A Hybrid COVID-19 Detection Model Using an Improved Marine Predators Algorithm and a Ranking-Based Diversity Reduction Strategy

MOHAMED ABDEL-BASSET[1], **REDA MOHAMED**[1], **MOHAMED ELHOSENY**[2], **RIPON K. CHAKRABORTTY**[3], **AND MICHAEL RYAN**[3]

[1]Faculty of Computers and Informatics, Zagazig University, Zagazig 44519, Egypt
[2]Faculty of Computers and Information, Mansoura University, Dakahlia 35516, Egypt
[3]Capability Systems Centre, School of Engineering and IT, University of New South Wales, Canberra, BC 2610 ACT, Australia

Corresponding author: Mohamed Abdel-Basset (mohamed.abdelbasset@fci.zu.edu.eg)

ABSTRACT Many countries are challenged by the medical resources required for COVID-19 detection which necessitates the development of a low-cost, rapid tool to detect and diagnose the virus effectively for a large numbers of tests. Although a chest X-Ray scan is a useful candidate tool the images generated by the scans must be analyzed accurately and quickly if large numbers of tests are to be processed. COVID-19 causes bilateral pulmonary parenchymal ground-glass and consolidative pulmonary opacities, sometimes with a rounded morphology and a peripheral lung distribution. In this work, we aim to extract rapidly from chest X-Ray images the similar small regions that may contain the identifying features of COVID-19. This paper therefore proposes a hybrid COVID-19 detection model based on an improved marine predators algorithm (IMPA) for X-Ray image segmentation. The ranking-based diversity reduction (RDR) strategy is used to enhance the performance of the IMPA to reach better solutions in fewer iterations. RDR works on finding the particles that couldn't find better solutions within a consecutive number of iterations, and then moving those particles towards the best solutions so far. The performance of IMPA has been validated on nine chest X-Ray images with threshold levels between 10 and 100 and compared with five state-of-art algorithms: equilibrium optimizer (EO), whale optimization algorithm (WOA), sine cosine algorithm (SCA), Harris-hawks algorithm (HHA), and salp swarm algorithms (SSA). The experimental results demonstrate that the proposed hybrid model outperforms all other algorithms for a range of metrics. In addition, the performance of our proposed model was convergent on all numbers of thresholds level in the Structured Similarity Index Metric (SSIM) and Universal Quality Index (UQI) metrics.

INDEX TERMS COVID-19 detection, marine predators algorithm, ranking-based reduction diversity, Kapur's entropy, image segmentation.

I. INTRODUCTION

Due to the limited diagnosis tools available, many countries are only able to apply the COVID-19 [1], [2] test for a limited number of citizens. Despite the great efforts to find an effective way for COVID-19 detection, the required medical resources in many countries represent a big challenge. Accordingly, there is an urgent need to identify a low-cost and rapid tool to detect and diagnose COVID-19 effectively.

Many attempts have been conducted to find a suitable and fast way to detect infected patients in an early stage. After making chest CT scans of 21 patients infected with COVID-19 in China, Guan *et al.* [2] found that CT scan analysis included bilateral pulmonary parenchymal ground-glass and consolidative pulmonary opacities, sometimes with a rounded morphology and a peripheral lung distribution. Consequently, COVID-19 diagnosis can be represented as an image segmentation problem to extract the main features of the disease. This segmentation problem can be solved by developing an algorithm that has the ability to extract the smaller similar regions that can indicate infection with the COVID-19 virus.

The associate editor coordinating the review of this manuscript and approving it for publication was Victor Hugo Albuquerque.

Segmentation of an image, separating image regions from each other, is an essential step in image processing [3] and computer vision [4] to focus on a specific region thereby increasing the accuracy of image analysis techniques. The image segmentation problem (ISP) is present in many fields such as: medical diagnosis [5], [6], object recognition [7], satellite image processing [8], remote sensing [9], historical documents [10], and historical newspapers [11], [12].

Several techniques have been proposed to provide an effective image segmentation tool, such as region-based segmentation [13], edge-based detection [14], feature selection-based clustering [15], and threshold-based segmentation [16]. Due to its simplicity, speed, and accuracy, threshold-based segmentation is widely used for image segmentation [3], [17], [18] using either a bi-level threshold or a multi-level threshold. In bi-level thresholding, the image is segmented into two regions: object and background. Although the bi-level threshold is very useful in subdividing the image into only two parts, many applications are interested in more than two regions. In that case, another threshold technique called multi-level threshold has been used to segment the image into more than two regions. Although increasing the number of regions extracted from the image, the time needed to segment the image increases exponentially with the number of regions of interest.

Threshold techniques are based on two approaches: parametric and non-parametric [19]. In a parametric approach, some parameters for each class in the image need to be computed using a probability density function. However, in a non-parametric approach, the technique searches for the optimal threshold values based on maximizing an appropriate function (such as Kapur's entropy [20], fuzzy entropy [21], and Otsu function [22]) without needing to calculate parameters at the outset.

Since processing time increases exponentially with increasing numbers of thresholds, traditional techniques will take considerable time to search for the optimal threshold. Consequently, meta-heuristic algorithms have been used as excellent stochastic meta-heuristic techniques to overcome the high processing time and accuracy problems [23]–[25]. Recently, many meta-heuristic algorithms have been proposed for image segmentation, such as genetic algorithm (GA) [26], particle swarm optimization (PSO) [27]–[29], ant-colony optimization algorithm [30], whale optimization algorithm (WOA) [31], honey bee mating (HBM) optimization [32], multi-verse optimizer [33], cuckoo search (CS) [34], symbiotic organisms search (SOS) [35], Harris hawks optimization algorithm (HHA) [36], and moth-flame optimization algorithm (MFA) [31], flower pollination algorithm (FPA) [37], crow search algorithm [38], grey wolf optimizer [39], bee colony algorithm (BCA) [40], locust search algorithm (LSA) [41] and firefly optimization algorithm (FFA) [42].

Singla and Patra [43] investigated the bounds and the potential thresholds that contain the optimal threshold values by using the cluster validity measure, and then used the GA algorithm to search for the optimal thresholds from the discovered bounds. GA has also been proposed [44] for image segmentation based on a simulated binary crossover to maximize Kapur's entropy for the medical image. Among swarm algorithms, PSO [45] has been proposed for image segmentation, in addition to improving its performance by cooperative and comprehensive learning to face the dimensionality curse and to reduce the premature convergence of the swarm, respectively. A modified PSO [46] has also been developed to improve its performance for solving ISP using adaptive inertia and the adaptive population. Ghamisi *et al.* [47] introduced fractional-order Darwinian PSO to solve the problem of the *n*-level threshold based on the Otsu function to maximize the variance between classes.

In [31], WOA and MFA were proposed for solving the image segmentation problem by maximizing Otsu's criterion, although only for small threshold levels up to 6. FFA [42] has also been applied to image segmentation but does not perform well for multi-level thresholding, so the improved FFA (IFFA) [48] has been proposed using the Cauchy mutation and neighborhood strategy to avoid being trapped in local optima and to enhance the exploration operation.

CS [34] has also been proposed for tackling the ISP by maximizing the Tsallis entropy. SOS [35] has been proposed for segmenting the color images, improved by opposite-based learning in an attempt to enhance its performance (ISOS). ABC [49] has been used for segmentation of satellite imagery based on maximizing various fitness functions—the technique has been modified by initializing the population using a chaotic search and using differential evolution as a novel search technique to improve the exploitation phase.

The Bacterial Foraging Algorithm (BFA) [50], relying on fuzzy entropy to switch the bacterium between exploitation and exploration operators, has been adapted for gray-scale image segmentation. Also, BFA [51] has been modified by moving the best bacteria to the subsequent iterations to accelerate the convergence to the optimal solution. Furthermore, BFA [52] has been integrated with PSO to support the global search capability and accelerate the convergence rate. In addition, the weak bacterium in BFA chooses a strong bacterium from the healthiest bacteria, then it moves near to the location of this strong selection. WOA [53] has been proposed for tackling liver image segmentation. WOA divides the liver image into a predetermined number of clusters based on the prospect liver position in the abdominal image defined by a statistical image. The problem of multi-level threshold segmentation [54] is handled as a multi-objective problem that maximized both Kapur's entropy and Otsu's function.

Although there are many existing methodologies for medical image segmentation, none of the works exposed at the literature was validated on an image with high threshold levels to observe its ability to segment an image with many similar regions. Subsequently, those algorithms may not be the best choice for searching for smaller homogenous regions in medical images that may contain the features of a disease such as COVID-19. This challenge motivates us to observe

the performance of some state-of-art algorithms proposed in the literature for tackling ISP. In addition, it leads us to propose a robust meta-heuristic algorithm, namely the improved marine predators algorithm (IMPA), that has a good ability to segment an image into many similar regions.

The contribution of this paper is two-fold. First, we propose a hybrid model for COVID-19 detection using an improved marine predators algorithm (IMPA) for overcoming the multi-threshold image segmentation problems of chest X-Ray images. Second, a new method, namely ranking-based diversity reduction (RDR), has been proposed to improve the MPA by moving the positions of the worst solutions to be near to the best solution. The proposed RDR is compared with other well-known algorithms using a set of chest X-Ray images. The experimental results show that MPA and IMPA are better able to solve the image segmentation problem compared with state-of-art algorithms in terms of fitness value and standard metrics. Additionally, it is competitive with EO in low numbers of threshold levels in terms of peak signal-to-noise-ratio (PSNR), and signal-to-noise-ratio (SNR), but has significantly better performance for high numbers of threshold levels. Along with EO, the performance of our proposed algorithm is convergent using the structured similarity index metric (SSIM) and the Universal Quality Index (UQI).

The remainder of the paper is organized as follows. In section 2, we explain the Kapur's entropy formulation. Then, section 3 provides a description of the marine predators algorithm. Section 4 describes the steps of adapting MPA for application to image segmentation. Section 5 provides the results and discussions and section 6 concludes the paper.

II. MULTILEVEL THRESHOLDING

As discussed earlier, image threshold techniques are categorized as bi-level or multilevel thresholding. In this work, optimal threshold values are obtained using a popular multilevel method, namely Kapur's entropy, which determines the optimal threshold values based on the entropy of the segmented regions [20]. Assuming that $[t_0, t_1, t_2, \ldots \ldots \ldots, t_n]$ represents the threshold values that segment the image into multiple regions, then Kapur's entropy method can be formulated in Eq. 1, Eq. 2, Eq. 3, Eq. 4, and Eq. 5.

$$T(t_0, t_1, t_2, \ldots \ldots \ldots, t_n) = T_0 + T_1 + T_2 + \ldots \ldots \ldots + T_n \quad (1)$$

where:

$$T_0 = -\sum_{i=0}^{t_0-1} \frac{X_i}{W_0} * \ln \frac{X_i}{W_0}, \quad X_i = \frac{N_i}{W}, \quad W_0 = \sum_{i=0}^{t_1-1} X_i \quad (2)$$

$$T_1 = -\sum_{i=t_0}^{t_1-1} \frac{X_i}{W_1} * \ln \frac{X_i}{W_1}, \quad X_i = \frac{N_i}{W}, \quad T_1 = \sum_{i=t_0}^{t_1-1} X_i \quad (3)$$

$$T_2 = -\sum_{i=t_1}^{t_2-1} \frac{X_i}{W_2} * \ln \frac{X_i}{W_2}, \quad X_i = \frac{N_i}{W}, \quad T_2 = \sum_{i=t_1}^{t_2-1} X_i \quad (4)$$

$$T_n = -\sum_{i=t_n}^{L-1} \frac{X_i}{W_n} * \ln \frac{X_i}{W_n}, \quad X_i = \frac{N_i}{W}, \quad T_n = \sum_{i=t_n}^{L-1} X_i \quad (5)$$

$T_0, T_1, T_2, \ldots \ldots \ldots,$ and T_n are the entropies of the distinct regions, and N_i indicates the number of pixels with a value of i, the grey level. $W_0, W_1, W_2, \ldots,$ and W_n are the probabilities of the regions relative to the number of pixels W found in the whole image.

To obtain the optimal threshold values, the function at Eq. 6 must be maximized.

$$F(t_0, t_1, t_2, \ldots \ldots \ldots, t_n) = \max\{T(t_0, t_1, t_2, \ldots \ldots \ldots, t_n)\} \quad (6)$$

Here, Eq.6 is used as a fitness function to obtain the optimal threshold values using the MPA illustrated in the next section.

III. MARINE PREDATORS ALGORITHM (MPA)

MPA has been proposed to simulate the optimal foraging mechanism for marine predators in finding their prey: predators use Lévy strategy when there is a low concentration of prey and Brownian movements when there is abundant prey [55]. The velocity ratio v from the prey to the predators represents the tradeoff between the Lévy and Brownian strategies:

1. At low-velocity, $v < 0.1$, the best strategy for the predators is to move in Lévy steps regardless of whether the prey is moving in Brownian or Lévy.
2. At unit velocity, $v = 1$, the predators should move in Brownian if the prey is moving in Lévy steps.
3. Finally, at high-velocity > 10, the best strategy for the predators is to remain motionless, regardless of whether the prey is moving in Brownian or Lévy.

The mathematical model of the MPA is as follows:.

In the first stage, a group of the prey will be initialized within the search space using the following equation:

$$\vec{X} = \vec{X}_{min} + rand(0, 1) * (\vec{X}_{max} - \vec{X}_{min}) \quad (7)$$

where $rand(0, 1)$ is a random number in the range of [0, 1], and $\vec{X}_{min}$ and $\vec{X}_{max}$ are the vectors including the upper and lower bounds for the search space of each dimension in the optimization problem.

After initializing the prey, the fitness of each predator is calculated, and the one that has the best fitness value is determined to be the top predator. Based on the survival of the fittest, the top predator is the best one in foraging, so it is used to construct a matrix known as *Elite*. This elite matrix can be formulated as follows:

$$Elite = \begin{bmatrix} X^I_{1,1} & X^I_{1,2} & \cdots & X^I_{1,d} \\ X^I_{2,1} & X^I_{2,2} & \cdots & X^I_{2,d} \\ . & . & . & . \\ . & . & . & . \\ X^I_{n,1} & X^I_{n,2} & \cdots & X^I_{n,d} \end{bmatrix}$$

where $\vec{X^I}$ represents the top predator vector and is replicated n times to build up an $n \times d$ Elite matrix, where n is the number

of the individuals in the population, and d is the number of dimensions.

Another matrix, namely *Prey*, has the same dimensions as *Elite* and is used by the predators to update their positions.

$$Prey = \begin{bmatrix} X_{1,1} & X_{1,2} & \ldots & X_{1,d} \\ X_{2,1} & X_{2,2} & \ldots & X_{2,d} \\ . & . & . & . \\ . & . & . & . \\ X_{n,1} & X_{n,2} & \ldots & X_{n,d} \end{bmatrix}$$

In the main loops of the MPA, the optimization process is divided into three stages based on the velocity ratio, and is modeled as follows:

A. HIGH VELOCITY RATIO

This is the exploration phase, and is formulated at Eq. 8 and Eq. 9:

$$while\ it\ < \frac{1}{3} * max\_iter$$

$$\vec{S}_i = \vec{R}_B \otimes \left(\overrightarrow{Elite_i} - \vec{R}_B \overrightarrow{\otimes Prey_i} \right) \quad (8)$$

$$\overrightarrow{prey_i} = \overrightarrow{prey_i} + P * \vec{R} \otimes \vec{S}_i \quad (9)$$

where $\vec{R}_B$ is a vector of random numbers created based on the normal distribution and represents the Brownian motion, $\otimes$ represent the entry-wise multiplication, P = 0.5, 0.5 constant is recommended from the original paper, is a constant number, R is a random numbers vector created uniformly, t is the current iteration, and t_{max} is the maximum number of iterations.

B. UNIT VELOCITY RATIO

This phase occurs in the intermediate phase of optimization process, where exploration is gradually changed to exploitation. The mathematical model of this phase is represented in Eq. 10, Eq. 11, Eq. 12, and Eq. 13.

$$while \frac{1}{3} * max_{iter} < it < \frac{2}{3} * max\_iter$$

- For the first half of the population

$$\vec{S}_i = \vec{R}_L \otimes \left(\overrightarrow{Elite_i} - \vec{R}_L \overrightarrow{\otimes Prey_i} \right) \quad (10)$$

$$\overrightarrow{prey_i} = \overrightarrow{prey_i} + P * \vec{R} \otimes \vec{S}_i \quad (11)$$

- For the second half of the population

$$\vec{S}_i = \vec{R}_B \otimes \left(\vec{R}_B \otimes \overrightarrow{Elite_i} - \overrightarrow{Prey_i} \right) \quad (12)$$

$$\overrightarrow{prey_i} = \overrightarrow{Elite_i} + P * CF \otimes \vec{S}_i \quad (13)$$

where $\vec{R}_L$ is the vector created using the Lé vy flight strategy. In this phase, the first half of prey would move with Lé vy steps, while the other half uses Brownian steps.

where CF is an adaptive parameter to control the step size and is generated using Eq. 14.

$$CF = (1 - \frac{it}{max\_iter})^{\left(2 \frac{it}{max\_iter}\right)} \quad (14)$$

C. LOW VELOCITY RATIO

This is the exploitation phase and is formulated using Eq. 15 and Eq. 16:

$$while\ it\ > \frac{2}{3} * max\_iter$$

$$\vec{S}_i = \vec{R}_L \otimes \left(\vec{R}_L \otimes \overrightarrow{Elite_i} - \overrightarrow{Prey_i} \right) \quad (15)$$

$$\overrightarrow{prey_i} = \overrightarrow{Elite_i} + P * CF \otimes \vec{S}_i \quad (16)$$

Some studies confirmed that the surrounding environment such as the eddy formulation, and fish aggregating devices (FADs) affects the behavior of the prey. As a result, the predators spend 80% of their time searching for their prey in the vicinity, while the remaining time, they search for the prey in another environment. This process is known as FADs and is calculated using Eq. 17.

$$\overrightarrow{prey_i} = \begin{cases} \overrightarrow{prey_i} + CF[\vec{X}_{min} + \vec{R} * (\vec{X}_{max} - \vec{X}_{min})] \otimes \vec{U} \\ \qquad \text{if } r < \text{FADs} \\ \overrightarrow{prey_i} + [\text{FADs}\,(1 - r) + r]\left(\overrightarrow{prey_{r1}} - \overrightarrow{prey_{r2}} \right) \\ \qquad \text{if } r > \text{FADs} \end{cases}$$

$$(17)$$

where r is a random number in the range of [0, 1]. $\vec{U}$ is the vector containing the arrays with 0 and 1 values. For each array in $\vec{U}$, a random number between 0 and 1 is generated and if the generated number is greater than 0.2, then this array is set to 1; otherwise it is set to 0. FADs = 0.2 indicates the influence of the FADs on the searching process.

MPA accomplishes memory saving by saving the old position of the prey. And, after updating the current solutions, the fitness values of each current solution and each old solution are compared, and if the fitness of the old one is better than the current one, they are swapped. The steps of MPA are listed in Algorithm 1.

IV. THE HYBRID PROPOSED MODEL
In this section, standard MPA and improved MPA (IMPA) have been developed for overcoming the multi-thresholding image segmentation problems. The steps of adaptation are shown in the next sections.

A. INITIALIZATION
In this phase, the number of prey N and the number threshold are predefined. Then each threshold is initialized randomly within 0 and 255 (the gray levels of the 8-bit image) using Eq. 18.

$$P_{i,j} = L_{min} + r * (L_{max} - L_{min}) \quad (18)$$

where L_{min}, and L_{max} indicate the upper and lower bounds of the gray level values in the image histogram, and r is a random number generated randomly in the range of [0, 1].

Algorithm 1 The Marine Predators Algorithm (MPA)

1. Initialize the population of prey $p_i (i = 1, 2, 3, \ldots, n)$
2. Set parameter's value
3. $P = 0.5$; Top_Predetor_fit $== MAX\_VALUEX$
4. Top_Predetor_Position $=$ NULL
5. **while** (it $<$ t$_{maxIter}$)
6.   **for** each i prey
7.   Calculate the fitness value of prey $if (\overrightarrow{p}_i)$
8.   **if** $(f(\overrightarrow{p}_i) <$ Top_Predetor_Best))
9.   Top_Predetor_Best $= f(\overrightarrow{p}_i)$
10.   Top_Predetor_Position $= \overrightarrow{p}_i$
11.   **End if**
12.   **End for**
13.   Construct The *Elitematrix*
14.   Accomplish the memory saving
15.   Assign *CF* using Eq. (14)
16.   **for** each *i* prey
17.   ***if*** $(it < \frac{1}{3} * t_{max})$
18.   Update the current $\overrightarrow{p}_i$ using Eq. (9)
19.   ***Elseif*** $(\frac{1}{3} * t_{max} < it < \frac{2}{3} * t_{max}$
20.   ***If*** $(i < \frac{1}{2} * n)$
21.   Update the current $\overrightarrow{p}_i$ using Eq. (11)
22.   ***Else***
23.   Update the current $\overrightarrow{p}_i$ using Eq. (13)
24.   **End if**
25.   ***Else***
26.   Update the current $\overrightarrow{p}_i$ using Eq. (16)
27.   **End if**
28.   **end for**
29.   **for** each i prey
30.   Calculate the fitness value of prey $if (\overrightarrow{p}_i)$
31.   **if** $(f(\overrightarrow{p}_i) <$ Top_Predetor_Best))
32.   Top_Predetor_Best $= f(\overrightarrow{p}_i)$
33.   Top_Predetor_Position $= \overrightarrow{p}_i$
34.   **End if**
35.   **end for**
36.   Accomplish the memory saving
37.   Accomplish the FADs for each predator $\overrightarrow{p}_i$ using Eq. (17)
38.   it $++$
39. **end while**

Algorithm 2 RDR

1. P : the number of prey
2. CR : a vector of size N and contain 0's value in the start
3. $i = 0$
4. $perIter = 3$
5. **while** $(i < N)$
6.   **if** (fit $(P_i) >$ fitLocal (P_i))
7. $CR_i ++$
8.   **else**
9. $CR_i = 0$
10.   **end if**
11. $i ++$
12. **end while**
13. **for** each *i* particle
14.   **if** $(CR_i > perIter)$
15.   Update P$_i$ toward the best one using Eq.19
16.   **end if**
17. **end for**

---

distance from the optimal solution using the **Eq.19.**

$$\overrightarrow{P_b} = \overrightarrow{P_b} + r * \left(\overrightarrow{P_b} - \overrightarrow{P_i}\right) \tag{19}$$

where $\overrightarrow{P_i}$ refers to a worst particle that fails in finding a better solution within a consecutive number of iteration, $\overrightarrow{P_b}$ refers to a vector of the best solution, and r is a number generated randomly in the range of [0, 1]. This technique that reduces the distance between the optimal solution and the particles that couldn't find a better solution within a consecutive number of iterations is called RDR. **Algorithm 2** illustrates the steps of the RDR technique.

In **Algorithm 2,** a vector of size equal to the number of prey is created and initialized in 0's value. Then the old fitness is compared with the current fitness, and if the old fitness is still better, the rank CR_i of the *i*th particle is increased by 1. Otherwise, it is reset to 0 again. This will help to identify the number of particles that couldn't reach better solutions within a consecutive number of iterations. After that, each particle couldn't find a better solution within the consecutive number of iterations CN, predefined, will be updated towards the best solution using Eq.19.

C. THE PROPOSED IMPA

The steps of adapting the IMPA using the RDR for overcoming multi-thresholding problems are illustrated in Fig 1. The initialization step is considered the first step for all meta-heuristic algorithms, so it is firstly used for initializing the prey randomly, as shown in Fig 1. Within the initialization step, the fitness of each prey would be calculated, and the one with the highest fitness value is defined as the Top_Predator_Best, and its position as the Top_Predator_Position. After that, the first stage of the primary optimization process will start to update the current positions using one of the updating equations illustrated in Section 2 at the expense of the current iteration and prey.

B. RANKING-BASED DIVERSITY REDUCTION TECHNIQUE (RDR)

Some particles may be far away from an optimal solution which will require a long time to find and the number of iterations may terminate before a better solution is reached. Therefore, we propose an algorithm to calculate the consecutive number of iterations in which each particle was not able to identify a better solution. After identifying the worst particles that fail to find a better solution within a consecutive number of iterations, in **Algorithm 2** those particles will be updated towards the best solution found so far to reduce the

FIGURE 1. Flowchart of IMPA for overcoming image segmentation problem.

After finishing the first stage of the optimization process, the fitness value of each prey is calculated, and memory saving is accomplished. Last but not least, the second stage of the optimization process implements the FADs methodology. FADs helps MPA dispose of local optima, subsequently finding better solutions. Finally, after the selected number of iterations, the RDR strategy is called to reduce diversity through the population, as elaborated in Section 4.2. The first and second stages of the optimization process, in addition to the RDR strategy, will be repeated until the termination criterion is satisfied.

Note that i in Fig. 1 indicates the current particle number, and N refers to the maximum number of particles.

Memory saving in MPA replaces the old solution with the current one if the current is better; otherwise the old one is used in the population to be updated toward another direction for finding better solutions. But what happen if the old one is always better? This means that the predator would stay in its position, motionless, and the distance with the best solution would not change. As long as the particles are far away of the best solution, the probability of finding a better solution reduces. Subsequently, a significant number of

a. Original image X1   b. Histogram of X1   c. Original image X2   d. Histogram of X2

e. Original image X3   f. Histogram of X3   g. Original image X4   h. Histogram of X4

l. Original image X5   m. Histogram of X5   n. Original image X6   o. Histogram of X6

p. Original image X7   q. Histogram of X7   r. Original image X8   s. Histogram of X8

t. Original image X9   u. Histogram of X9

FIGURE 2. Illustration the original images and their histograms used in our experiment.

FIGURE 3. Comparison of the STD values obtained by each algorithm.

TABLE 1. PSNR values obtained by each algorithm.

Im	T	IMPA	MPA	EO[57]	HHA [36]	WOA [31]	SCA [56]	SSA [58]	Im	T	IMPA	MPA	EO[57]	HHA [36]	WOA [31]	SCA [56]	SSA [58]
Ct1	10	25.6438	25.6114	25.6593	25.6228	**25.6636**	25.0539	25.3485	X1	10	**26.4396**	26.4077	26.0003	25.3781	25.3792	24.2924	25.9790
	20	29.6535	29.5767	**29.8714**	29.2280	29.4436	28.4885	29.0232		20	**31.7614**	31.1351	31.4536	30.2321	30.6570	29.3323	31.2057
	30	31.5667	31.4428	**31.6654**	31.0503	31.3505	30.3652	30.8446		30	**33.7695**	33.3255	33.6722	32.5147	32.8609	31.3919	32.9712
	40	32.5000	**32.6214**	32.6610	32.0961	32.3612	31.3514	31.9752		40	**34.8530**	34.3624	34.6389	33.7115	34.0642	32.7077	33.9689
	50	33.1628	**33.1895**	33.1239	32.7074	32.9174	32.1816	32.7346		50	**35.3744**	34.9688	35.3043	34.4811	34.8323	33.7750	34.5409
	60	33.5250	**33.5690**	33.4733	33.0939	33.2423	32.7975	32.9721		60	**35.6864**	35.5034	35.5637	34.9351	35.0762	34.3679	35.0276
	80	**33.8581**	33.7931	33.7883	33.5721	33.6808	33.2338	33.5221		80	**35.8838**	35.7419	35.8604	35.6048	35.6561	35.2083	35.4195
	100	**33.9716**	33.9371	33.9151	33.8406	33.8524	33.6864	33.8266		100	35.9787	35.9363	**36.0043**	35.7992	35.8689	35.5021	35.6632
Ct2	10	**26.3581**	26.1257	26.2771	26.2469	26.0932	25.3036	25.3677	X2	10	20.8763	20.8798	**20.8803**	20.8363	20.8573	20.5113	20.7013
	20	29.9333	29.8607	**30.2452**	29.4571	29.9520	28.9911	29.3778		20	21.5820	21.5748	**21.5911**	21.5245	21.5464	21.3895	21.4827
	30	32.3178	32.4930	**33.0093**	31.8064	32.3767	31.4079	31.7477		30	21.7437	21.7465	**21.7689**	21.6947	21.7366	21.5473	21.6463
	40	**34.0805**	33.8359	34.0564	33.2113	33.5550	32.5865	33.0408		40	21.8254	21.8047	**21.8378**	21.7960	21.8145	21.7538	21.7429
	50	**34.9983**	34.6699	34.8803	34.0286	34.3049	33.6192	34.2083		50	**21.8687**	21.8600	21.8681	21.8321	21.8464	21.7738	21.8154
	60	**35.5023**	35.3832	35.2021	34.6440	34.8051	34.2315	34.9505		60	21.8780	21.8707	**21.8891**	21.8595	21.8738	21.8301	21.8294
	80	**35.9231**	35.7243	35.6008	35.3656	35.5102	35.0862	35.2402		80	**21.9029**	21.8937	21.9025	21.8833	21.8947	21.8754	21.8763
	100	**35.9859**	36.0128	35.9388	35.7682	35.8485	35.4451	35.6093		100	**21.9093**	21.9056	21.9092	21.9020	21.9064	21.8962	21.8867
Ct3	10	25.3088	25.4251	**25.4945**	25.0604	25.2573	24.5937	25.3582	X3	10	26.0695	**26.1092**	26.0839	26.1641	26.0702	25.5570	25.7961
	20	29.2327	29.3267	**29.7444**	28.7699	29.0285	28.0222	29.0224		20	31.2699	31.4847	**31.7178**	31.1614	31.3774	30.0055	29.9148
	30	31.1300	31.1550	**31.3199**	30.5842	30.7580	29.9384	30.5462		30	34.6792	34.6760	**35.0608**	33.9729	34.4152	32.4621	32.4027
	40	32.0583	31.9607	**32.0765**	31.5027	31.7284	31.0153	31.7085		40	36.7248	36.9596	**37.1281**	35.8774	36.2978	34.4223	34.7787
	50	**32.5801**	32.4934	32.4213	31.9803	32.1967	31.6665	32.0975		50	38.6822	38.7828	**38.7647**	37.1489	37.6665	36.3515	36.2394
	60	**32.8778**	32.7538	32.7354	32.4353	32.4822	32.1358	32.4285		60	**40.0923**	39.7226	39.9296	38.2266	38.9595	37.2993	38.0785
	80	**33.0505**	33.0253	33.0000	32.7370	32.9206	32.5732	32.8327		80	**41.2113**	41.1223	41.0330	39.9809	40.7269	39.6003	39.6961
	100	**33.1543**	33.1316	33.1116	33.0136	33.0560	32.8704	33.0025		100	**42.1024**	42.0139	41.9592	41.3227	41.7962	40.1951	40.4920
Ct4	10	24.9202	24.9105	**25.0204**	24.9932	25.0112	24.3884	24.6093	X4	10	25.0917	25.0672	**25.7366**	25.2783	25.4134	25.1794	25.1261
	20	28.7235	28.7318	**28.8147**	28.6994	28.8868	27.6519	27.6428		20	27.2812	27.3797	**27.5745**	26.9743	26.8693	26.8318	27.4334
	30	30.1727	30.0947	**30.4051**	29.9998	30.2122	29.2940	28.6977		30	**28.4108**	28.1666	28.4055	27.8591	27.8288	27.8215	28.1297
	40	30.7513	30.8867	**31.0502**	30.7188	30.9422	30.1117	29.6410		40	28.7189	28.5647	**28.7469**	28.2718	28.3160	28.1095	28.5156
	50	31.2558	31.2829	**31.3387**	31.0564	31.2429	30.6949	30.2932		50	**28.8820**	28.7655	28.8800	28.4901	28.5137	28.4503	28.6864
	60	**31.4774**	31.4865	31.4513	31.3422	31.4985	31.1129	30.8719		60	28.8953	28.8468	**28.9449**	28.6880	28.6779	28.5253	28.7378
	80	31.7519	31.7621	**31.7816**	31.7059	31.7489	31.4269	31.4074		80	**28.9638**	28.9177	28.9577	28.8827	28.8557	28.7813	28.8366
	100	31.8636	**31.8708**	31.7756	31.7967	31.8458	31.6709	31.6038		100	**29.0049**	28.9687	29.0049	28.9277	28.9294	28.9005	28.9368
Ct5	10	25.9479	25.8624	**25.9493**	25.8632	25.7995	25.1196	25.4050									
	20	29.8747	29.7453	**30.2898**	29.4749	29.8257	28.9057	28.9889									
	30	32.0100	32.0282	**32.4423**	31.6852	32.0339	31.1892	30.6350									
	40	33.3348	33.3868	**33.5671**	32.7428	33.1892	32.1122	32.3377									
	50	34.2223	34.2814	**34.3885**	33.5403	33.6921	33.1321	33.2755									
	60	**34.7817**	34.7564	34.6739	34.1727	34.3080	33.7635	33.9079									
	80	**35.1748**	35.1650	35.1262	34.7413	34.9482	34.5249	34.5881									
	100	**35.3424**	35.3531	35.2286	35.1508	35.2174	34.9130	35.0585									

iterations would be neglected. To solve this problem, the RDR strategy is used to move the particle that failed to find a better solution within a consecutive number of iterations, gradually toward the best solution even if the updated position isn't better than the old one. This will help the particle in exploring whether other regions may contain a better solution. Because the best solution is unified for all the members, the diversity between the members of the population will reduce when the particles move toward it. Accordingly, many better solutions may be generated, due to the exploration of more regions by the particles that couldn't find better solution within a consecutive number of iterations.

V. RESULTS AND DISCUSSION

In this section, the conducted experiments are offered and discussed to show the superiority of our proposed algorithm for solving ISP. This section is organized as follows:

- **Section A.** Describes Test Images

FIGURE 4. Average PSNR values obtained by each algorithm on selected threshold levels from 10 to 100.

- **Section B.** discusses Stability Analysis of all the compared algorithms.
- **Section C.** discusses the results of The Peak Signal to Noise Ratio (PSNR) metric.
- **Section D.** discusses the results of the Signal to Noise Ratio (PSNR) metric.
- **Section E.** demonstrates the outcomes of the Structures similarity index metric (SSIM).
- **Section F.** elaborates the results of the universal quality index (UQI).
- **Section G.** demonstrate the obtained Kapure's entropy values
- **Section H.** shows some segmented images using IMPA, and MPA

A. DESCRIPTION OF TEST IMAGES

In our experiment, eight COVID-19 Chest images taken from https://github.com/ieee8023/covid-chestxray-dataset are used to validate the performance of our proposed algorithm and other algorithms in extracting the similar regions.

These images are labelled X1, X2, X3, X4, X5, X6, X7, X8, and X9. The original images and the histogram of each are shown in Fig.2. We compared our proposed model and selected state-of-art algorithms: SCA [56], WOA [31], EO [57], HHA [36] and SSA [58] using the same parameters and running environment. The population size N was set to 20, and the maximum iterations t_{max} set to 150 for a fair comparison. The experiments are performed on a desktop computer equipped with Windows 7 ultimate platform and 1 GB memory space. The RDR strategy is implemented on each particle that exceeds 3 iterations (CN = 3) without a better solution.

B. STABILITY ANALYSIS

To measure the dispersion of the results obtained by each algorithm, the standard deviation (Std) is calculated using Eq. 20.

$$Std = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(f_i - \bar{f}\right)^2} \qquad (20)$$

TABLE 2. SNR values obtained by each algorithm.

Im	T	IMPA	MPA	EO[57]	HHA [36]	WOA [31]	SCA [56]	SSA [58]	Im	T	IMPA	MPA	EO[57]	HHA [36]	WOA [31]	SCA [56]	SSA [58]
Ct1	10	10.8240	10.7786	10.8778	10.7495	10.7820	9.9459	10.4284	X1	10	10.6334	10.8049	10.2043	9.1994	9.5566	8.5098	10.4864
	20	17.4175	17.3917	17.9046	17.0106	17.0999	15.5589	16.5632		20	20.5694	20.0733	20.3042	17.5131	18.0261	16.4368	15.2409
	30	22.2459	21.9222	22.8575	20.8455	21.4097	19.0977	20.6808		30	26.5105	25.2425	25.9821	21.6466	23.9526	20.4738	17.2882
	40	25.2272	25.1935	25.3231	23.6141	24.5227	22.4483	23.3454		40	30.2926	29.1103	29.6285	26.3234	26.8573	23.8391	19.3932
	50	26.8063	27.1618	26.8225	25.1452	26.2737	24.4223	25.5653		50	32.0761	31.2943	31.4070	28.9808	30.2410	26.7129	21.5355
	60	28.3107	28.0683	28.0373	26.6751	27.4298	25.5240	26.6113		60	33.6086	32.3895	32.8834	30.6751	31.0281	29.0854	22.3165
	80	29.3228	29.0125	29.1183	28.3127	28.4868	27.6240	28.1309		80	34.3374	33.8719	34.1858	32.5410	33.1873	31.3298	23.9892
	100	29.7972	29.6562	29.6316	29.3327	29.2809	28.6254	29.2171		100	34.9276	34.6412	34.6057	33.8629	34.2499	32.9533	24.7140
Ct2	10	13.2361	12.7460	12.9408	13.0049	13.0303	12.0739	12.0083	X2	10	5.8820	5.8718	5.8702	5.8050	5.8361	5.6404	10.9554
	20	20.4176	20.5281	21.7423	19.1905	20.2597	18.4100	19.0179		20	6.5623	6.5496	6.5569	6.4543	6.5273	6.3630	17.8530
	30	27.4094	27.9633	29.4428	26.0650	27.1676	24.5532	24.8258		30	6.7526	6.7362	6.7699	6.6998	6.7374	6.6066	21.6100
	40	33.7004	33.1953	33.8089	30.4968	31.5892	29.0149	30.5808		40	6.8653	6.8522	6.8671	6.8114	6.8386	6.7532	27.0730
	50	36.7602	36.4549	36.5976	33.5335	34.3721	30.9607	34.5072		50	6.9095	6.9009	6.9095	6.8860	6.8945	6.8169	28.9120
	60	39.5965	38.8336	38.0979	35.6575	36.9086	33.6730	36.1078		60	6.9395	6.9322	6.9438	6.9016	6.9291	6.8630	31.7581
	80	41.4819	41.0862	40.6519	38.9296	39.6565	37.6209	39.0347		80	6.9667	6.9650	6.9666	6.9471	6.9557	6.9316	34.9387
	100	42.0208	42.2317	41.8943	40.9642	41.0937	39.5952	41.0296		100	6.9827	6.9796	6.9800	6.9620		6.9513	36.2522
Ct3	10	10.7088	10.6294	10.8627	10.3970	10.4540	9.3811	10.1963	X3	10	13.5002	13.2116	13.0320	13.2529	13.0777	12.1990	11.0452
	20	17.1397	17.2531	18.1058	16.4281	16.9194	15.4037	15.9523		20	24.8626	25.2788	25.5137	23.6436	25.0664	20.9306	19.4441
	30	21.8271	21.5179	22.4702	20.6409	21.0605	19.0334	20.0211		30	37.4303	36.4281	37.9659	33.3239	34.9028	27.9437	24.0457
	40	24.2833	24.1751	24.4793	22.5780	22.9051	20.4875	23.2512		40	49.1392	48.9013	50.1542	40.7477	42.6703	35.7066	27.2235
	50	25.9867	25.4154	25.5696	24.3743	24.6076	22.7335	24.8419		50	60.2658	59.4602	59.5526	48.0404	54.0166	42.8576	29.0866
	60	26.8013	26.4296	26.3118	25.2730	25.7686	24.4078	25.4008		60	70.9716	68.8028	66.2353	55.3538	60.8479	50.0217	30.6795
	80	27.4246	27.1625	27.2031	26.5442	26.8317	25.6777	26.6347		80	80.1626	78.9483	80.0544	68.4719	74.3533	63.1169	31.9762
	100	27.7901	27.6192	27.4659	27.1407	27.1630	26.7232	27.0462		100	88.0085	90.1688	82.8517	77.9799	83.2123	70.9763	33.6479
Ct4	10	10.9000	10.9284	11.0123	11.0230	11.1165	10.3880	10.4284	X4	10	8.1361	8.4347	8.7699	8.4424	8.6813	8.3840	5.7619
	20	17.5245	17.2812	17.6530	17.4189	17.7865	15.5524	16.5632		20	11.2343	10.9065	11.4610	10.5505	10.4609	10.2462	6.4290
	30	20.8501	20.8875	21.1782	20.6200	20.8453	18.7619	20.6808		30	12.7251	12.5197	12.7841	11.6103	12.0625	11.3950	6.6575
	40	22.5714	22.3695	23.1355	22.0724	22.6847	20.9587	23.3454		40	13.2048	13.1761	13.3454	12.6216	12.6054	12.4319	6.7531
	50	23.8669	23.9407	24.1753	23.3263	23.7954	22.2017	25.5653		50	13.6000	13.3924	13.5202	13.1921	13.1169	12.8830	6.8421
	60	24.5506	24.5702	24.8386	23.8685	24.5009	23.5310	26.6113		60	13.7012	13.5994	13.7367	13.2129	13.3289	13.1332	6.8887
	80	25.2961	25.3651	25.2164	25.0001	25.2380	24.3972	28.1309		80	13.7380	13.7187	13.7247	13.6217	13.6082	13.4898	6.9243
	100	25.6519	25.6600	25.6033	25.4650	25.6415	25.0475	29.2171		100	13.8342	13.7926	13.8842	13.7255	13.6892	13.6272	6.9549
Ct5	10	12.1281	12.1480	12.1333	11.8906	11.9761	10.8363	12.0083									
	20	19.3158	19.2837	20.4187	19.0932	19.1566	17.4755	19.0179									
	30	25.0686	24.7065	26.4808	24.1239	25.2583	21.5859	24.8258									
	40	29.9916	29.6286	30.2685	27.7150	29.0109	26.2733	30.5808									
	50	33.0794	32.7930	33.2645	31.0167	31.5333	29.0920	34.5072									
	60	34.9871	34.6218	34.9864	32.8950	33.2601	30.7869	36.1078									
	80	36.9259	36.6796	36.4429	34.8612	35.5948	34.2273	39.0347									
	100	37.5329	37.5768	37.0237	36.5845	37.0026	35.8944	41.0296									

TABLE 3. SSIM values of each algorithm.

Im	T	IMPA	MPA	EO[57]	HHA [36]	WOA [31]	SCA [56]	SSA [58]	Im	T	IMPA	MPA	EO [57]	HHA [36]	WOA [31]	SCA [56]	SSA [58]
1	10	0.9880	0.9879	0.9880	0.9879	0.9880	0.9861	0.9870	6	10	0.9804	0.9803	0.9771	0.9724	0.9717	0.9637	0.9268
	20	0.9931	0.9930	0.9934	0.9926	0.9929	0.9916	0.9922		20	0.9915	0.9906	0.9912	0.9882	0.9895	0.9859	0.9337
	40	0.9944	0.9943	0.9944	0.9940	0.9942	0.9934	0.9937		40	0.9933	0.9928	0.9931	0.9915	0.9921	0.9899	0.9348
	60	0.9948	0.9949	0.9949	0.9946	0.9948	0.9941	0.9944		60	0.9939	0.9935	0.9937	0.9928	0.9932	0.9916	0.9348
	80	0.9951	0.9952	0.9951	0.9949	0.9950	0.9946	0.9949		80	0.9942	0.9939	0.9941	0.9934	0.9938	0.9927	0.9350
	100	0.9953	0.9953	0.9952	0.9951	0.9952	0.9949	0.9950		100	0.9943	0.9942	0.9942	0.9938	0.9939	0.9932	0.9353
	120	0.9954	0.9954	0.9954	0.9952	0.9953	0.9951	0.9952		120	0.9944	0.9943	0.9943	0.9941	0.9942	0.9939	0.9351
	150	0.9954	0.9954	0.9954	0.9954	0.9954	0.9953	0.9954		150	0.9944	0.9944	0.9944	0.9942	0.9943	0.9940	0.9352
2	10	0.9884	0.9881	0.9883	0.9882	0.9879	0.9851	0.9862	7	10	0.9293	0.9293	0.9296	0.9294	0.9293	0.9269	0.9844
	20	0.9929	0.9929	0.9932	0.9922	0.9929	0.9912	0.9920		20	0.9347	0.9348	0.9340	0.9340	0.9342	0.9332	0.9905
	40	0.9945	0.9947	0.9949	0.9940	0.9945	0.9936	0.9939		40	0.9355	0.9357	0.9351	0.9348	0.9353	0.9350	0.9926
	60	0.9952	0.9952	0.9953	0.9948	0.9950	0.9943	0.9945		60	0.9355	0.9356	0.9352	0.9351	0.9353	0.9352	0.9942
	80	0.9954	0.9954	0.9955	0.9951	0.9953	0.9949	0.9952		80	0.9355	0.9353	0.9352	0.9352	0.9353	0.9351	0.9948
	100	0.9957	0.9957	0.9956	0.9954	0.9954	0.9951	0.9955		100	0.9356	0.9356	0.9351	0.9353	0.9352	0.9353	0.9954
	120	0.9957	0.9957	0.9957	0.9956	0.9957	0.9955	0.9955		120	0.9353	0.9355	0.9352	0.9354	0.9353	0.9353	0.9958
	150	0.9958	0.9958	0.9957	0.9957	0.9957	0.9956	0.9956		150	0.9353	0.9354	0.9351	0.9353	0.9352	0.9351	0.9959
3	10	0.9847	0.9850	0.9852	0.9835	0.9840	0.9816	0.9845	8	10	0.9853	0.9855	0.9854	0.9855	0.9853	0.9828	0.9223
	20	0.9910	0.9912	0.9916	0.9902	0.9907	0.9888	0.9905		20	0.9929	0.9932	0.9934	0.9928	0.9931	0.9906	0.9400
	40	0.9927	0.9927	0.9928	0.9921	0.9923	0.9912	0.9919		40	0.9948	0.9948	0.9949	0.9943	0.9947	0.9931	0.9424
	60	0.9932	0.9931	0.9932	0.9927	0.9930	0.9922	0.9929		60	0.9953	0.9954	0.9954	0.9950	0.9952	0.9942	0.9441
	80	0.9935	0.9934	0.9934	0.9930	0.9932	0.9928	0.9930		80	0.9957	0.9958	0.9957	0.9953	0.9955	0.9950	0.9448
	100	0.9936	0.9936	0.9935	0.9933	0.9934	0.9931	0.9933		100	0.9959	0.9959	0.9959	0.9956	0.9958	0.9953	0.9450
	120	0.9937	0.9937	0.9936	0.9935	0.9936	0.9934	0.9936		120	0.9960	0.9960	0.9960	0.9958	0.9960	0.9958	0.9453
	150	0.9937	0.9937	0.9937	0.9936	0.9937	0.9936	0.9936		150	0.9961	0.9961	0.9961	0.9960	0.9961	0.9958	0.9458
4	10	0.9860	0.9858	0.9862	0.9862	0.9863	0.9840	0.9849	9	10	0.9250	0.9244	0.9293	0.9241	0.9256	0.9230	0.9268
	20	0.9914	0.9915	0.9915	0.9914	0.9916	0.9897	0.9894		20	0.9393	0.9397	0.9409	0.9358	0.9349	0.9342	0.9337
	40	0.9926	0.9925	0.9928	0.9924	0.9926	0.9915	0.9904		40	0.9444	0.9433	0.9442	0.9409	0.9406	0.9406	0.9348
	60	0.9928	0.9930	0.9931	0.9928	0.9931	0.9923	0.9914		60	0.9454	0.9447	0.9454	0.9426	0.9431	0.9417	0.9348
	80	0.9931	0.9931	0.9932	0.9930	0.9932	0.9927	0.9921		80	0.9458	0.9453	0.9461	0.9434	0.9438	0.9433	0.9350
	100	0.9933	0.9933	0.9933	0.9932	0.9933	0.9930	0.9927		100	0.9460	0.9457	0.9461	0.9447	0.9448	0.9435	0.9353
	120	0.9934	0.9935	0.9935	0.9934	0.9935	0.9932	0.9932		120	0.9461	0.9459	0.9461	0.9456	0.9454	0.9451	0.9351
	150	0.9935	0.9935	0.9934	0.9935	0.9935	0.9934	0.9933		150	0.9461	0.9459	0.9461	0.9458	0.9458	0.9457	0.9352
5	10	0.9890	0.9889	0.9890	0.9886	0.9887	0.9867	0.9879									
	20	0.9938	0.9937	0.9942	0.9934	0.9938	0.9925	0.9928									
	40	0.9951	0.9951	0.9953	0.9948	0.9951	0.9945	0.9937									
	60	0.9956	0.9956	0.9957	0.9953	0.9956	0.9949	0.9949									
	80	0.9959	0.9959	0.9960	0.9956	0.9957	0.9954	0.9954									
	100	0.9961	0.9961	0.9960	0.9959	0.9959	0.9957	0.9957									
	120	0.9962	0.9962	0.9962	0.9960	0.9961	0.9959	0.9959									
	150	0.9962	0.9962	0.9962	0.9962	0.9962	0.9961	0.9961									

where n is the number of runs, f_i indicates the fitness value of the i-th run, and $\bar{f}$ is the mean of the fitness value obtained within all the runs. Note that the lower value of Std metric refers to better stability.
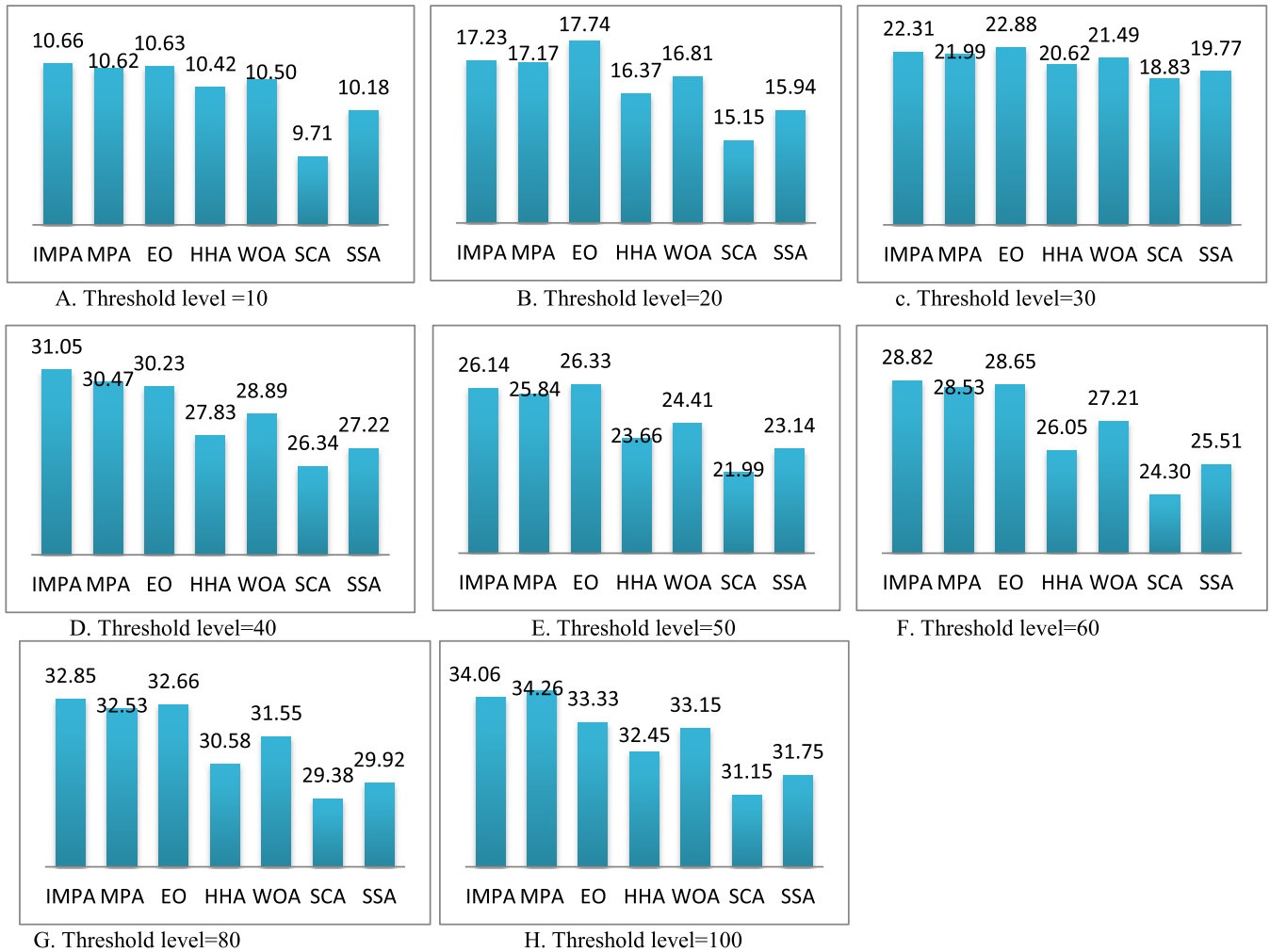
FIGURE 5. Average SNR values obtained by each algorithm for selected threshold levels of 10 to 100.

To check the stability of our proposed model, the average of Std values was calculated for each algorithm using 20 independent runs on all test images and all the threshold levels and introduced in Fig. 3, which shows that IMPA and MPA have lower Std values compared with the other algorithms investigated. As a result, both IMPA and MPA provide results with better consistency and stability.

C. PEAK SIGNAL TO NOISE RATIO (PSNR)

PSNR is an indicator used to evaluate the similarity of the predicted image with the original by calculating the ratio between the square of 255 and the mean square error between the original image and the predicted one. This metric can be calculated using Eq. 21 and Eq. 22.

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right) \qquad (21)$$

where *MSE* is the mean squared error which is calculated as follows:

$$MSE = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} |A(i,j) - S(i,j)|}{M * N} \qquad (22)$$

where $A(i,j)$, $S(i,j)$ represent the gray level of the predicted and original images, respectively. M, and N are the number of columns and rows of the image matrix. The greater value of the PSNR refers to a better quality of the predicted image. The average PSNR values obtained over 20 runs by each algorithm using Kapur's entropy are listed in Table 1, which shows that both IMPA and MPA have the best performance in 40 cases out of 72, while IMPA alone has the best performance in 31 cases. With small threshold levels, proposed IMPA algorithm is competitive with the EO algorithm. In contrast, the proposed algorithm presents the best PSNR values with an increase in the number of thresholds level. Based on this analysis, the proposed algorithm can determine the relevant threshold values for each image, especially for
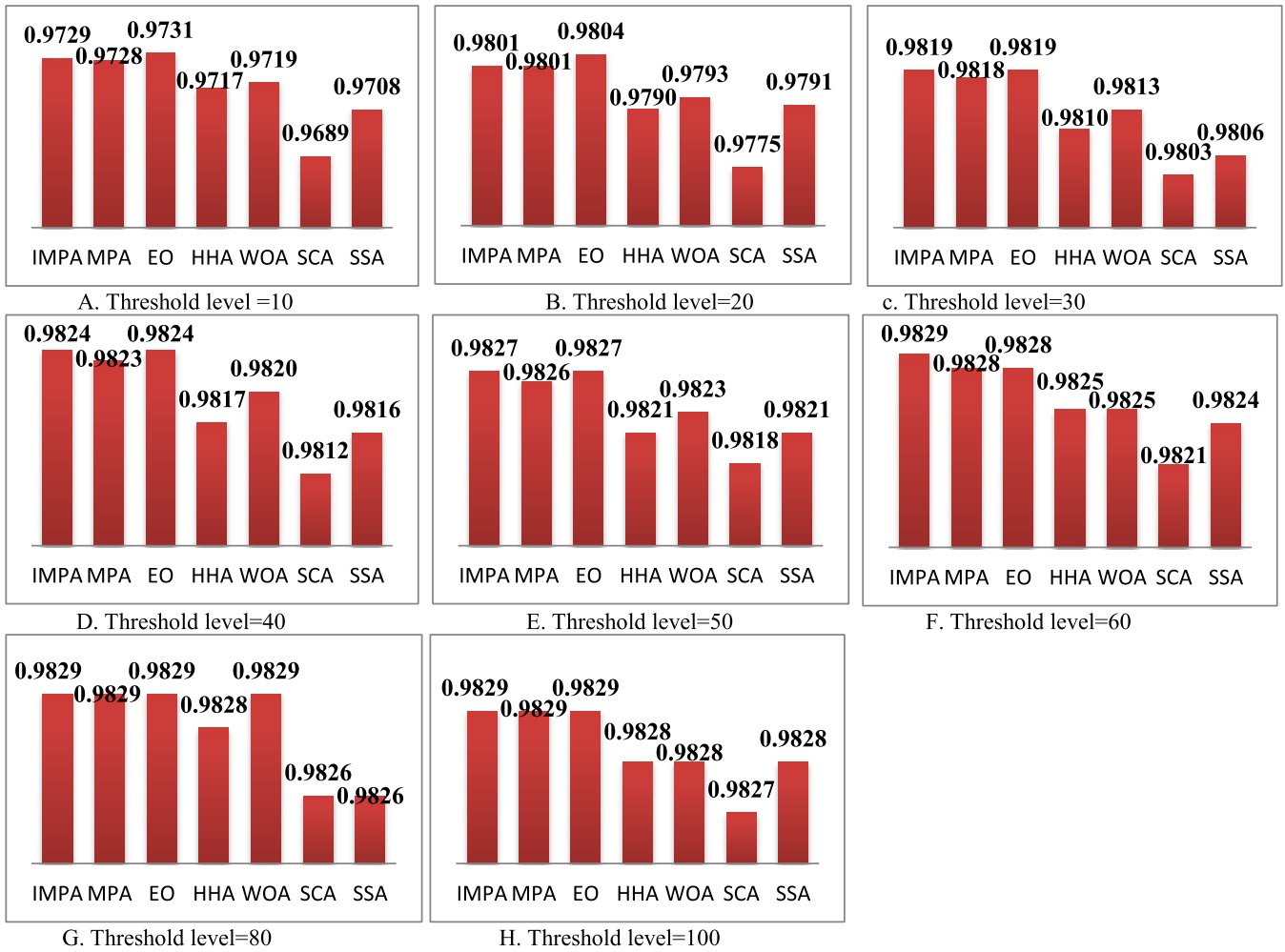
FIGURE 6. Average SSIM values obtained by each algorithm for selected threshold levels between 10 and 100.

the images with high threshold levels, and subsequently, the segmented image generated by IMPA is very close to the original. Fig. 4 shows the average of the PSNR values across 20 runs, from which it can be seen that the proposed IMPA algorithm has the best performance for high threshold levels, and its performance is competitive of EO and MPA for small threshold levels.

D. SIGNAL TO NOISE RATIO (SNR)

SNR [59] is the error summation method that is used to measure the quality of the predicted images by calculating the ratio of the error between the original and the segmented images, and is computed using the Eq. 23.

$$SNR = 10 \log_{10} \left(\frac{I^2}{SE^2} \right) \qquad (23)$$

where I is the average of the intensities of the original image and is calculated using Eq. 24.

$$I = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} X(i,j)}{M * N} \qquad (24)$$

and *SE* is the squared error and is calculated using Eq. 25.

$$SE = \sum_{i=1}^{M} \sum_{j=1}^{N} |X(i,j) - Y(i,j)| \qquad (25)$$

where $X(i,j)$, $Y(i,j)$ represent the original and the segmented images, respectively. Note that the higher value of *SNR* refers to better performance.

The average of SNR values obtained over 20 runs by each algorithm using Kapur's entropy are listed in Table 2, which shows that IMPA is competitive with EO for small threshold levels and is superior for high threshold levels, as shown in Fig.5.

E. STRUCTURED SIMILARITY INDEX METRIC (SSIM)

The SSIM [60] metric is used to calculate the difference between the structure of the segmented and original image, which takes into consideration the structure similarity, brightness, and contrast distortion between the original and segmented images. The mathematical model of SSIM is

TABLE 4. Average UQI values of each algorithm.

Im	T	IMPA	MPA	EO[57]	HHA [36]	WOA [31]	SCA [56]	SSA [58]	Im	T	IMPA	MPA	EO[57]	HHA [36]	WOA [31]	SCA [56]	SSA [58]
Ct1	10	0.9897	0.9896	0.9898	0.9895	0.9896	0.9875	0.9889	X6	10	0.9787	0.9810	0.9748	0.9668	0.9706	0.9623	0.9796
	20	0.9949	0.9949	0.9951	0.9947	0.9947	0.9935	0.9942		20	0.9923	0.9920	0.9922	0.9888	0.9901	0.9881	0.9914
	30	0.9963	0.9962	0.9964	0.9959	0.9961	0.9951	0.9958		30	0.9941	0.9938	0.9940	0.9916	0.9932	0.9909	0.9930
	40	0.9968	0.9968	0.9968	0.9965	0.9967	0.9962	0.9962		40	0.9948	0.9946	0.9947	0.9937	0.9939	0.9926	0.9940
	50	0.9970	0.9971	0.9970	0.9967	0.9969	0.9966	0.9968		50	0.9951	0.9949	0.9949	0.9943	0.9947	0.9937	0.9943
	60	0.9972	0.9972	0.9972	0.9970	0.9971	0.9968	0.9969		60	0.9953	0.9951	0.9951	0.9947	0.9948	0.9943	0.9947
	80	0.9973	0.9973	0.9973	0.9972	0.9972	0.9971	0.9971		80	0.9953	0.9953	0.9953	0.9950	0.9951	0.9947	0.9949
	100	0.9973	0.9973	0.9973	0.9973	0.9973	0.9972	0.9973		100	0.9954	0.9953	0.9953	0.9952	0.9953	0.9950	0.9952
Ct2	10	0.9906	0.9901	0.9903	0.9903	0.9903	0.9884	0.9887	X7	10	0.9313	0.9312	0.9312	0.9304	0.9309	0.9295	0.9294
	20	0.9949	0.9949	0.9955	0.9941	0.9948	0.9933	0.9940		20	0.9364	0.9362	0.9358	0.9352	0.9359	0.9352	0.9348
	30	0.9964	0.9966	0.9969	0.9961	0.9964	0.9956	0.9957		30	0.9371	0.9372	0.9365	0.9361	0.9367	0.9363	0.9364
	40	0.9973	0.9972	0.9972	0.9967	0.9970	0.9965	0.9967		40	0.9372	0.9370	0.9367	0.9364	0.9367	0.9364	0.9364
	50	0.9975	0.9974	0.9974	0.9971	0.9972	0.9967	0.9972		50	0.9372	0.9371	0.9368	0.9368	0.9367	0.9369	0.9364
	60	0.9977	0.9976	0.9975	0.9973	0.9975	0.9971	0.9973		60	0.9370	0.9370	0.9365	0.9368	0.9367	0.9367	0.9363
	80	0.9978	0.9977	0.9977	0.9976	0.9976	0.9974	0.9976		80	0.9368	0.9370	0.9366	0.9367	0.9367	0.9368	0.9365
	100	0.9978	0.9978	0.9978	0.9977	0.9977	0.9976	0.9977		100	0.9368	0.9367	0.9366	0.9368	0.9366	0.9368	0.9368
Ct3	10	0.9871	0.9869	0.9873	0.9858	0.9859	0.9823	0.9856	X8	10	0.9899	0.9895	0.9892	0.9894	0.9892	0.9865	0.9878
	20	0.9930	0.9932	0.9935	0.9923	0.9928	0.9913	0.9919		20	0.9965	0.9967	0.9967	0.9959	0.9966	0.9938	0.9948
	30	0.9947	0.9946	0.9949	0.9942	0.9944	0.9933	0.9937		30	0.9982	0.9981	0.9983	0.9977	0.9980	0.9961	0.9968
	40	0.9952	0.9952	0.9953	0.9947	0.9948	0.9939	0.9949		40	0.9989	0.9988	0.9989	0.9982	0.9985	0.9976	0.9976
	50	0.9955	0.9954	0.9954	0.9951	0.9952	0.9946	0.9953		50	0.9991	0.9991	0.9991	0.9987	0.9990	0.9982	0.9984
	60	0.9956	0.9956	0.9955	0.9953	0.9954	0.9951	0.9953		60	0.9993	0.9993	0.9993	0.9989	0.9992	0.9986	0.9987
	80	0.9957	0.9957	0.9957	0.9955	0.9956	0.9954	0.9956		80	0.9994	0.9994	0.9994	0.9993	0.9994	0.9991	0.9991
	100	0.9958	0.9957	0.9957	0.9956	0.9957	0.9956	0.9956		100	0.9995	0.9995	0.9994	0.9994	0.9995	0.9993	0.9993
Ct4	10	0.9880	0.9880	0.9881	0.9881	0.9883	0.9864	0.9870	X9	10	0.9270	0.9283	0.9309	0.9260	0.9285	0.9249	0.9264
	20	0.9934	0.9932	0.9935	0.9933	0.9936	0.9919	0.9913		20	0.9424	0.9409	0.9437	0.9372	0.9371	0.9340	0.9394
	30	0.9944	0.9945	0.9946	0.9944	0.9945	0.9935	0.9923		30	0.9468	0.9460	0.9469	0.9418	0.9437	0.9409	0.9455
	40	0.9948	0.9947	0.9950	0.9947	0.9949	0.9943	0.9932		40	0.9474	0.9474	0.9478	0.9450	0.9453	0.9449	0.9465
	50	0.9951	0.9951	0.9952	0.9950	0.9951	0.9946	0.9941		50	0.9482	0.9477	0.9481	0.9469	0.9467	0.9454	0.9465
	60	0.9952	0.9952	0.9953	0.9951	0.9952	0.9950	0.9944		60	0.9485	0.9482	0.9485	0.9468	0.9472	0.9466	0.9473
	80	0.9953	0.9954	0.9953	0.9953	0.9954	0.9952	0.9950		80	0.9483	0.9483	0.9483	0.9479	0.9479	0.9476	0.9476
	100	0.9954	0.9954	0.9954	0.9954	0.9954	0.9953	0.9951		100	0.9486	0.9484	0.9486	0.9482	0.9480	0.9479	0.9480
Ct5	10	0.9907	0.9907	0.9907	0.9902	0.9905	0.9878	0.9890									
	20	0.9955	0.9955	0.9959	0.9953	0.9954	0.9943	0.9946									
	30	0.9968	0.9967	0.9970	0.9965	0.9968	0.9957	0.9954									
	40	0.9974	0.9974	0.9974	0.9971	0.9973	0.9968	0.9968									
	50	0.9977	0.9977	0.9977	0.9975	0.9975	0.9972	0.9971									
	60	0.9978	0.9978	0.9978	0.9976	0.9977	0.9974	0.9974									
	80	0.9979	0.9979	0.9979	0.9978	0.9979	0.9977	0.9978									
	100	0.9980	0.9980	0.9979	0.9979	0.9979	0.9979	0.9979									

TABLE 5. Fitness values of each algorithm.

Im	T	IMPA	MPA	EO	HHA	WOA [31]	SCA [56]	SSA [58]	Im	T	IMPA	MPA	EO	HHA	WOA [31]	SCA [56]	SSA [58]
Ct1	10	33.3836	33.3791	33.3788	33.3673	33.3835	33.0613	33.3116	X6	10	31.5893	31.5929	31.5752	31.2919	31.3899	30.7541	31.1677
	20	50.6666	50.7408	50.6118	50.0839	50.3998	48.6168	50.0565		20	47.2966	47.0362	47.3958	46.4763	46.8973	44.6209	46.2646
	30	63.2219	63.0729	62.7617	61.4903	62.3863	58.8067	61.9757		30	58.0215	57.7706	58.1852	56.3265	57.2249	52.6965	54.8890
	40	72.0141	71.8148	71.4438	69.0027	70.7450	65.1804	69.7906		40	65.2540	64.5915	65.0492	62.6099	63.7088	58.9034	60.8165
	50	78.1178	77.6420	76.5053	74.3391	75.7645	69.7310	75.0080		50	69.9040	68.8916	69.4750	66.7810	68.2375	61.9954	65.2348
	60	81.7290	81.4173	80.1712	77.8921	78.7307	73.6387	77.0840		60	72.4115	71.7330	71.7199	68.7843	70.1651	64.5805	67.4993
	80	84.3752	82.7900	82.9483	80.6207	82.9251	76.2387	80.0392		80	73.8612	72.4551	72.9647	71.1998	72.8018	67.3230	69.2689
	100	83.8191	82.5376	82.3885	81.5434	83.3713	77.9723	80.9316		100	72.8533	71.4457	71.2847	71.3912	73.0903	67.5624	69.5892
Ct2	10	32.9948	33.0503	33.0207	32.9895	33.0003	32.5761	32.8997	X7	10	31.1518	31.1570	31.1576	31.0972	31.1250	30.6794	30.9437
	20	50.7444	50.7180	50.4685	50.0280	50.3551	45.8961	50.1893		20	48.0638	48.0734	48.1347	47.6678	47.9632	45.9222	47.0744
	30	62.9013	62.6455	62.7304	61.4564	62.3250	58.7272	61.5795		30	59.6533	59.3953	59.8302	58.0685	59.2733	54.7545	57.3738
	40	71.8339	71.3278	71.1538	68.9371	70.0454	65.2889	69.3460		40	67.5255	66.7608	67.6199	65.3957	66.5310	61.2539	63.7609
	50	78.0826	77.3373	76.7628	73.8218	75.8373	69.8954	74.5778		50	72.4357	72.0532	72.5209	69.0589	70.7139	64.7523	68.6399
	60	81.9169	80.3250	80.3818	77.3270	78.8840	73.3069	78.0291		60	75.5314	74.7870	75.5839	72.4266	73.6599	67.5709	70.5475
	80	84.1614	83.3880	82.0502	80.5783	82.9959	76.6019	79.9523		80	77.3389	76.1367	77.0588	74.0920	76.4432	70.4026	72.8325
	100	84.1750	83.0583	82.3519	81.1241	84.2952	77.9796	80.2345		100	76.7677	75.3350	75.8684	74.4560	77.2512	71.1332	73.0065
Ct3	10	32.8796	32.9977	32.9013	32.7559	32.7881	32.3223	32.5762	X8	10	33.1172	33.1965	33.1940	33.1845	33.1962	32.8727	33.1530
	20	50.0643	50.0544	50.0320	49.4083	49.6320	47.6843	49.5135		20	50.5809	50.6196	50.6264	50.3357	50.5401	48.6413	49.8881
	30	62.3648	62.2672	62.3457	60.6139	61.3919	57.8846	60.7238		30	62.8081	62.9281	62.3849	61.7474	62.5851	58.5744	61.5774
	40	70.9823	70.8978	70.3592	68.3540	69.6190	64.9980	68.6979		40	71.8516	71.7460	71.0376	69.2304	70.9029	65.5907	69.8655
	50	77.4592	76.5634	75.6136	72.9337	75.2009	69.1971	73.8070		50	77.9587	77.5786	76.2312	73.8804	75.9637	70.2503	74.5393
	60	81.4128	80.1103	79.4793	77.0907	78.2720	72.9562	76.9950		60	82.0902	80.8981	79.7576	77.7272	79.9423	73.2119	78.3120
	80	83.6163	83.0158	82.1438	79.6195	82.7296	76.4800	79.4819		80	84.4951	82.9751	82.2213	80.8753	83.8355	76.5397	80.4860
	100	83.6217	81.6677	81.9086	81.0315	83.0729	77.4564	79.7233		100	84.1934	82.5087	82.6473	81.9399	84.8653	78.0873	80.1949
Ct4	10	34.0596	34.0604	34.0698	34.0556	34.0642	33.7030	33.9965	X9	10	30.2267	30.2082	30.2673	30.1879	30.2103	29.8635	30.1062
	20	51.3501	51.3218	51.3354	51.1577	51.3971	49.4443	50.7032		20	46.8422	46.6803	46.8562	45.7739	45.9310	44.1466	45.8979
	30	63.7392	63.5213	63.6234	62.4526	63.4963	59.3442	62.3605		30	57.9428	57.7300	58.1393	56.0265	56.6398	52.9713	56.3791
	40	72.4528	72.1003	72.2635	70.1696	71.9924	66.0286	70.6134		40	65.1922	65.1086	65.3392	62.1173	63.3764	58.7390	62.7586
	50	78.8880	78.3060	77.0876	74.9167	76.9559	70.7051	75.3911		50	70.0605	69.5722	70.3854	65.6610	66.7122	62.5229	66.1647
	60	82.5116	81.9368	80.5221	78.2999	80.7298	74.2779	78.2886		60	72.1979	71.4663	72.6608	68.3612	69.7424	65.1107	68.0328
	80	85.1952	83.9329	84.0062	82.4015	84.3660	77.1946	81.2387		80	73.4339	72.1343	73.7919	71.0006	72.2233	67.3730	69.5737
	100	85.0031	83.0850	81.2806	82.3355	85.0956	78.4746	81.0291		100	73.1347	71.2715	72.9829	71.1972	72.1598	67.8850	69.4808
Ct5	10	33.1003	33.0785	33.0879	33.0808	33.0409	32.6580	32.9509									
	20	50.5406	50.4425	50.3396	49.9905	50.3133	48.4944	50.0702									
	30	62.8457	62.6845	62.4136	61.4570	62.2537	58.6242	61.7189									
	40	71.6470	71.6542	71.0002	68.4977	70.0888	65.2492	69.7716									
	50	77.9361	77.4284	77.1878	73.6602	74.6824	69.7864	74.7814									
	60	81.6466	80.8228	80.4929	77.2242	78.5282	73.1465	77.3013									
	80	83.8592	82.8844	82.6728	80.4211	82.0528	76.3083	80.1077									
	100	83.6500	82.4052	81.1981	81.1517	83.4398	77.5154	80.2067									

formulated as in Eq.26.

$$SSIM(O, S) = \frac{(2\mu_o \mu_s + a)(2\sigma_{os} + b)}{(\mu_o^2 + \mu_s^2 + a)(\sigma_o^2 + \sigma_s^2 + b)} \quad (26)$$

where μ_o, μ_s are the mean intensities of the original and segmented image; σ_o and σ_s are the standard deviation of the original and segmented image; σ_{os} is the co-variance between the predicted and original image; and a and b are constant

FIGURE 7. Average UQI of each algorithm on selected threshold levels from 10 to 100.

values equal to 0.001 and 0.003 respectively. A higher value of SSIM indicates better results.

The average SSIM values obtained over 20 runs by each compared algorithm using Kapur's entropy are listed in Table 3, from which it can be identified that both IMPA and MPA are competitive with EO for both small and high thresholds levels. Fig. 6 shows the average of the SSIM values over 20 runs.

F. UNIVERSAL QUALITY INDEX (UQI)

UQI [61] is an indicator used to measure the quality of the segmented image based on three factors: loss of correlation, brightness, and contrast distortion instead of the error summation between the original and segmented. The mathematical model of UQI is formulated as in Eq. 27.

$$\text{UQI}(O, S) = \frac{(4\sigma_{\text{os}}\mu_{\text{o}}\mu_{\text{s}})}{\left(\mu_{\text{o}}^2 + \mu_{\text{s}}^2\right)\left(\sigma_{\text{o}}^2 + \sigma_{\text{s}}^2\right)} \qquad (27)$$

O, and S refer to the original and segmented images, μ_o, μ_s are the mean intensities of the original and segmented image; σ_o and σ_s are the standard deviation of the original and

segmented image; σ_{os} is the co-variance between the predicted and original image. A higher value of UQI indicates better results.

The average UQI values obtained over 20 runs by each algorithm using Kapur's entropy are listed in Table 4, which shows that both IMPA outperforms all the other algorithms in 26 of 72 cases, while achieves the same values as EO in 15 cases. Meanwhile, MPA outperforms both EO and IMPA in 2 cases of 72. Further, EO outperforms our proposed IMPA in 19 cases of 72. The proposed IMPA therefore achieves high quality for the segmented images especially for the images with the upper threshold levels. Fig. 7 introduces the average of the UQI values obtained over 20 run at each threshold level.

G. FITNESS VALUES USING KAPUR'S ENTROPY

Table 5 shows the average of the fitness values across 20 runs obtained by each algorithm using Kapur's entropy. It can be seen that both IMPA and MPA outperform the other algorithms in 55 cases of 72, while IMPA alone could outperform in 50 cases of the 72, presenting the best fitness
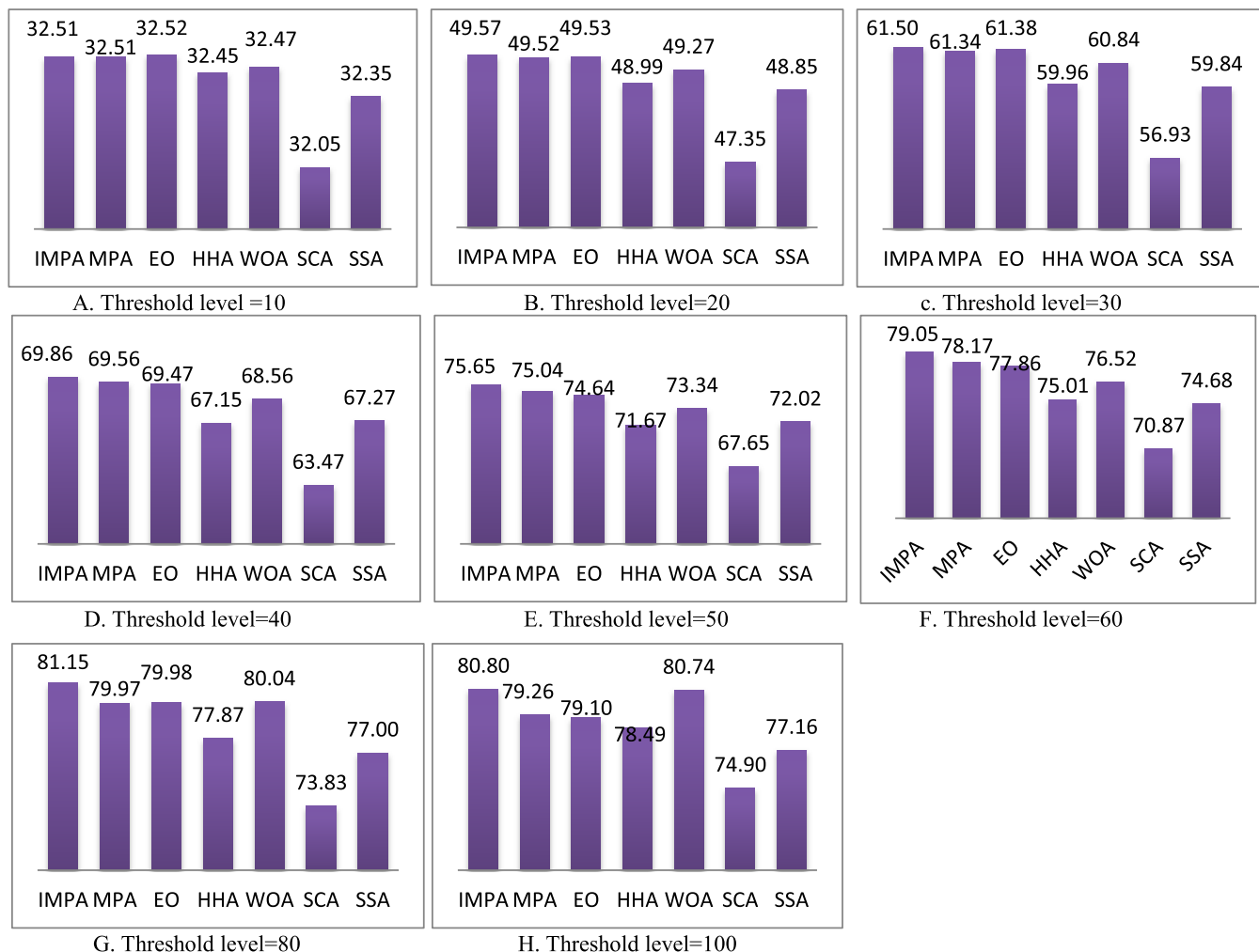
FIGURE 8. Average fitness values of each algorithm on selected threshold levels from 10 to 100.
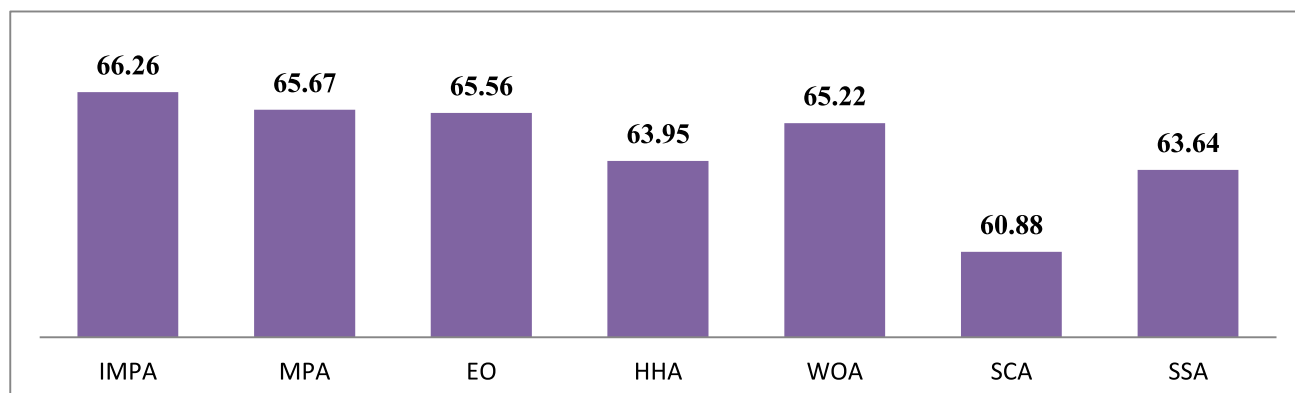
FIGURE 9. Average fitness values obtained by each algorithm on all threshold levels (10 to 100).

values with all threshold levels in most cases. Fig. 8 shows the average of the fitness values within 20 times obtained by each algorithm using Kapur's entropy for selected threshold levels from 10 to 100. Fig. 9 presents the average across 20 runs of Kapur's entropy for all thresholds levels, from which it can be seen that the proposed IMPA algorithm outperforms all other algorithms investigated.

H. CONVERGENCE RATE

The convergence toward the best solution is illustrated in Fig.10; at the outset of iterations, MPA has high exploration capabilities, so the convergence rate toward the best solution is low compared with the other algorithms, as shown in Fig.10. After that, at the intermediate phase of the optimization process specifically between maximum iterations
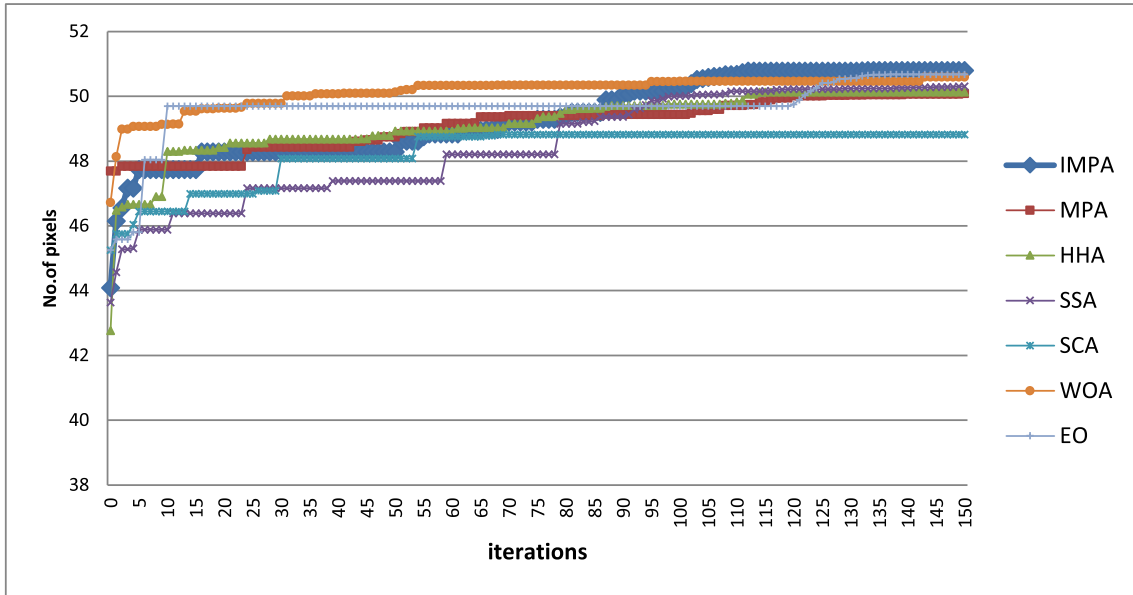
a. Threshold level=20
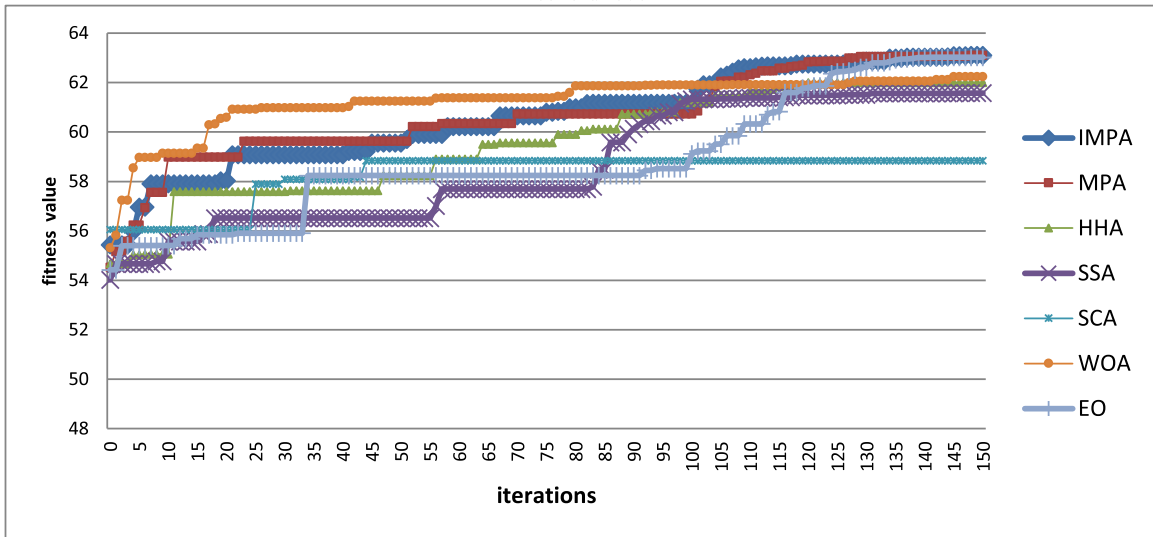
b. Threshold level=30
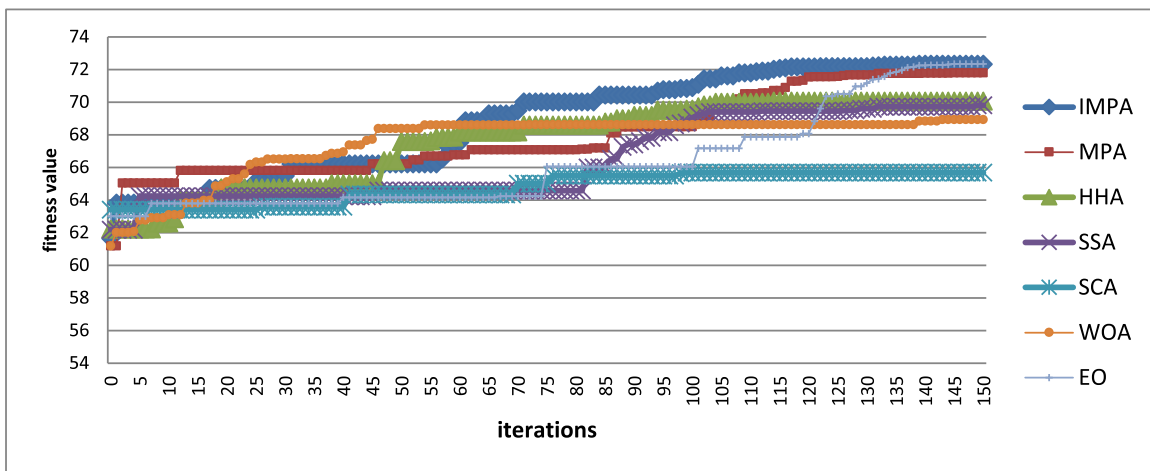
c. Threshold level=40

FIGURE 10. Convergence rate towards the best value obtained by each algorithm using Kapur's entropy.
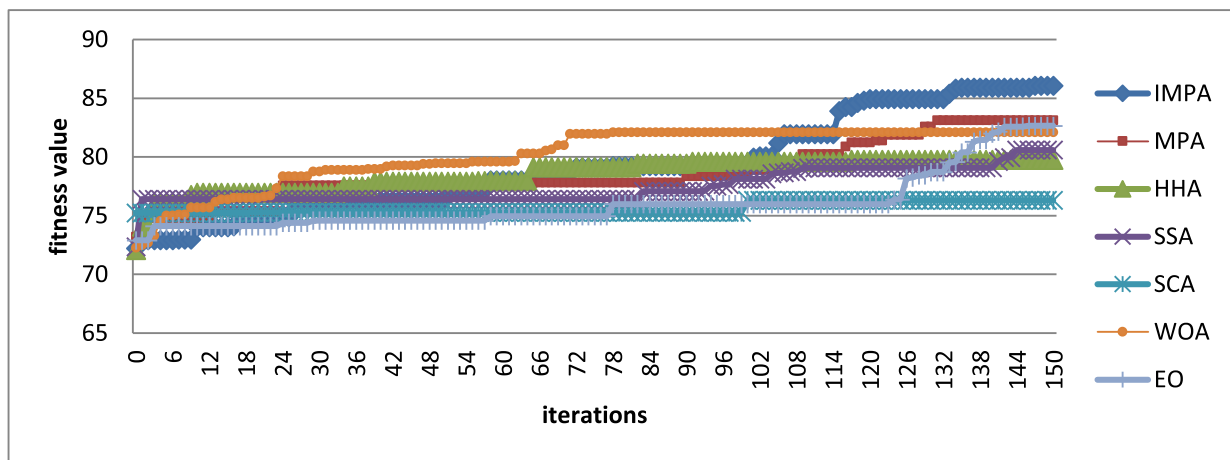
d. Threshold level=80
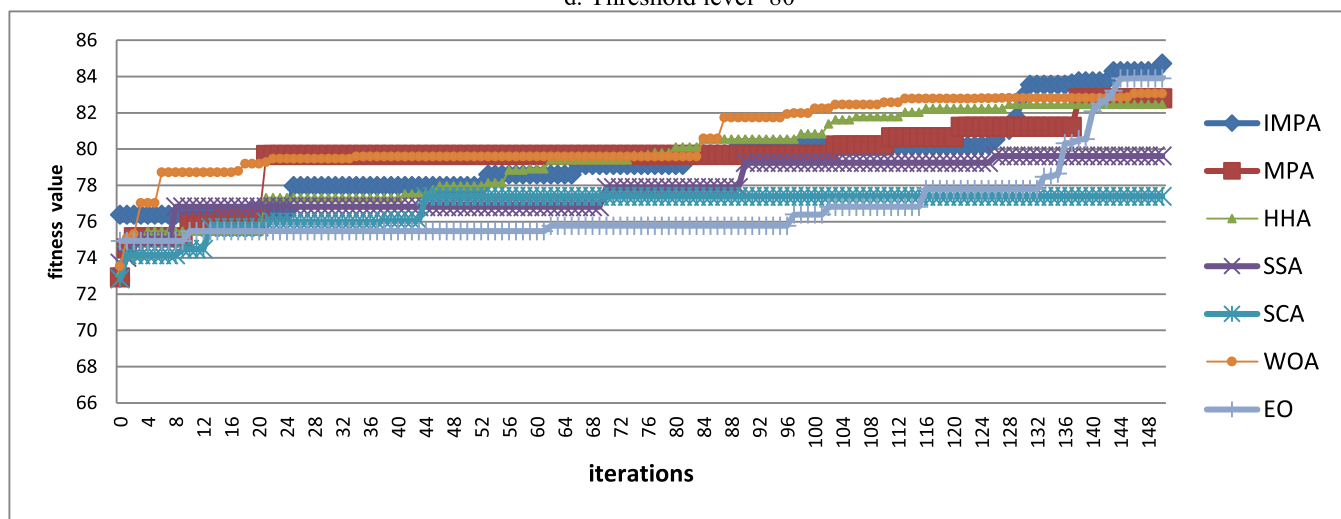
e. Threshold level=100

FIGURE 10. *(Continued.)* **Convergence rate towards the best value obtained by each algorithm using Kapur's entropy.**

and maximum iterations, MPA is between the exploration and exploitation operators, where it divides the population into two parts: the first part will move using the exploration operator and while the second will be moved using the exploitation operator. So in this case, MPA moves faster toward the best solution, and the convergence rate increases, this is illustrated in Fig.10 at the half of the iterations. In the final stage, all the prey would be moved with the exploitation step, so the convergence rate increases significantly towards the best solution.

However, MPA still suffers from low convergence due to spending many iterations in exploration, so RDR is used to help IMPA to achieve a high convergence rate toward the optimal solution as shown in Fig.10. Further, IMPA can outperform all the other algorithms in convergence rate for all threshold levels, especially for high threshold levels. In Figure 10, the convergence rate is shown for all algorithms for the threshold levels 20, 30, 40, 80, and 100. For threshold level 20, WOA has a higher convergence rate, but after 100 iterations, the performance of WOA drops, while IMPA increases significantly. For threshold level 20, MPA couldn't

outperform WOA. For threshold levels 30, 40, 80, and 100, IMPA, and MPA could outperform all the other algorithms in convergence rate during the second half of iterations.
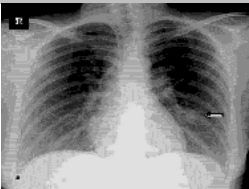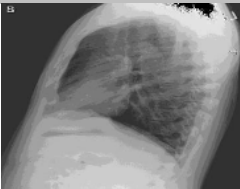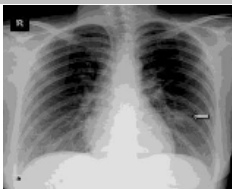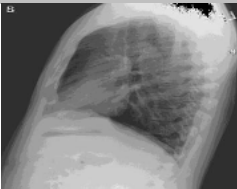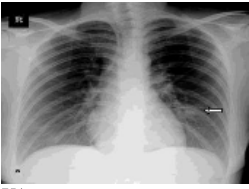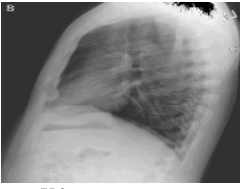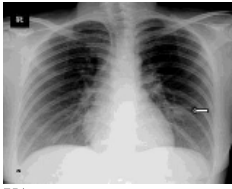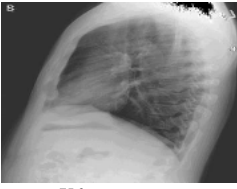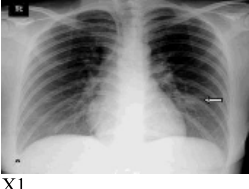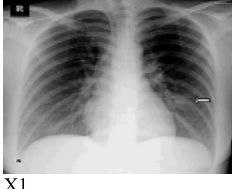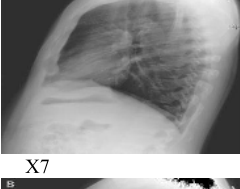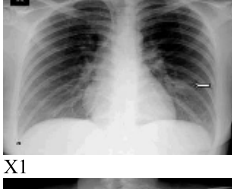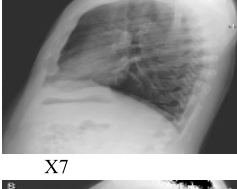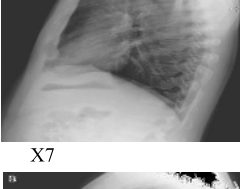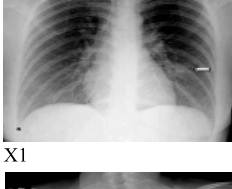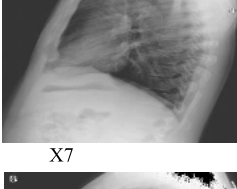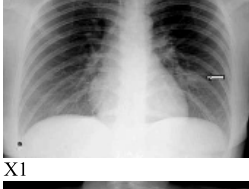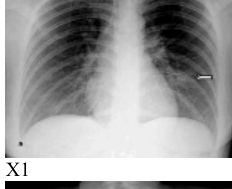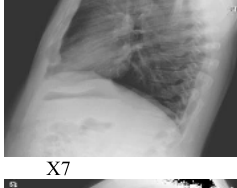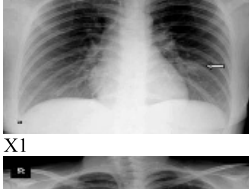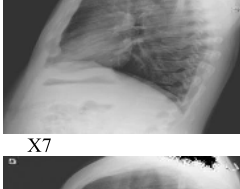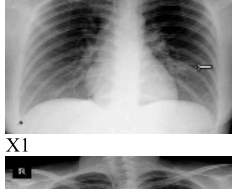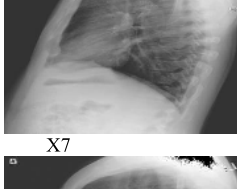
I. SEGMENTED IMAGES OF THE PROPOSED MODEL

This section shows a graphical comparison between MPA and IMPA to illustrate better the performance improvement. Table 6 shows the segmented images obtained by the proposed IMPA algorithm and MPA. All the results of the performance metrics discussed before confirm that IMPA could produce higher quality segmented images than MPA. As a result, the segmented images produced by IMPA, and introduced in Table 6 is better than the images produced by MPA, and introduced also in Table 6 It is noticeable in Table 6 that IMPA outperforms MPA for all threshold levels.

VI. CONCLUSION AND FUTURE WORK

In this paper, we proposed a new hybrid model to detect the COVOD-19 using an improved marine predators algorithm (IMPA) and a ranking-based diversity reduction (RDR) strategy to obtain the number of particles that can't find a

TABLE 6. The segmented images obtained by the proposed IMPA algorithm.

Threshold level	Segmented images			
	IMPA		MPA	
10	X1	X6	X1	X6
20	X1	X6	X1	X6
30	X1	X7	X1	X7
40	X1	X7	X1	X7
50	X1	X7	X1	X7
60	X1	X7	X1	X7
80	X1	X7	X1	X7
100	X1	X7	X1	X7

better solution within a consecutive number of iterations. Our model works on the x-ray images to extract similar small regions, in an attempt to obtain the regions that may contain COVID-19. Extracting these regions can be treated as an image segmentation problem. The performance of our proposed IMPA algorithm was compared with five state-of-art algorithms—whale optimization algorithm (WOA), sine-cosine algorithm (SCA), salp swarm algorithm (SSA), Harris hawks algorithm (HHA), and Equilibrium optimizer (EO)—using a set of chest X-Ray images with threshold levels between 10 and 100. The performance of our proposed IMPA algorithm is shown to outperform all other investigated algorithms in the fitness values, Std, and a range of threshold metrics. In addition, the performance of our proposed model and EO was shown to be convergent on all the thresholds level in SSIM and UQI metrics.

In the future work, the proposed algorithm can be applied to color image segmentation and different medical applications.

REFERENCES

[1] H. Liu, F. Liu, J. Li, T. Zhang, D. Wang, and W. Lan, "Clinical and CT imaging features of the COVID-19 pneumonia: Focus on pregnant women and children," *J. Infection*, vol. 80, no. 5, pp. e7–e13, May 2020.

[2] C. S. Guan, Z. B. Lv, S. Yan, Y. N. Du, H. Chen, L. G. Wei, R. M. Xie, and B. D. Chen, "Imaging features of coronavirus disease 2019 (COVID-19): Evaluation on thin-section CT," *Acad. Radiol.*, vol. 27, no. 5, pp. 609–613, May 2020.

[3] J. Kuruvilla, D. Sukumaran, A. Sankar, and S P Joy, "A review on image processing and image segmentation," in *Proc. Int. Conf. Data Mining Adv. Comput. (SAPIENCE)*, Mar. 2016, pp. 198–203.

[4] R. Hu, M. Rohrbach, S. Venugopalan, and T. Darrell, "Utilizing large scale vision and text datasets for image segmentation from referring expressions," 2016, *arXiv:1608.08305*. [Online]. Available: http://arxiv.org/abs/1608.08305

[5] M. Mittal, "Image segmentation using deep learning techniques in medical images," in *Proc. Advancement Mach. Intell. Interact. Med. Image Anal.* Singapore: Springer, 2020, pp. 41–63.

[6] Z. Zhang, C. Wu, S. Coleman, and D. Kerr, "DENSE-INception U-net for medical image segmentation," *Comput. Methods Programs Biomed.*, vol. 192, Aug. 2020, Art. no. 105395.

[7] X. Wang and X. D. M. Wang Wilkes, "An efficient image segmentation algorithm for object recognition using spectral clustering," in *Proc. Mach. Learning-based Natural Scene Recognit. Mobile Robot Localization Unknown Environ.* Singapore: Springer, 2020, pp. 215–234.

[8] C. G. Karydas, "Optimization of multi-scale segmentation of satellite imagery using fractal geometry," *Int. J. Remote Sens.*, vol. 41, no. 8, pp. 2905–2933, Dec. 2019.

[9] T. Su and S. Zhang, "Local and global evaluation for remote sensing image segmentation," *ISPRS J. Photogramm. Remote Sens.*, vol. 130, pp. 256–276, Aug. 2017.

[10] M. Alberti, M. Seuret, V. Pondenkandath, R. Ingold, and M. Liwicki, "Historical document image segmentation with LDA-initialized deep neural networks," in *Proc. 4th Int. Workshop Historical Document Imag. Process. (HIP)*, 2017, pp. 95–100.

[11] A. Naoum, J. Nothman, and J. Curran, "Article segmentation in digitised newspapers with a 2D Markov model," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 1007–1014.

[12] R. Barman, M. Ehrmann, S. Clematide, S. Ares Oliveira, and F. Kaplan, "Combining visual and textual features for semantic segmentation of historical newspapers," 2020, *arXiv:2002.06144*. [Online]. Available: http://arxiv.org/abs/2002.06144

[13] A. Aksac, T. Ozyer, and R. Alhajj, "Complex networks driven salient region detection based on superpixel segmentation," *Pattern Recognit.*, vol. 66, pp. 268–279, Jun. 2017.

[14] P. Prathusha and S. Jyothi, "A Novel edge detection algorithm for fast and efficient image segmentation," in *Data Engineering and Intelligent Computing*. Singapore: Springer, 2018, pp. 283–291.

[15] B. N. Narayanan, R. C. Hardie, T. M. Kebede, and M. J. Sprague, "Optimized feature selection-based clustering approach for computer-aided detection of lung nodules in different modalities," *Pattern Anal. Appl.*, vol. 22, no. 2, pp. 559–571, Oct. 2017.

[16] J. Han, C. Yang, X. Zhou, and W. Gui, "A new multi-threshold image segmentation approach using state transition algorithm," *Appl. Math. Model.*, vol. 44, pp. 588–601, Apr. 2017.

[17] D. Oliva, E. Cuevas, G. Pajares, D. Zaldivar, and V. Osuna, "A multilevel thresholding algorithm using electromagnetism optimization," *Neurocomputing*, vol. 139, pp. 357–381, Sep. 2014.

[18] S. Arora, J. Acharya, A. Verma, and P. K. Panigrahi, "Multilevel thresholding for image segmentation through a fast statistical recursive algorithm," *Pattern Recognit. Lett.*, vol. 29, no. 2, pp. 119–125, Jan. 2008.

[19] A. Dirami, K. Hammouche, M. Diaf, and P. Siarry, "Fast multilevel thresholding for image segmentation through a multiphase level set method," *Signal Process.*, vol. 93, no. 1, pp. 139–153, Jan. 2013.

[20] J. N. Kapur, P. K. Sahoo, and A. K. C. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," *Comput. Vis., Graph., Image Process.*, vol. 29, no. 1, p. 140, Jan. 1985.

[21] D. Oliva and M. A. S. Elaziz Hinojosa, "Fuzzy entropy approaches for image segmentation," in *Metaheuristic Algorithms for Image Segmentation: Theory Applications*. Springer, 2019, pp. 141–147.

[22] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.

[23] M. Abdel-Basset, G. Manogaran, D. El-Shahat, and S. Mirjalili, "A hybrid whale optimization algorithm based on local search strategy for the permutation flow shop scheduling problem," *Future Gener. Comput. Syst.*, vol. 85, pp. 129–145, Aug. 2018.

[24] G. I. Sayed, A. E. Hassanien, and A. T. Azar, "Feature selection via a novel chaotic crow search algorithm," *Neural Comput. Appl.*, vol. 31, no. 1, pp. 171–188, Apr. 2017.

[25] R. M. Rizk-Allah, A. E. Hassanien, M. Elhoseny, and M. Gunasekaran, "A new binary salp swarm algorithm: Development and application for optimization tasks," *Neural Comput. Appl.*, vol. 31, no. 5, pp. 1641–1663, Jul. 2018.

[26] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "A new genetic algorithm for solving optimization problems," *Eng. Appl. Artif. Intell.*, vol. 27, pp. 57–69, Jan. 2014.

[27] C. Guo and H. Li, "Multilevel thresholding method for image segmentation based on an adaptive particle swarm optimization algorithm," in *Proc. Australas. Joint Conf. Artif. Intell.* Berlin, Germany: Springer, 2007, pp 654–658.

[28] L. Xiong, G. Tang, Y.-C. Chen, Y.-X. Hu, and R.-S. Chen, "Color disease spot image segmentation algorithm based on chaotic particle swarm optimization and FCM," *J. Supercomput.*, pp. 1–15, Jan. 2020, doi: 10.1007/s11227-020-03171-8.

[29] Di Martino, F. and S. Sessa, "PSO image thresholding on images compressed via fuzzy transforms," *Inf. Sci.*, vol. 506, pp. 308–324, Jan. 2020.

[30] A. Kaveh and S. Talatahari, "An improved ant colony optimization for constrained engineering design problems," *Eng. Comput.*, vol. 27, no. 1, pp. 155–182, Jan. 2010.

[31] M. A. E. Aziz, A. A. Ewees, and A. E. Hassanien, "Whale optimization algorithm and moth-flame optimization for multilevel thresholding image segmentation," *Expert Syst. Appl.*, vol. 83, pp. 242–256, Oct. 2017.

[32] M.-H. Horng, "Multilevel minimum cross entropy threshold selection based on the honey bee mating optimization," *Expert Syst. Appl.*, vol. 37, no. 6, pp. 4580–4592, Jun. 2010.

[33] P. Kandhway and A. K. Bhandari, "Spatial context cross entropy function based multilevel image segmentation using multi-verse optimizer," *Multimedia Tools Appl.*, vol. 78, no. 16, pp. 22613–22641, Apr. 2019.

[34] S. Agrawal, R. Panda, S. Bhuyan, and B. K. Panigrahi, "Tsallis entropy based optimal multilevel thresholding using cuckoo search algorithm," *Swarm Evol. Comput.*, vol. 11, pp. 16–30, Aug. 2013.

[35] F. Chakraborty, D. Nandi, and P. K. Roy, "Oppositional symbiotic organisms search optimization for multilevel thresholding of color image," *Appl. Soft Comput.*, vol. 82, Sep. 2019, Art. no. 105577.

[36] X. Bao, H. Jia, and C. Lang, "A novel hybrid Harris hawks optimization for color image multilevel thresholding segmentation," *IEEE Access*, vol. 7, pp. 76529–76546, 2019.

[37] R. Wang, Y. Zhou, C. Zhao, and H. Wu, "A hybrid flower pollination algorithm based modified randomized location for multi-threshold medical image segmentation," *Bio-Med. Mater. Eng.*, vol. 26, no. s1, pp. S1345–S1351, Aug. 2015.

[38] D. Oliva, S. Hinojosa, E. Cuevas, G. Pajares, O. Avalos, and J. Gálvez, "Cross entropy based thresholding for magnetic resonance brain images using crow search algorithm," *Expert Syst. Appl.*, vol. 79, pp. 164–180, Aug. 2017.

[39] X. Yao, Z. Li, L. Liu, and X. Cheng, "Multi-threshold image segmentation based on improved grey wolf optimization algorithm," *IOP Conf., Earth Environ. Sci.*, vol. 252, Jul. 2019, Art. no. 042105.

[40] F. Huo, X. Sun, and W. Ren, "Multilevel image threshold segmentation using an improved bloch quantum artificial bee colony algorithm," *Multimedia Tools Appl.*, vol. 79, nos. 3–4, pp. 2447–2471, Nov. 2019.

[41] E. Cuevas and F. A. Fausto González, "Locust search algorithm applied to multi-threshold segmentation," in *New Advancements Swarm Algorithms: Operators Applications*. Cham, Switzerland: Springer, 2020, pp. 211–240.

[42] Erdmann, H., "A study of a firefly meta-heuristics for multithreshold image segmentation," in *Developments in Medical Image Processing and Computational Vision*. Cham, Switzerland: Springer, 2015, pp. 279–295.

[43] A. Singla and S. Patra, "A fast automatic optimal threshold selection technique for image segmentation," *Signal, Image Video Process.*, vol. 11, no. 2, pp. 243–250, Jul. 2016.

[44] S. Manikandan, K. Ramar, M. W. Iruthayarajan, and K. G. Srinivasagan, "Multilevel thresholding for segmentation of medical brain images using real coded genetic algorithm," *Measurement*, vol. 47, pp. 558–568, Jan. 2014.

[45] M. Maitra and A. Chatterjee, "A hybrid cooperative–comprehensive learning based PSO algorithm for image segmentation using multi-level thresholding," *Expert Syst. Appl.*, vol. 34, no. 2, pp. 1341–1350, Feb. 2008.

[46] Y. Liu, C. Mu, W. Kou, and J. Liu, "Modified particle swarm optimization-based multilevel thresholding for image segmentation," *Soft Comput.*, vol. 19, no. 5, pp. 1311–1327, Jun. 2014.

[47] P. Ghamisi, M. S. Couceiro, F. M. L. Martins, and J. A. Benediktsson, "Multilevel image segmentation based on fractional-order darwinian particle swarm optimization," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2382–2394, May 2014.

[48] K. Chen, Y. Zhou, Z. Zhang, M. Dai, Y. Chao, and J. Shi, "Multilevel image segmentation based on an improved firefly algorithm," *Math. Problems Eng.*, vol. 2016, pp. 1–12, Feb. 2016.

[49] A. K. Bhandari, A. Kumar, and G. K. Singh, "Modified artificial bee colony based computationally efficient multilevel thresholding for satellite image segmentation using Kapur's, Otsu and Tsallis functions," *Expert Syst. Appl.*, vol. 42, no. 3, pp. 1573–1601, Feb. 2015.

[50] N. Sanyal, A. Chatterjee, and S. Munshi, "An adaptive bacterial foraging algorithm for fuzzy entropy based image segmentation," *Expert Syst. Appl.*, vol. 38, no. 12, pp. 15489–15498, Nov. 2011.

[51] P. D. Sathya and R. Kayalvizhi, "Modified bacterial foraging algorithm based multilevel thresholding for image segmentation," *Eng. Appl. Artif. Intell.*, vol. 24, no. 4, pp. 595–615, Jun. 2011.

[52] K. Tang, X. Xiao, J. Wu, J. Yang, and L. Luo, "An improved multilevel thresholding approach based modified bacterial foraging optimization," *Int. J. Speech Technol.*, vol. 46, no. 1, pp. 214–226, Aug. 2016.

[53] A. Mostafa, A. E. Hassanien, M. Houseni, and H. Hefny, "Liver segmentation in MRI images based on whale optimization algorithm," *Multimedia Tools Appl.*, vol. 76, no. 23, pp. 24931–24954, Apr. 2017.

[54] E. Aziz, A. A. Ewees, A. E. Hassanien, M. Mudhsh, and S. Xiong, "Multi-objective whale optimization algorithm for multilevel thresholding segmentation," in *Proc. Adv. Soft Comput. Mach. Learn. Image Process.* Cham, Switzerland: Springer, 2018, pp. 23–39.

[55] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine predators algorithm: A nature-inspired Metaheuristic," *Expert Syst. Appl.*, vol. 152, Aug. 2020, Art. no. 113377.

[56] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowl.-Based Syst.*, vol. 96, pp. 120–133, Mar. 2016.

[57] M. Abdel-Basset, V. Chang, and R. Mohamed, "A novel equilibrium optimization algorithm for multi-thresholding image segmentation problems," *Neural Comput. Appl.*, pp. 1–34, Mar. 2020, doi: 10.1007/s00521-020-04820-y.

[58] S. Wang, H. Jia, X. Peng, "Modified salp swarm algorithm based multilevel thresholding for color image segmentation," *Math. Biosciences Eng.*, vol. 17, no. 1, pp. 700–724, 2020.

[59] V. Rawat, A. jain, and V. shrimali, "Investigation and assessment of disorder of ultrasound B-mode images," 2010, *arXiv:1003.1827*. [Online]. Available: http://arxiv.org/abs/1003.1827

[60] A. Hore and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *Proc. 20th Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 2366–2369.

[61] K. Egiazarian, "New full-reference quality metrics based on HVS," in *Proc. 2nd Int. Workshop Video Process. Quality Metrics*, 2006, pp. 1–4.

MOHAMED ABDEL-BASSET received the B.Sc., M.Sc., and Ph.D. degrees in operations research from the Faculty of Computers and Informatics, Zagazig University, Egypt. He is an Associate Professor with the Faculty of Computers and Informatics, Zagazig University. He has published more than 200 articles in international journals and conference proceedings. His current research interests are optimization, operations research, data mining, computational intelligence, applied statistics, decision support systems, robust optimization, engineering optimization, multiobjective optimization, swarm intelligence, evolutionary algorithms, and artificial neural networks. He is working on the application of multiobjective and robust metaheuristic optimization techniques. He is also an/a Editor/Reviewer in different international journals and conferences.

REDA MOHAMED received the B.Sc. degree from the Department of Computer Science, Faculty of Computers and Informatics, Zagazig University, Egypt. His research interests include robust optimization, multiobjective optimization, swarm intelligence, evolutionary algorithms, and artificial neural networks. He is working on the application of multiobjective and robust meta-heuristic optimization techniques in computational intelligence.

MOHAMED ELHOSENY is currently an Assistant Professor with the Faculty of Computers and Information, Mansoura University, where he is also the Director of the Distributed Sensing and Intelligent Systems Laboratory. He has been appointed as an ACM Distinguished Speaker, from 2019 to 2022. Collectively, he has authored/coauthored over 85 ISI journal articles in high-ranked and prestigious journals, such as the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, the IEEE TRANSACTIONS ON RELIABILITY, *Future Generation Computer Systems* (Elsevier), and *Neural Computing and Applications* (Springer). He has authored/edited conference proceedings, book chapters, and ten books published by Springer and Taylor & Francis. His research interests include smart cities, network security, artificial intelligence, the Internet of Things, and intelligent systems.

RIPON K. CHAKRABORTTY received the B.Sc. and M.Sc. degrees in industrial and production engineering and the Ph.D. degree in computer science from the Bangladesh University of Engineering and Technology, in 2009, 2013, and 2017, respectively. He is a Lecturer in system engineering and project management with the School of Engineering and Information Technology, University of New South Wales (UNSW) at Canberra, Australia. He has written two book chapters and over 45 technical journal articles and conference papers. His research interests include a wide range of topics in operations research, optimization problems, project management, supply chain management, and information systems management.

MICHAEL RYAN is the Director of the Capability Systems Centre, University of New South Wales at Canberra. He lectures and regularly consults in a range of subjects, including communications systems, systems engineering, requirements engineering, and project management. He is the Co-Chair of the Requirements Working Group, International Council on Systems Engineering (INCOSE). He is a Fellow of Engineers Australia, the International Council on Systems Engineering, and the Institute of Managers and Leaders. He is the author or coauthor of 12 books, three book chapters, and over 250 technical articles and reports.

● ● ●