Multimedia Tools and Applications (2024) 83:41051–41081
https://doi.org/10.1007/s11042-023-17221-9

Adapted Arithmetic Optimization Algorithm for Multi-level Thresholding Image Segmentation: A case Study of chest X-ray Images

Mohammad Otair[1] · Laith Abualigah[2,3,4,5,6,7,8] · Saif Tawfiq[1] · Mohammad Alshinwan[9] · Absalom E. Ezugwu[10] · Raed Abu Zitar[11] · Putra Sumari[7]

Received: 4 August 2022 / Revised: 21 June 2023 / Accepted: 21 September 2023 /
Published online: 11 October 2023
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract
Particularly in recent years, there has been increased interest in determining the ideal thresholding for picture segmentation. The best thresholding values are found using various techniques, including Otsu and Kapur-based techniques. These techniques work well for bi-level thresholding, but when used to find the appropriate thresholds for multi-level thresholding, there will be issues with long calculation times, high computational costs, and the need for accuracy improvements. This work investigates the capability of the Arithmetic Optimization Algorithm to discover the best multilayer thresholding for picture segmentation to circumvent this issue. The leading mathematical arithmetic operators' distributional nature is used by the AOA method. The picture histogram was used to construct the candidate solutions in the modified algorithms, which were then updated according to the algorithm's features. The solutions are evaluated using Otsu's fitness function throughout the optimization process. The picture histogram is used to display the algorithm's potential solutions. The proposed approach is tested on five frequent photos from the Berkeley University database. The fitness function, root-mean-squared error, peak signal-to-noise ratio, and other widely used assessment metrics were utilized to assess the performance of the suggested segmentation approach. Many benchmark pictures were employed to verify the suggested technique's effectiveness and evaluate it against other well-known optimization methods described in the literature.

Keywords Image segmentation · Multi-level thresholding · Meta-heuristic algorithms · Arithmetic optimization algorithm

1 Introduction

Dividing an image into multiple sections or regions is called image segmentation [1, 2]. This means that the total number of pixels in that segment is similar regarding some criteria, such as color and density [3]. The pixel property and its information about closeness to the rest of the pixels are essential parameters of any algorithm that can be used

Extended author information available on the last page of the article

in segmentation techniques. Pixel similarity can also represent an image area. If similarity breaks, it signifies edges in the image [4]. The segmentation process separates objects based on the pixel density that forms them [5]. The basic idea is to separate pixels into homogeneous classes formed as a result of the common characteristics of the pixel [6].

One of the most important factors affecting image processing is dividing pixels into different groups. It allows us to separate and identify the objects in the image [1, 7]. Different ways are used to segment the images; the simplest is the Threshold method [8]. The image histogram and the number of thresholds are the most important factors influencing the Threshold method. Thresholds are used to break the graph into different classes. In the case of a single threshold, the pixels are divided into two classes and named a bi-level threshold. If the number of thresholds is more than one, there will be more than two classes, which is called a multi-level threshold (MTH). The main problem in multi-level thresholds is finding the best thresholds in the image relative to the number of classes (or called the objects in the image) [9, 10].

Several measures are used to determine the optimum thresholds using variance or entropy. Otsu method (1979) uses contrast, while the Kapoor method uses entropy [11, 12]. Each of these methods is effective in the case of one threshold. When the number of thresholds increases to more than one, it is necessary to use meta-heuristic algorithms (MA) because they provide accurate results in most cases.

The Meta-heuristic takes advantage of random operators because it belongs to the family of stochastic improvements [13–16]. This helps them avoid random solutions while solving real problems, which usually have many local options. These advantages have a role in making Meta-heuristic to be included in various applications of science and industry [17, 18]. Meta-heuristic optimization algorithms are based on two basic tasks, which are 1) Exploration: which represents the ability of the algorithm to explore non-local (global) search areas. 2) Exploitation: represents the ability of the algorithm to explore high-quality optimization solutions within those global solutions. An appropriate balance between these strategies (exploration and exploitation) must exist. All optimization algorithms based on the population principle use these two strategies, but they differ according to each algorithm's mechanisms and operating factors [19, 20]. Among the most important classifications on which the meta-heuristics are based on swarm intelligence algorithms are physics-based or human-based methods and evolutionary algorithms [21, 22].

Image segmentation is dividing an image into sections [23]. The splitting process is based on the similarity of pixels in some aspects, including density and color. The segmentation process recognizes existing objects and borders in an image. In the segmentation process, the output is represented by pixels that form together to represent the object. Segmentation is stopped when the object in the image is isolated [23]. Image segmentation is an important and difficult problem at the same time. It is an important problem in image processing, artificial intelligence, and pattern recognition. It is also a major step in computer vision technology. It is often used in medical applications for tumor detection, tissue volume measurement, diagnosis and planning, computer surgery, and more. It is also used to reorganize the face, locate objects in satellites, and reorganize fingerprints, filter distortions in the image, and manipulate images by remote sensing [24]. So, segmentation is an important process in analyzing images for various fields and applying them on a large scale [25].

The extremely effective computer-aided detection approach for acute lymphoblastic leukemia (ALL) detection is presented in this work [26]. The suggested strategy consists of two separate phases. White blood cells (WBCs) are precisely segmented in the initial stage using tiny blood pictures. Important properties, such as form and texture, are retrieved from the segmented cells in the second step. Based on the collected characteristics, Nave Bayes and k-nearest neighbor classifier algorithms are used to categorizing

the segmented cells into normal and pathological categories. The performance of the suggested technique is thoroughly assessed using the well-known ALL-IDB dataset, which comprises microscopic blood pictures. The results of extensive testing show the suggested approach's excellent performance, outperforming state-of-the-art approaches with a remarkable accuracy rate of 98.7%.

This project aims to create an automated system that is incredibly efficient at correctly recognizing acute lymphoblastic leukemia cells [27]. The suggested strategy consists of two key phases. The goal of the first phase, which involves segmenting white blood cells, is to identify the target cells precisely. The segmented cells are then used to extract a wide range of relevant aspects, including shape, geometrical, statistical, and discrete cosine transform (DCT) data. These characteristics form the basis for classifying the segmented cells as normal or aberrant using a variety of classification approaches. Extensive tests are carried out using the well-known ALL-IDB dataset, which consists of microscopic blood pictures, to assess the performance of the suggested technique. The trial outcomes demonstrate the effectiveness of the suggested strategy, with a remarkable accuracy rate of 97.45%. These results demonstrate the superiority of the suggested method over currently used techniques, emphasizing its potential to advance the precise detection of cells considerably.

The specific local features that are present in different viewpoints are frequently ignored by existing approaches, which reduces their ability to discriminate when faced with noisy samples of the data [28]. They suggested a novel method called the multi-view low-rank representation method to overcome this issue by concurrently utilizing the internal relationships and unique local structures of several perspectives. They illustrate the efficiency of our suggested strategy in contrast to related state-of-the-art approaches by comprehensive tests carried out on several datasets, especially in tasks like classification and clustering. The results show how the suggested approach has the potential to increase performance in difficult situations while overcoming the drawbacks of current strategies by adding particular local structures. In this study [29], a unique methodology known as resilient low-rank representation using residual projection is introduced. By investigating the underlying structure of residues, as opposed to current approaches that regard residues as noise or outliers, our suggested method intends to identify robust characteristics. Our solution consistently beats comparative approaches in terms of classification outcomes after extensive tests on corrupted datasets. These results demonstrate the potency of our suggested approach as a robust approach to addressing noisy datasets, demonstrating its ability to provide superior robust low-rank representation in the face of noise.

One of the most important problems in image analysis and processing is segmenting pixels into categories. Segmentation aims to identify elements, isolate them from the background, and distinguish between pixels to improve contrast. Several important methods are used to solve this problem. The most common segmentation techniques researchers still use nowadays are Threshold, Histogram, Edge detection, Watershed Transformation, clustering, and Region-based methods.

The segmentation method separates pixels into different classes depending on their intensity level and according to one or more threshold values. One of this method's most important advantages is its ease of implementation, as it is one of the simplest partitioning methods. This is because the segmented images created from the threshold are characterized by small storage space, fast processing, and ease of processing. Into consideration, the segmentation using thresholds takes two important factors: the histogram and the threshold set. This method adopts the histogram to divide the image into its different categories.

The segmentation process is important and has a main role in wide fields, and one such process is image processing. Most researchers want to produce new methodologies to

obtain more accurate excised organisms. This study gives practical and academic importance for researchers when achieving all objectives and helping anyone interested in image segmentation and optimization algorithms. An image is divided into classes pertaining to various objects using a multi-level threshold. Consequently, extracting more data and objects from the segmented image is feasible. In order to solve the optimization challenge of determining the best thresholding for picture segmentation in a multi-level thresholding state, Otsu's approach has been applied as a fitness function. The study employed a brand-new technique called the Arithmetic Optimization Algorithm (AOA) to overcome this issue. This study determined the ideal threshold for multi-level image segmentation using the Arithmetic Optimization Algorithm (AOA) [30].

This study focuses on chest X-ray pictures and proposes a unique method to tackle the difficult job of image segmentation. The major goal of the study is to provide a better Arithmetic Optimization Algorithm (AOA) to increase the precision and effectiveness of multilayer thresholding strategies. The importance of picture segmentation in medical applications is highlighted in the paper's first paragraph, focusing on chest X-ray analysis to diagnose various lung ailments. It highlights the necessity for reliable segmentation algorithms to precisely distinguish lung sections from the surrounding tissue and any potential anomalies. The new AOA method is presented to get beyond the constraints of the pre-existing multi-level thresholding strategies. The method uses arithmetic optimization principles to choose the best threshold values, integrating arithmetic progression and optimization techniques. By identifying the ideal threshold configuration that optimizes the similarity between segmented areas and ground truth pictures, it seeks to minimize the objective function. Using a dataset of chest X-ray pictures, a thorough case study is undertaken to verify the efficiency of the suggested approach. The experimental findings show that the modified AOA performs better than cutting-edge techniques in terms of segmentation accuracy, resilience, and computing efficiency. It performs better in properly defining lung areas and spotting anomalies, which leads to more precise diagnosis and treatment choices. The paper's conclusion covers the possible uses and future possibilities of the enhanced AOA algorithm. It emphasizes its importance in medical image processing and encourages more research to gauge its performance across various medical imaging modalities and datasets. Ultimately, this study provides insightful information into picture segmentation, gives a possible remedy for enhanced chest X-ray analysis, and raises the general medical diagnosis standard.

The rest of this paper is organized as follows. Section 2 shows the proposed method. Section 3 shows the experiments and results. Section 4 shows the conclusion and future work directions.

2 The proposed method

According to the problem mentioned before, it is necessary to use an optimization algorithm to deal with multi-level thresholding to find the optimum threshold. The optimal values of thresholds are set by maximizing some standards such as the Otsu standard [31]. In the Otsu-based method, the best value of the threshold is maximizing the variance between classes. The choice of optimal thresholds in multi-level categories was a challenge over decades. When the number of thresholds decreases, the classic techniques will be appropriate, but the problem is at increasing the number of thresholds. This affects segmentation accuracy; AOA algorithms are used and tested on several images to avert this weakness.

In this section, we presented the formulation of the problem, AOA, and its main items. Beginning with initialize parameter and solution. Each solution's exploration and exploitation will be enhanced to determine the best fitness, as shown in Fig. 1.

2.1 Formulation of the problem

Parametric and Non-parametric approaches are used to set the optimal thresholds. In Parametric, several statistical parameters, e.g. (time-consuming, accuracy), should be calculated for each class in the image. In Non-parametric, the optimal values of thresholds are set by maximizing some standard, such as the Otsu standard.

In the Otsu-based method, the best value of the threshold is maximizing the variance between classes. Kapur measures the homogeneity of the classes using maximizing entropy. So, choosing optimal thresholds in multi-level categories is a Nonparametric-hard problem, and it has been a challenge for decades [32]. When the number of thresholds decreases, the classic techniques will be appropriate, but the problem is increasing the number of thresholds. And this affects segmentation accuracy.

The problem description of multi-level thresholding is presented. The gray level of each image is presented as I, which should be segmented into $K+1$ classes. In each segmentation step, the pixels of the image can be divided into I classes as given in Eq. (1). The results are given by determined K thresholds ($t_1, t_2, \ldots, t_K$).

$$\begin{aligned} G_0 &= \{I_{ij} \in I | 0 \le I_{ij} \le t_1 - 1\} \\ G_1 &= \{I_{ij} \in I | t_1 \le I_{ij} \le t_2 - 1\} \\ G_k &= \{I_{ij} \in I | t_k \le I_{ij} \le L - 1\} \end{aligned} \tag{1}$$

G_K is *k-th* class of the tested image *(I)*, *I (i, j)* is the gray level of the pixel *(i, j)*, t_K where $(k = 1, \ldots, K)$ that defines the *k-th* threshold value, and L is the gray levels of *I*. All the used levels are presented as in *(0, 1, …, L—1)*.

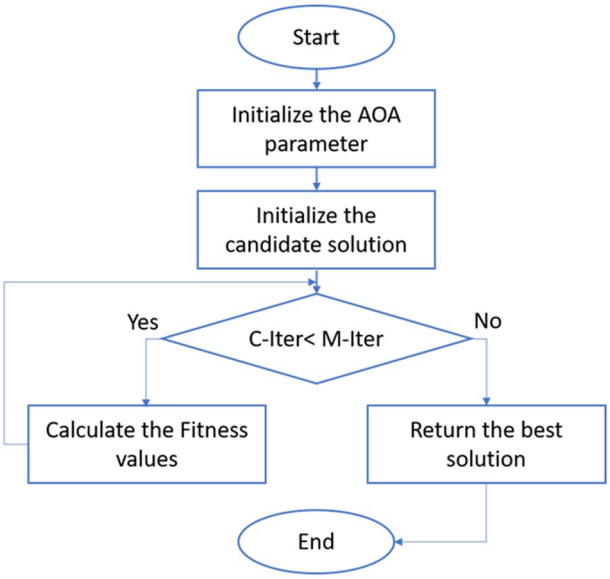


Fig. 1 The general steps of the proposed method

Multi-level thresholding aims to find the best threshold levels that split the pixels of I into many groups; these parameters are found using optimization techniques. Otsu's approach is one of the popular techniques and well-known optimization functions. Both multi-level and bi-level segmentation issues can use it. Otsu's method finds the best threshold values for I in multi-level thresholding by maximizing Eq. (2). It can be presented as follows:

$$t_1^*, t_2^*, ..., t_k^* = \max_{t_1,....,t_k} F(t_1, t_2, ..., t_k) \tag{2}$$

where $F\ (t_1, ..., t_K)$ is the intra-class difference that is presented as follows:

$$F = \sum_{i=0}^{k} A_i(\eta_1 - \eta_2)^2 \tag{3}$$

$$A_i = \sum_{j=t_i}^{t_{i+1}-1} P_j \tag{4}$$

$$\eta_i = \sum_{j=t_i}^{t_{i+1}-1} i\frac{P_j}{A_j}, where\ P_i = h(i)/N \tag{5}$$

where η_1 is to mean intensity of the tested image (I), with $t_0=0$ and $t_{K+1}=L$. $h(i)$ is the frequency and P_j is the probability of the *ith* gray level. N presents the used pixels in the evaluated image (I).

2.2 Arithmetic optimization algorithm (AOA)

The principle of optimization processes for different algorithms is based on creating a set of randomly candidate solutions. The essence of optimization methods is to optimize new solutions with a set of rules that differ from one algorithm to another. These solutions are frequently evaluated to find the best solution. And because these algorithms seek the best solution, it became important not to be satisfied with one run. The probability of reaching an optimal solution increases with a sufficient number of random solutions plus several optimization iterations.

Arithmetic Optimization Algorithm (AOA): It is a new meta-heuristic algorithm. The behavior of AOA is the distribution of the main arithmetic operators in mathematics, including Multiplication (M), Division (D), Subtraction (S), and Addition (A)). AOA is implemented to execute the optimization processes in a massive range of search spaces. The algorithm performs 29 benchmark functions and many real-world engineering design problems to show its applicability. The analysis of convergence behaviors, performance, and the computational complexity of the algorithm is rated by different scenarios.

Many algorithms are used in the optimization field, each with its methods and rules for optimization. The optimization process is generally divided into two basic stages: exploration and exploitation. The first stage (exploration) covers a wide range of research areas, using agents to avoid local solutions, while the second stage, which is exploitation, is to reach the optimal solution obtained from the previous stage. concerning the proposed algorithm in our study, arithmetic factors in mathematics represent both the exploration and exploitation phases as in (1) addition (A "+"), (2) subtraction (S "-"), (3) division (D /"), and (4) (M "×"). The AOAa algorithm is a population-based algorithm capable of solving optimization problems without accounting for their derivatives.

The improvement process starts with several candidate solutions (X), as shown in Matrix (1).

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & x_{1,n} \\ x_{2,1} & & x_{2,j} & x_{2,n} \\ \vdots & \ddots & \vdots & \\ x_{N-1,1} & \cdots & x_{N-1,j} & x_{N-1,n-1} \\ x_{N,1} & & x_{N,j} & x_{N,n} \end{bmatrix} \tag{6}$$

At first, it is important to select the search phase (i.e., exploration or exploitation). The Math Optimizer Accelerated (M O A) function is a coefficient calculated by Eq. (7) and used in these phases.

$$MOA(C_Iter) = Min + C_Iter \times \left(\frac{Max - Min}{C_Iter}\right) \tag{7}$$

M O A (C_Iter) is the function value at the tth iteration, calculated in Eq. (7). *C_Iter* refers to the current iteration between 1 to *M_Iter. M_Iter* refers to the maximum number of iterations, *Min* and *Max* refer to the minimum and maximum values of the accelerated function, respectively. Figure 2 shows the steps of the AOA process.

2.3 Exploration phase

The two primary search techniques are represented in Eq. (8), and the exploration operators of the AOA approach discover a better answer by exploring the search area randomly in many places.

$$x_{i,j}(C_{Iter}) = \begin{cases} best(x_j) \div MOP \times \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right), r_3 < 0.5 \\ best(x_j) \times MOP \times \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right), otherwise \end{cases} \tag{8}$$

$$MOA(C_Iter) = 1 - \frac{C_Iter^{1/\propto}}{M_Iter^{1/\propto}}) \tag{9}$$

2.4 Exploitation phase

In this stage, a deep search is used to use the search space fully. The other operator (A) will not be taken into consideration until the first operator (S) in this phase (first rule in Eq. (10)) completes its present duty, which is conditioned by r<3 0.5. If not, the S will be substituted with the second operator (A) to complete the present job.

$$x_{i,j}(C_{Iter}) = \begin{cases} best(x_j) - MOP \times \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right), r_3 < 0.5 \\ best(x_j) + MOP \times \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right), otherwise \end{cases} \tag{10}$$

2.5 Pseudo-code of the arithmetic optimization algorithm (AOA)

This can be summed up: the AOA algorithm begins with random solutions. Factors D, M, S, and A estimate where solutions are close to optimal. Then each solution revamps its site to approach the best solution. The factor MOA will change its value from 0.2 to 0.9. Whenever the value of MOA<r_1, it moves away from the near-optimal solution. If it is MOA>r_1, it

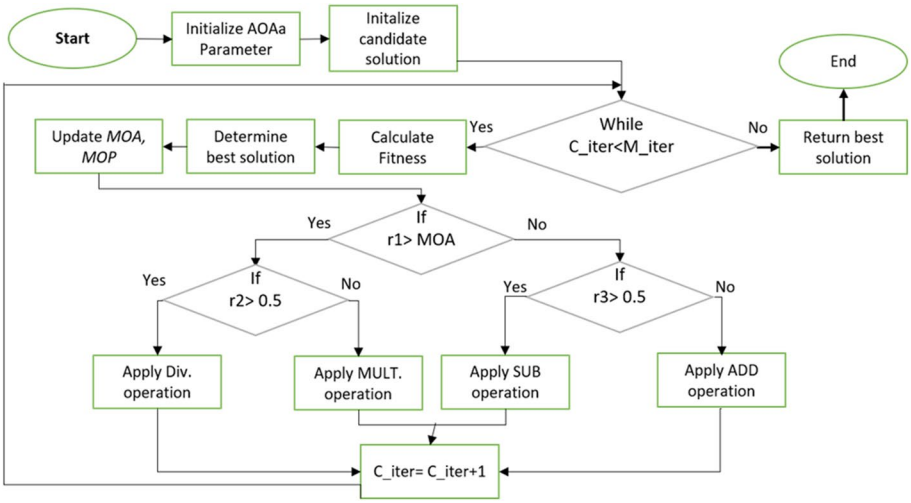


Fig. 2 Flowchart of the Arithmetic Optimization Algorithm

approaches the near-optimal solution. Eventually, the AOA algorithm is stopped by reaching the criterion, as shown in Algorithm 1.

3 Experiments and results

The proposed method is evaluated and compared with similar methods from the literature.

3.1 Experimental settings

In order to be fair, the comparison of all algorithms employed in the study uses identical cessation circumstances (maximum iterations set to 100), an entire run of 30, and a population size of (25). Images are processed using the following thresholds for the performance evaluation of all test experiments: 2, 3, 4, 5, and 6. The decision to use these criteria demonstrated how well the algorithm (AOA) performed compared to more established swarm-based segmentation techniques.

3.2 Performance measures

Four parameters are employed to assess the segmented image's fitness. The following criteria were used to evaluate the suggested algorithms:

1. The execution time: To evaluate and examine each method, the consumption time for each image segment is computed.
2. The fitness function value: Using each solution's present location as a starting point, the fitness value is generated to evaluate and analyze each solution. By maximizing the fit-

ness function, Otsu's method for multi-level thresholding determines the ideal threshold values for a picture.

$$F = \sum_{i=0}^{k} A_i(\eta_1 - \eta_2)^2 \tag{11}$$

$$A_i = \sum_{j=t_i}^{t_{i+1}-1} P_j \tag{12}$$

$$\eta_i = \sum_{j=t_i}^{t_{i+1}-1} i\frac{P_j}{A_j}, where\ P_i = h(i)/N \tag{13}$$

where η_1 points to the mean intensity of image I, with $t_0=0$ and $t_{K+1}=L$. $h(i)$ is the frequency, and P_j is the probability of the *ith* gray level. N is all the pixels in the evaluated I.

3. The Peak Signal-to-Noise Ratio (PSNR) metric, dependent on the picture's pixel intensity and indicates the input images' fitness, is used to quantify the difference between the source images and features extracted. Following is a definition of the PSNR.

$$PSNR = 20\log_{10}\left(\frac{255}{RMSE}\right), (in\ dB) \tag{14}$$

RMSE refers to the root-mean-squared error detected as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{M}\sum_{j=1}^{Q}(I(i,j) - Seg(i,j))^2}{M * Q}} \tag{15}$$

I and Seg are the original image and segmented images with a size of $(M\times Q)$, respectively. The best $PSNR$ is the highest value.

4. The Structural Similarity Index (SSIM) is utilized to assess the similarity between (I) and (Seg) image, which is presented as follows.

$$SSMI(I, Seg) = \frac{(2\mu_I\mu_{Seg} + C_1)(2\sigma_{I.Seg} + C_2)}{(\mu_I^2 + \mu_{Seg}^2 + C_1)(\sigma_I^1 + \sigma_{Seg}^2 + C_2)} \tag{16}$$

μI and μSeg refer to the mean intensity of (I) and (Seg); while σI and σSeg refer to the standard deviation of (I) and (Seg), respectively; σI, Seg is a variance of (I) and (Seg). $c1$ and $c2$ are constants, that $c_1=6.5025$ and $c_2=58.52252$. The best $SSIM$ value is the highest.

5. Friedman ranking test: The Friedman test is the one-way ANOVA with repeated measurements' non-parametric alternative. When the dependent variable being measured is ordinal, it is utilized to examine changes across groups. Additionally, it may be applied to continuous data that deviates from the presumptions required to do a one-way ANOVA with repeated measurements.

3.3 Parameters setting

The parameter settings for each method utilized in the ensuing fairness are displayed in Table 1. These parameters are set in the source of each algorithm.

Table 1 The parameters of all algorithms and their values

Algorithm	Parameters setting
SSA	c_2 and c_3 are random value [1,0]
MPA	$FAD_S = 0.2$, $P = 0.5$
PSO	$wMax = 0.9$, $wMin = 0.2$, $c_1 = 2$, $c_2 = 2$
WOA	a [2,0], b = 1, t [-1,1]
AOA	$\mu = 0.499$, $\alpha = 5$
AO	$alpha = 0.1$, $delta = 0.1$;
AOAa	$Alpha = 5$, $Mu = 0.499$

3.4 Benchmark images

In this research, all algorithms are tested on the eight images: four are COVID-19 case images, and four are NORMAL case images. These images are from a dataset published on the link access on 9 Apr 2021: https://www.kaggle.com/tawsifurrahman/covid19-radiography-database. These images are named NORMAL1, NORMAL2, NORMAL3, NORMAL4, and COVID1, COVID2, COVID3, COVID4 as shown in Fig. 3 with size (299×299) and extension PNG for all. Our research will examine the algorithms on several common gray-scale and color images from the Berkeley University database. Also, we examine the algorithm on the COVID-19 images of the dataset published on the link access on 9 Apr 2021.

3.5 The results and discussions

The algorithm's performance (AOAa) will be discussed in this part, along with comparisons to other, more well-known methods for the segmentation issue (fitness value, PSNR, SSIM, time). Eight photos are subjected to five threshold applications (2, 3, 4, 5, and 6). The outcomes are compared using six techniques (i.e., AO, MPA, SSA, WOA, PSO, and AOA). For the comparable approaches employing the PSNR and SSIM Mean values for each picture, the results regarding PSNR, SSIM, CPU time, Fitness values, and the Friedman ranking test are provided. These are typical performance assessment metrics for picture segmentation, particularly for approaches using many levels of thresholds. The comparison results are given in Tables 2, 3, 4, 5, 6, 7 and 8. The discussion depends on each

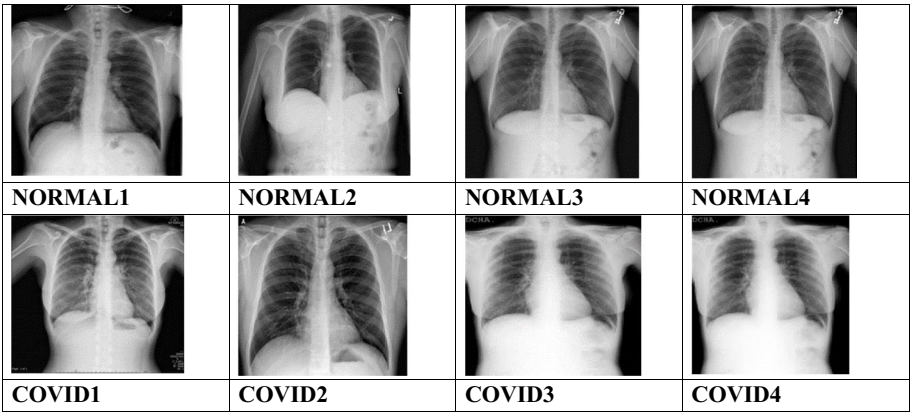


Fig. 3 Source images of (COVID and NORMAL cases)

Table 2 Comparison of *PSNR* for COVID, NORMAL cases images

	K		AO	WOA	SSA	AOA	AOAa	MPA	PSO
COVID-1	2	MEAN	**15.2623**	12.77967	12.16236	14.72392	14.25551	13.90585	13.77227
		STD	0.457807	2.879741	0.812893	1.631453	0.749422	1.673632	2.029729
	3	MEAN	**18.09861**	16.06347	15.24879	14.23023	17.07609	14.00016	15.62706
		STD	0.495808	0.921241	1.989558	2.637373	1.363071	1.214327	0.895679
	4	MEAN	16.36127	**17.91047**	17.08453	16.0158	15.70021	16.52431	15.27945
		STD	0.46831	0.722467	1.623999	1.901651	1.678525	1.567743	2.587032
	5	MEAN	18.57704	18.00793	17.93063	17.44464	18.91183	18.24343	**19.21467**
		STD	1.204432	0.038961	0.329087	2.838713	1.181614	2.173416	1.481966
	6	MEAN	18.28938	**20.37572**	18.86296	20.26783	19.66328	19.64373	18.07998
		STD	1.275444	0.94462	2.296606	1.238636	0.666739	1.033956	0.754216
COVID-2	2	MEAN	**14.42125**	12.92168	13.16038	12.29714	13.6162	12.70389	13.50345
		STD	0.504	0.737138	1.476946	1.038113	0.883273	2.473826	0.806705
	3	MEAN	14.7861	14.09509	14.4383	15.20345	**15.51641**	14.6161	13.71934
		STD	2.693257	0.49987	2.55045	1.901503	0.659547	1.404978	2.027998
	4	MEAN	16.27777	16.04338	15.08413	16.6797	17.0365	**17.45457**	15.51276
		STD	1.509429	0.595869	1.567154	1.039551	0.189918	0.664825	1.899308
	5	MEAN	17.33639	18.03994	17.00822	**18.3406**	15.85981	18.22424	16.86964
		STD	1.992179	1.295728	2.463494	1.553394	2.028002	1.23363	1.092169
	6	MEAN	14.81803	19.10307	18.04605	18.58039	19.03539	**19.62357**	18.4588
		STD	1.469314	0.405443	1.884078	1.001871	1.101765	1.304284	1.3484
COVID-3	2	MEAN	11.16933	10.83157	13.22884	**14.44292**	11.1988	13.65758	12.98808
		STD	2.423073	1.894036	1.82483	0.399431	1.966076	2.046013	2.060039
	3	MEAN	14.51449	14.58047	**14.86989**	9.564635	14.43983	13.0029	14.58103
		STD	2.245033	0.743261	2.143962	1.48891	3.068949	1.104968	1.434026
	4	MEAN	12.26771	13.33694	14.83319	**16.88782**	15.17265	14.39665	16.51477
		STD	5.378315	1.03325	1.313469	0.49478	0.990431	3.322576	2.463693
	5	MEAN	16.92287	16.55397	15.69802	14.38048	17.9417	**18.89579**	16.25714
		STD	2.17281	0.595055	4.706241	3.999957	2.841534	0.458308	1.322994
	6	MEAN	**19.66963**	17.27966	17.61387	15.1862	17.93219	17.99446	16.34047
		STD	0.207749	2.820685	3.020165	2.373307	3.267133	0.647447	3.535097
COVID-4	2	MEAN	**12.72154**	12.06296	10.6879	10.82645	11.42273	11.96649	12.63232
		STD	1.533756	1.435759	0.776152	1.176126	2.254029	1.364063	0.222884
	3	MEAN	11.18061	**14.82788**	13.48392	12.57109	12.76725	14.69765	11.02622
		STD	3.311761	2.647345	1.329151	1.186902	0.769867	0.609141	5.002406
	4	MEAN	12.69527	15.1221	**16.43382**	16.22043	14.38071	14.76656	15.41012
		STD	0.630882	2.007462	1.224077	1.773017	3.509376	2.007426	2.203012
	5	MEAN	16.02119	15.84609	15.67564	16.83676	16.18513	**18.84953**	15.50862
		STD	2.993478	3.097997	1.622925	0.363412	3.044245	0.595802	1.777893
	6	MEAN	16.93975	16.7439	**19.116**	16.7057	18.14345	17.47733	15.65147
		STD	2.762957	3.053883	1.670625	4.475576	2.401002	2.817783	0.303474

Table 2 (continued)

	K		AO	WOA	SSA	AOA	AOAa	MPA	PSO
NORMAL-1	2	MEAN	12.21721	13.51486	13.18181	13.80757	**13.96536**	13.89225	14.41324
		STD	1.483247	0.480803	0.997374	1.030454	0.98858	0.647945	0.473116
	3	MEAN	13.69298	**16.29859**	15.26288	14.69812	15.38509	12.06488	16.14744
		STD	1.834143	0.983107	1.690518	0.467485	0.820318	3.587307	0.388066
	4	MEAN	15.48356	**16.72871**	16.36059	16.17531	16.61682	14.98963	14.6245
		STD	1.056256	1.03766	0.363913	2.915623	1.407711	1.249653	0.939832
	5	MEAN	**19.81565**	18.24369	18.10848	15.7174	14.7646	17.78201	13.79051
		STD	0.367772	1.694373	1.554052	1.511312	1.843742	1.09404	5.123731
	6	MEAN	17.3734	**19.35193**	16.74214	16.11937	15.58373	13.92041	15.15231
		STD	0.589653	0.210532	0.40632	2.988868	4.565528	4.00794	4.533339
NORMAL-2	2	MEAN	**12.97159**	12.37534	12.29419	12.93748	12.94565	14.09742	12.26681
		STD	1.555674	1.580217	1.863271	1.209149	2.355322	0.544052	1.057802
	3	MEAN	14.51758	13.80351	14.76289	14.82943	14.63665	**16.26853**	14.12476
		STD	1.431234	0.98482	0.77459	0.906443	1.237538	1.420477	0.817917
	4	MEAN	15.76711	**17.82441**	16.27103	16.10153	15.80668	15.87	16.96044
		STD	1.485758	1.518574	1.136031	1.077229	1.513132	0.8759	2.444868
	5	MEAN	17.99759	17.82201	17.17272	17.19482	**18.14099**	17.75371	17.99913
		STD	1.087937	0.309315	0.648124	1.095208	1.141993	2.208951	1.415045
	6	MEAN	19.25699	**19.78352**	18.4251	17.67516	18.6501	18.38545	18.41697
		STD	2.010477	0.920201	0.234704	1.428555	1.55162	1.851369	1.667974
NORMAL-3	2	MEAN	13.64529	11.77763	10.62252	12.84539	13.2891	12.58018	**14.07368**
		STD	2.07489	0.797641	2.657105	2.158322	1.266557	1.754184	0.594326
	3	MEAN	14.8877	14.20757	14.96163	15.06826	**16.33906**	13.96985	13.96591
		STD	1.755432	2.253108	0.460852	1.825546	0.794285	0.774239	1.230897
	4	MEAN	17.03732	**17.47441**	15.88255	16.24592	14.75569	16.30044	16.63864
		STD	1.092134	1.320735	0.749032	1.485013	3.072776	2.927526	1.688564
	5	MEAN	16.73914	17.83166	15.7339	17.41436	17.40532	16.65377	**18.44109**
		STD	1.736396	1.361916	1.112252	0.766642	1.107332	0.805777	1.190947
	6	MEAN	18.62769	18.66607	17.83616	18.31547	18.76981	19.02392	**19.28842**
		STD	0.417386	1.188573	1.365392	2.666428	0.550711	1.869489	1.07549
NORMAL-4	2	MEAN	14.42355	12.84311	**14.5452**	14.14464	13.64685	12.12876	13.5101
		STD	0.738135	0.794227	0.287968	0.878197	0.672658	2.750113	0.46632
	3	MEAN	**15.42658**	12.89072	15.01992	14.87762	14.64615	15.18174	14.93892
		STD	0.860717	0.887621	2.062571	1.496616	1.24102	0.800181	1.654095
	4	MEAN	16.04607	16.28479	**17.8475**	15.63419	17.50957	15.86906	15.16286
		STD	2.543886	0.364581	1.267257	1.903894	1.242821	1.999883	2.028429
	5	MEAN	17.05958	17.4702	**19.02356**	18.6923	16.8066	18.86523	17.3854
		STD	0.848082	0.970442	1.067506	1.016923	1.323133	1.032608	2.1991
	6	MEAN	16.56732	18.3278	17.89078	**19.73605**	17.53022	19.47673	18.60612
		STD	0.761405	1.595749	0.4339	1.435588	0.695685	0.030226	1.08701

Bold font refer to the best obtained result

Table 3 Comparison of *SSIM* for COVID, NORMAL cases images

	K		AO	WOA	SSA	AOA	AOAa	MPA	PSO
COVID-1	2	MEAN	0.429675	0.40297	0.376196	0.458719	**0.46946**	0.444653	0.460506
		STD	0.034065	0.059947	0.030316	0.011146	0.051275	0.023983	0.024026
	3	MEAN	0.498063	0.443777	**0.532249**	0.411628	0.472553	0.395412	0.467073
		STD	0.005337	0.040703	0.070879	0.109241	0.041128	0.04514	0.015658
	4	MEAN	0.463947	0.507542	**0.536568**	0.462778	0.44335	0.475494	0.533161
		STD	0.02138	0.014043	0.032464	0.042108	0.064226	0.021184	0.05675
	5	MEAN	0.52982	0.52172	0.534607	0.504675	**0.562163**	0.561966	0.555674
		STD	0.032872	0.044831	0.025498	0.079676	0.040096	0.048949	0.046626
	6	MEAN	0.542774	0.588358	0.557355	**0.624412**	0.5685	0.576641	0.591855
		STD	0.054907	0.0314	0.08599	0.09104	0.042096	0.052018	0.019438
COVID-2	2	MEAN	**0.509214**	0.422248	0.425588	0.370877	0.431636	0.360834	0.439632
		STD	0.017173	0.067832	0.057058	0.097219	0.017339	0.134443	0.073197
	3	MEAN	0.481667	0.414796	0.515935	0.456117	**0.584209**	0.460597	0.418115
		STD	0.085589	0.01758	0.079211	0.085157	0.049065	0.089635	0.056822
	4	MEAN	0.529687	0.541124	0.493851	0.520688	**0.553416**	0.546894	0.480378
		STD	0.124984	0.037822	0.10514	0.014728	0.057987	0.017256	0.063804
	5	MEAN	0.583153	0.60667	0.550325	**0.626196**	0.540152	0.586562	0.535726
		STD	0.082105	0.023342	0.080763	0.007564	0.020859	0.004549	0.052696
	6	MEAN	0.576617	0.627816	0.562125	0.624522	**0.646665**	0.646609	0.597459
		STD	0.006343	0.026981	0.088716	0.023419	0.073929	0.032317	0.066562
COVID-3	2	MEAN	0.373957	0.339003	0.495862	0.512223	0.363736	**0.56946**	0.450022
		STD	0.142917	0.091282	0.080725	0.016875	0.103842	0.062944	0.080966
	3	MEAN	0.530742	0.551768	**0.610587**	0.280486	0.526578	0.455782	0.558857
		STD	0.125196	0.022165	0.029914	0.08827	0.124872	0.027358	0.124729
	4	MEAN	0.437292	0.465522	0.523525	0.601394	**0.603229**	0.531071	0.585213
		STD	0.324268	0.031428	0.049376	0.029155	0.07619	0.156657	0.104579
	5	MEAN	0.636514	0.589412	0.519043	0.563353	**0.672892**	0.672963	0.601557
		STD	0.062881	0.010432	0.17201	0.081167	0.091527	0.027416	0.023636
	6	MEAN	0.688058	0.602256	0.631671	0.552423	**0.695619**	0.635441	0.565898
		STD	0.022429	0.080002	0.073866	0.096402	0.036592	0.03781	0.100983
COVID-4	2	MEAN	0.581631	0.60622	0.56273	0.56627	0.561971	0.581557	**0.609844**
		STD	0.077199	0.059084	0.032745	0.097838	0.054661	0.049568	0.030158
	3	MEAN	0.583518	0.675019	0.64054	0.603578	**0.695809**	0.591213	0.537971
		STD	0.043304	0.031402	0.116596	0.100248	0.017547	0.04803	0.152867
	4	MEAN	0.595079	0.63306	0.673079	0.648143	**0.673139**	0.593861	0.640991
		STD	0.049769	0.03235	0.025289	0.066341	0.053269	0.032985	0.069876
	5	MEAN	0.685021	0.694551	0.72816	0.696754	**0.705425**	0.703296	0.65466
		STD	0.083246	0.054632	0.044351	0.067018	0.023137	0.029351	0.044562
	6	MEAN	0.682693	0.661963	0.697253	0.653201	**0.691739**	0.679539	0.691851
		STD	0.027564	0.077946	0.024858	0.094735	0.055611	0.090696	0.039219

Table 3 (continued)

	K		AO	WOA	SSA	AOA	AOAa	MPA	PSO
NORMAL-1	2	MEAN	0.377072	0.433945	0.425974	0.448437	0.44067	0.439367	**0.45975**
		STD	0.146139	0.084738	0.05663	0.082113	0.077503	0.072743	0.040375
	3	MEAN	0.487931	0.495169	0.466183	0.461924	**0.508414**	0.350841	0.485511
		STD	0.052161	0.090261	0.09588	0.028882	0.059348	0.186978	0.069474
	4	MEAN	0.449308	0.497662	0.500883	0.494075	**0.503744**	0.43292	0.405417
		STD	0.069547	0.053722	0.040597	0.13073	0.045387	0.074601	0.032333
	5	MEAN	**0.622331**	0.586756	0.527278	0.454854	0.455494	0.575692	0.406074
		STD	0.014378	0.043584	0.078913	0.04693	0.0591	0.068585	0.225934
	6	MEAN	0.521997	0.582859	0.549776	0.506106	**0.584034**	0.426425	0.46821
		STD	0.060609	0.03107	0.088384	0.125498	0.124547	0.160948	0.194706
NORMAL-2	2	MEAN	0.440137	0.443875	**0.448085**	0.380422	0.433494	0.425825	0.361047
		STD	0.108188	0.064598	0.031175	0.042702	0.03671	0.041938	0.024426
	3	MEAN	0.478387	0.499265	0.441933	0.506195	**0.531474**	0.518201	0.507956
		STD	0.071283	0.042977	0.040663	0.03042	0.048546	0.041512	0.050541
	4	MEAN	0.457856	0.585825	0.600818	0.497838	**0.594393**	0.574744	0.507555
		STD	0.029951	0.044236	0.048041	0.062741	0.028649	0.017915	0.06425
	5	MEAN	0.637749	0.590795	0.571383	0.554001	**0.603174**	0.58265	0.578656
		STD	0.010376	0.081043	0.081152	0.04878	0.034464	0.044166	0.042985
	6	MEAN	0.620989	0.610829	**0.657615**	0.554396	0.594018	0.579489	0.577239
		STD	0.080172	0.021174	0.094464	0.080524	0.089916	0.042093	0.075686
NORMAL-3	2	MEAN	0.579632	0.525593	0.470218	0.496973	**0.656299**	0.528933	0.55688
		STD	0.067865	0.084187	0.209488	0.121069	0.076646	0.144974	0.043189
	3	MEAN	0.659135	0.637236	**0.659628**	0.671445	0.650855	0.578848	0.623327
		STD	0.043666	0.017713	0.0457	0.090691	0.036711	0.125382	0.065171
	4	MEAN	0.727387	0.723672	0.70059	0.724118	**0.728359**	0.660755	0.688485
		STD	0.051295	0.035443	0.012628	0.049346	0.083821	0.089128	0.021552
	5	MEAN	0.671559	0.666467	0.750507	0.702174	**0.790825**	0.701738	0.693761
		STD	0.088484	0.039542	0.046332	0.044752	0.094626	0.059367	0.068361
	6	MEAN	0.746381	0.745939	0.749069	0.726662	0.749735	0.750085	0.745973
		STD	0.022251	0.073073	0.061598	0.055838	0.053908	0.024152	0.016091
NORMAL-4	2	MEAN	0.451074	0.357137	0.463249	0.414633	0.452664	0.355667	**0.47036**
		STD	0.041746	0.04894	0.047747	0.017746	0.016278	0.137518	0.054919
	3	MEAN	0.459787	0.401018	**0.538735**	0.476954	0.473534	0.484241	0.416771
		STD	0.04384	0.068847	0.063085	0.064052	0.085712	0.071878	0.038135
	4	MEAN	0.530538	0.510717	0.56348	0.502788	**0.566659**	0.53399	0.477989
		STD	0.03545	0.024968	0.01622	0.050147	0.027072	0.146038	0.124327
	5	MEAN	0.568866	0.509539	0.604917	0.579062	**0.630124**	0.586291	0.613234
		STD	0.079211	0.053329	0.047724	0.033739	0.058409	0.026048	0.034766
	6	MEAN	0.523548	0.612574	0.61613	0.640129	**0.692208**	0.612804	0.689183
		STD	0.083762	0.079615	0.084044	0.061391	0.02547	0.01572	0.029119

Bold font refer to the best obtained result

Table 4 Average CPU time (seconds) for COVID-19 cases

	K	AO	WOA	SSA	AOA	MPA	PSO	AOAa
COVID-1	2	0.239373	0.040097	0.03062	0.036374	76.88261	11.06229	6.020129
	3	0.230218	0.044229	0.035656	0.043114	106.0657	15.30059	7.626052
	4	0.238687	0.122027	0.064104	0.077583	175.7069	27.53357	16.09753
	5	0.247421	0.113222	0.082588	0.091326	140.0356	26.58181	13.93868
COVID-2	2	0.248864	2.307623	0.087376	0.079854	22.60579	3.105797	9.946913
	3	0.249791	0.061295	0.047198	0.055619	101.6172	17.13897	7.643711
	4	0.263115	0.061844	0.045043	0.054286	129.8905	23.27711	10.81031
	5	0.266202	0.082559	0.077439	0.081497	28.96452	6.003071	20.01818
COVID-3	2	0.036374	0.055243	0.042374	0.050997	79.58729	10.50436	6.968161
	3	0.043114	0.043226	0.038953	0.046039	105.7169	17.60654	9.140672
	4	0.077583	0.05905	0.041318	0.049201	131.4959	23.11287	10.81287
	5	0.091326	0.058529	0.046055	0.058149	21.57042	32.8968	13.90128
COVID-4	2	0.036374	0.311161	0.245086	0.248864	2.90873	0.723672	2.736707
	3	0.043114	0.291892	0.263487	0.249791	4.504445	1.12395	4.09117
	4	0.077583	0.309512	0.303814	0.263115	10.71171	1.121382	4.379006
	5	0.091326	0.31204	0.259287	0.266202	5.274792	1.188409	5.35945

Table 5 Average CPU time (seconds) for NORMAL cases

	K	AO	WOA	SSA	AOA	MPA	PSO	AOAa
NORMAL-1	2	0.044229	0.251967	0.245896	0.239373	5.940693	1.146158	1.91396
	3	0.122027	0.232458	0.228359	0.230218	7.630269	1.79152	2.651281
	4	0.113222	0.241309	0.231332	0.238687	5.203141	2.080605	3.733159
	5	2.307623	0.25015	0.245332	0.247421	11.48492	2.594214	4.058729
NORMAL-2	2	0.061295	0.311161	0.245086	0.248864	2.90873	0.723672	2.736707
	3	0.061844	0.291892	0.263487	0.249791	4.504445	1.12395	4.09117
	4	0.082559	0.309512	0.303814	0.263115	10.71171	1.121382	4.379006
	5	0.055243	0.31204	0.259287	0.266202	5.274792	1.188409	5.35945
NORMAL-3	2	0.043226	0.040097	0.03062	0.036374	76.88261	11.06229	6.020129
	3	0.05905	0.044229	0.035656	0.043114	106.0657	15.30059	7.626052
	4	0.058529	0.122027	0.064104	0.077583	175.7069	27.53357	16.09753
	5	0.311161	0.113222	0.082588	0.091326	140.0356	26.58181	13.93868
NORMAL-4	2	0.291892	0.040097	0.03062	0.036374	76.88261	11.06229	6.020129
	3	0.309512	0.044229	0.035656	0.043114	106.0657	15.30059	7.626052
	4	0.31204	0.122027	0.064104	0.077583	175.7069	27.53357	16.09753
	5	0.040097	0.113222	0.082588	0.091326	140.0356	26.58181	13.93868

Table 6 The best fitness values were obtained from the algorithms on COVID-19 cases

	K	AO	WOA	SSA	AOA	MPA	PSO	AOAa
COVID-1	2	1775.373	2430.33	2434.255	2382.197	2467.48	2428.952	2469.495
	3	1884.819	2638.255	2645.489	2495.657	3249.076	2602.186	3254.056
	4	1938.726	2726.173	2727.67	2572.067	3565.544	2746.819	3636.873
	5	2418.37	2746.354	2731.698	2703.694	3896.637	2736.831	4318.764
COVID-2	2	2913.571	2326.881	2328.222	2251.314	2348.447	2349.319	2351.324
	3	3094.46	2417.967	2417.967	2315.081	3051.351	2432.396	3244.085
	4	3156.338	2472.045	2483.902	2254.491	3146.329	2480.248	3584.116
	5	3209.725	2527.195	2483.655	2433.25	3722.265	2485.464	4033.038
COVID-3	2	1769.214	5203.927	5203.927	5194.795	233.783	5227.777	5235.423
	3	1881.986	5381.305	5374.99	5334.478	6916.685	5353.02	7354.29
	4	1907.119	5451.083	5447.217	5418.614	7628.84	5427.893	9026.392
	5	1950.474	5519.578	5509.297	5506.294	8698.647	5506.08	9259.896
COVID-4	2	2430.33	1009.721	1009.721	727.3778	1033.654	1023.687	1034.68
	3	2638.255	1069.997	1036.101	915.2615	1231.034	1038.308	1384.995
	4	2726.173	1087.979	1008.752	1044.375	1581.942	1057.043	1594.859
	5	2746.354	1103.255	1106.493	1079.437	1659.359	1150.208	1816.25

Table 7 The best fitness values obtained from the algorithms on NORMAL cases

	K	AO	WOA	SSA	AOA	MPA	PSO	MPASSA
NORMAL-1	2	2382.197	2430.33	2434.255	2382.197	2467.48	2428.952	2469.495
	3	2495.657	2638.255	2645.489	2495.657	3249.076	2602.186	3254.056
	4	2572.067	2726.173	2727.67	2572.067	3565.544	2746.819	3636.873
	5	2703.694	2746.354	2731.698	2703.694	3896.637	2736.831	4318.764
NORMAL-2	2	2251.314	2326.881	2328.222	2251.314	2348.447	2349.319	2351.324
	3	2315.081	2417.967	2417.967	2315.081	3051.351	2432.396	3244.085
	4	2254.491	2472.045	2483.902	2254.491	3146.329	2480.248	3584.116
	5	2433.25	2527.195	2483.655	2433.25	3722.265	2485.464	4033.038
NORMAL-3	2	5194.795	5203.927	5203.927	5194.795	5233.783	5227.777	5235.423
	3	5334.478	5381.305	5374.99	5334.478	6916.685	5353.02	7354.29
	4	5418.614	5451.083	5447.217	5418.614	7628.84	5427.893	9026.392
	5	5506.294	5519.578	5509.297	5506.294	8698.647	5506.08	9259.896
NORMAL-4	2	727.3778	1758.845	1755.602	1705.031	1778.65	1769.214	1778.743
	3	915.2615	1883.002	1847.382	1811.518	2362.56	1881.986	2405.533
	4	1044.375	1923.829	1918.6	1911.359	2538.1	1907.119	2898.64
	5	1079.437	1953.451	1955.783	1895.952	2859.001	1950.474	3228.724

Table 8 The results of the Friedman ranking test for the comparative methods using the PSNR values for the COVID-19 experiment

PSNR_Rank	K	AO	WOA	SSA	AOA	AOAa	MPA	PSO
COVID1	2	1	6	7	2	3	4	5
	3	1	3	5	6	2	7	4
	4	4	1	2	5	6	3	7
	5	3	5	6	7	2	4	1
	6	6	1	5	4	2	3	7
	SUM	15	16	25	24	15	21	24
	Mean rank	3	3.2	5	4.8	3	42	4.8
	Final Rank	1	2	5	4	**1**	3	4
COVID2	2	1	5	4	7	2	6	3
	3	3	6	5	2	1	4	7
	4	4	5	7	3	2	1	6
	5	4	3	7	1	5	2	6
	6	7	3	6	4	2	1	5
	SUM	19	22	29	17	12	14	27
	Mean rank	3.8	4.4	5.8	3.4	2.4	2.8	5.4
	Final Rank	4	5	7	**3**	**1**	2	6
COVID3	2	6	7	3	1	5	2	4
	3	4	3	1	7	5	6	2
	4	7	6	5	1	2	4	3
	5	4	5	6	7	1	3	2
	6	1	5	4	7	2	3	6
	SUM	22	26	19	23	15	18	17
	Mean rank	4.4	5.2	3.8	4.6	3	3.6	3.4
	Final Rank	5	7	4	6	**1**	3	2
COVID4	2	1	3	7	6	4	5	2
	3	6	1	3	5	4	2	7
	4	7	6	1	2	4	5	3
	5	4	5	6	2	3	1	7
	6	3	5	1	6	2	4	7
	SUM	21	20	18	21	17	17	26
	Mean rank	4.2	4	3.6	4.2	3.4	3.4	5.2
	Final Rank	4	3	2	4	**1**	**1**	5

Bold font refer to the best obtained result

measure given in this section as well. Figures 4 and 5 show the image's histogram for the algorithm (AOAa) at k equal to 2, 3, 4, 5, and 6.

In most situations, the AOAa algorithm produces the best results. This is so that each image may be treated as a different optimization issue. In some instances, the outcomes might vary as a result of chance. According to the No-Free Lunch Theorem NFL, it is also challenging to identify a single appropriate method for all optimization situations. Each image also has a different level histogram. Due to the multimodality of the histogram, segmenting pictures also becomes challenging. The segmented picture is built by calculating each class level value from the mean of the class members.

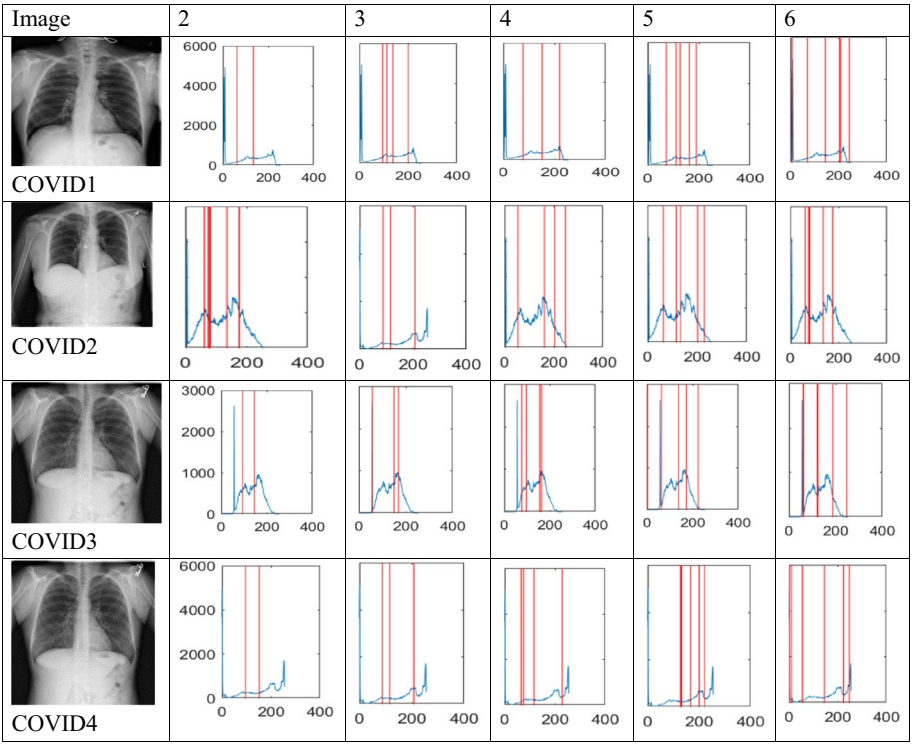


Fig. 4 Histogram for COVID Images experiments for proposed technique in threshold 2,3,4,5,6

Thus, the (AOAa) method produced better results for each picture regarding PSNR, SSIM, and fitness values at each level. This algorithm may not be able to segment the picture in various situations and levels. Due to these factors, we compared statistical data to find the most effective method for all photos. The histogram results for the proposed technique illustrate that the AOAa algorithm distributes thresholds along the histogram in each level of threshold, as shown in Figs. 4 and 5.

In terms of CPU time measurement: The temporal complexity of each algorithm was taken into account, and its total performance was evaluated in terms of exhaustion time. Tables 4 and 5 compare the AOAa method's time complexity to other methods utilizing COVID and Normal scenarios. It demonstrates that the suggested method takes an average amount of time compared to existing methods. Before the MPA and PSO algorithms, it came in fourth. Furthermore, after SSA, AOA, WOA, and AO. While WOA at all levels with all photos took the least time to execute, SSA was the least time-consuming at all levels.

Regarding PSNR measure According to the testing results of threshold two, as in, the AOAa got the second-best value after AO for COVID1 and COVID2 images. But it got the fifth and fourth levels for COVID3 and COVID4, respectively. And we got the best value for the NORMAL1 image, while it got the third level for NORMAL2, NORMAL3, and NORMAL4. The testing results of threshold 3, the AOAa got the second-best value after AO for COVID-19. In COVID2, the image got the best value. And we got the best value for the NORMAL3 image, while it got the third level for NORMAL1, NORMAL2, and

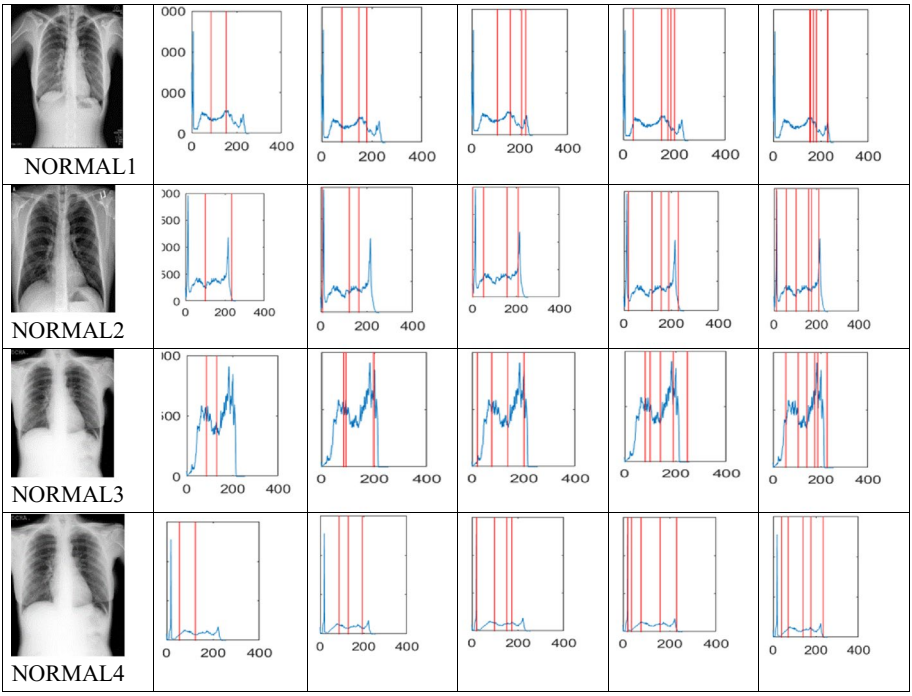


Fig. 5 Histogram for NORMAL Images experiments for proposed technique in threshold 2,3,4,5,6

NORMAL4. The testing results of the thresholds 5, 6, and 6, the AOAa ranges between the first and second levels and sometimes reaches the third level.

Regarding SSIM measure According to the testing results of threshold two, the AOAa got the best value for COVID-19 and the third level for COVID-2 images. But it got the fifth and sixth levels for COVID-4 and COVID-3, respectively. And we got the best value for the NORMAL-3 image, while it got the third and fourth level for NORMAL-1, NORMAL-4, and NORMAL-2, respectively. The testing results of threshold 3, the AOAa got the best value for COVID-19. COVID2 and COVID4 got the best value. At the same time, COVID3 got the third value. NORMAL1 and NORMAL2 got the best level, while they got the third level for NORMAL3 and NORMAL4. The testing results of the thresholds 4, 5, and 6 the AOAa at almost got the best level.

Table 9 Summary of Friedman ranking test for the comparative methods using the PSNR values for COVID-19 experiment

PSNR_Rank	AO	WOA	SSA	AOA	AOAa	MPA	PSO
COVID1	1	2	5	4	1	3	4
COVID2	4	5	7	3	1	2	6
COVID3	5	7	4	6	1	3	2
COVID4	4	3	2	4	1	1	5

In terms of fitness value measure The fitness function values of all algorithms were mentioned and analyzed over the variance of thresholds. The results of all the fitness values at level 2 of all algorithms are almost at the same level. Whereas in levels 3, 4, and 5, the percentage of the difference is very large between the AO and AOA, WOA, PSO, and SSA algorithms, while the ratio is close to the MPA algorithm. These results show the AOAa did not fix at the same level at a different threshold.

In terms of the Friedman ranking test The results for comparison of methods of the Friedman ranking test for the (AOAa, MPA, SSA, WOA, AOA, AO, and PSO) using the PSNR values are given in Tables 8, 9, 10 and 11 for PSNR and Tables 12, 13, 14 and 15 for SSIM.

Table 10 The results of the Friedman ranking test for the comparative methods using the PSNR values for the NORMAL experiment

PSNR_Rank	K	AO	WOA	SSA	AOA	AOAa	MPA	PSO
NORMAL1	2	6	7	5	4	1	3	2
	3	6	1	4	5	3	7	2
	4	5	1	3	4	2	6	7
	5	1	4	2	5	3	6	7
	6	2	1	3	4	5	7	6
	SUM	20	14	17	22	14	29	24
	Mean rank	4	2.8	3.4	4.4	2.8	5.8	4.8
	Final Rank	3	1	2	4	1	6	5
NORMAL2	2	1	6	7	4	3	2	5
	3	4	7	6	2	3	1	5
	4	7	1	2	3	5	6	4
	5	3	4	6	7	1	5	2
	6	2	1	5	7	3	6	4
	SUM	17	19	26	23	15	20	20
	Mean rank	3.4	3.8	5.2	4.6	3	4	4
	Final Rank	2	3	6	5	1	4	4
NORMAL3	2	2	6	7	4	3	5	1
	3	4	5	3	2	1	6	7
	4	2	1	7	5	4	6	3
	5	5	3	7	4	2	6	1
	6	5	4	7	6	3	2	1
	SUM	18	19	31	21	13	25	13
	Mean rank	3.6	3.8	6.2	4.2	2.6	5	2.6
	Final Rank	2	3	6	4	1	5	1
NORMAL4	2	2	6	1	3	4	7	5
	3	1	7	4	6	3	2	5
	4	4	3	1	6	2	5	7
	5	6	4	1	3	5	2	7
	6	7	3	6	1	4	2	5
	SUM	20	23	13	19	18	18	29
	Mean rank	4	4.6	2.6	3.8	3.6	3.6	5.8
	Final Rank	4	5	1	3	2	2	6

Table 11 Summary of Friedman ranking test for the comparative methods using the PSNR values for NORMAL experiment

PSNR_Rank	AO	WOA	SSA	AOA	AOAa	MPA	PSO
NORMAL1	3	1	2	4	1	6	5
NORMAL2	2	3	6	5	1	4	4
NORMAL3	2	3	6	4	1	5	1
NORMAL4	4	5	1	3	2	2	6

Table 12 The results of the Friedman ranking test for the comparative methods using the SSIM values for COVID experiment

SSIM_Rank	K	AO	WOA	SSA	AOA	AOAa	MPA	PSO
COVID1	2	5	6	7	3	1	4	2
	3	2	4	1	5	3	7	6
	4	6	3	1	7	4	5	2
	5	5	6	4	7	1	2	3
	6	6	3	7	1	5	4	2
	SUM	24	22	20	23	14	22	15
	Mean rank	4.8	4.4	4	4.6	2.8	4.4	3
	Final Rank	6	4	3	5	1	4	2
COVID2	2	1	5	4	6	3	7	2
	3	3	5	2	4	1	6	7
	4	5	2	6	4	1	3	7
	5	3	2	6	1	5	4	7
	6	5	4	6	3	1	2	7
	SUM	17	18	24	18	11	22	30
	Mean rank	3.4	3.6	4.8	3.6	2.2	4.4	6
	Final Rank	2	3	5	3	1	4	6
COVID3	2	5	7	3	2	6	1	4
	3	4	3	1	7	5	6	2
	4	7	6	5	2	1	4	3
	5	3	5	7	6	1	2	4
	6	2	4	4	7	1	3	6
	SUM	21	25	20	24	14	16	19
	Mean rank	4.2	5	4	4.8	2.8	3.2	3.8
	Final Rank	5	7	4	6	1	2	3
COVID4	2	3	2	7	6	5	4	1
	3	6	2	3	4	1	5	7
	4	6	5	2	3	1	7	4
	5	6	5	1	4	2	3	7
	6	4	6	2	7	1	5	3
	SUM	25	20	15	24	10	24	22
	Mean rank	5	4	3	4.8	2	4.8	4.4
	Final Rank	6	3	2	5	1	1	4

Table 13 The summary of the Friedman ranking test for the comparative methods using the SSIM values for COVID experiment

SSIM_Rank	AO	WOA	SSA	AOA	AOAa	MPA	PSO
COVID1	6	4	3	5	1	4	2
COVID2	2	3	5	3	1	4	6
COVID3	5	7	4	6	1	2	3
COVID4	6	3	2	5	1	1	4

Table 14 The results of the Friedman ranking test for the comparative methods using the SSIM values for NORMAL experiment

SSIM_Rank	K	AO	WOA	SSA	AOA	AOAa	MPA	PSO
NORMAL1	2	7	5	6	2	3	4	1
	3	3	2	5	6	1	7	4
	4	5	3	2	4	1	6	7
	5	1	2	4	6	5	3	7
	6	4	2	3	5	1	7	6
	SUM	20	14	20	23	11	27	25
	Mean rank	4	2.8	4	4.6	2.2	5.4	5
	Final Rank	3	2	3	4	**1**	6	5
NORMAL2	2	3	2	1	6	4	5	7
	3	6	5	7	4	1	2	3
	4	7	3	1	6	2	4	5
	5	1	3	6	7	2	4	5
	6	2	3	1	7	4	5	6
	SUM	19	16	16	30	13	20	26
	Mean rank	3.8	3.2	3.2	6	2.6	4	5.2
	Final Rank	3	2	2	**6**	**1**	4	5
NORMAL3	2	2	5	7	6	1	4	3
	3	3	5	2	1	4	7	6
	4	2	4	5	3	1	7	6
	5	6	7	2	3	1	4	5
	6	4	5	3	7	2	1	6
	SUM	17	26	19	20	9	23	26
	Mean rank	3.4	5.2	3.8	4	1.8	4.6	5.2
	Final Rank	2	6	3	4	**1**	5	6
NORMAL4	2	4	6	2	5	3	7	1
	3	5	7	1	3	4	2	6
	4	4	5	2	6	1	3	7
	5	6	7	3	5	1	4	2
	6	7	5	4	3	1	6	2
	SUM	26	30	12	22	10	22	18
	Mean rank	5.2	6	2.4	4.4	2	4.4	3.6
	Final Rank	5	6	2	4	**1**	**4**	3

Bold font refer to the best obtained result

Table 15 Summarizes the Friedman ranking test for the comparative methods using the SSIM values for the NORMAL experiment

SSIM_Rank	AO	WOA	SSA	AOA	AOAa	MPA	PSO
NORMAL1	3	2	3	4	1	6	5
NORMAL2	3	2	2	6	1	4	5
NORMAL3	2	6	3	4	1	5	6
NORMAL4	5	6	2	4	1	4	3

These tables show each result's rank, summation, mean ranking, and the final ranking. In these tables of PSNR value and SSIM tables, the AOAa got the first ranking in all tests, followed by different algorithms for each image, and the result is shown in Figs. 6, 7, 8 and 9.

Table 9 summarizes the proposed technique implementation and how it was ahead of all algorithms. Figure 6 illustrate the execution of all algorithm for each PSNR-COVID case.

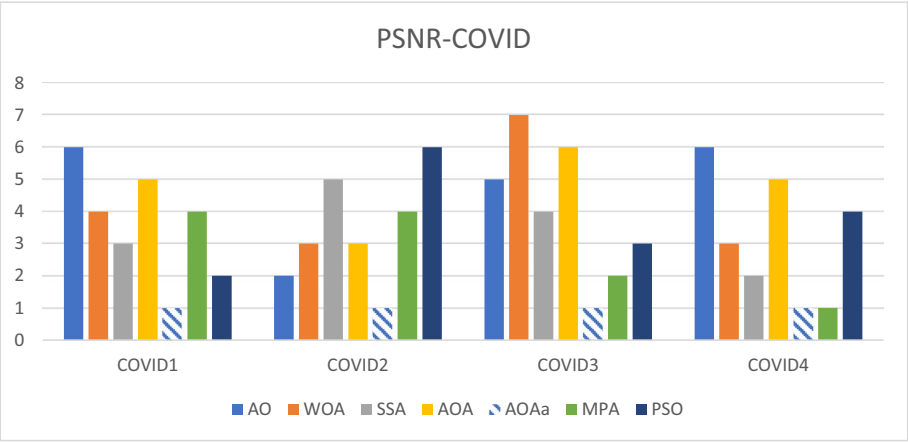


Fig. 6 Friedman ranking for PSNR-COVID

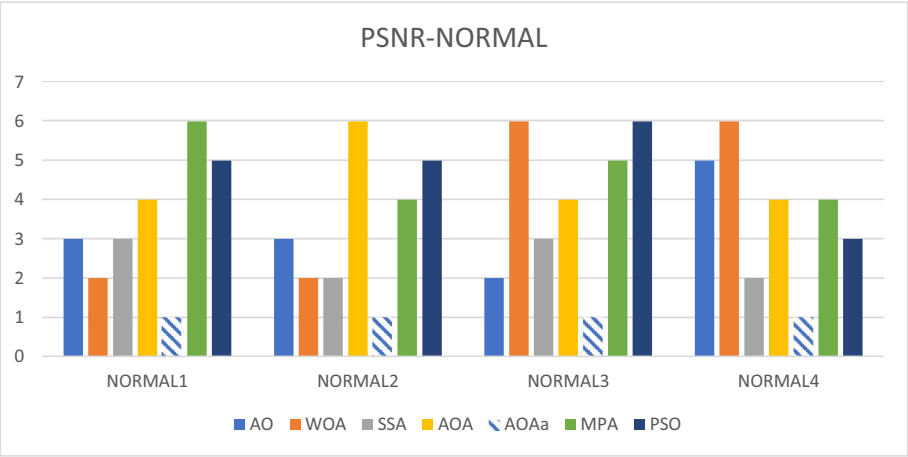


Fig. 7 Friedman ranking for PSNR-NORMAL

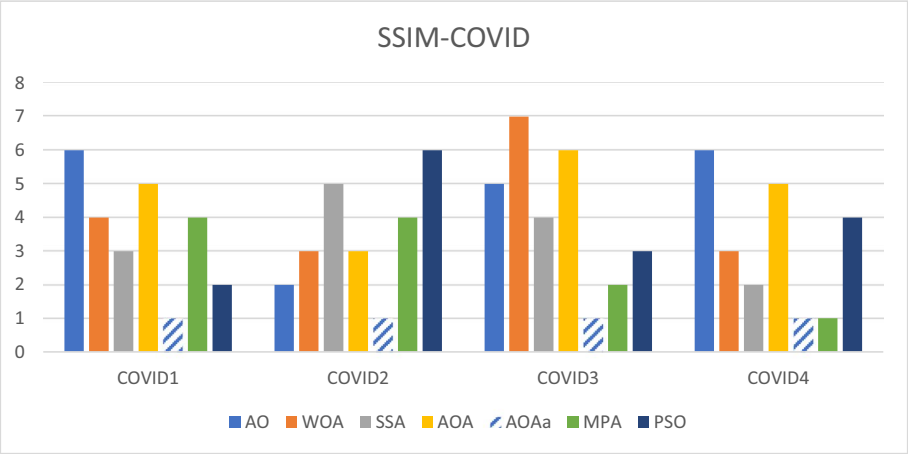


Fig. 8 Friedman ranking for SSIM-COVID

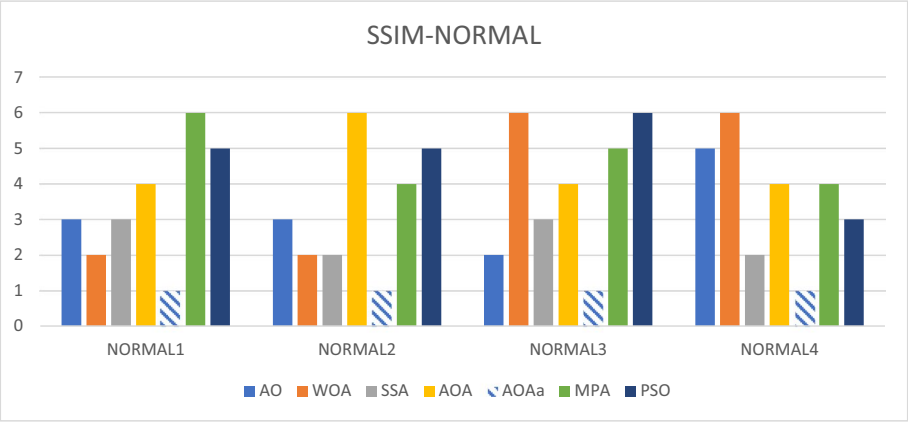


Fig. 9 Friedman ranking for SSIM-NORMAL

Table 11 summarizes the proposed technique implementation and how it was ahead of all algorithms. Figure 7 illustrate the execution of all algorithm for each PSNR-NORMAL case. Tables 12, 13, and 14 summarize the proposed technique implementation and how it was ahead of all algorithms. Figures 7 and 8 illustrate the execution of all algorithm for each SSIM-COVID case.

Table 15 summarizes the proposed technique implementation and how it was ahead of all algorithms. Figure 9 illustrate the execution of all algorithm for each SSIM-NORMAL case (Figs. 10, 11, 12 and 13).

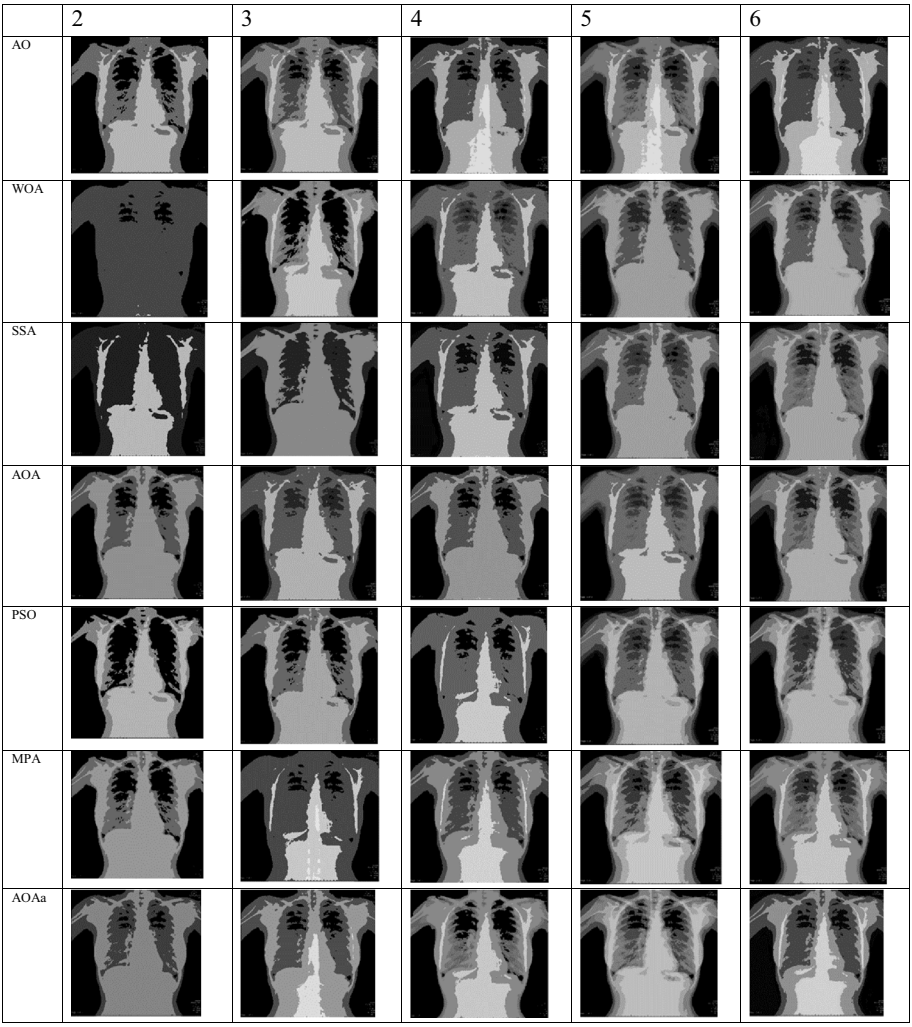


Fig. 10 COVID1 segmented images

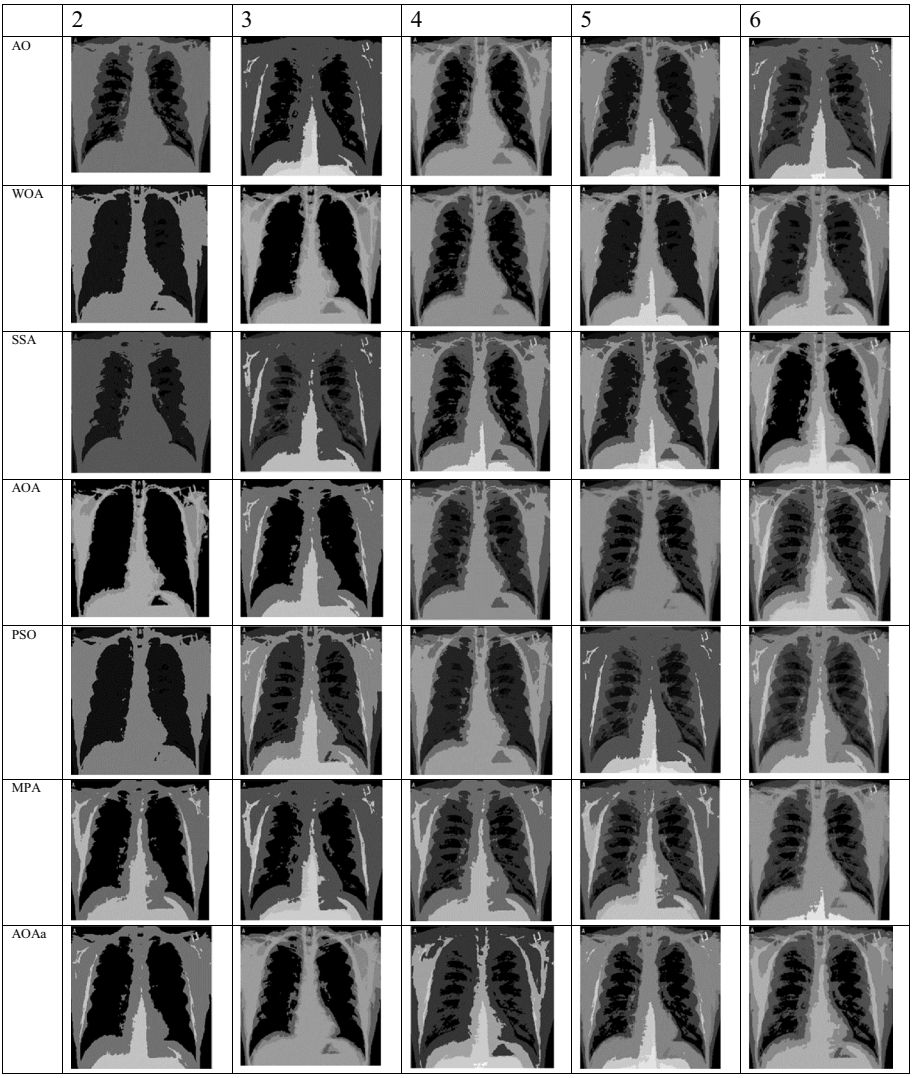


Fig. 11 COVID2 segmented images

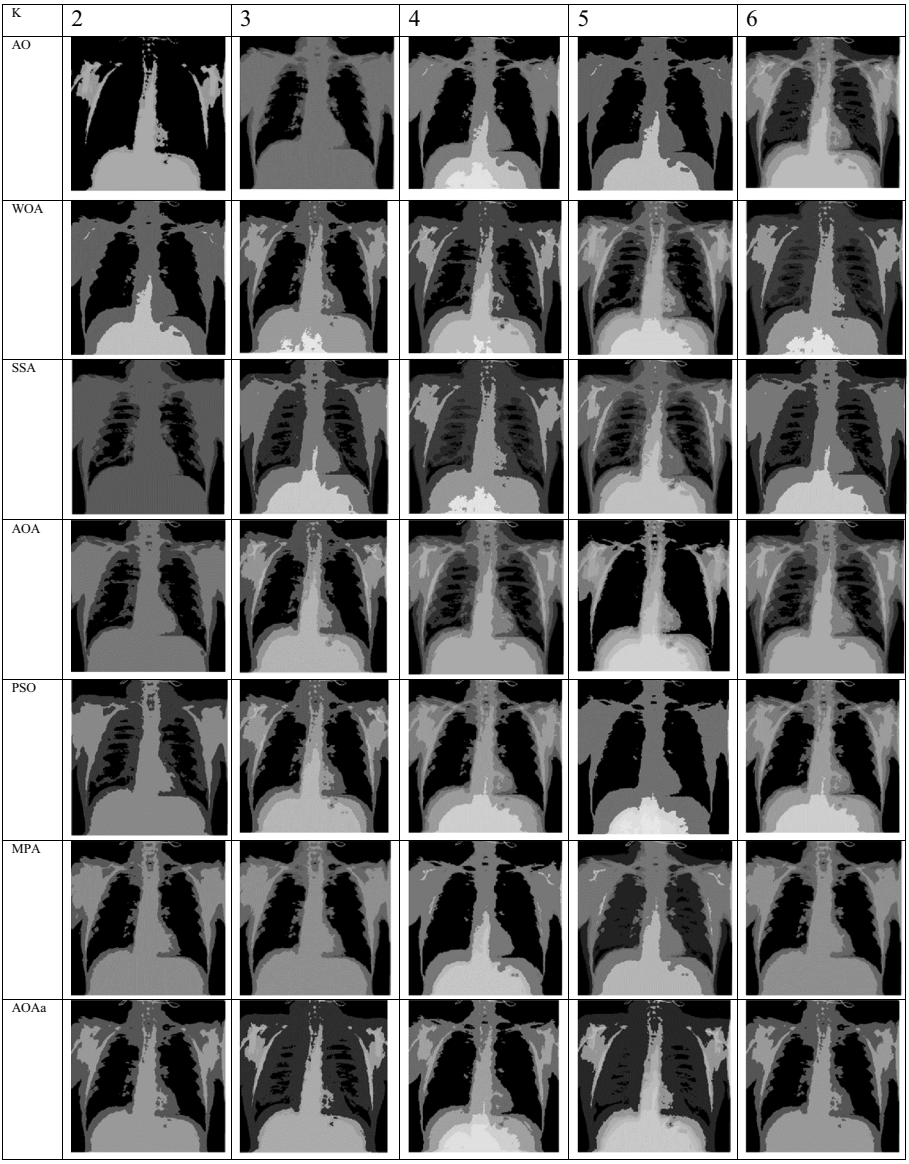


Fig. 12 NORMAL1 segmented images

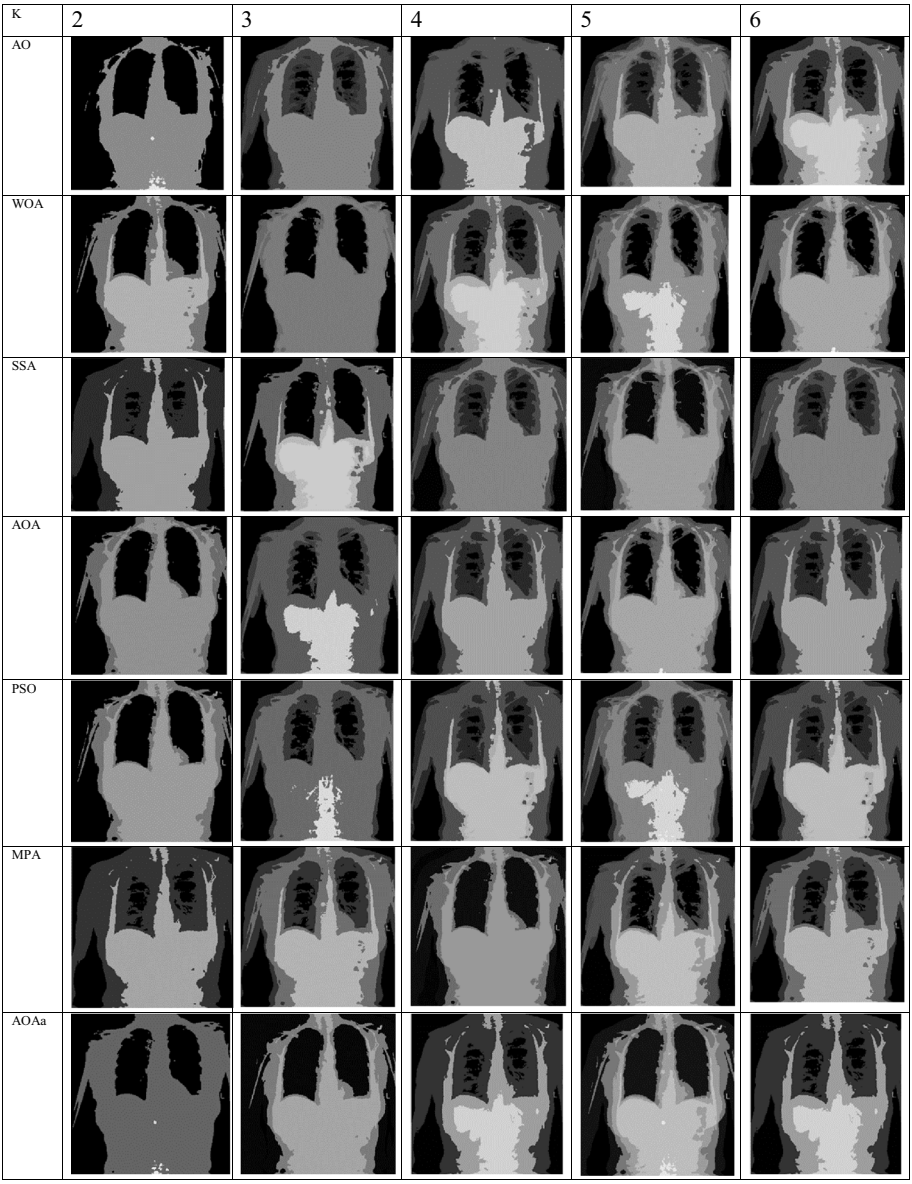


Fig. 13 NORMAL2 segmented images

4 Conclusion and future work

The technique of multi-level thresholding is one of the techniques for segmenting images. In many applications, it is referred to as the preprocessing stage. Identifying the optimal or best threshold for picture segmentation is a challenge in this study's scenario of multilayer thresholding. As a fitness function, Otsu's function has been used. In order to segment images using a multilayer threshold, an AOAa algorithm was employed. It makes

use of the characteristics of mathematical operations. This issue has been resolved using the suggested algorithm. Its goal is to establish the ideal threshold value that improves Otsu's function. The algorithm's outcomes have been contrasted with those of the MPA, SSA, WOA, AOA, AO, and PSA algorithms. The PSNR, SSSIM, and Friedman ranking tests have been used to determine the effectiveness of algorithms. All studies used eight benchmark photos for COVID-19 and NORMAL instances images. According to the study's findings, the AOAa approach performs well during segmentation. This superiority is due mainly to the fact that it avoids becoming trapped at a local optimum. All of this suggests that this method has a strong capacity to locate the image's ideal solution. Future uses of the suggested approach to address various image processing issues include (visualization, computer vision, computer-aided diagnostics, image classification, etc.). The finesses of the pictures are a very significant difficulty in image segmentation in all of these applications. Low resolutions, excessive noise, or poor contrast may bring on this issue. In order to increase the quality of that situation and to achieve the best segmentation results, preprocessing the photos is therefore essential.

Data availability Data is available from the authors upon reasonable request.

Declarations

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

Informed consent Informed consent was obtained from all individual participants included in the study.

Conflict of interest The authors declare that there is no conflict of interest regarding the publication of this paper.

References

1. Cheng H-D et al (2001) Color image segmentation: advances and prospects. Pattern Recogn 34(12):2259–2281
2. Liu Q, Li N, Jia H, Qi Q, Abualigah L (2023) A chimp-inspired remora optimization algorithm for multilevel thresholding image segmentation using cross entropy. Artif Intell Rev 1–58. https://doi.org/10.1007/s10462-023-10498-0
3. Alaiad AI, Mugdadi EA, Hmeidi II, Obeidat N, Abualigah L (2023) Predicting the severity of COVID-19 from lung CT images using novel deep learning. J Med Biol Eng 43(2):135–146
4. Chakraborty S, Saha AK, Ezugwu AE, Agushaka JO, Zitar RA, Abualigah L (2023) Differential evolution and its applications in image processing problems: A comprehensive review. Arch Comput Methods Eng 30(2):985–1040
5. Ye Q, Gao W, Zeng W (2003) Color image segmentation using density-based clustering. In: 2003 International Conference on Multimedia and Expo. ICME'03.Proceedings (Cat. No. 03TH8698). IEEE, Baltimore, MD
6. AbdElaziz M, Ewees AA, Oliva D (2020) Hyper-heuristic method for multilevel thresholding image segmentation. Expert Syst Appl 146:113201
7. Resma KB, Nair MS (2021) Multilevel thresholding for image segmentation using Krill Herd Optimization algorithm. J King Saud Univ Comput Inf Sci 33(5):528–541
8. Senthilkumaran N, Vaithegi S (2016) Image segmentation by using thresholding techniques for medical images. Comput Sci Eng: Int J 6(1):1–13
9. Singh S, Mittal N, Singh H (2020) A multilevel thresholding algorithm using LebTLBO for image segmentation. Neural Comput Appl 32(21):16681–16706
10. Zaitoun NM, Aqel MJ (2015) Survey on image segmentation techniques. Procedia Comput Sci 65:797–806

11. Bhandari AK, Kumar A, Singh GK (2015) Modified artificial bee colony based computationally efficient multilevel thresholding for satellite image segmentation using Kapur's, Otsu and Tsallis functions. Expert Syst Appl 42(3):1573–1601
12. Al-Shourbaji I et al (2023) Artificial ecosystem-based optimization with dwarf mongoose optimization for feature selection and global optimization problems. Int J Comput Intell Syst 16(1):1–24
13. Mohammadi M et al (2016) A game-based meta-heuristic for a fuzzy bi-objective reliable hub location problem. Eng Appl Artif Intell 50:1–19
14. Zare M et al (2023) A modified Particle Swarm Optimization algorithm with enhanced search quality and population using Hummingbird Flight patterns. Decis Anal J 7:100251
15. Abualigah L, Habash M, Hanandeh ES, Hussein AM, Shinwan MA, Zitar RA, Jia H (2023) Improved reptile search algorithm by Salp swarm algorithm for medical image segmentation. J Bionic Eng 1–25. https://doi.org/10.1007/s42235-023-00332-2
16. Nama S et al (2023) Boosting particle swarm optimization by backtracking search algorithm for optimization problems. Swarm Evol Comput 79:101304
17. Dinkar SK et al (2021) Opposition-based Laplacian equilibrium optimizer with application in image segmentation using multilevel thresholding. Expert Syst Appl 174:114766
18. Yang Z, Wu A (2020) A non-revisiting quantum-behaved particle swarm optimization based multilevel thresholding for image segmentation. Neural Comput Appl 32(16):12011–12031
19. Abualigah L, Diabat A (2021) Advances in sine cosine algorithm: a comprehensive survey. Artif Intell Rev 54(4):2567–2608
20. Abualigah L et al (2020) Salp swarm algorithm: a comprehensive survey. Neural Comput Appl 32(15):11195–11215
21. Abualigah L, Diabat A (2020) A comprehensive survey of the Grasshopper optimization algorithm: results, variants, and applications. Neural Comput Appl 32(19):15533–15556
22. Abualigah L (2020) Multi-verse optimizer algorithm: a comprehensive survey of its results, variants, and applications. Neural Comput Appl 32(16):12381–12401
23. Yu H, He F, Pan Y (2020) A scalable region-based level set method using adaptive bilateral filter for noisy image segmentation. Multimed Tools Appl 79(9):5743–5765
24. Yin S, Zhang Y, Karim S (2018) Large scale remote sensing image segmentation based on fuzzy region competition and Gaussian mixture model. IEEE Access 6:26069–26080
25. Wang Z, Wang E, Zhu Y (2020) Image segmentation evaluation: a survey of methods. Artif Intell Rev 53(8):5637–5674
26. Abdulla AA (2020) Efficient computer-aided diagnosis technique for leukaemia cancer detection. IET Image Proc 14(17):4435–4440
27. Mohammed ZF, Abdulla AA (2021) An efficient CAD system for ALL cell identification from microscopic blood images. Multimed Tools Appl 80(4):6355–6368
28 Wang ZY et al (2021) Multi-view intrinsic low-rank representation for robust face recognition and clustering. IET Image Processing 15(14):3573–3584
29. Hui K-F et al (2022) Robust low-rank representation via residual projection for image classification. Knowl-Based Syst 241:108230
30. Abualigah L et al (2021) The arithmetic optimization algorithm. Comput Methods Appl Mech Eng 376:113609
31. Otsu N (1979) A threshold selection method from gray-level histograms. IEEE Trans Syst Man Cybern 9(1):62–66
32. Dirami A et al (2013) Fast multilevel thresholding for image segmentation through a multiphase level set method. Signal Process 93(1):139–153

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.

Authors and Affiliations

Mohammad Otair[1] · Laith Abualigah[2,3,4,5,6,7,8] · Saif Tawfiq[1] · Mohammad Alshinwan[9] · Absalom E. Ezugwu[10] · Raed Abu Zitar[11] · Putra Sumari[7]

✉ Laith Abualigah
aligah.2020@gmail.com

Mohammad Otair
Dean@khawarizmi.edu.jo

Mohammad Alshinwan
moshanwan@yahoo.com

Absalom E. Ezugwu
Ezugwua@ukzn.ac.za

Raed Abu Zitar
raed.zitar@sorbonne.ae

Putra Sumari
putra@usm.my

1 Khawarizmi University Technical College, Amman 11953, Jordan

2 Computer Science Department, Prince Hussein Bin Abdullah Faculty for Information Technology, Al Al-Bayt University, Mafraq 25113, Jordan

3 Hourani Center for Applied Scientific Research, Al-Ahliyya Amman University, Amman 19328, Jordan

4 MEU Research Unit, Middle East University, Amman 11831, Jordan

5 Applied Science Research Center, Applied Science Private University, Amman 11931, Jordan

6 School of Engineering and Technology, Sunway University Malaysia, 27500 Petaling Jaya, Malaysia

7 School of Computer Sciences, Universiti Sains Malaysia, 11800 Pulau Pinang, Malaysia

8 Department of Electrical and Computer Engineering, Lebanese American University, Byblos 13-5053, Lebanon

9 Faculty of Information Technology, Applied Science Private University, Amman 11931, Jordan

10 Unit for Data Science and Computing, North-West University, 11 Hoffman Street, Potchefstroom 2520, South Africa

11 Sorbonne Center of Artificial Intelligence, Sorbonne University-Abu Dhabi, 38044 Abu Dhabi, United Arab Emirates