technique for multilevel color image thresholding based on modified fuzzy entropy and Lévy flight firefly algorithm. Comput Electr Eng 70:476–495

70. Yang Z, Angus W (2020) A non-revisiting quantum-behaved particle swarm optimization based multilevel thresholding for image segmentation. Neural Comput Appl 32(16):12011–12031

71. Singh S, Mittal N, Singh H (2020) A multilevel thresholding algorithm using LebTLBO for image segmentation. Neural Comput Appl 32:16681–16706

72. Ashish Kumar Bhandari (2020) A novel beta differential evolution algorithm-based fast multilevel thresholding for color image segmentation. Neural Comput Appl 32(9):4583–4613

73. Houssein EH, Helmy BE, Oliva D, Elngar AA, Shaban H (2021) A novel black widow optimization algorithm for multilevel thresholding image segmentation. Expert Syst Appl 167:114159

74. Chakraborty F, Roy PK, Nandi D (2019) Oppositional elephant herding optimization with dynamic Cauchy mutation for multilevel image thresholding. Evolut Intell 12(3):445–467

75. Glasbey CA (1993) An analysis of histogram-based thresholding algorithms. CVGIP Graph Models Image Process 55(6):532–537

76. Tubishat M, Idris N, Shuib L, Abushariah MAM, Mirjalili S (2020) Improved salp swarm algorithm based on opposition based learning and novel local search algorithm for feature selection. Expert Syst Appl 145:113122

77. Elaziz MA, Oliva D, Xiong S (2017) An improved opposition-based sine cosine algorithm for global optimization. Expert Syst Appl 90:484–500

78. Turgut OE (2021) A novel chaotic Manta-Ray foraging optimization algorithm for thermo-economic design optimization of an air-fin cooler. SN Appl Sci 3(1):1–36

79. Akay B (2013) A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding. Appl Soft Comput 13(6):3066–3091

80. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13(4):600–612

81. Wilcoxon F (1992) Individual comparisons by ranking methods. In: Breakthroughs in statistics. Springer, pp 196–202

82. Liao C, Li S, Luo Z (2006) Gene selection using Wilcoxon rank sum test and support vector machine for cancer classification. In: International conference on computational and information science. Springer, pp 57–66

83. Zhao J, Zhang Y, He X, Xie P (2020) Covid-CT-dataset: a CT scan dataset about covid-19. arXiv preprint arXiv:2003.13865

84. Cohen JP, Morrison P, Dao L, Roth K, Duong TQ, Ghassemi M (2020) Covid-19 image data collection: prospective predictions are the future. arXiv preprint arXiv:2006.11988

85. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67

86. Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM (2017) Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. Adv Eng Softw 114:163–191

87. Faramarzi A, Heidarinejad M, Stephens B, Mirjalili S (2020) Equilibrium optimizer: a novel optimization algorithm. Knowl-Based Syst 191:105190

88. Arcuri A, Fraser G (2013) Parameter tuning or default values? An empirical investigation in search-based software engineering. Empir Softw Eng 18(3):594–623