



# Harris hawks optimization for COVID-19 diagnosis based on multi-threshold image segmentation

Mohammad Hashem Ryalat<sup>1</sup> · Osama Dorgham<sup>1,5</sup> · Sara Tedmori<sup>2</sup> · Zainab Al-Rahamneh<sup>1</sup> · Nijad Al-Najdawi<sup>1</sup> · Seyedali Mirjalili<sup>3,4</sup>

Received: 8 January 2022 / Accepted: 22 November 2022 / Published online: 1 December 2022  
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2022

## Abstract

Digital image processing techniques and algorithms have become a great tool to support medical experts in identifying, studying, diagnosing certain diseases. Image segmentation methods are of the most widely used techniques in this area simplifying image representation and analysis. During the last few decades, many approaches have been proposed for image segmentation, among which multilevel thresholding methods have shown better results than most other methods. Traditional statistical approaches such as the Otsu and the Kapur methods are the standard benchmark algorithms for automatic image thresholding. Such algorithms provide optimal results, yet they suffer from high computational costs when multilevel thresholding is required, which is considered as an optimization matter. In this work, the Harris hawks optimization technique is combined with Otsu's method to effectively reduce the required computational cost while maintaining optimal outcomes. The proposed approach is tested on a publicly available imaging datasets, including chest images with clinical and genomic correlates, and represents a rural COVID-19-positive (COVID-19-AR) population. According to various performance measures, the proposed approach can achieve a substantial decrease in the computational cost and the time to converge while maintaining a level of quality highly competitive with the Otsu method for the same threshold values.

**Keywords** Harris hawks optimization · Multilevel thresholding · Image segmentation · Otsu method · Covid-19 · CT images

## 1 Introduction

Among the wide variety of sciences that can offer support during global pandemics, digital image processing is the one that can assist many aspects of medicine. Medical

imaging systems are used to identify or study the occurrence/development of certain diseases. The domain of medical imaging is developing rapidly due to advances in image processing techniques (e.g., applied machine learning) that include image recognition, segmentation,

✉ Mohammad Hashem Ryalat  
ryalat@bau.edu.jo

✉ Osama Dorgham  
o.dorgham@bau.edu.jo; o.dorgham@skylineuniversity.ac.ae

✉ Seyedali Mirjalili  
seyedali.mirjalili@griffithuni.edu.au

Sara Tedmori  
s.tedmori@psut.edu.jo

Zainab Al-Rahamneh  
zainab@bau.edu.jo

Nijad Al-Najdawi  
n.al-najdawi@bau.edu.jo

<sup>1</sup> Prince Abdullah Bin Ghazi Faculty of Information and Communication Technology, Al-Balqa Applied University, Al-Salt 19117, Jordan

<sup>2</sup> King Hussein School of Computing Sciences, Princess Sumaya University for Technology, Amman 11941, Jordan

<sup>3</sup> Centre for Artificial Intelligence Research and Optimisation, Torrens University, Adelaide, SA 5000, Australia

<sup>4</sup> Yonsei Frontier Lab, Yonsei University, Seoul, South Korea

<sup>5</sup> School of Information Technology, Skyline University College, Sharjah, United Arab Emirates

enhancement, and analysis. Among various modalities, computed tomography (CT) can be used effectively to screen and monitor diagnosed cases and has a direct impact on health outcomes because it assists healthcare professionals to administer rapid, accurate, and large-scale diagnostic screenings [18]. CT scans of the chest may be helpful in diagnosing COVID-19 in patients with a high clinical suspicion of infection but who have not been recommended for routine screening. In this type of medical imaging, cross-sectional tomographic images (slices) of the body are produced by using multiple X-ray measurements. Chest CT scans of patients with COVID-19-associated pneumonia usually show ground-glass opacification, possibly with consolidation.

Among various available medical image processing techniques and algorithms, image segmentation methods are the best tools to simplify image representation and analysis. Image segmentation is considered an essential part of the pattern recognition, computer vision, and medical image processing techniques [13, 30]. In image segmentation, the image is divided into regions of interest, and the segmentation process can be conducted by using shape, size, color, texture, illumination, etc. [16]. Image segmentation approaches can be categorized into four main types: clustering, region-based, texture analysis, and histogram thresholding. Among these various segmentation approaches, histogram thresholding is known for its robustness, simplicity, and accuracy and therefore is widely applied in image segmentation. In histogram thresholding, the histogram data are extracted from a given image and the best threshold values are used to categorize pixels in different regions [12]. For automatic image thresholding, traditional statistical approaches such as the Otsu and Kapur methods are the typical benchmark algorithms.

Multilevel thresholding methods deliver better results when compared to other traditional thresholding methods [14, 17]. A wide range of multilevel segmentation methods have been proposed in the literature [40, 50]. Furthermore, obtaining optimal threshold values is considered a practical optimization problem. Therefore, many nature-inspired algorithms have been presented in recent years for solving difficult practical optimization issues. Because of their adaptability and efficiency, those algorithms have attracted the interest of both scientists and researchers.

Despite their simple design, nature-inspired algorithms are effective at addressing extremely difficult optimization problems. Those algorithms are an essential component of modern global optimization algorithms, artificial intelligence, and informatics in general. Many metaheuristic algorithms have the property of reaching a global optimum for the problem after a relatively small number of iterations. Therefore, various evolutionary and swarm-based

strategies have been combined with statistical thresholding methods such as the Kapur method and the Otsu method in order to find the optimal threshold values for multilevel segmentation [52]. The Harris hawks optimization (HHO) algorithm is a well-known swarm-based, gradient-free optimization technique. HHO has received much interest from scientists and researchers in terms of its performance, quality of results, and its acceptable convergence in dealing with different applications in real-world problems.

In this work, the Harris hawks optimization technique is combined with Otsu's method to effectively reduce the required computational cost while maintaining optimal results. The proposed method is validated using publically accessible imaging datasets, which include chest scans with clinical and genetic correlates and reflect a COVID-19-positive (COVID-19-AR) population in rural areas. For the same threshold values, the suggested strategy can achieve a significant reduction in computing cost and time to converge while retaining a level of quality that is highly competitive with the Otsu method, according to different performance criteria.

The content of the paper can be summarized as follows: Sect. 2 presents a literature review of CT-based methods for efficient lung segmentation. Section 3 describes the implementation of Harris hawks optimization algorithm for multilevel image segmentation. Section 4 outlines the Harris hawks optimization for multi-threshold image segmentation evaluation with some confirmatory ranking results and analysis. Section 5 presents the evaluation of results. Finally, Sect. 6 presents the conclusions of this paper.

## 2 Literature review

In the field of CT imaging, the accurate segmentation of lung can lead to the accurate diagnosis of lung infection and to the correct detection and classification of lung nodules. For this reason, various CT-based methods for efficient lung segmentation have been proposed in the literature. For example, region-growing techniques have been applied in works such as Netto et al. [31], while other works such as Keshani et al. [26] have applied active contours techniques. On the other hand, authors in Messay et al. [32], Pu et al. [43], and Tan et al. [49] have utilized thresholding strategies such as rule-based, fuzzy inference methods, and intensity in order to detect nodules in CT images. Template matching methods have also been widely used in some works such as Akram et al. [3], Jo et al. [24], and Serhat Ozekes and Ucan [47]. Meanwhile, other works such as da Silva Sousa et al. [8] and Narayanan et al. [37] have designed composite discriminative feature techniques to classify and detect lung nodules through the use of

different classification methods. In addition, some other machine learning methods have been investigated for lung nodule detection, such as those presented in Furqan et al. [19], Gruetzemacher et al. [21], Jiang et al. [23], Pehrson et al. [42], and Zhang et al. [53].

Recently, contextual image information (i.e., channel and spatial) is shown to be effective for semantic segmentation as proposed by Li et al. [28], which interactively explores the spatial contextual information and the channel contextual information. In their work, the spatial contextual module is exploited to uncover the spatial contextual dependency between pixels by exploring the correlation between pixels and categories, while the semantic features are extracted using the channel contextual module by modeling the long-term semantic dependence between channels. Results show that the two context modules are adaptively integrated to achieve better outcomes, by connecting them in tandem for interactive training for mutual communication between the two dimensions.

Image thresholding techniques can be categorized as either bi-level or multilevel. In bi-level techniques, designated objects are differentiated by using a single threshold value. In multilevel techniques, the image is divided into multiple regions using multiple threshold values [17]. Therefore, the multilevel thresholding techniques give more accurate segmentation results as compared to the bi-level ones [27]. Moreover, the optimal threshold values in multilevel thresholding techniques can be classified into parametric and nonparametric approaches [38].

Traditional image segmentation approaches such as the Otsu [39] and the Kapur et al. [25] methods use the maximization of the variance between classes and the histogram entropy to obtain the best possible threshold values. Although those methods obtain optimal results, they impose high computational cost that increases as the number of thresholds is increased. More recently, the use of metaheuristic algorithms in combination with traditional statistical approaches has been proved to work effectively and to reduce the computational cost when multilevel thresholding is required.

Well-known automatic thresholding algorithms, such as the algorithms in the Otsu and the Kapur methods, function by performing a set of steps. These steps include processing the input image, obtaining the image histogram, computing the threshold value, and finally converting the image pixels into white in regions where the saturation is greater than the obtained threshold and into black otherwise. However, these algorithms utilize different methods to compute the threshold value. For instance, the algorithm in Otsu's method processes the image histogram and segments the objects by minimizing the variance on each of the classes. The obtained histogram contains two expressed peaks that represent different ranges of intensity value.

In Otsu's method, the image histogram is separated into two clusters by a threshold that is defined by minimizing the weighted variance of the classes denoted by  $\sigma_w^2(t)$ , where  $\sigma_w^2(t) = w_1(t)\sigma_1^2(t) + w_2(t)\sigma_2^2(t)$ , which is defined as the Otsu model, where  $w_1(t), w_2(t)$  are the probabilities of the two classes divided by a threshold ( $t$ ), where  $(0 \leq t \leq 255)$ . Otsu's method defines two alternatives to find the threshold. The first alternative is to minimize the within-class variance  $\sigma_w^2(t)$ , and the second is to maximize the between-class variance using  $\sigma_b^2(t) = w_1(t)w_2(t)[\mu_1(t) - \mu_2(t)]^2$ , where  $\mu_i$  is a mean of class  $i$ . The cluster probability function is used to calculate the probability  $P$  for each pixel value in the two separated clusters  $C_1, C_2$  as  $w_1(t) = \sum_{i=1}^t P(i)$  and  $w_2(t) = \sum_{i=t+1}^I P(i)$ , respectively [48].

Digital images can be represented as an intensity function  $f(x, y)$  that includes gray-level values, with a total number of pixels  $n$  and the total number of pixels with a specified gray-level  $i$ . Therefore, the probability of the occurrence of gray-level  $i$  is  $P(i) = n_i/n$ . The pixel intensity values for  $C_1$  and  $C_2$  are within the range of  $[1, t]$  and  $[t + 1, I]$ , respectively, where  $I$  is the maximum intensity value (i.e., 255). The means for  $C_1, C_2$ , denoted by  $\mu_1(t), \mu_2(t)$ , are obtained by  $\mu_1(t) = \sum_{i=1}^t iP(i)/w_1(t)$  and  $\mu_2(t) = \sum_{i=t+1}^I iP(i)/w_2(t)$ , respectively. Therefore the  $(\sigma_1^2, \sigma_2^2)$  values can be obtained by  $\sigma_1^2(t) = \sum_{i=1}^t [i - \mu_1(t)]^2 P(i)/w_1(t)$ , and  $\sigma_2^2(t) = \sum_{i=t+1}^I [i - \mu_2(t)]^2 P(i)/w_2(t)$ , respectively [48].

Obtaining optimal threshold values is considered a practical optimization problem. Therefore, many nature-inspired algorithms have been presented in recent years for solving difficult practical optimization problems [5–7, 33–35, 44–46]. Some of the metaheuristic algorithms are designed for image processing applications and image thresholding specifically include [2], which proposed the use of bird-mating optimization together with the Otsu and the Kapur methods as a strategy to find the best thresholds for image segmentation. On the other hand, Mousavirad and Ebrahimpour-Komleh [36] used the human mental search algorithm with Otsu's and Kapur's methods. Another nature-inspired approach was put forward in Pare et al. [41], in which Rényi entropy is used for locating the best thresholds using the bat algorithm. Other works have proposed using different algorithms for image segmentation, such as Liang et al. [29], which used Tsallis entropy and the grasshopper optimization algorithm, Baby Resma and Nair [4] which used the krill herd optimization algorithm and [54] which combined the firefly optimization algorithm to reduce the computation time and enhance the accuracy of image segmentation. A comprehensive review of available metaheuristic algorithms can be found in Pare et al. [41].

As presented in this section, there are several approaches that have been applied in the recent years for medical image segmentation. The first group of those approaches employed the concept of region-growing and active contours techniques to segment images and isolate the region-of-interest areas. Although this group of techniques can lead to a good level of accuracy, those approaches need to specific knowledge in the domain, and at the mean time they usually need to interact with human medical experts. This adds a pressure on human experts and also delays the process of treatment.

There are other different approaches such as template matching methods and composite discriminative feature that have been also utilized by a few researchers. The main drawback of those approaches is they cannot be generalized to be used in different imaging modalities and they do not lead to convincing outcomes in some cases.

On the other hand, many researchers utilized thresholding techniques to detect nodules in CT images as a base for the image segmentation. In these techniques, the maximization of the variance between classes is usually employed to segment images. Otsu method is one of the most popular techniques which is used in this domain. Although utilizing Otsu method in multilevel segmentation leads to competitive results in the medical image segmentation, the high computational cost needed by this approach is a challenging task that researchers are still trying to overcome; hence, there is the importance of employing optimization. The authors of this research employed the HHO with Otsu criterion as an objective function to overcome the high computational cost and in the meantime to achieve optimal outcomes.

The Harris hawks optimization (HHO) algorithm proposed in Heidari et al. [22] is a new metaheuristic algorithm that has the potential to be used for multilevel image thresholding. This algorithm is swarm based and has been developed to efficiently handle continuous optimization tasks and produce high-quality solutions. The HHO algorithm is still relatively new and has not been tested sufficiently on real-world problems. In this research, therefore, it is applied to the multilevel image segmentation of chest images of COVID-19 patients. Its segmentation results are then analyzed and compared against those obtained by the Otsu method. The authors in this research claim that the use of metaheuristic algorithms in image segmentation domain lowers the amount of computations required to locate the best threshold configuration.

### 3 Implementation of Harris hawks optimization algorithm for multilevel image segmentation

Harris hawk's optimization is a population-based swarm intelligence algorithm. It mimics the hunting strategy of Harris hawks, which is mathematically modeled to address different optimization problems. A detailed explanation of the background and fundamentals of the algorithm can be found in Heidari et al. [22]. Similar to other meta-heuristics, HHO consists of two main phases: diversification (exploration) and intensification (exploitation) which mimics the attacking strategy of Harris hawks when they are hunting their prey, where the attacking strategy is changed based on the circumstances of the prey. The attacking strategy that is simulated in HHO is explained in the following subsections. Figure 1 illustrates the phases of the HHO algorithm.

#### 3.1 Diversification phase (exploration)

In HHO, the Harris hawks represent the solutions; the best solution in each iteration represents the prey. Harris hawks perch randomly in some places, and they are one of the two strategies to attack their prey. The perch positions of the Harris hawks are based on the positions of other family members, as modeled in Eq. (1) for the condition  $q < 0.5$ , or they perch randomly in tall trees, which is modeled by Eq. (1) for the condition  $q \geq 0.5$ . The first rule in the model (see Eq. 1) represents the random generation of solutions, and the second rule represents the difference between the position of the best solution (prey) and the average location of the group (hawks):

$$X(t+1) = \begin{cases} X_{\text{rand}}(t) - r_1 |X_{\text{rand}}(t) - 2r_2 X(t)| & q \geq 0.5 \\ (X_{\text{rabbit}}(t) - X_m(t)) - r_3(LB + r_4(UB - LB)) & q < 0.5 \end{cases} \quad (1)$$

where  $X(t)$  is the position vector of a hawk in the  $t$ th iteration,  $X_{\text{rand}}(t)$  is a randomly selected hawk from the current population,  $X_{\text{rabbit}}(t)$  is the position of the rabbit prey,  $X(t)$  is the current position vector of the hawks,  $r_1$ ,  $r_2$ ,  $r_3$ ,  $r_4$ , and  $q$  are random numbers that are updated in each iteration, UB and LB are the upper bound and lower bound of the variables, respectively, and  $X_m$  is the average position of the current population of hawks. The average position  $X_m$  can be defined as shown in Eq. (2):

$$X_m(t) = \frac{1}{N} \sum_{i=1}^N X_i(t). \quad (2)$$

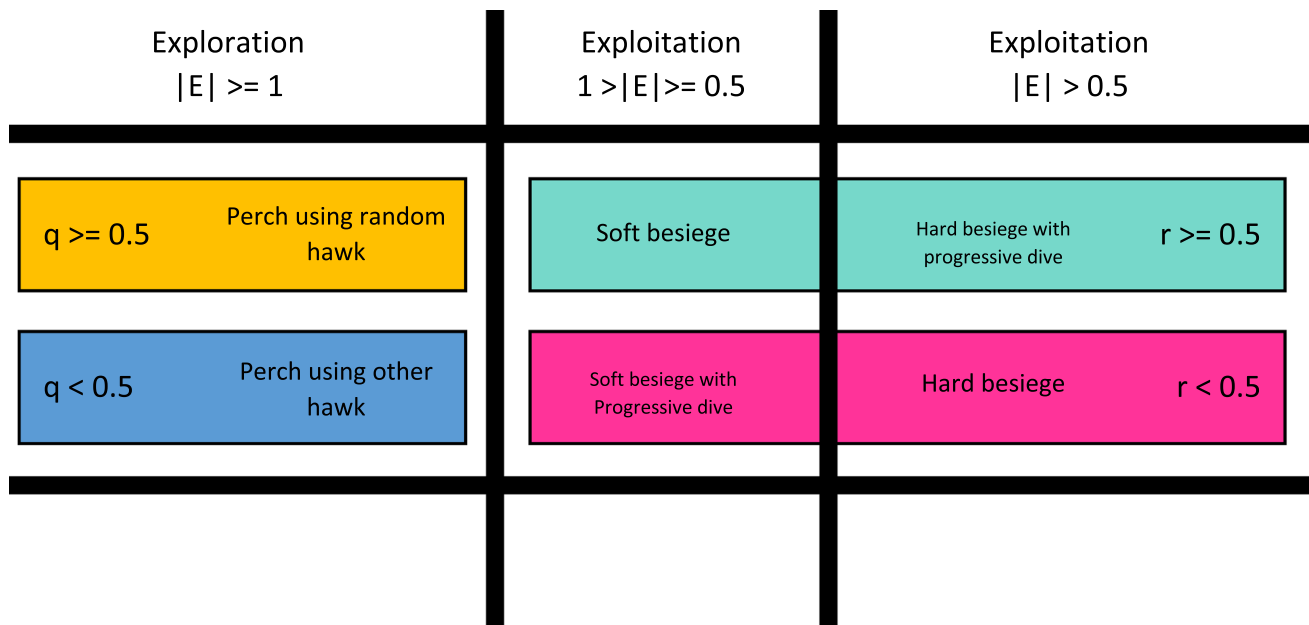


Fig. 1 Phases of Harris hawks optimization [22]

### 3.2 Switch between diversification (exploration) and intensification (exploitation)

The HHO algorithm can switch between exploration and exploitation according to the escaping energy  $E$  of the prey. The mathematical model for the energy of the prey can be defined as shown in Eq. (3):

$$E = 2E_0(1 - t/T) \quad (3)$$

In HHO,  $E_0$  randomly changes inside the interval  $(-1, 1)$  at each iteration. This requires  $E$  to be decreased linearly proportional to the number of iterations.

### 3.3 Intensification phase (exploitation)

Harris hawks execute a surprise dive to pounce on their prey. However, the prey has the power or capability to escape from this risky situation. The prey's chance of escaping attack can be represented by  $r$  as follows:

$$\text{Escape capability} = \begin{cases} \text{successfully escapes} & \text{if } r < 0.5 \\ \text{unsuccessfully escapes} & \text{if } r \geq 0.5 \end{cases}$$

#### 3.3.1 Soft (smooth) besiege strategy

If the prey has some energy, it tries to escape from the hawks by doing random jumps. However, the Harris hawks surround the prey softly to exhaust it and then execute a surprise attack. This process can happen when the chance of the prey escaping,  $r$ , equals  $r \geq 0.5$ , (i.e., unsuccessful) and the escaping energy of the prey,  $E$ , equals  $E \geq 0.5$ .

This process can be modeled by Eqs. (4) and (5) as follows:

$$X(t+1) = \Delta X(t) - E|2(1 - r_5)X_{\text{rabbit}}(t) - X(t)| \quad (4)$$

$$\Delta X(t) = X_{\text{rabbit}}(t) - X(t), \quad (5)$$

where  $\Delta X(t)$  is the difference between the current location and the vector of the rabbit in iteration  $t$  and  $r_5$  is a random number between 0 and 1, which represents the random bounce force of the rabbit throughout the escaping criteria.

#### 3.3.2 Hard besiege strategy

If the prey has a little escaping energy ( $|E| < 0.5$ ) and it becomes exhausted (unsuccessfully escaping,  $r \geq 0.5$ ), the Harris hawks surround the prey and perform a surprise attack. This situation can be modeled by Eq. (6) as follows:

$$X(t+1) = X_{\text{rabbit}}(t) - E|\Delta X(t)| \quad (6)$$

#### 3.3.3 Soft (smooth) besiege strategy and progressive quick pounce

If the prey has some energy to escape (which implies that  $|E| \geq 0.5$ ), it can successfully escape ( $r < 0.5$ ). In this case, the Harris hawks use a smooth (soft) besiege to attack the prey. During the escaping process, a simulation of the zigzag motion of the prey can be performed using a Lévy flight (LF) operator. Based on the tricky movements of the prey, the Harris hawks try to modify their pouncing strategy gradually.



The Harris hawks can perform the soft besiege by deciding their next position as shown in Eq. (7):

$$Y = X_{\text{rabbit}}(t) - E|2(1 - r5)X_{\text{rabbit}}(t) - X(t)| \quad (7)$$

The Harris hawks try to adjust their movement by comparing the current pounce result and the previous one. If the result is not good, they will pounce based on the LF as shown in Eq. (8):

$$Z = Y + S \times \text{LF}(D), \quad (8)$$

where  $D$  is the dimension of problem,  $S$  is a random vector of size  $1 \times D$ , and LF is the Lévy flight function. Based on the previous assumption of the soft besiege, the Harris hawks update their position by Eq. (9) as follows:

$$X(t+1) = \begin{cases} Y & \text{if } F(Y) < F(X(t)) \\ Z & \text{if } F(Z) < F(X(t)) \end{cases} \quad (9)$$

### 3.3.4 Hard besiege strategy and progressive quick pounce

The Harris hawks apply the hard besiege strategy when the prey has a little energy to escape ( $|E| < 0.5$ ) and it also has a chance to successfully escape ( $r < 0.5$ ). To perform this

strategy, the Harris hawks try to reduce the distance between their average position  $X_m$  and that of the prey. The overall process can be modeled by Eq. (10) as follows:

$$X(t+1) = \begin{cases} Y & \text{if } F(Y) < F(X(t)) \\ Z & \text{if } F(Z) < F(X(t)) \end{cases}, \quad (10)$$

where  $Y$  and  $Z$  are obtained using the new rules in Eqs. (11) and (12), respectively:

$$Y = X_{\text{rabbit}}(t) - E|2(1 - r5)X_{\text{rabbit}}(t) - X_m(t)| \quad (11)$$

$$Z = Y + S \times \text{LF}(D) \quad (12)$$

### 3.3.5 Pseudocode of Harris hawks optimization algorithm for multilevel thresholding

The pseudocode of the procedures of the HHO algorithm employed for multilevel thresholding in this research is described in Algorithm 1.

---

#### Algorithm 1: Harris hawks optimization (HHO) algorithm for multilevel thresholding

---

**Inputs:** The population size  $N$ , maximum number of iterations  $T$ , pixel values, number of threshold values and initialization of the random population  $X_i$  ( $i = 1, 2, \dots, N$ )

**Output:** The location of the rabbit, its fitness value, and threshold values which maximize the Otsu criteria.

```

1: Set the parameters of the population size  $N$ , initial energy  $E_0$ , iteration number  $t$  and maximum iteration number
2: for  $i = 1$  to  $N$  do
3:   Generate the initial population  $X_i(t)$ 
4:   Compute the fitness function (i.e., threshold values) for each solution in the population  $X_i(t)$ 
5: end for
6: Set the best solution (i.e., threshold values) in the population to  $X_{\text{rabbit}}(t)$  (i.e., best threshold values)
7: repeat
8:   for  $i = 1$  to  $N$  do
9:     Update the energy of a prey  $E$  as shown in Eq. 2
10:    if ( $|E| \geq 1$ ) then
11:      Update the position of each solution as shown in Eq. 1
12:    Else
13:      if ( $|E| < 1$ ) then
14:        if ( $r \geq 0.5$ ) &&  $|E| \geq 0.5$  then
15:          Update the position of each solution as shown in Eq. 4
16:        else if ( $r \geq 0.5$ ) &&  $|E| < 0.5$  then
17:          Update the position of each solution as shown in Eq. 6
18:        else if ( $r < 0.5$ ) &&  $|E| \geq 0.5$  then
19:          Update the position of each solution as shown in Eq. 9
20:        else if ( $r < 0.5$ ) &&  $|E| < 0.5$  then
21:          Update the position of each solution as shown in Eq. 10
22:        end if
23:      end if
24:    end if
25:    Compute the fitness function for each solution in the population  $X_i(t)$ 
26:  end for
27:  Set the overall best solution in the population to  $X_{\text{rabbit}}(t)$ 
28:  Set  $t = t + 1$ 
29: until Termination criteria are satisfied
30: return The overall best solution  $X_{\text{rabbit}}(t)$  (i.e., final threshold values)

```

---

There are three primary phases in the HHO algorithm. The following are the processes and their computational complexity (i.e., time complexity) [20].

1. Initiation process  $\rightarrow O(N)$ .
2. Updating the locations of all Harris hawks  $\rightarrow O(T \times N \times D)$ .
3. Determining the best location  $\rightarrow O(T \times N)$ .

where  $T$  is the maximum iteration,  $N$ : population size,  $D$ : the dimension of the problem.

Therefore, the total complexity of the HHO algorithm can be described as  $O(N \times (T + TD + 1))$ .

## 4 Experiment and results

The World Health Organization declared the outbreak of the novel coronavirus (2019-nCoV) an ongoing global pandemic in March 2020. As of today, tens of millions of cases have been confirmed, and with millions of lives lost due to COVID-19, it is becoming one of the deadliest pandemics in history.

This pandemic has shown that public health is not only a medical problem; it has also become the main common concern of all scientific fields. Therefore, interdisciplinary teams of scientists from all over the world have conducted novel research and studies to find efficient solutions that can be used to control the consequences of the pandemic and to prevent its return and the emergence of similar pandemics.

### 4.1 Dataset

In order to evaluate the efficiency of the proposed approach, a publicly available imaging dataset was obtained from the Cancer Imaging Archive. The obtained dataset consists of a set of CT images ( $762 \times 762$  pixels with a spacing  $1.08 \times 1.08$  mm and  $0.98 \times 0.98$  mm and a slice thickness 3.14 mm) [9–11]. The dataset contains chest imaging with clinical and genomic correlates and

represents a rural COVID-19-positive population (COVID-19-AR). The dataset contains a total of about 105 patients. For the purpose of this research, 20 randomly selected patient records (hereinafter referred to as 10 individual datasets) were used to test and evaluate the proposed approach.

### 4.2 Experimental settings

Prior to the experiment, the initial parameter values were set and tuned experimentally (see Table 1). In the experiment, the proposed algorithm starts by generating 30 different possible solutions, where each solution represents the set of possible thresholds. The number of iterations is set to 150 iterations to allow the analysis of the convergence behavior of the algorithm. In order to evaluate the stability and the reliability of the generated outcomes, the experiment is repeated 25 times. The lower bound (LB) and the upper bound (UB) are set to 0 and 255, respectively (for grayscale images), and the value of the dimension parameter in the algorithm is set dynamically in accordance with the number of thresholds in each specific experiment.

### 4.3 Results

Table 2 presents a sample from each dataset and presents a comparison between the outcomes generated by the Otsu method as a ground truth against the threshold values generated by the proposed approach. To keep Table 2 short and readable, the outcomes of only the first 10 datasets are presented in the table. However, the full details of all the experimental outcomes can be accessed via the URL linked to this paper (<https://sites.google.com/view/hhocovid19/home>). In Table 2, the third and fourth columns present the Otsu threshold values and the threshold values generated by the proposed approach, respectively. The fourth column presents the most frequent threshold values that appeared after the 25th execution of the algorithm. The last column provides the number of times (NOT), which is the number of times out of 25 that the most frequent threshold values appeared.

It is obvious from Table 2 that the values generated by the proposed approach are either the same or very close to the Otsu threshold values. Moreover, even in the cases where there is no complete match between the outcomes of the Otsu method and those generated by the proposed approach, the difference is negligible.

The two main inputs that are fed into the proposed approach are the group of CT images and the number of thresholds. The optimal threshold values are generated as output. The optimizer is employed to maximize the value of the between-class variance, as suggested by the Otsu criterion.

**Table 1** Initial parameter values of the HHO algorithm

| Parameter name       | Parameter value      |
|----------------------|----------------------|
| Population size      | 30                   |
| Number of iterations | 150                  |
| Number of runs       | 25                   |
| LB                   | 0                    |
| UB                   | 255                  |
| Dimension            | Number of thresholds |

**Table 2** Threshold values generated by the Otsu method versus threshold values generated by the proposed approach


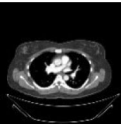
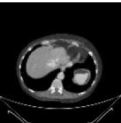
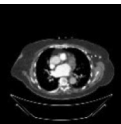
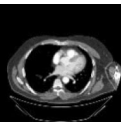
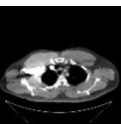

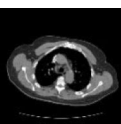
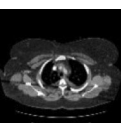
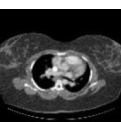
| ID       | Image   | Otsu method threshold value | Proposed approach threshold value | NOT |
|----------|---|-----------------------------|-----------------------------------|-----|
| 16406488 |    | 56, 173                     | 56, 173                           | 24  |
|          |   | 45, 110, 187                | 43, 109, 186                      | 24  |
|          |   | 39, 100, 150, 213           | 39, 100, 150, 213                 | 16  |
|          |   | 33, 85, 116, 156, 216       | 32, 83, 115, 156, 215             | 11  |
| 16406490 |    | 45, 159                     | 45, 159                           | 25  |
|          |   | 26, 88, 181                 | 26, 88, 180                       | 14  |
|          |   | 25, 84, 146, 211            | 24, 69, 113, 187                  | 10  |
|          |   | 23, 66, 106, 157, 217       | 23, 66, 106, 157, 217             | 5   |
| 16406498 |    | 32, 110                     | 32, 110                           | 25  |
|          |   | 29, 96, 165                 | 29, 96, 165                       | 18  |
|          |   | 28, 77, 124, 175            | 26, 76, 124, 175                  | 8   |
|          |   | 25, 75, 121, 163, 209       | 25, 75, 121, 163, 209             | 13  |
| 16406502 |    | 36, 143                     | 36, 142                           | 25  |
|          |   | 24, 81, 170                 | 24, 81, 170                       | 13  |
|          |   | 24, 76, 138, 206            | 23, 76, 138, 205                  | 17  |
|          |   | 22, 62, 100, 150, 212       | 21, 62, 100, 150, 211             | 5   |
| 16406503 |    | 50, 154                     | 50, 154                           | 25  |
|          |   | 28, 92, 175                 | 28, 92, 175                       | 22  |
|          |   | 28, 88, 153, 214            | 27, 88, 152, 212                  | 17  |
|          |   | 24, 68, 109, 160, 216       | 23, 68, 109, 160, 216             | 6   |
| 16406513 |   | 61, 172                     | 60, 171                           | 25  |
|          |   | 31, 98, 185                 | 31, 98, 185                       | 24  |
|          |   | 29, 90, 145, 208            | 29, 90, 145, 207                  | 8   |
|          |   | 24, 69, 108, 151, 211       | 23, 68, 108, 151, 210             | 13  |
| 16407187 |  | 61, 170                     | 61, 170                           | 25  |
|          |   | 31, 97, 181                 | 31, 97, 180                       | 20  |
|          |   | 30, 92, 143, 201            | 28, 81, 124, 187                  | 22  |
|          |   | 27, 75, 115, 155, 210       | 26, 75, 115, 155, 210             | 13  |
| 16424071 |  | 63, 177                     | 63, 177                           | 25  |
|          |   | 22, 86, 182                 | 22, 86, 181                       | 24  |
|          |   | 19, 63, 110, 185            | 19, 63, 110, 184                  | 12  |
|          |   | 19, 61, 107, 158, 216       | 19, 61, 107, 157, 215             | 7   |
| 16424081 |  | 58, 169                     | 58, 169                           | 25  |
|          |   | 20, 78, 176                 | 20, 78, 176                       | 25  |
|          |   | 17, 61, 108, 178            | 17, 58, 108, 178                  | 17  |
|          |   | 15, 42, 78, 119, 189        | 14, 41, 77, 118, 188              | 14  |
| 16424106 |  | 29, 104                     | 29, 104                           | 25  |
|          |   | 26, 86, 164                 | 26, 85, 163                       | 18  |
|          |   | 23, 67, 113, 177            | 23, 67, 113, 177                  | 21  |
|          |   | 20, 55, 92, 137, 195        | 20, 55, 92, 137, 195              | 19  |

Table 3 presents the average fitness values (i.e., between-class variance) that are achieved when the proposed approach is applied to the 10 datasets. Figure 2 displays the histograms of five different images picked randomly from different datasets along with the threshold values which are represented as red vertical lines. These threshold values represent the outcome of applying the proposed approach.

## 5 Evaluation of results

In order to evaluate the efficiency of the proposed approach, the optimal thresholds obtained from the experiment are supplied to the evaluation process as an input. The evaluation starts, as illustrated in Fig. 3, by generating a segmented image in accordance with the optimal thresholds. The generated segmented image and



**Table 3** Average between-class variance value for each dataset over 25 runs

| #  | Patient ID | # Of thr. | Avg. fitness     | #  | Patient ID | # Of thr. | Avg. fitness     |
|----|------------|-----------|------------------|----|------------|-----------|------------------|
| 1  | 16406488   | 2         | 1,995,968,548.14 | 11 | 16424120   | 2         | 1,325,763,624.57 |
|    |            | 3         | 2,025,402,282.08 |    |            | 3         | 1,373,514,362.89 |
|    |            | 4         | 2,040,028,586.97 |    |            | 4         | 1,389,606,981.23 |
|    |            | 5         | 2,045,857,463.56 |    |            | 5         | 1,398,038,015.74 |
| 2  | 16406490   | 2         | 2,298,088,159.24 | 12 | 16434363   | 2         | 3,524,501,058.11 |
|    |            | 3         | 2,436,038,694.92 |    |            | 3         | 3,710,226,862.57 |
|    |            | 4         | 2,453,919,131.42 |    |            | 4         | 3,745,843,560.59 |
|    |            | 5         | 2,466,483,199.24 |    |            | 5         | 3,766,274,305.97 |
| 3  | 16406498   | 2         | 2,076,199,600.90 | 13 | 16434368   | 2         | 2,320,389,483.76 |
|    |            | 3         | 2,130,226,824.33 |    |            | 3         | 2,404,024,664.02 |
|    |            | 4         | 2,154,189,341.66 |    |            | 4         | 2,430,796,897.84 |
|    |            | 5         | 2,162,355,593.61 |    |            | 5         | 2,441,260,652.15 |
| 4  | 16406502   | 2         | 1,940,207,310.35 | 14 | 16434369   | 2         | 2,602,673,414.62 |
|    |            | 3         | 2,040,921,175.96 |    |            | 3         | 2,749,944,666.81 |
|    |            | 4         | 2,062,553,214.82 |    |            | 4         | 2,777,323,399.19 |
|    |            | 5         | 2,074,264,187.22 |    |            | 5         | 2,798,949,958.41 |
| 5  | 16406503   | 2         | 3,012,563,719.35 | 15 | 16434411   | 2         | 2,830,098,693.27 |
|    |            | 3         | 3,151,883,519.94 |    |            | 3         | 2,914,157,752.27 |
|    |            | 4         | 3,178,821,393.68 |    |            | 4         | 2,931,386,952.33 |
|    |            | 5         | 3,195,958,229.64 |    |            | 5         | 2,939,713,611.57 |
| 6  | 16406513   | 2         | 3,265,136,378.30 | 16 | 16434453   | 2         | 3,225,924,532.16 |
|    |            | 3         | 3,371,021,181.84 |    |            | 3         | 3,389,121,940.72 |
|    |            | 4         | 3,405,659,240.79 |    |            | 4         | 3,415,523,456.42 |
|    |            | 5         | 3,422,308,473.62 |    |            | 5         | 3,434,394,061.24 |
| 7  | 16407187   | 2         | 2,340,736,567.15 | 17 | 16445122   | 2         | 2,355,216,836.24 |
|    |            | 3         | 2,438,359,772.60 |    |            | 3         | 2,492,331,087.81 |
|    |            | 4         | 2,455,628,797.20 |    |            | 4         | 2,514,181,165.77 |
|    |            | 5         | 2,468,027,367.69 |    |            | 5         | 2,524,959,336.26 |
| 8  | 16424071   | 2         | 2,299,778,005.74 | 18 | 16445138   | 2         | 2,812,376,551.29 |
|    |            | 3         | 2,407,441,253.39 |    |            | 3         | 2,962,616,506.72 |
|    |            | 4         | 2,428,542,182.15 |    |            | 4         | 2,995,377,758.74 |
|    |            | 5         | 2,437,592,717.67 |    |            | 5         | 3,012,410,682.07 |
| 9  | 16424081   | 2         | 1,857,758,173.25 | 19 | 16445144   | 2         | 2,269,545,641.31 |
|    |            | 3         | 1,994,137,148.17 |    |            | 3         | 2,380,108,941.03 |
|    |            | 4         | 2,026,029,797.17 |    |            | 4         | 2,401,661,550.92 |
|    |            | 5         | 2,041,342,618.95 |    |            | 5         | 2,413,854,491.02 |
| 10 | 16424106   | 2         | 2,724,258,369.10 | 20 | 16445168   | 2         | 3,891,245,700.61 |
|    |            | 3         | 2,860,700,119.06 |    |            | 3         | 3,941,055,911.03 |
|    |            | 4         | 2,899,143,433.02 |    |            | 4         | 3,965,460,390.77 |
|    |            | 5         | 2,921,678,998.55 |    |            | 5         | 3,976,172,936.86 |

the Otsu segmented image (i.e., ground truth) are then compared and tested to determine the degree of similarity.

Three different similarity metrics are used to evaluate the accuracy of the experiment: the peak signal-to-noise ratio (PSNR), the structural similarity index (SSIM), and the Jaccard index. These similarity metrics are explained in the following subsections.

### 5.1 Peak signal-to-noise ratio (PSNR)

The complications associated with subjective quality assessment necessitate the use of automated objective quality assessment methods to measure the quality differences between processed images. Among the well-known objective quality assessment methods is the PSNR, which

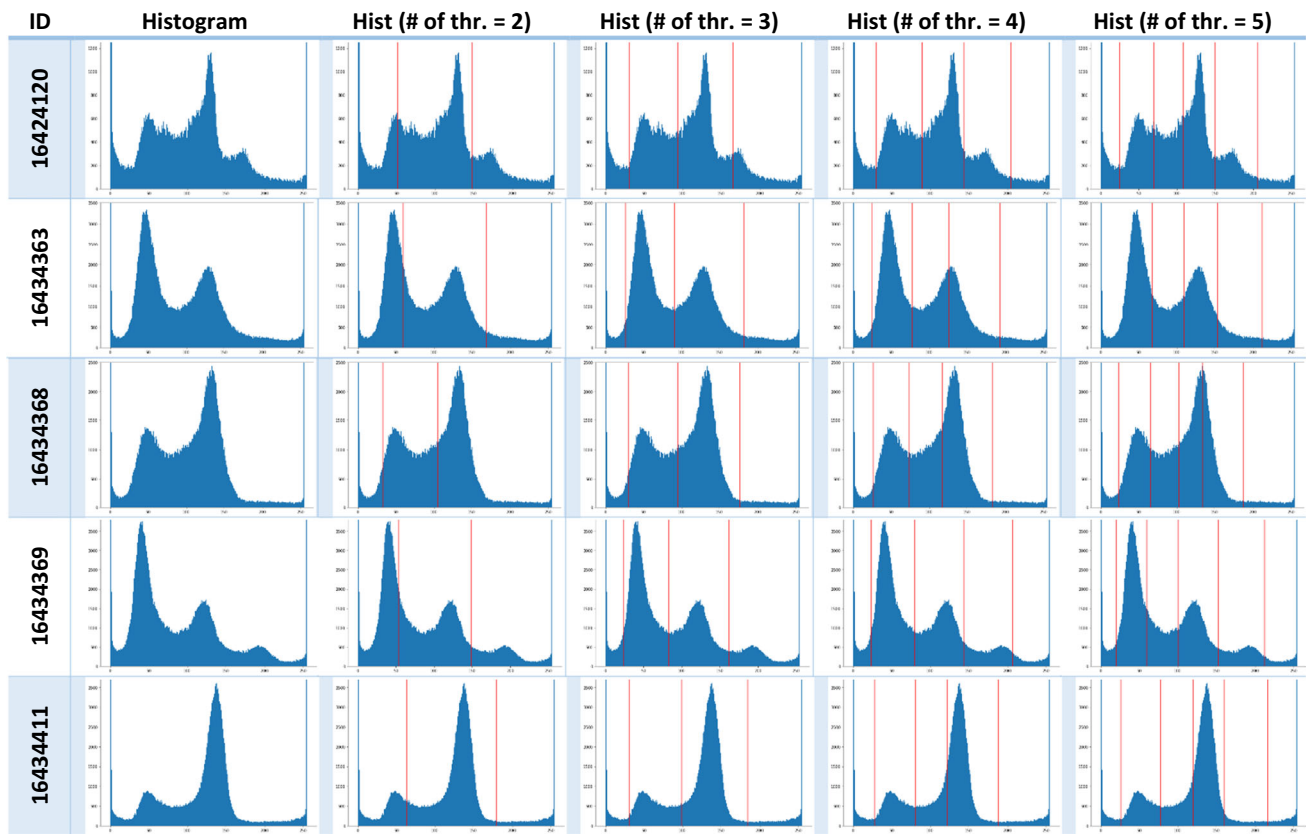


Fig. 2 Histograms of five images from different datasets with a different number of thresholds

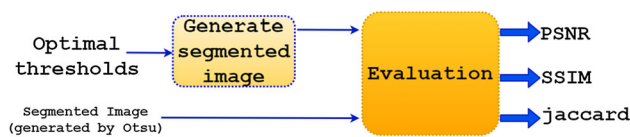


Fig. 3 Basic inputs and outputs in the evaluation process

is based on the mean squared error (MSE) and can be calculated as follows [51]:

$$\text{MSE} = \sum_{i=1}^N (x_i - y_i)^2,$$

where  $N$  is the number of pixels in the given image (both images must be of the same size) and  $x_i, y_i$  are the pixel values in both of the images to be tested.

The PSNR can be calculated as follows:

$$\text{PSNR} = 20 \log_{10} K / \sqrt{\text{MSE}}$$

where  $K$  is the maximum range of the image pixel values ( $K = 255$  for grayscale images).

The PSNR and MSE are high-quality measures for testing the proposed approach. The PSNR is widely used in digital image processing applications. In this work, the PSNR gives a value of 100 dB under one condition only, i.e., when both images (to be compared) are identical,

which leads to a division by 0, that is controlled by a conditional statement. When the PSNR result exceeds 30 dB, the human visual system would find it more difficult to differentiate between the two images. However, when the PSNR result is lower than 30 dB, this indicates that the human visual system would have the ability to notice the differences between the two images.

## 5.2 Structural similarity index (SSIM)

The SSIM is another well-known measure that can be used to assess the quality of the output generated by the proposed approach. The SSIM is based on the computation of three factors: luminance, contrast, and structure. The SSIM can be defined as follows [1]:

$$\text{SSIM}(x, y) = [l(x, y)]^\alpha \times [c(x, y)]^\beta \times [s(x, y)]^\gamma,$$

where  $(x, y) = (2\mu_x\mu_y + C_1)/(2\mu_x^2 + C_1)$ ,  $c(x, y) = (2\sigma_x\sigma_y + C_2)/(2\sigma_x^2 + C_2)$ , and  $s(x, y) = (\sigma_{xy} + C_3)/(\sigma_x\sigma_y + C_3)$ , where  $\mu_x, \mu_y, \sigma_x, \sigma_y$ , and  $\sigma_{xy}$  are the local means, standard deviations, and cross-covariance of the two compared images, respectively.

If  $\alpha = \beta = \gamma = 1$  and  $C_3 = C_2/2$ , the equation can be simplified as follows:

**Table 4** Average value of the PSNR (20 datasets, 25 different runs for each experiment)

| #  | Patient ID | # Of thr. | PSNR     | #  | Patient ID | # Of thr. | PSNR     |
|----|------------|-----------|----------|----|------------|-----------|----------|
| 1  | 16406488   | 2         | 99.32582 | 11 | 16424120   | 2         | 100      |
|    |            | 3         | 74.86861 |    |            | 3         | 81.04659 |
|    |            | 4         | 94.14886 |    |            | 4         | 78.23629 |
|    |            | 5         | 72.37319 |    |            | 5         | 82.44525 |
| 2  | 16406490   | 2         | 100      | 12 | 16434363   | 2         | 100      |
|    |            | 3         | 82.19359 |    |            | 3         | 77.21508 |
|    |            | 4         | 59.37486 |    |            | 4         | 70.61255 |
|    |            | 5         | 78.81656 |    |            | 5         | 73.18433 |
| 3  | 16406498   | 2         | 100      | 13 | 16434368   | 2         | 100      |
|    |            | 3         | 94.89771 |    |            | 3         | 97.4049  |
|    |            | 4         | 74.64346 |    |            | 4         | 76.29251 |
|    |            | 5         | 89.6506  |    |            | 5         | 68.79373 |
| 4  | 16406502   | 2         | 82.22882 | 14 | 16434369   | 2         | 100      |
|    |            | 3         | 90.61202 |    |            | 3         | 100      |
|    |            | 4         | 78.26527 |    |            | 4         | 56.58543 |
|    |            | 5         | 76.76858 |    |            | 5         | 83.73876 |
| 5  | 16406503   | 2         | 100      | 15 | 16434411   | 2         | 100      |
|    |            | 3         | 97.62194 |    |            | 3         | 97.41837 |
|    |            | 4         | 73.58808 |    |            | 4         | 74.54786 |
|    |            | 5         | 76.39956 |    |            | 5         | 57.7601  |
| 6  | 16406513   | 2         | 74.13531 | 16 | 16434453   | 2         | 100      |
|    |            | 3         | 80.21965 |    |            | 3         | 77.90913 |
|    |            | 4         | 77.67605 |    |            | 4         | 68.96586 |
|    |            | 5         | 70.98543 |    |            | 5         | 74.70691 |
| 7  | 16407187   | 2         | 100      | 17 | 16445122   | 2         | 83.64803 |
|    |            | 3         | 81.48877 |    |            | 3         | 97.52143 |
|    |            | 4         | 58.98134 |    |            | 4         | 81.85069 |
|    |            | 5         | 81.63743 |    |            | 5         | 65.45271 |
| 8  | 16424071   | 2         | 100      | 18 | 16445138   | 2         | 100      |
|    |            | 3         | 84.32184 |    |            | 3         | 83.75593 |
|    |            | 4         | 76.03031 |    |            | 4         | 88.94338 |
|    |            | 5         | 70.7011  |    |            | 5         | 78.00936 |
| 9  | 16424081   | 2         | 100      | 19 | 16445144   | 2         | 100      |
|    |            | 3         | 100      |    |            | 3         | 94.8936  |
|    |            | 4         | 66.61292 |    |            | 4         | 77.19462 |
|    |            | 5         | 68.36883 |    |            | 5         | 76.91706 |
| 10 | 16424106   | 2         | 100      | 20 | 16445168   | 2         | 100      |
|    |            | 3         | 72.29532 |    |            | 3         | 82.15853 |
|    |            | 4         | 95.68653 |    |            | 4         | 79.26923 |
|    |            | 5         | 92.65646 |    |            | 5         | 76.03681 |

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

The SSIM results are given in values that range between [0, 1], where 1 indicates a perfect match between the two images and 0 indicates no match.

### 5.3 Jaccard similarity index

The Jaccard similarity index measures the similarity of two sets of data (two images) on a range from 0 to 100%. This similarity index compares the pixels in two images to see which of them are shared and which are distinct. It counts the number of pixels that are shared by the two images, counts the total number of pixels in both images, divides

**Table 5** Average value of the SSIM (20 datasets, 25 different runs for each experiment)

| #  | Patient ID | # Of thr. | SSIM     | #  | Patient ID | # Of thr. | SSIM     |
|----|------------|-----------|----------|----|------------|-----------|----------|
| 1  | 16406488   | 2         | 0.999923 | 11 | 16424120   | 2         | 1        |
|    |            | 3         | 0.992272 |    |            | 3         | 0.998286 |
|    |            | 4         | 0.999545 |    |            | 4         | 0.996793 |
|    |            | 5         | 0.991701 |    |            | 5         | 0.997164 |
| 2  | 16406490   | 2         | 1        | 12 | 16434363   | 2         | 1        |
|    |            | 3         | 0.997395 |    |            | 3         | 0.993362 |
|    |            | 4         | 0.897879 |    |            | 4         | 0.98207  |
|    |            | 5         | 0.991361 |    |            | 5         | 0.988317 |
| 3  | 16406498   | 2         | 1        | 13 | 16434368   | 2         | 1        |
|    |            | 3         | 0.999283 |    |            | 3         | 0.999411 |
|    |            | 4         | 0.991874 |    |            | 4         | 0.992309 |
|    |            | 5         | 0.997574 |    |            | 5         | 0.980276 |
| 4  | 16406502   | 2         | 0.997756 | 14 | 16434369   | 2         | 1        |
|    |            | 3         | 0.997689 |    |            | 3         | 1        |
|    |            | 4         | 0.99476  |    |            | 4         | 0.818353 |
|    |            | 5         | 0.995071 |    |            | 5         | 0.99308  |
| 5  | 16406503   | 2         | 1        | 15 | 16434411   | 2         | 1        |
|    |            | 3         | 0.999434 |    |            | 3         | 0.999388 |
|    |            | 4         | 0.987911 |    |            | 4         | 0.990381 |
|    |            | 5         | 0.990183 |    |            | 5         | 0.851039 |
| 6  | 16406513   | 2         | 0.986831 | 16 | 16434453   | 2         | 1        |
|    |            | 3         | 0.996997 |    |            | 3         | 0.993229 |
|    |            | 4         | 0.994794 |    |            | 4         | 0.967014 |
|    |            | 5         | 0.984312 |    |            | 5         | 0.982524 |
| 7  | 16407187   | 2         | 1        | 17 | 16445122   | 2         | 0.998781 |
|    |            | 3         | 0.997659 |    |            | 3         | 0.999349 |
|    |            | 4         | 0.878351 |    |            | 4         | 0.997061 |
|    |            | 5         | 0.996419 |    |            | 5         | 0.94744  |
| 8  | 16424071   | 2         | 1        | 18 | 16445138   | 2         | 1        |
|    |            | 3         | 0.997648 |    |            | 3         | 0.998709 |
|    |            | 4         | 0.991567 |    |            | 4         | 0.996321 |
|    |            | 5         | 0.959043 |    |            | 5         | 0.994599 |
| 9  | 16424081   | 2         | 1        | 19 | 16445144   | 2         | 1        |
|    |            | 3         | 1        |    |            | 3         | 0.999081 |
|    |            | 4         | 0.954849 |    |            | 4         | 0.995619 |
|    |            | 5         | 0.971602 |    |            | 5         | 0.994153 |
| 10 | 16424106   | 2         | 1        | 20 | 16445168   | 2         | 1        |
|    |            | 3         | 0.978961 |    |            | 3         | 0.998022 |
|    |            | 4         | 0.997128 |    |            | 4         | 0.996986 |
|    |            | 5         | 0.99353  |    |            | 5         | 0.990996 |

the number of shared pixels by the total number of pixels, and finally multiplies the number by 100. This percentage gives the similarity between the two images. This similarity index is simple and vulnerable to small samples sizes. It can be defined as follows [15]:

$$J(X, Y) = |X \cap Y| / |X \cup Y|$$

## 5.4 Similarity results

Table 4 presents the average value of the PSNR between each segmented image (i.e., generated by the proposed approach) and its respective ground-truth image (i.e., generated by the Otsu method) that was obtained after repeating the same experiment 25 times. The table shows

**Table 6** Average value of the Jaccard index (20 datasets, 25 different runs for each experiment)

| #  | Patient ID | # Of thr. | Jaccard  | #  | Patient ID | # Of thr. | Jaccard  |
|----|------------|-----------|----------|----|------------|-----------|----------|
| 1  | 16406488   | 2         | 0.999955 | 11 | 16424120   | 2         | 1        |
|    |            | 3         | 0.981613 |    |            | 3         | 0.993089 |
|    |            | 4         | 0.999077 |    |            | 4         | 0.993548 |
|    |            | 5         | 0.958543 |    |            | 5         | 0.988738 |
| 2  | 16406490   | 2         | 1        | 12 | 16434363   | 2         | 1        |
|    |            | 3         | 0.997526 |    |            | 3         | 0.992903 |
|    |            | 4         | 0.645924 |    |            | 4         | 0.978777 |
|    |            | 5         | 0.986191 |    |            | 5         | 0.984631 |
| 3  | 16406498   | 2         | 1        | 13 | 16434368   | 2         | 1        |
|    |            | 3         | 0.999675 |    |            | 3         | 0.999716 |
|    |            | 4         | 0.989342 |    |            | 4         | 0.994441 |
|    |            | 5         | 0.995199 |    |            | 5         | 0.959493 |
| 4  | 16406502   | 2         | 0.996841 | 14 | 16434369   | 2         | 1        |
|    |            | 3         | 0.999401 |    |            | 3         | 1        |
|    |            | 4         | 0.993744 |    |            | 4         | 0.544799 |
|    |            | 5         | 0.988634 |    |            | 5         | 0.992978 |
| 5  | 16406503   | 2         | 1        | 15 | 16434411   | 2         | 1        |
|    |            | 3         | 0.999844 |    |            | 3         | 0.999431 |
|    |            | 4         | 0.980146 |    |            | 4         | 0.982615 |
|    |            | 5         | 0.981625 |    |            | 5         | 0.617592 |
| 6  | 16406513   | 2         | 0.989879 | 16 | 16434453   | 2         | 1        |
|    |            | 3         | 0.995785 |    |            | 3         | 0.994461 |
|    |            | 4         | 0.992992 |    |            | 4         | 0.965739 |
|    |            | 5         | 0.973302 |    |            | 5         | 0.973598 |
| 7  | 16407187   | 2         | 1        | 17 | 16445122   | 2         | 0.996711 |
|    |            | 3         | 0.995345 |    |            | 3         | 0.999807 |
|    |            | 4         | 0.678648 |    |            | 4         | 0.995945 |
|    |            | 5         | 0.993707 |    |            | 5         | 0.867911 |
| 8  | 16424071   | 2         | 1        | 18 | 16445138   | 2         | 1        |
|    |            | 3         | 0.997077 |    |            | 3         | 0.997058 |
|    |            | 4         | 0.985951 |    |            | 4         | 0.997613 |
|    |            | 5         | 0.889522 |    |            | 5         | 0.992204 |
| 9  | 16424081   | 2         | 1        | 19 | 16445144   | 2         | 1        |
|    |            | 3         | 1        |    |            | 3         | 0.999021 |
|    |            | 4         | 0.93825  |    |            | 4         | 0.986873 |
|    |            | 5         | 0.969069 |    |            | 5         | 0.993515 |
| 10 | 16424106   | 2         | 1        | 20 | 16445168   | 2         | 1        |
|    |            | 3         | 0.986755 |    |            | 3         | 0.997408 |
|    |            | 4         | 0.998627 |    |            | 4         | 0.993114 |
|    |            | 5         | 0.992681 |    |            | 5         | 0.978976 |

the average PSNR values when 2, 3, 4, and 5 thresholds are applied to segment the datasets.

The PSNR values presented in Table 4 indicate that the proposed approach is able to achieve very accurate results, where the PSNR value of 100 dB clearly indicates a high level of accuracy.

Tables 5 and 6 present the average values of the SSIM and Jaccard index, respectively. Three main observations

can be made about these outcomes. First, there is a high match between the segmented images and the ground-truth images when the ratio between the matched pixels and the total number of pixels is calculated. Second, there is a general tendency by the proposed approach to achieve better results when the number of thresholds is relatively small. Third, most of the records in Tables 5 and 6 are segmented at accuracies greater than 98%, which clearly



**Table 7** Average total time for 150 iterations and the time to converge (in sec) for each run

| #  | Patient ID | # Of thr. | Total time | Time to converge | #  | Patient ID | # Of thr. | Total time | Time to converge |
|----|------------|-----------|------------|------------------|----|------------|-----------|------------|------------------|
| 1  | 16406488   | 2         | 99.33306   | 14.70808         | 11 | 16424120   | 2         | 103.087    | 11.59913         |
|    |            | 3         | 99.4228    | 32.68711         |    |            | 3         | 102.4821   | 21.29996         |
|    |            | 4         | 100.0869   | 47.43099         |    |            | 4         | 101.1351   | 42.59933         |
|    |            | 5         | 100.8867   | 59.37802         |    |            | 5         | 101.1553   | 64.76624         |
| 2  | 16406490   | 2         | 120.8951   | 9.951318         | 12 | 16434363   | 2         | 132.9661   | 14.82821         |
|    |            | 3         | 120.9023   | 48.20626         |    |            | 3         | 133.4617   | 45.54483         |
|    |            | 4         | 125.0947   | 59.53491         |    |            | 4         | 133.6005   | 53.71276         |
|    |            | 5         | 122.7947   | 65.88505         |    |            | 5         | 137.0266   | 70.90094         |
| 3  | 16406498   | 2         | 119.2373   | 13.70028         | 13 | 16434368   | 2         | 123.3992   | 9.125473         |
|    |            | 3         | 119.8075   | 42.18315         |    |            | 3         | 124.3      | 42.3352          |
|    |            | 4         | 119.7914   | 62.41246         |    |            | 4         | 123.3049   | 59.64479         |
|    |            | 5         | 119.052    | 64.44114         |    |            | 5         | 121.9602   | 64.79661         |
| 4  | 16406502   | 2         | 122.1041   | 19.12062         | 14 | 16434369   | 2         | 140.0901   | 17.89299         |
|    |            | 3         | 120.5432   | 37.49903         |    |            | 3         | 135.9724   | 37.27247         |
|    |            | 4         | 121.0931   | 53.75128         |    |            | 4         | 135.4112   | 53.74089         |
|    |            | 5         | 121.3464   | 76.15637         |    |            | 5         | 135.3814   | 64.44001         |
| 5  | 16406503   | 2         | 129.1814   | 17.3181          | 15 | 16434411   | 2         | 116.8461   | 28.92963         |
|    |            | 3         | 128.1719   | 43.99658         |    |            | 3         | 116.4991   | 42.26079         |
|    |            | 4         | 129.4383   | 63.27366         |    |            | 4         | 116.7608   | 61.45203         |
|    |            | 5         | 129.0465   | 73.70594         |    |            | 5         | 116.8804   | 60.08481         |
| 6  | 16406513   | 2         | 120.871    | 10.91851         | 16 | 16434453   | 2         | 132.6529   | 9.854632         |
|    |            | 3         | 123.0058   | 30.695           |    |            | 3         | 131.05     | 46.19639         |
|    |            | 4         | 125.866    | 59.49019         |    |            | 4         | 131.469    | 45.90509         |
|    |            | 5         | 126.3875   | 73.85112         |    |            | 5         | 132.551    | 61.38263         |
| 7  | 16407187   | 2         | 118.8549   | 19.50727         | 17 | 16445122   | 2         | 118.8132   | 16.70446         |
|    |            | 3         | 118.9825   | 44.72838         |    |            | 3         | 116.1046   | 31.02404         |
|    |            | 4         | 118.6215   | 50.22636         |    |            | 4         | 115.3362   | 44.98582         |
|    |            | 5         | 118.9379   | 71.74427         |    |            | 5         | 116.0373   | 62.20557         |
| 8  | 16424071   | 2         | 127.3556   | 12.85624         | 18 | 16445138   | 2         | 147.0102   | 16.6875          |
|    |            | 3         | 127.5319   | 54.22377         |    |            | 3         | 147.54     | 35.463           |
|    |            | 4         | 127.1523   | 53.90645         |    |            | 4         | 147.6751   | 53.03595         |
|    |            | 5         | 133.0665   | 89.02275         |    |            | 5         | 149.806    | 75.82007         |
| 9  | 16424081   | 2         | 146.3925   | 14.39858         | 19 | 16445144   | 2         | 113.0392   | 17.31427         |
|    |            | 3         | 159.8886   | 40.56328         |    |            | 3         | 111.7535   | 40.76643         |
|    |            | 4         | 147.8028   | 52.99171         |    |            | 4         | 112.3271   | 46.3613          |
|    |            | 5         | 150.3438   | 78.41631         |    |            | 5         | 112.0688   | 65.08288         |
| 10 | 16424106   | 2         | 158.2036   | 15.30983         | 20 | 16445168   | 2         | 114.4999   | 23.31525         |
|    |            | 3         | 165.5247   | 37.40362         |    |            | 3         | 114.7055   | 43.63497         |
|    |            | 4         | 162.9541   | 52.89718         |    |            | 4         | 115.0052   | 62.77938         |
|    |            | 5         | 162.7715   | 81.35365         |    |            | 5         | 115.9229   | 62.69237         |

confirms the robustness of the results generated by the proposed approach.

## 6 Discussion and analysis

The average total time (in sec) required to process each image is shown in Table 7, from which it can be noted that the proposed approach converges in the early stages before

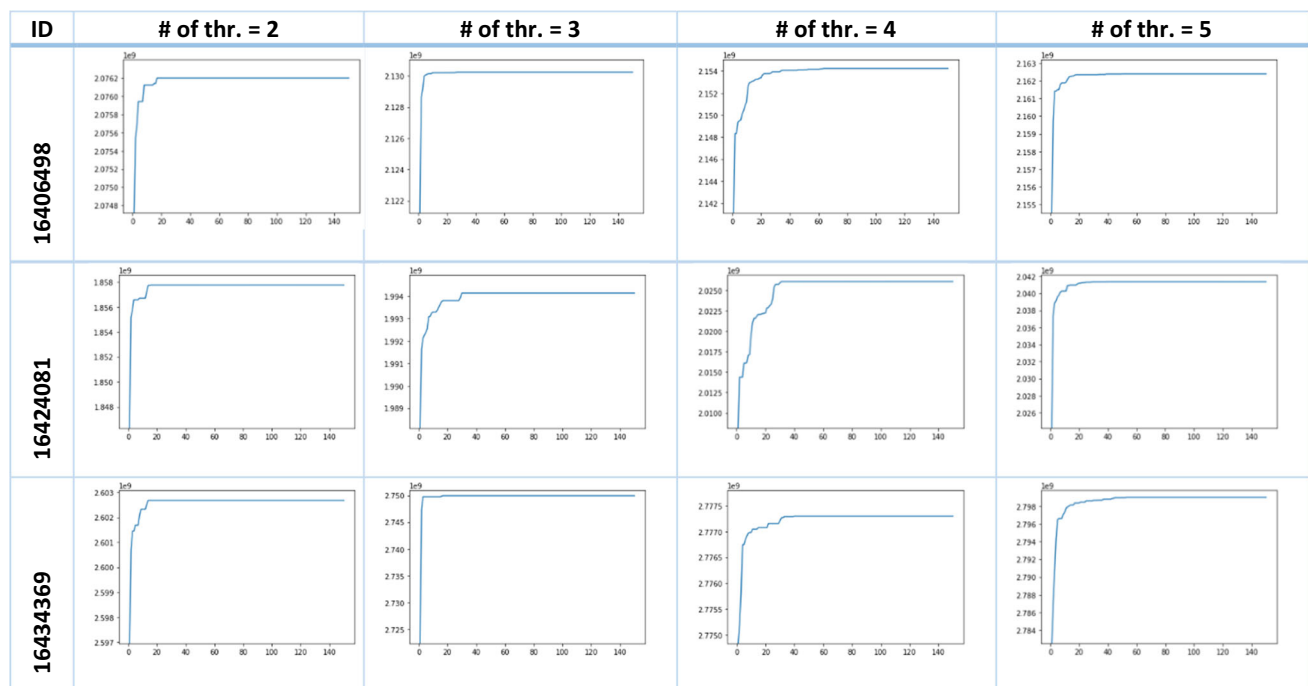


Fig. 4 Convergence plots of sample images

reaching 150 iterations. As shown in the table, there is a significant difference between the total time to complete 150 iterations and the time needed to converge. This further reinforces the findings on the effectiveness and suitability of the proposed approach for medical image segmentation because it can find a fast and accurate solution in a domain where both time and accuracy are crucial factors in real-time medical applications.

As shown in Table 7, there is a positive correlation between the time to converge and the number of thresholds. Therefore, the time to converge is proportional to the number of segmented clusters. It is remarkable that the time to converge in most images is about 7–15% of the total time when the number of thresholds = 2. Moreover, even when the number of thresholds = 5, which is a case considered to require a longer time for convergence to be achieved, the time to converge in most images is about 50–55% of the total time.

Figure 4 shows the convergence plots of three sample images at different threshold values. It is evident that the convergence behavior is present in all of the images regardless of the number of thresholds.

Table 8 presents the average number of iterations at which the convergence starts for each dataset. It can be seen that the number of iterations required to converge is proportional to the number of thresholds. However, as also shown in the table, a small number of iterations can lead to the required convergence and achieve optimal results.

The main contribution of this research is to achieve the optimal results provided by Otsu's algorithm while trying to reduce the massive number of computations it require. A set of experiments have been designed in order to compare and prove (a) the accuracy of outcomes and how they are matching the outcomes generated by Otsu, (b) the degree of similarity between the proposed approach and Otsu, and (c) the required time that the proposed approach need to reach the solution.

In Table 2, the threshold values generated by the Otsu method (i.e., the third column) are compared with the threshold values generated by the proposed approach (i.e., the fourth column). The values in the two columns are either the same or very close which proves the accuracy of outcomes generated by the proposed approach. Furthermore, Table 3 presents the average between-class variance which is an indicator to the accuracy of the outcomes. All results and comparisons are presented in details in the URL linked to this paper (<https://sites.google.com/view/hhocovid19/home>).

In Tables 4, 5, and 6, the degree of similarity between each segmented image (i.e., generated by the proposed approach) and its respective ground-truth image (i.e., generated by the Otsu method) is presented using PSNR, SSIM, and Jaccard index, respectively. The outcomes in these tables confirm again the accurate results that are generated by the proposed approach and how they are competitive when compared with the original method.

**Table 8** Average number of iterations at which convergence starts

| #               | Patient ID | # Of thr.     | Avg. # of iter. | #             | Patient ID    | # Of thr. | Avg. # of iter. |
|-----------------|------------|---------------|-----------------|---------------|---------------|-----------|-----------------|
| 1               | 16406488   | 2             | 19.76           | 11            | 16424120      | 2         | 13.84           |
|                 |            | 3             | 48.88           |               |               | 3         | 30.36           |
|                 |            | 4             | 71.76           |               |               | 4         | 64.24           |
|                 |            | 5             | 87.48           |               |               | 5         | 95.04           |
| 2               | 16406490   | 2             | 8.2             | 12            | 16434363      | 2         | 13.68           |
|                 |            | 3             | 60.88           |               |               | 3         | 51.36           |
|                 |            | 4             | 72.36           |               |               | 4         | 61.6            |
|                 |            | 5             | 80.72           |               |               | 5         | 78.68           |
| 3               | 16406498   | 2             | 14.52           | 13            | 16434368      | 2         | 7.08            |
|                 |            | 3             | 53.32           |               |               | 3         | 52              |
|                 |            | 4             | 79.12           |               |               | 4         | 73.16           |
|                 |            | 5             | 81.52           |               |               | 5         | 80.56           |
| 4               | 16406502   | 2             | 22.08           | 14            | 16434369      | 2         | 16.68           |
|                 |            | 3             | 47.48           |               |               | 3         | 41.56           |
|                 |            | 4             | 66.84           |               |               | 4         | 60.52           |
|                 |            | 5             | 93.16           |               |               | 5         | 72.56           |
| 5               | 16406503   | 2             | 17.56           | 15            | 16434411      | 2         | 34.2            |
|                 |            | 3             | 51.92           |               |               | 3         | 54.52           |
|                 |            | 4             | 73.28           |               |               | 4         | 79.24           |
|                 |            | 5             | 86.16           |               |               | 5         | 77.68           |
| 6               | 16406513   | 2             | 9.72            | 16            | 16434453      | 2         | 7.12            |
|                 |            | 3             | 36.92           |               |               | 3         | 53.04           |
|                 |            | 4             | 71.64           |               |               | 4         | 53.36           |
|                 |            | 5             | 87.72           |               |               | 5         | 71.16           |
| 7               | 16407187   | 2             | 22.28           | 17            | 16445122      | 2         | 18.36           |
|                 |            | 3             | 56.32           |               |               | 3         | 40.28           |
|                 |            | 4             | 64.68           |               |               | 4         | 59.72           |
|                 |            | 5             | 90.2            |               |               | 5         | 80.72           |
| 8               | 16424071   | 2             | 11.52           | 18            | 16445138      | 2         | 14.32           |
|                 |            | 3             | 62.8            |               |               | 3         | 35.96           |
|                 |            | 4             | 64.2            |               |               | 4         | 54.8            |
|                 |            | 5             | 99.2            |               |               | 5         | 76.64           |
| 9               | 16424081   | 2             | 11.6            | 19            | 16445144      | 2         | 20.84           |
|                 |            | 3             | 38.44           |               |               | 3         | 54.2            |
|                 |            | 4             | 55.12           |               |               | 4         | 63.32           |
|                 |            | 5             | 79.08           |               |               | 5         | 87.08           |
| 10              | 16424106   | 2             | 11.08           | 20            | 16445168      | 2         | 29.48           |
|                 |            | 3             | 33.72           |               |               | 3         | 56.88           |
|                 |            | 4             | 49.96           |               |               | 4         | 82.12           |
|                 |            | 5             | 75.64           |               |               | 5         | 81.4            |
| # of thresholds |            | # Of thr. = 2 | # Of thr. = 3   | # Of thr. = 4 | # Of thr. = 5 |           |                 |
| # of iterations |            | 16.196        | 48.042          | 66.052        | 83.12         |           |                 |

In order to present the required time that the proposed approach needs, Tables 7 and 8 present the average total time and the average number of iterations at which convergence starts since the original method (i.e., Otsu's original method) follows the brute force approach in which all combinations are evaluated.

## 7 Conclusion

Digital image processing algorithms have been supporting the medical field over the last few decades, and a huge number of algorithms have been specifically designed to assist radiologists and specialists to identify or study the

occurrence or the development of diseases. The universal transmission of COVID-19 has encouraged researchers in this domain to develop medical imaging solutions to help medical experts with decision making and diagnosis.

Image segmentation methods that are based on thresholding are considered an essential part of various domains including medical image processing. However, multilevel thresholding methods give improved segmentation results when compared to the standard single thresholding methods. Among the wide variety of available segmentation methods, the traditional standard benchmark statistical methods give optimal results at the cost of intensive computation when used for multilevel thresholding. Therefore, various metaheuristic optimization algorithms have been proposed and combined with statistical thresholding algorithms to reduce the number of required computations. In this research, the recent metaheuristic HHO algorithm was combined with the standard benchmark Otsu algorithm to perform image segmentation using multilevel image thresholding. The HHO algorithm is swarm based and has been proposed to efficiently handle continuous optimization tasks and produce high-quality solutions. The proposed approach was tested on a publicly available imaging dataset that contains chest images with clinical and genomic correlates and represents a rural COVID-19-positive population (COVID-19-AR).

Three well-known similarity metrics (PSNR, SSIM, and Jaccard index) were used to evaluate the accuracy of the results generated by the proposed approach against those produced by the Otsu method. After running the experiments 25 times, the average value of the PSNR was around 97%, 88%, 76%, and 76% when the number of thresholds was 2, 3, 4, and 5, respectively. The average value of the SSIM showed an overlap match of 99.9%, 99.7%, 97.9%, and 97.2% for 2, 3, 4, and 5 thresholds, respectively, and the Jaccard index showed a similarity of 99.9%, 99.6%, 95.4%, and 93.2% for the same number of thresholds. These results indicate that the proposed approach can achieve a massive reduction in the computational cost and simultaneously match the quality of images produced by the Otsu method for the same threshold values. It is worth mentioning here that the number of iterations in the experiment was limited to 150 due to the limitation of time. Thus, we envisage that forcing the algorithm to iterate a larger number of times will lead to better outcomes in terms of the degree of matching quality.

## Declarations

**Conflicts of interest** The authors declare no conflict of interest.

**Data availability** Publication of the annotated images database is available through this public repository (<https://sites.google.com/view/hhocovid19/data-set>).

**Code availability** The source code is available at this public repository (<https://sites.google.com/view/hhocovid19/source-code>).

## References

1. Abdel-Basset M, Mohamed R, Elhoseny M, Chakraborty RK, Ryan M (2020) A hybrid COVID-19 detection model using an improved marine predators algorithm and a ranking-based diversity reduction strategy. *IEEE Access* 8:79521–79540
2. Ahmadi M, Kazemi K, Aarabi A, Niknam T, Helfroush MS (2019) Image segmentation using multilevel thresholding based on modified bird mating optimization. *Multimed Tools Appl* 78(16):23003–23027. <https://doi.org/10.1007/s11042-019-7515-6>
3. Akram S, Javed MY, Akram MU, Qamar U, Hassan A (2016) Pulmonary nodules detection and classification using hybrid features from computerized tomographic images. *J Med Imaging Health Inform* 6(1):252–259. <https://doi.org/10.1166/jmihi.2016.1600>
4. Baby Resma KP, Nair MS (2018) Multilevel thresholding for image segmentation using krill herd optimization algorithm. *J King Saud Univ Comput Inf Sci*. <https://doi.org/10.1016/j.jksuci.2018.04.007>
5. Chen H, Heidari AA, Zhao X, Zhang L, Chen H (2020) Advanced orthogonal learning-driven multi-swarm sine cosine optimization: framework and case studies. *Expert Syst Appl* 144:113113. <https://doi.org/10.1016/j.eswa.2019.113113>
6. Chen H, Li S, Heidari AA, Wang P, Li J, Yang Y, Wang M, Huang C (2020) Efficient multi-population outpost fruit fly-driven optimizers: framework and advances in support vector machines. *Expert Syst Appl* 142:112999. <https://doi.org/10.1016/j.eswa.2019.112999>
7. Chen H, Yang C, Heidari AA, Zhao X (2020) An efficient double adaptive random spare reinforced whale optimization algorithm. *Expert Syst Appl* 154:113018. <https://doi.org/10.1016/j.eswa.2019.113018>
8. da Silva Sousa JRF, Silva AC, de Paiva AC, Nunes RA (2010) Methodology for automatic detection of lung nodules in computerized tomography images. *Comput Methods Programs Biomed* 98(1):1–14. <https://doi.org/10.1016/j.cmpb.2009.07.006>
9. Desai S, Baghal A, Wongsurawat T, Jenjaroenpun P, Powell T, Al-Shukri S, Gates K, Farmer P, Rutherford M, Blake G, Nolan T (2020) Chest imaging representing a COVID-19 positive rural U.S. population. *Sci Data* 7(1):414. <https://doi.org/10.1038/s41597-020-00741-6>
10. Desai S, Baghal A, Wongsurawat T, Al-Shukri S, Gates K, Farmer P, Rutherford M, Blake GD, Nolan T, Powell T, Sexton K, Bennett W, Prior F (2020) Data from chest imaging with clinical and genomic correlates representing a rural COVID-19 positive population data set. The Cancer Imaging Archive
11. Desai S, Baghal A, Wongsurawat T, Al-Shukri S, Gates K, Farmer P, Rutherford M, Blake GD, Nolan T, Powell T, Sexton K, Bennett W, Prior F (2020) Chest imaging with clinical and genomic correlates representing a rural COVID-19 positive population
12. Dorgham O, Fisher M, Laycock S (2009) Performance of a 2D–3D image registration system using (lossy) compressed x-ray CT. *Ann BMVA* 3:1–11
13. Dorgham O, Ryalat MH, Naser MA (2020) Automatic body segmentation for accelerated rendering of digitally reconstructed

- radiograph images. *Inform Med Unlocked* 20:100375. <https://doi.org/10.1016/j.imu.2020.100375>
14. Dorgham OM, Alweshah M, Ryalat MH, Alshaer J, Khader M, Alkhalaileh S (2021) Monarch butterfly optimization algorithm for computed tomography image segmentation. *Multimed Tools Appl*. <https://doi.org/10.1007/s11042-020-10147-6>
  15. Eelbode T, Bertels J, Berman M, Vandermeulen D, Maes F, Bisschops R, Blaschko MB (2020) Optimization for medical image segmentation: theory and practice when evaluating with dice score or jaccard index. *IEEE Trans Med Imaging* 39(11):3679–3690
  16. Elaziz MA, Lu S (2019) Many-objectives multilevel thresholding image segmentation using knee evolutionary algorithm. *Expert Syst Appl* 125:305–316. <https://doi.org/10.1016/j.eswa.2019.01.075>
  17. Elaziz MA, Oliva D, Ewees AA, Xiong S (2019) Multi-level thresholding-based grey scale image segmentation using multi-objective multi-verse optimizer. *Expert Syst Appl* 125:112–129. <https://doi.org/10.1016/j.eswa.2019.01.047>
  18. Fisher M, Dorgham O, Laycock SD (2013) Fast reconstructed radiographs from octree-compressed volumetric data. *Int J Comput Assist Radiol Surg* 8(2):313–322. <https://doi.org/10.1007/s11548-012-0783-5>
  19. Furqan S, Gulistan R, Alejandro FF (2019) Computer-aided detection of lung nodules: a review. *J Med Imaging* 6(2):1–11. <https://doi.org/10.1117/1.JMI.6.2.020901>
  20. Gezici H, Livatyah H (2022) Chaotic Harris hawks optimization algorithm. *J Comput Des Eng* 9(1):216–245. <https://doi.org/10.1093/jcde/qwab082>
  21. Gruetzemacher R, Gupta A, Paradise D (2018) 3D deep learning for detecting pulmonary nodules in CT scans. *J Am Med Inform Assoc* 25(10):1301–1310. <https://doi.org/10.1093/jamia/ocy098>
  22. Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019) Harris hawks optimization: algorithm and applications. *Futur Gener Comput Syst* 97:849–872. <https://doi.org/10.1016/j.future.2019.02.028>
  23. Jiang H, Ma H, Qian W, Gao M, Li Y (2018) An automatic detection system of lung nodule based on multigroup patch-based deep learning network. *IEEE J Biomed Health Inform* 22(4):1227–1237. <https://doi.org/10.1109/JBHI.2017.2725903>
  24. Jo HH, Hong H, Mo Goo J (2014) Pulmonary nodule registration in serial CT scans using global rib matching and nodule template matching. *Comput Biol Med* 45:87–97. <https://doi.org/10.1016/j.combiomed.2013.10.028>
  25. 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. [https://doi.org/10.1016/0734-189X\(85\)90125-2](https://doi.org/10.1016/0734-189X(85)90125-2)
  26. Keshani M, Azimifar Z, Tajeripour F, Boostani R (2013) Lung nodule segmentation and recognition using SVM classifier and active contour modeling: a complete intelligent system. *Comput Biol Med* 43(4):287–300. <https://doi.org/10.1016/j.combiomed.2012.12.004>
  27. Kumar AS, Kumar A, Bajaj V, Singh, GK (2018) Fractional-order darwinian swarm intelligence inspired multilevel thresholding for mammogram segmentation. Paper presented at the 2018 international conference on communication and signal processing (ICCSP)
  28. Li Z, Sun Y, Zhang L, Tang J (2021) CTNet: context-based tandem network for semantic segmentation. *IEEE Trans Pattern Anal Mach Intell*. <https://doi.org/10.1109/TPAMI.2021.3132068>
  29. Liang H, Jia H, Xing Z, Ma J, Peng X (2019) Modified grasshopper algorithm-based multilevel thresholding for color image segmentation. *IEEE Access* 7:11258–11295. <https://doi.org/10.1109/ACCESS.2019.2891673>
  30. Shapiro LG, Stockman GC (2001) Computer vision, 1st edn. Prentice Hall, Hoboken
  31. Magalhães Barros Netto S, Corrêa Silva A, Acatauassú Nunes R, Gattass M (2012) Automatic segmentation of lung nodules with growing neural gas and support vector machine. *Comput Biol Med* 42(11):1110–1121. <https://doi.org/10.1016/j.combiomed.2012.09.003>
  32. Messay T, Hardie RC, Rogers SK (2010) A new computationally efficient CAD system for pulmonary nodule detection in CT imagery. *Med Image Anal* 14(3):390–406. <https://doi.org/10.1016/j.media.2010.02.004>
  33. 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. <https://doi.org/10.1016/j.advengsoft.2017.07.002>
  34. Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67. <https://doi.org/10.1016/j.advengsoft.2016.01.008>
  35. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>
  36. Mousavirad SJ, Ebrahimpour-Komleh H (2020) Human mental search-based multilevel thresholding for image segmentation. *Appl Soft Comput* 97:105427. <https://doi.org/10.1016/j.asoc.2019.04.002>
  37. Narayanan BN, Hardie RC, Kebede TM, Sprague MJ (2019) Optimized feature selection-based clustering approach for computer-aided detection of lung nodules in different modalities. *Pattern Anal Appl* 22(2):559–571. <https://doi.org/10.1007/s10044-017-0653-4>
  38. 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. <https://doi.org/10.1007/s00500-017-2794-1>
  39. Otsu N (1979) A threshold selection method from gray-level histograms. *IEEE Trans Syst Man Cybern* 9(1):62–66
  40. Pare S, Kumar A, Singh GK, Bajaj V (2020) Image segmentation using multilevel thresholding: a research review. *Iran J Sci Technol Trans Electr Eng* 44(1):1–29. <https://doi.org/10.1007/s40998-019-00251-1>
  41. Pare S, Bhandar AK, Kumar A, Singh GK (2019) Rényi's entropy and bat algorithm based color image multilevel thresholding. In: Machine intelligence and signal analysis. Advances in intelligent systems and computing, pp 71–84, Springer, Singapore
  42. Pehrson LM, Nielsen MB, Ammitzbøl Lauridsen C (2019) Automatic pulmonary nodule detection applying deep learning or machine learning algorithms to the LIDC-IDRI database: a systematic review. *Diagnostics*. <https://doi.org/10.3390/diagnostics9010029>
  43. Pu J, Zheng B, Leader JK, Wang X-H, Gur D (2008) An automated CT based lung nodule detection scheme using geometric analysis of signed distance field. *Med Phys* 35(8):3453–3461. <https://doi.org/10.1118/1.2948349>
  44. Ryalat MH, Laycock S, Fisher M (2017) Automatic removal of mechanical fixations from ct imagery with particle swarm optimisation. Paper presented at the bioinformatics and biomedical engineering, Cham
  45. Ryalat MH, Laycock S, Fisher M (2017) A fast and automatic approach for removing artefacts due to immobilisation masks in X-ray CT. Paper presented at the 2017 IEEE EMBS international conference on biomedical and health informatics (BHI). IEEE, New York
  46. Saremi S, Mirjalili S, Lewis A (2017) Grasshopper optimisation algorithm: theory and application. *Adv Eng Softw* 105:30–47. <https://doi.org/10.1016/j.advengsoft.2017.01.004>



47. Serhat Ozekes OO, Ucan ON (2007) Nodule detection in a lung region that's segmented with using genetic cellular neural networks and 3D template matching with fuzzy rule based thresholding. *Korean J Radiol* 9(1):1–9
48. Sharma A, Kumar S, Singh SN (2019) Brain tumor segmentation using DE embedded OTSU method and neural network. *Multidimens Syst Signal Process* 30(3):1263–1291. <https://doi.org/10.1007/s11045-018-0603-3>
49. Tan M, Deklerck R, Jansen B, Bister M, Cornelis J (2011) A novel computer-aided lung nodule detection system for CT images. *Med Phys* 38(10):5630–5645. <https://doi.org/10.1118/1.3633941>
50. Tarkhaneh O, Shen H (2019) An adaptive differential evolution algorithm to optimal multi-level thresholding for MRI brain image segmentation. *Expert Syst Appl* 138:112820. <https://doi.org/10.1016/j.eswa.2019.07.037>
51. Tedmori S, Al-Najdawi N (2012) Lossless image cryptography algorithm based on discrete cosine transform. *Int Arab J Inf Technol* 9(5):471–478
52. Upadhyay P, Chhabra JK (2020) Kapur's entropy based optimal multilevel image segmentation using crow search algorithm. *Appl Soft Comput* 97:105522. <https://doi.org/10.1016/j.asoc.2019.105522>
53. Zhang G, Jiang S, Yang Z, Gong L, Ma X, Zhou Z, BaoLiu CQ (2018) Automatic nodule detection for lung cancer in CT images: a review. *Comput Biol Med* 103:287–300. <https://doi.org/10.1016/j.combiomed.2018.10.033>
54. Zhou C, Tian L, Zhao H, Zhao K (2015) A method of two-dimensional otsu image threshold segmentation based on improved firefly algorithm. Paper presented at the 2015 IEEE international conference on cyber technology in automation, control, and intelligent systems (CYBER). IEEE, New York

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.