Journal of Bionic Engineering (2023) 20:2276–2316
https://doi.org/10.1007/s42235-023-00391-5

RESEARCH ARTICLE

An Efficient Multilevel Threshold Image Segmentation Method for COVID-19 Imaging Using Q-Learning Based Golden Jackal Optimization

Zihao Wang[1] · Yuanbin Mo[1,2] · Mingyue Cui[1]

Received: 6 February 2023 / Revised: 23 April 2023 / Accepted: 26 April 2023 / Published online: 14 June 2023

Abstract

From the end of 2019 until now, the Coronavirus Disease 2019 (COVID-19) has been rampaging around the world, posing a great threat to people's lives and health, as well as a serious impact on economic development. Considering the severely infectious nature of COVID-19, the diagnosis of COVID-19 has become crucial. Identification through the use of Computed Tomography (CT) images is an efficient and quick means. Therefore, scientific researchers have proposed numerous segmentation methods to improve the diagnosis of CT images. In this paper, we propose a reinforcement learning-based golden jackal optimization algorithm, which is named QLGJO, to segment CT images in furtherance of the diagnosis of COVID-19. Reinforcement learning is combined for the first time with meta-heuristics in segmentation problem. This strategy can effectively overcome the disadvantage that the original algorithm tends to fall into local optimum. In addition, one hybrid model and three different mutation strategies were applied to the update part of the algorithm in order to enrich the diversity of the population. Two experiments were carried out to test the performance of the proposed algorithm. First, compare QLGJO with other advanced meta-heuristics using the IEEE CEC2022 benchmark functions. Secondly, QLGJO was experimentally evaluated on CT images of COVID-19 using the Otsu method and compared with several well-known meta-heuristics. It is shown that QLGJO is very competitive in benchmark function and image segmentation experiments compared with other advanced meta-heuristics. Furthermore, the source code of the QLGJO is publicly available at https://github.com/Vang-z/QLGJO.

Keywords COVID-19 · Bionic algorithm · Golden jackal optimization · Image segmentation · Otsu and Kapur method

1 Introduction

COVID-19 was an unprecedented human pandemic in recent decades [1, 2]. The first case of COVID-19 was reported in Wuhan, China, in December 2019, but there is no conclusive evidence as far as its origin is concerned [3]. Due to its severe transmissibility, 600 million infections have accumulated worldwide to date, with irreversible effects on the global economy and human lives [4]. Cough, fever, sore throat, headache, and physical fatigue are all symptoms of COVID-19 [5]. In addition, COVID-19 can cause fibrosis of the lung tissue and lead to various complications that seriously threaten the health of patients [6]. Therefore, according to the above, healthcare providers need to have the means to detect this disease in time to prevent and interrupt the spread of the epidemic. The most common assay is Polymerase Chain Reaction (PCR), which is frequently used by medical institutions all over the world [7]. However, this work is time-consuming, costly, and most importantly, it has the potential for false-positive results. Considering that COVID-19 causes fibrosis in patients' lung tissues, chest CT has become a more reliable means of detection [8]. CT images can clearly reflect the internal anatomy of the body tissues and also the nature of

✉ Yuanbin Mo
moyuanbin2020@gxmzu.edu.cn

Zihao Wang
vang-z@stu.gxmzu.edu.cn

Mingyue Cui
202120812001865@stu.gxmzu.edu.cn

[1] School of Artificial Intelligence, Guangxi Minzu University, Nanning, China

[2] Guangxi Key Laboratory of Hybrid Computation and IC Design Analysis, Guangxi Minzu University, Nanning 530006, China

the lesions, which is a great help for the clinical counterpart treatment. The reasonable segmentation of CT images can further improve the efficiency of disease diagnosis by medical personnel. Therefore, image segmentation has become a common technique in medical imaging [9].

Segmentation is a key step in image processing and it plays an indispensable role in different fields such as remote sensing [10], feature selection [11], computer vision [12], medical imaging [13], and cryptography [14], etc. The main goal of image segmentation is to find a suitable threshold to segment an image into multiple regions containing the same features based on texture, color, brightness, or contrast. There are numerous approaches to solving the segmentation problem, such as edge-based [15], region-based [16], threshold-based [17], and feature clustering-based [18]. However, according to a large literature summary, the threshold-based segmentation method relies on its simplicity and efficiency to be the preferred solution [19–21]. Threshold segmentation is divided into two types in terms of broad categories: one is bi-level threshold segmentation and the other is multi-level threshold segmentation. Bi-level threshold segmentation is the simplest segmentation method and is now well established. It segments an image region into two classes by determining an optimal threshold value. On the other hand, multilevel thresholding is used to segment an image into several different parts by maximizing or minimizing the objective function for a given number of thresholds to determine the threshold value. The most common multilevel thresholding methods include Fuzzy entropy [9], Kapur's entropy [22], Tsallis entropy [23], and Otsu method [24]. Although multilevel thresholding is theoretically an upgraded version of bi-level threshold segmentation, it is constrained by the practical problem that as the number of thresholds increases and the image size becomes larger, it becomes impractical to rely on the computing power to exhaust every possible solution. Therefore, more and more scholars are using meta-heuristics instead of traditional mathematical computational methods to reduce the time cost in complex image processing problems.

In recent decades, meta-heuristics have made a splash in industry with their simple and efficient performance, especially for a variety of challenging optimization problems. Many researchers have worked to improve the performance of meta-heuristics, and a large number of meta-heuristics have emerged during this period. The most common meta-heuristics for multilevel threshold image segmentation problems include: Particle Swarm Optimization (PSO) [25], Differential Evolution (DE) [26], Grey Wolf Optimizer (GWO) [27], Monarch Butterfly Optimization (MBO) [28], Whale Optimization Algorithm (WOA) [29], Multi-Verse Optimizer (MVO) [30], Harris Hawk Optimization (HHO) [31], Black Widow Optimization Algorithm (BWOA) [32], Slime Mould Algorithm (SMA) [33], Hunger Games Search (HGS) [34], RUNge Kutta Optimizer (RUN) [35], Marine Predators Algorithm (MPA) [36], Weighted Mean of Vectors (INFO) [37], and Rime Optimization Algorithm (RIME) [38] etc. In addition, the recently proposed Golden Jackal Optimization (GJO) [39] has also shown superior performance. For the GJO, Rezaie et al. [40] used the GJO algorithm to solve the model parameter estimation problem for fuel cells, and the experimental results showed that the proposed improved method outperformed other classical algorithms in terms of optimal model estimation. Houssein et al. [41] introduced the opposition-based learning mechanism into the GJO which is named IGJO. The experimental results show that IGJO outperforms WOA, SOS, SSA, HHO, GTO, and MPA in multilevel threshold segmentation. Zhang et al. [42] proposed an enhanced GJO algorithm and used it to solve the adaptive infinite impulse response system identification problem, and the experimental results showed that the enhanced GJO algorithm was able to obtain higher computational accuracy in this problem. By investigating the above literature, we can observe that, due to its flexibility and versatility, the GJO has received much attention from scholars in various fields since it was proposed, and has solved various complex engineering problems. However, these literatures also reflect the disadvantages of GJO. Firstly, the convergence speed is slow in high-dimensional complex problems. Then, GJO is easily influenced by the initial population. At last, like most meta-heuristics, it is easy to fall into local optimum.

The following is a review of some recent outstanding contributions in the field of image segmentation using meta-heuristics. Huo, Sun, and Ren [43] used the improved Bloch Quantum Artificial Bee Colony Algorithm for multilevel thresholding of images and verified the comprehensive performance of the algorithm on grayscale images. In [44], Xing used Gaussian mutation, Lévy flight, and opposition-based learning to improve the Emperor Penguin Optimization, and the performance of the improved algorithm was enhanced for multilevel thresholding segmentation of color images. Upadhyay et al. [45] used the Crow Search Algorithm to optimize the objective function for computing Kapur entropy, and experimented on different threshold values of 2, 4, 8, 16, and 32, which obtained superior results. Elaziz et al. [46] proposed a hybrid algorithm to determine the optimal threshold for threshold segmentation, which combines the Whale Optimization Algorithm and the Volleyball Premier League Algorithm. The experimental results show that the proposed algorithm outperforms other methods in several performance metrics such as Peak Signal to Noise Ratio and Structural Similarity Index. Meanwhile, Elaziz et al. [47] improves the Marine Predators Algorithm using quantum theory, which greatly enhances the global search capability of the algorithm. Better results were obtained in the multilevel threshold segmentation task. Houssein et al. [12] proposed an image segmentation method based on the

Black Widow Optimization Algorithm, the Kapur entropy and the Otsu method were used as the objective function, respectively. The experimental results show that the proposed algorithm has more reliable performance compared with the classical algorithm. Meanwhile, the application of meta-heuristics in the field of image segmentation is investigated in more depth by Houssein et al. in the following literature. In [48], a local escaping operator is combined with the Tunicate Swarm Algorithm to propose a method for global optimization and image segmentation. In [49], the opposition-based learning mechanism is introduced for the Marine Predators Algorithm. The proposed algorithm is applied to the multilevel threshold segmentation problem. Experimental results show that the proposed algorithm outperforms other algorithms being compared in terms of performance. In [50], an effective multilevel thresholding segmentation method is proposed using an improved Equilibrium Optimizer, which is used for segmentation of medical images. The experimental results show that the proposed method can effectively solve the segmentation problem of medical images. In [51], the Salp Swarm Algorithm is combined with the Marine Predators Algorithm to determine the optimal threshold for the multilevel thresholding segmentation problem. In [13], the Chimp Optimization Algorithm is enhanced based on opposition-based learning and Lévy flight, and the improved algorithm outperforms other algorithms in multilevel threshold segmentation. Liu et al. [52] proposed a novel Ant Colony Optimization for COVID-19 image segmentation, and the experimental results show that the algorithm can further improve the diagnosis of COVID-19. Bhandari [53] proposed an algorithm using two objective functions of a multilevel threshold segmentation method based on Beta-difference Evolution, which can retrieve the best threshold accurately and efficiently. Wu et al. [54] proposed an improved Teaching–learning-based Optimization Algorithm for the multilevel threshold segmentation problem. The experimental results show that the algorithm can segment high-resolution X-ray images perfectly. He and Huang [55] proposed a Krill Herd Algorithm for solving multilevel thresholding segmentation of color images by using the Otsu method, Kapur entropy, and Tsallis entropy as the objective function. The experimental results show that the proposed algorithm has more accurate and stable performance on Kapur's entropy. Ren et al. [56] improved the differential evolution (DE) and proposed an algorithm called MDE. This algorithm can improve the convergence accuracy and the ability to leap out of local optimum to some extent. And the segmentation experiments were conducted on breast cancer and skin cancer pathology images, and the experimental results illustrate that the proposed method can provide an efficient segmentation procedure for pathology medical images. Hosny et al. [57] combine Coronavirus Optimization Algorithm (COVIDOA) and Harris Hawks Optimization Algorithm (HHOA) to solve the segmentation problem, and the proposed method compensates the shortcomings of COVIDOA and HHOA mutually to some extent, and the improved performance of the proposed algorithm over the COVIDOA and HHOA algorithms was demonstrated by five test problems in the IEEE CEC 2019 benchmark problem, and the quality of the segmented images of the proposed algorithm was better than the other methods in the segmentation experiments. Zhu et al. [58] proposed an improved WOA with Levy operator and chaotic random variation strategy to improve the ability of the algorithm to jump out of the local optimum and explore the search space, and the proposed method has excellent performance in both benchmark test sets and image segmentation experiments compared with other variants of WOA. Emam et al. [59] proposed an improved RSA algorithm by integrating Reptile Search Algorithm (RSA) and RUN algorithm, which introduced the ESQ mechanism of RUN into RSA, improved the convergence speed and the ability to jump out of the local optimal. Through CEC2020 benchmark test set and brain magnetic resonance imaging segmentation experiments, it is proved that the proposed algorithm has strong optimization ability. Han et al. [60] used an improved MVO to maximize the Kapur's entropy, and the experimental results proved that the proposed method is highly competitive with other meta-heuristics for benchmark functions and image segmentation experiments. Xing et al. [61] improved WOA by introducing a Quasi-Opposition-Based Learning and Gaussian Barebone Mechanism, and the improved method beat other algorithms in the image segmentation experiments. According to the above literature, we can further understand that the meta-heuristics has been widely used in image segmentation. However, at present, there are still many shortcomings, most meta-heuristics converge slowly and easy to fall into local optimality. Therefore, we need to further improve the meta-heuristic algorithm to improve the efficiency of the algorithm to deal with problems.

In recent years, many researchers have combined machine learning methods and meta-heuristics to improve the performance of meta-heuristics. Among them, combining methods such as Support Vector Machine (SVM) [62], Self-Organizing Maps (SOM) [63], and Reinforcement Learning (RL) [64] with meta-heuristics has achieved amazing success and provided a new direction for the development of meta-heuristics. In addition, RL has become more popular among research scholars in recent studies [65]. RL enables an agent to learn autonomously by interacting with the environment, and the combination with metaheuristics often leads to better performance. For example, Qu et al. [66] combined Grey Wolf Optimizer with RL and succeeded in achieving unexpected success in the UAV 3D path planning problem. In [67], the authors used Policy Iteration (PI) and Grey Wolf Optimizer to train the Neural Networks (NNs). Experimental results show that the proposed method has a better solution. Sadeg et al. [68] proposed a Reinforcement Learning based Bee Swarm Optimization for feature selection. Experimental results show that the proposed algorithm can yield satisfactory results for large instances. Chen et al. [69] introduced a method for solving the flexible job-shop scheduling problem, combining RL with the Genetic Algorithm for the first time, and achieved satisfactory results in this problem.

As mentioned above, at this present, meta-heuristics have been widely used in the field of image segmentation, and the combination of RL and meta-heuristics has been recognized by a wide range of researchers. However, as far as we know, there is a lack of research that combines RL with meta-heuristics to solve image segmentation. Therefore, in view of the problem that golden jackal optimization algorithm is easy to fall into local optimality, in this study, we propose a reinforcement learning-based improved golden jackal optimization to advance the research of medical image segmentation. The main contributions of this paper are as follows:

(1) QLGJO: An enhanced version of the GJO based on reinforcement learning (Q-Learning), which is named QLGJO, is proposed to be used to advance the research of CT images of COVID-19.
(2) Three mutation strategies are proposed to improve the exploration performance of GJO. In addition, a new update strategy is introduced in the original algorithm to further balance the exploration and exploitation.
(3) The performance comparison experiment of QLGJO and other advanced meta-heuristics were conducted on IEEE CEC2022. The experimental data reveal that the performance of QLGJO is better than the others.
(4) The Otsu method (maximum interclass variance method) is used as the objective function, and it is optimized using QLGJO.
(5) Peak Signal to Noise Ratio ($PSNR$) [70], Structural Similarity Index ($SSIM$) [71], and Feature Similarity Index ($FSIM$) [72] are used as metrics for the segmentation experiments to verify the effectiveness of the different algorithms.
(6) The performance of the proposed method was evaluated with six different meta-heuristics at thresholds of 8, 12, 16, and 20. Experimental results show that the proposed method has superior advantages and can be further extended to other classes of medical imaging diagnostics.

The rest of this paper is organized as follows: Sect. 2 contains the materials and methods. Section 3 suggest the

Algorithm 1 Pseudo-code of the GJO.

```
Inputs: The population size $N$, the max iterations $T$, the threshold number $L$, and the image $I_{in}$.
Outputs: The best solution, which contains the best thresholds and the image after segmentation.
Initialization the population $\boldsymbol{Pos}$ with threshold $L$ according Eq.(1).
while $(t < T)$
    Calculating the fitness using Otsu method.
    $\overrightarrow{\boldsymbol{Pos_m}}(t) = best(\boldsymbol{Pos}(t))$
    $\overrightarrow{\boldsymbol{Pos_{fm}}}(t) = second\_best(\boldsymbol{Pos}(t))$
    foreach $(\boldsymbol{Pos}(t))$
        Calculate the Evasion Energy $(\vec{\boldsymbol{E}})$ according Eq.(4).
        if $(|\vec{\boldsymbol{E}}| \geq 1)$
            Update the population according Eq.(2), Eq.(3), and Eq.(7).
        else
            Update the population according Eq.(8), Eq.(9), and Eq.(7).
        end if
    end foreach
    $t = t + 1$
end while
return the best solution.
```

QLGJO algorithm. Section 4 introduces, discusses and analyzes the performance of the proposed method. At last, Sect. 5 summarizes the study and provides suggestions for future work.

2 Preliminaries

In this section, we introduced the main framework of the GJO algorithm and some basic concepts of reinforcement learning. In addition, the objective function used in this study and the dataset of COVID-19 are also described.

2.1 Golden Jackal Optimization

GJO was proposed by Chopra and Ansari [39] in 2022 as a meta-heuristic based on swarm intelligence. Compared with other meta-heuristics, GJO provides a fresh strategy to solve optimization problems. The GJO is inspired by the collaborative hunting behavior of golden jackals, which hunt in pairs. Generally, the male jackal leads the female jackal in hunting, first finding and approaching the prey, then surrounding and chasing the prey, and finally hunting the prey. GJO simulates the hunting process of the golden jackal in two phases: first, the search phase, which involves finding and tracking the prey. Next is the exploitation phase, in which the golden jackal surrounds the prey and hunts it. The mathematical description of GJO will be discussed in the following subsections.

2.1.1 Initialization

As mentioned above, GJO is a meta-heuristic based on swarm intelligence. Therefore, the initialization process of GJO is similar to most meta-heuristics. Equation (1) depicts the initialization process of the GJO.

$$\begin{aligned} \overrightarrow{Pos_k} &= \overrightarrow{LB} + \overrightarrow{rand} \circ \left(\overrightarrow{UB} - \overrightarrow{LB}\right),\ k = 1, 2, \dots, n \\ Pos &= \left[\overrightarrow{Pos_1}, \overrightarrow{Pos_2}, \dots, \overrightarrow{Pos_n}\right]^T \end{aligned} \tag{1}$$

where $\boldsymbol{Pos}$ represents the positions of all individuals in the population, $\vec{Pos}_k$ stands for the position of the k th individual, n is the population size, $\vec{LB}$ and $\vec{\boldsymbol{UB}}$ denote the upper and lower boundaries of the environment, respectively. $\boldsymbol{rand}$ is a n-dimensional random vector between 0 and 1. Moreover, the symbol $\circ$ indicates the Hadamard product [73].

2.1.2 Exploration

In the exploration phase, male golden jackals lead female golden jackals to find and track their prey. The positions of the male and female golden jackets are shown by Eq. (2) and Eq. (3), respectively. The position update scheme for individual k is shown in Eq. (7).

$$\overrightarrow{\boldsymbol{Pos}_1}(t) = \overrightarrow{\boldsymbol{Pos}_m}(t) - \overrightarrow{\boldsymbol{E}} \circ \left|\overrightarrow{\boldsymbol{Pos}_m}(t) - \overrightarrow{\boldsymbol{RL}} \circ \overrightarrow{\boldsymbol{Pos}_k}(t)\right| \tag{2}$$

$$\overrightarrow{\boldsymbol{Pos}_2}(t) = \overrightarrow{\boldsymbol{Pos}_{fm}}(t) - \overrightarrow{\boldsymbol{E}} \circ \left|\overrightarrow{\boldsymbol{Pos}_{fm}}(t) - \overrightarrow{\boldsymbol{RL}} \circ \overrightarrow{\boldsymbol{Pos}_k}(t)\right| \tag{3}$$

where t is the current iteration, $\vec{\boldsymbol{Pos}}_k(t)$ stands for the position of the t th iteration for the k th individual. $\vec{\boldsymbol{Pos}}_m(t)$ and $\vec{\boldsymbol{Pos}}_{fm}(t)$ denote the male and female jackals of the current iteration, respectively, which are best and second-best individuals of the population. $\vec{\boldsymbol{RL}}$ is a n-dimensional random vector which is based on Lévy flight [74], which is discussed

Algorithm 2 Pseudo-code of the Q-Learning

```
Inputs: The max iterations $T$.
Outputs: The Q-table.
Initialization the Q-table.
Choose a random initial state $s_t$.
while ($t < T$)
    Choose the best action $a_t$ with the current state $s_t$ according Q-table.
    Verify the reward value based on action $a_t$.
    Find the new state $s_{t+1}$.
    Update the Q-table according Eq.(10), and update the state $s_t \leftarrow s_{t+1}$
end while
return the Q-table.
```

in detail in [36]. $\vec{E}$ is the Evading Energy of individual which is calculated as Eq. (4), Eq. (5), and Eq. (6).

$$\vec{\boldsymbol{E}} = E_1 \cdot \vec{\boldsymbol{E}}_{\mathbf{0}} \tag{4}$$

$$E_1 = 1.5 \times (1 - t/T) \tag{5}$$

$$\overrightarrow{\boldsymbol{E}_{\mathbf{0}}} = 2 \cdot \overrightarrow{\boldsymbol{rand}} - 1 \tag{6}$$

where E_1 stands for the decreasing energy of the individual, $\vec{\boldsymbol{E}}_0$ denotes the initial value of the decreasing energy. T is the maximum number of the iterations.

$$\overrightarrow{\boldsymbol{Pos}_{\boldsymbol{k}}}(t+1) = \frac{\overrightarrow{\boldsymbol{Pos}_{\mathbf{1}}}(t) + \overrightarrow{\boldsymbol{Pos}_{\mathbf{2}}}(t)}{2} \tag{7}$$

2.1.3 Exploitation

In the exploitation phase, the golden jackal attacks the prey tracked in the exploration phase, and when the evading energy of the prey decays to 0, the golden jackal will hunt the prey. The mathematical model of this phase is similar to that of the exploration phase, which is shown below.

$$\overrightarrow{\boldsymbol{Pos}_{\mathbf{1}}}(t) = \overrightarrow{\boldsymbol{Pos}_{\boldsymbol{m}}}(t) - \overrightarrow{\boldsymbol{E}} \circ \left| \overrightarrow{\boldsymbol{RL}} \circ \overrightarrow{\boldsymbol{Pos}_{\boldsymbol{m}}}(t) - \overrightarrow{\boldsymbol{Pos}_{\boldsymbol{k}}}(t) \right| \tag{8}$$

$$\overrightarrow{\boldsymbol{Pos}_{\mathbf{2}}}(t) = \overrightarrow{\boldsymbol{Pos}_{\boldsymbol{fm}}}(t) - \overrightarrow{\boldsymbol{E}} \circ \left| \overrightarrow{\boldsymbol{RL}} \circ \overrightarrow{\boldsymbol{Pos}_{\boldsymbol{fm}}}(t) - \overrightarrow{\boldsymbol{Pos}_{\boldsymbol{k}}}(t) \right| \tag{9}$$

Please note that all the definitions in this section have the same meanings as those mentioned in the previous section. Therefore, it will not be repeated any further. And the pseudo-code of the GJO is given in Algorithm 1.

2.2 Reinforcement Learning

Reinforcement Learning (RL), also referred to as reactive learning, evaluative learning, or augmented learning, which is one of the paradigms and methodologies of machine learning [75]. RL is a learning mechanism that learns how to map from states to actions in order to maximize the obtained reward. As scientific research continues to advance, RL can currently be divided into two main categories: policy-based methods and value-based methods. While policy-based methods do not include value functions, the opposite is true for value-based methods. A typical representative of value-based methods is the Q-Learning algorithm. This algorithm uses a matrix called the Q-table to record the Q-values for different states (S). The Q-table is randomly initialized, and before each iteration, the algorithm selects the best action to maximize the Q-value among all actions (A) for state (S) to obtain the best reward. In addition, Eq. (10) shows the Bellman Equation [76], which is the specific update formula for the Q-table.

$$Q_{t+1}(s_t, a_t) \leftarrow Q_t(s_t, a_t) + \lambda[r_{t+1} + \gamma \max(Q_t(s_{t+1}, a)) - Q_t(s_t, a_t)] \tag{10}$$

where s_t and s_{t+1} represent the current state and the next state, respectively. λ stands for the learning rate, and γ is the discount factor, which all of them are between 0 and 1. r_{t+1} denotes the reward or penalty which is the agent receives depending on the current action. $Q_t(s_t, a_t)$ indicates the Q-value of the selected action in the current state. $\max(Q_t(s_t, a))$ represents the maximum Q-value of all actions in the current state. Finally, $Q_{t+1}(s_t, a_t)$ is the Q-value which is pre-estimated for the next state. Therefore, three components are required to solve the problem with Q-Learning: the reward table (R-table), the Q-table, and the Bellman Equation. In addition, the pseudo-code of the Q-Learning is shown in Algorithm 2.

2.3 Thresholding Methods

2.3.1 Otsu's Methods

The Otsu method was proposed in 1979 [77], which segmented images by maximizing variance between classes. In other words, the Otsu method is a nonparametric segmentation method that divides an image into different regions based on the intensity of the pixels. Assume that L is the pixel intensity levels of an image which has size of $M \times N$.

$$n = n_0 + n_1 + \ldots + n_{L-1} \tag{11}$$

$$Ph_i = \frac{n_i}{n}, \sum_{i=0}^{L-1} Ph_i = 1 \tag{12}$$

where n indicates the total number of image pixels, n_i is the number of pixels for intensity level i, and probability distribution of all intensity levels is represented by Ph_i.

Assume there is a threshold th, in which th is between 0 and $L-1$, then the image can be divided into two classes according to th. The first class, C_1, contains all pixels with pixel intensity levels between $[0, th]$, while C_2 contains the rest of the pixels.

$$\omega_1(th) = \sum_{i=0}^{th} Ph_i,\ \omega_2(th) = \sum_{i=th+1}^{L-1} Ph_i = 1 - Ph_1(th) \tag{13}$$

where $\omega_1(th)$ and $\omega_2(th)$ represent the cumulative probability distributions for C_1 and C_2, respectively.

$$\mu_1(th) = \sum_{i=0}^{th} iP(i|C_1) = \sum_{i=0}^{th} \frac{iP(C_1|i)P(i)}{P(C_1)} = \frac{1}{\omega_1(th)} \sum_{i=0}^{th} iPh_i \tag{14}$$

$$\mu_2(th) = \sum_{i=th+1}^{L-1} iP(i|C_2) = \sum_{i=th+1}^{L-1} \frac{iP(C_2|i)P(i)}{P(C_2)} = \frac{1}{\omega_2(th)} \sum_{i=th+1}^{L-1} iPh_i \tag{15}$$

$$\mu_{th} = \sum_{i=0}^{th} iPh_i,\ \mu_T = \sum_{i=0}^{L-1} iPh_i \tag{16}$$

where $\mu_1(th)$ and $\mu_2(th)$ indicate the mean intensity levels for C_1 and C_2, respectively. μ_{th} denotes the mean intensity level from 0 to th. μ_T represents the mean intensity level for the whole image. In summary, Eq. (17) is derived without difficulty. Hence, the objective function of the maximizing variance between classes can be expressed by Eq. (18).

$$\begin{aligned} &\omega_1(th) + \omega_2(th) = 1 \\ &\omega_1(th) \cdot \mu_1(th) + \omega_2(th) \cdot \mu_2(th) = \mu_G \end{aligned} \tag{17}$$

$$\begin{aligned} \sigma_B^2 &= \frac{(\mu_G \omega_1(th) - \mu_{th})^2}{\omega_1(th)(1 - \omega_1(th))} \\ &= \omega_1(th)(\mu_1(th) - \mu_T)^2 + \omega_2(th)(\mu_2(th) - \mu_T)^2 \\ &= \omega_1(th)\omega_2(th)(\mu_1(th) - \mu_2(th))^2 \end{aligned} \tag{18}$$

$$\sigma_B^2(th^*) = \max_{0 \le th \le L-1} \sigma_B^2(th) \tag{19}$$

Therefore, according to Eq. (18), we are able to compute a th^* to maximize the σ_B^2, which can be further expressed using Eq. (19). In conclusion, the Otsu method can be considered as a maximization problem, which means that the Otsu method could be further optimized using the meta-heuristics.

2.3.2 Kapur's Entropy

Kapur's entropy was proposed by Kapur in 1985 [78], which segmented the image based on the probability distribution of the image histogram. Also considering the bi-level threshold segmentation problem, the objective function of Kapur's entropy is defined as shown below:

$$\max F_{Kapur}(th) = H_1 + H_2 \tag{20}$$

where H_1 and H_2 represent the Kapur's entropy of the pixel intensity at $[0, th]$ and $[th, L-1]$, respectively, which are computed as follows:

$$H_1 = \sum_{i=0}^{th} \frac{Ph_i}{\omega_1(th)} \ln\left(\frac{Ph_i}{\omega_1(th)}\right),\ H_2 = \sum_{i=th+1}^{L-1} \frac{Ph_i}{\omega_2(th)} \ln\left(\frac{Ph_i}{\omega_2(th)}\right) \tag{21}$$

where ln is the natural logarithm, and the rest definitions in this formula have the same meanings as those mentioned in the previous section.

2.4 The COVID-19 Dataset

As mentioned in the previous section, this study was conducted to promote the scientific research for COVID-19. Therefore, we evaluated the performance of the proposed algorithm using chest CT images of COVID-19. CT images were obtained from the dataset [79], which has 349 CT images containing clinical findings of COVID-19 from 216 patients. In this study, the proposed algorithm and other comparison algorithms were evaluated on 12 randomly selected images from this dataset in order to test and compare the performance between each algorithm.

3 The Proposed Algorithm

Global exploration capabilities and local exploitation capabilities intrinsically affect the performance of meta-heuristics. However, the two capabilities conflicted with each other for the majority of meta-heuristics. It means that there is no way for a meta-heuristic to perform local exploitation while performing global exploration. Therefore, how to balance the exploration and exploitation for meta-heuristics has become a key factor in improving the performance. At this stage, the meta-heuristics usually use a composite function calculated from the current and the maximum number of iterations to control exploration and exploitation. For example, the evading energy (E) in the GJO is used to control the exploration and exploitation. When $\left|\vec{E}\right| > 1$, the algorithm performs exploration, vice versa, it performs exploitation. Hence, we could realize that the search process of GJO is unified, which means that all of the individuals are transformed from global search to local exploitation. This phenomenon implies a strong possibility that the algorithm will be trapped in local optimal. Therefore, an independent search process can improve the performance more effectively, whereas reinforcement learning can conveniently achieve this goal.

3.1 The Q-Learning Strategy

In this study, we consider the individual in the population as the agent of RL, while the search space is regarded as the environment, the state (s) represents the current updated position scheme of the individual, and the action (a) stands for the change process of state (s). Each individual has three operations to update their own position: exploration, exploitation, and hybrid mode. They adaptively choose the update strategy based on their learning experience. If the fitness of an individual increases after the updating operation, then it could receive positive feedback, and vice versa, it must receive a penalty. The Q-table has been designed as a 3 × 3 matrix, where the rows and columns denote state and action, respectively. In addition, it is important to note that each individual has a separate Q-table in order to ensure the independence of the learning process. Figure 1 illustrates the update process of an individual in Q-Learning mode. The individual is currently in the "Exploration" state, and by comparing the action feedback of the "Exploration" state, we can anticipate that the individual will get the most rewards when the next state is in the "Hybrid". Therefore, the individual will switch to the "Hybrid" state. In addition, the value of the Q-table will be updated by Eq. (10).

With the analysis of Fig. 1 combined with Eq. (10), we realize that the learning rate λ can influence the Q-Learning algorithm enormously. A higher learning rate can cause individuals to forget the experiences which have been gained. On the other hand, a lower learning rate can prevent individuals from learning from the environment to change their behavior. Therefore, learning rate should be dynamically adjusted from a higher value to a lower value during the iteration, which can effectively increase the learning ability of the individual. In this study, the adjustment formula for the learning rate is shown as follows:

$$\lambda = \frac{\lambda_{initial} + \lambda_{funal}}{2} - \frac{\lambda_{initial} - \lambda_{funal}}{2} \cos\left(\pi\left(1 - \frac{t}{T}\right)\right) \quad (22)$$

where $\lambda_{initial}$ and λ_{final} stand for the initial and final value of λ, which have been set to 0.9 and 0.1, respectively. t and T represent the current and maximum iteration number, respectively. Moreover, the reward parameter r has been determined by fitness, which is set to 1 if the fitness is improved, otherwise -1.

Algorithm 3 Pseudo-code of the proposed algorithm.

```
Inputs: The population size $N$, the max iterations $T$, the threshold number $L$, and the image $I_{in}$.
Outputs: The best solution, which contains the best thresholds and the image after segmentation.
Initialization the population $\boldsymbol{Pos}$ with threshold $L$ according Eq.(1).
Initialization the Q-table: $Q(s,a) = 0$ and the initial state $s$.
while ($t < T$)
    Calculating the fitness using Otsu method.
    $\overrightarrow{\boldsymbol{Pos_m}}(t) = best(\boldsymbol{Pos}(t))$
    $\overrightarrow{\boldsymbol{Pos_{fm}}}(t) = second\_best(\boldsymbol{Pos}(t))$
    foreach ($\boldsymbol{Pos}(t)$)
        Calculate the Evasion Energy ($\vec{\boldsymbol{E}}$) according Eq.(4).
        Choose the best action $a$ for the current state $s$ according Q-table.
        switch (action)
            case "Exploration":
                Update the population according Eq.(2), Eq.(3), and Eq.(26).
            case "Hybrid":
                Update the population according Eq.(23), Eq.(24), and Eq.(25).
            case "Exploitation":
                Update the population according Eq.(8), Eq.(9), and Eq.(27).
        end switch
        Update the Q-table for each individual according Eq.(10).
        Update the next state $s$.
    end foreach
    $t = t + 1$
end while
return the best solution.
```

3.2 The New Update Mode and Mutation Strategy

In the original GJO algorithm, the exploration operation is performed at the beginning of the iteration, which is described by Eq. (2) and Eq. (3). The exploitation operation, on the other hand, is performed at the end of the exploration operation, which is depicted using Eq. (8) and Eq. (9). In this paper, RL is introduced to coordinate the exploration and exploitation selection processes and inherit the original exploration and exploitation mechanism. However, in order to further balance exploration and exploitation, this study proposes a new mode, called the hybrid mode. In the hybrid mode, the population will be divided into two classes, in which one class keeps exploration while the other class starts exploitation. In addition, in the hybrid model, to further enhance the diversity of the population, a variation mechanism is designed to update the position of individuals in the population. The specific update strategy of the hybrid model is described by Eq. (23), Eq. (24), and Eq. (25).

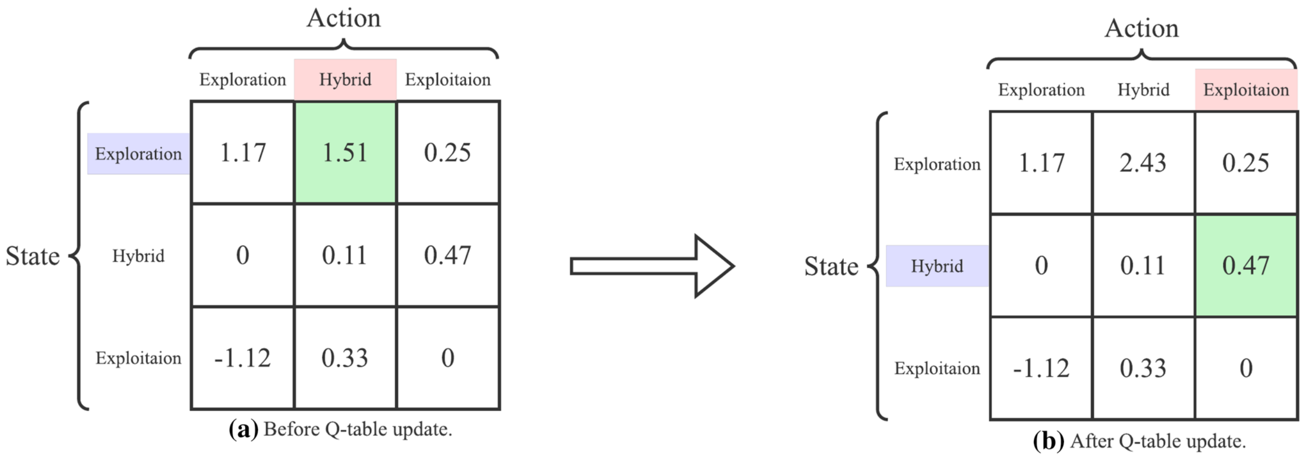
$$
\begin{aligned}
&\overrightarrow{\boldsymbol{Pos}_{\boldsymbol{\alpha}}}(t)=\frac{\overrightarrow{\boldsymbol{Pos}_{\mathbf{1}}}(t)+\overrightarrow{\boldsymbol{Pos}_{\mathbf{2}}}(t)}{2}\\
&\overrightarrow{\boldsymbol{Pos}_{\boldsymbol{\beta}}}(t)=\overrightarrow{\boldsymbol{Pos}_{\boldsymbol{r1}}}(t)+\overrightarrow{\boldsymbol{rand}}\circ(\overrightarrow{\boldsymbol{Pos}_{\boldsymbol{r2}}}(t)-\overrightarrow{\boldsymbol{Pos}_{\boldsymbol{r3}}}(t))\\
&\overrightarrow{\boldsymbol{Pos}_{\boldsymbol{k}}}(t+1)=\begin{cases}\overrightarrow{\boldsymbol{Pos}_{\boldsymbol{\alpha}}}(t), \text{fitness}(\overrightarrow{\boldsymbol{Pos}_{\boldsymbol{\alpha}}}(t))\text{fitness}(\overrightarrow{\boldsymbol{Pos}_{\boldsymbol{\beta}}}(t))\\ \overrightarrow{\boldsymbol{Pos}_{\boldsymbol{\beta}}}(t), otherwise\end{cases}
\end{aligned}
\tag{25}
$$

where r is a random number between 0 and 1. $\vec{\boldsymbol{Pos}}_{\boldsymbol{\alpha}}(t)$ represents the updated position of the t th iteration for the k th individual. $\vec{\boldsymbol{Pos}}_{\boldsymbol{\beta}}(t)$ denotes the mutation position of the t th iteration for the k th individual. $\vec{\boldsymbol{rand}}$ stands for a random vector between 0 and 1. $\vec{\boldsymbol{Pos}}_{r1}(t)$, $\vec{\boldsymbol{Pos}}_{r2}(t)$, and $\vec{\boldsymbol{Pos}}_{r3}(t)$ are three random individuals, respectively.

For both exploration and exploitation strategies, we present two mutation strategies to improve the performance of the algorithm, respectively. Equation (26) is applied after exploration to assist the population in exploring more search space. On the other hand, Eq. (27) is used to mutate after

$$
\overrightarrow{\boldsymbol{Pos}_{\mathbf{1}}}(t)=\begin{cases}\overrightarrow{\boldsymbol{Pos}_{\boldsymbol{m}}}(t)-\vec{\boldsymbol{E}}\circ\left|\overrightarrow{\boldsymbol{Pos}_{\boldsymbol{m}}}(t)-\vec{\boldsymbol{RL}}\circ\overrightarrow{\boldsymbol{Pos}_{\boldsymbol{k}}}(t)\right|, r<0.5\\ \overrightarrow{\boldsymbol{Pos}_{\boldsymbol{m}}}(t)-\overrightarrow{\ \boldsymbol{E}\ }\circ\left|\overrightarrow{\boldsymbol{RL}}\circ\overrightarrow{\boldsymbol{Pos}_{\boldsymbol{m}}}(t)-\overrightarrow{\boldsymbol{Pos}_{\boldsymbol{k}}}(t)\right|, r\geq 0.5\end{cases}
\tag{23}
$$

$$
\overrightarrow{\boldsymbol{Pos}_{\mathbf{2}}}(t)=\begin{cases}\overrightarrow{\boldsymbol{Pos}_{\boldsymbol{fm}}}(t)-\overrightarrow{\ \boldsymbol{E}\ }\circ\left|\overrightarrow{\boldsymbol{Pos}_{\boldsymbol{fm}}}(t)-\overrightarrow{\ \boldsymbol{RL}\ }\circ\overrightarrow{\boldsymbol{Pos}_{\boldsymbol{k}}}(t)\right|, r<0.5\\ \overrightarrow{\boldsymbol{Pos}_{\boldsymbol{fm}}}(t)-\overrightarrow{\ \boldsymbol{E}\ }\circ\left|\overrightarrow{\ \boldsymbol{RL}\ }\circ\overrightarrow{\boldsymbol{Pos}_{\boldsymbol{fm}}}(t)-\overrightarrow{\boldsymbol{Pos}_{\boldsymbol{k}}}(t)\right|, r\geq 0.5\end{cases}
\tag{24}
$$



Action

State

	Exploration	Hybrid	Exploitaion
Exploration	1.17	1.51	0.25
Hybrid	0	0.11	0.47
Exploitaion	-1.12	0.33	0

(a) Before Q-table update.

Action

State

	Exploration	Hybrid	Exploitaion
Exploration	1.17	2.43	0.25
Hybrid	0	0.11	0.47
Exploitaion	-1.12	0.33	0

(b) After Q-table update.

Fig. 1 The update process of the Q-table

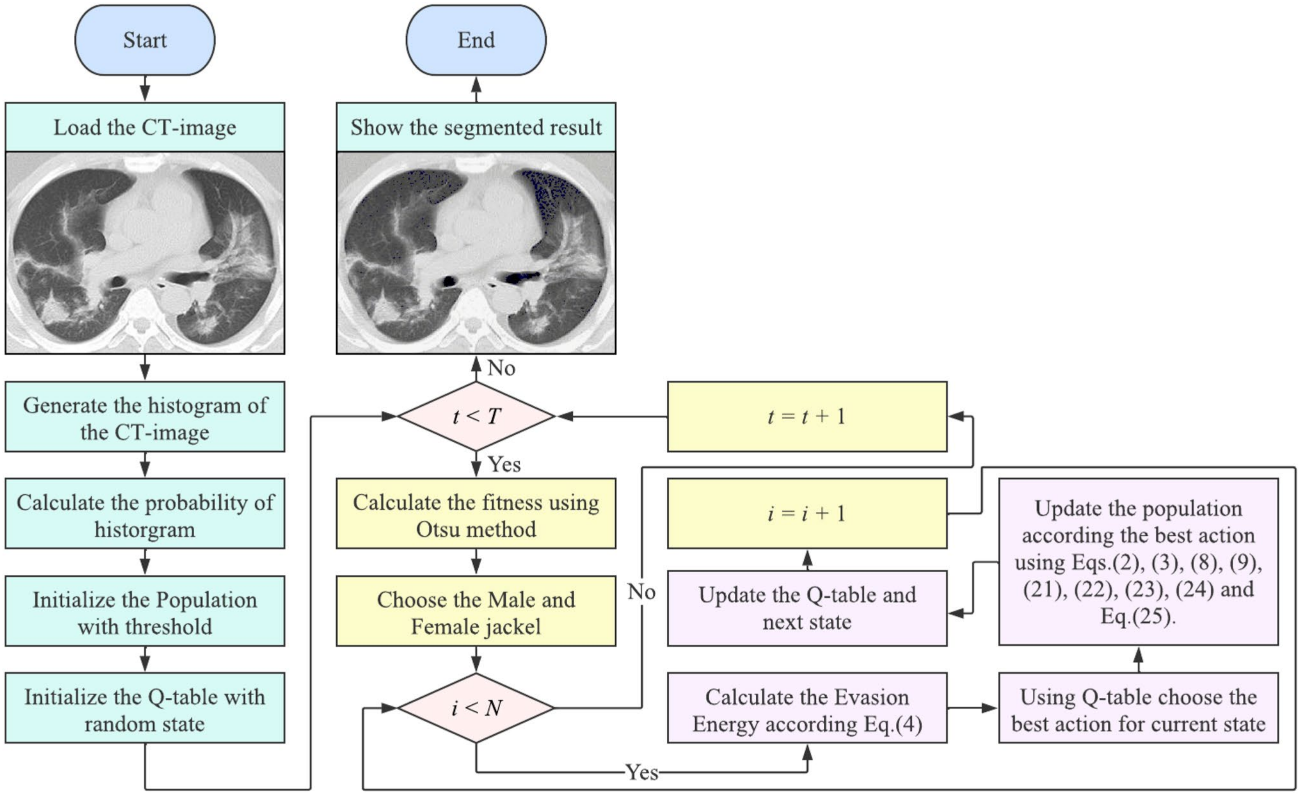

Start
Load the CT-image
Generate the histogram of the CT-image
Calculate the probability of historgram
Initialize the Population with threshold
Initialize the Q-table with random state
End
Show the segmented result
No
$t < T$
Yes
Calculate the fitness using Otsu method
Choose the Male and Female jackel
$i < N$
No
Yes
$t = t + 1$
$i = i + 1$
Update the Q-table and next state
Calculate the Evasion Energy according Eq.(4)
Update the population according the best action using Eqs.(2), (3), (8), (9), (21), (22), (23), (24) and Eq.(25).
Using Q-table choose the best action for current state

Fig. 2 Flowchart of the proposed method

the exploitation to improve the diversity of the population and prevent individuals from being trapped in local optima.

$$
\begin{aligned}
&\overrightarrow{\boldsymbol{Pos}_\alpha}(t) = \frac{\overrightarrow{\boldsymbol{Pos}_1}(t) + \overrightarrow{\boldsymbol{Pos}_2}(t)}{2}\\
&\overrightarrow{\boldsymbol{Pos}_\beta}(t) = \overrightarrow{\boldsymbol{Pos}_{r1}}(t) + \overrightarrow{\boldsymbol{rand}} \circ (\overrightarrow{\boldsymbol{Pos}_{cr2}}(t) - \overrightarrow{\boldsymbol{Pos}_{r3}}(t)) + \overrightarrow{\boldsymbol{rand}} \circ (\overrightarrow{\boldsymbol{Pos}_{r4}}(t) - \overrightarrow{\boldsymbol{Pos}_{r5}}(t))\\
&\overrightarrow{\boldsymbol{Pos}_k}(t+1) = \begin{cases} \overrightarrow{\boldsymbol{Pos}_\alpha}(t), \text{fitness}(\overrightarrow{\boldsymbol{Pos}_\alpha}(t))\text{fitness}(\overrightarrow{\boldsymbol{Pos}_\beta}(t)) \\ \overrightarrow{\boldsymbol{Pos}_\beta}(t), \textit{otherwise} \end{cases}
\end{aligned}
\tag{26}
$$

where $\overrightarrow{\boldsymbol{Pos}_{r4}}(t)$ and $\overrightarrow{\boldsymbol{Pos}_{r5}}(t)$ are two random individuals, respectively. In addition, the meanings of the rest variables are all consistent with those mentioned in Eq. (25).

$$
\begin{aligned}
&\overrightarrow{\boldsymbol{Pos}_\alpha}(t) = \frac{\overrightarrow{\boldsymbol{Pos}_1}(t) + \overrightarrow{\boldsymbol{Pos}_2}(t)}{2}\\
&\overrightarrow{\boldsymbol{Pos}_\beta}(t) = \overrightarrow{\boldsymbol{Pos}_k}(t) + \overrightarrow{\boldsymbol{rand}} \circ (\overrightarrow{\boldsymbol{Pos}_{r2}}(t) - \overrightarrow{\boldsymbol{Pos}_{r3}}(t))\\
&\overrightarrow{\boldsymbol{Pos}_k}(t+1) = \begin{cases} \overrightarrow{\boldsymbol{Pos}_\alpha}(t), \text{fitness}(\overrightarrow{\boldsymbol{Pos}_\alpha}(t))\text{fitness}(\overrightarrow{\boldsymbol{Pos}_\beta}(t)) \\ \overrightarrow{\boldsymbol{Pos}_\beta}(t), \textit{otherwise} \end{cases}
\end{aligned}
\tag{27}
$$

It is worth noting that Eq. (27) is more similar to Eq. (25), and all variables have the same meanings. However, the variable $\overrightarrow{\boldsymbol{Pos}_k}(t)$ is used instead of the variable $\overrightarrow{\boldsymbol{Pos}_{r1}}(t)$. It allows the mutation of individuals to be more dependent on the current position, which enables the population to increase diversity while maintaining the original convergence. Furthermore, the pseudo-code and flowchart of the proposed method are given in Algorithm 3 and Fig. 2, respectively.

3.3 Computational Complexity

3.3.1 Time Complexity

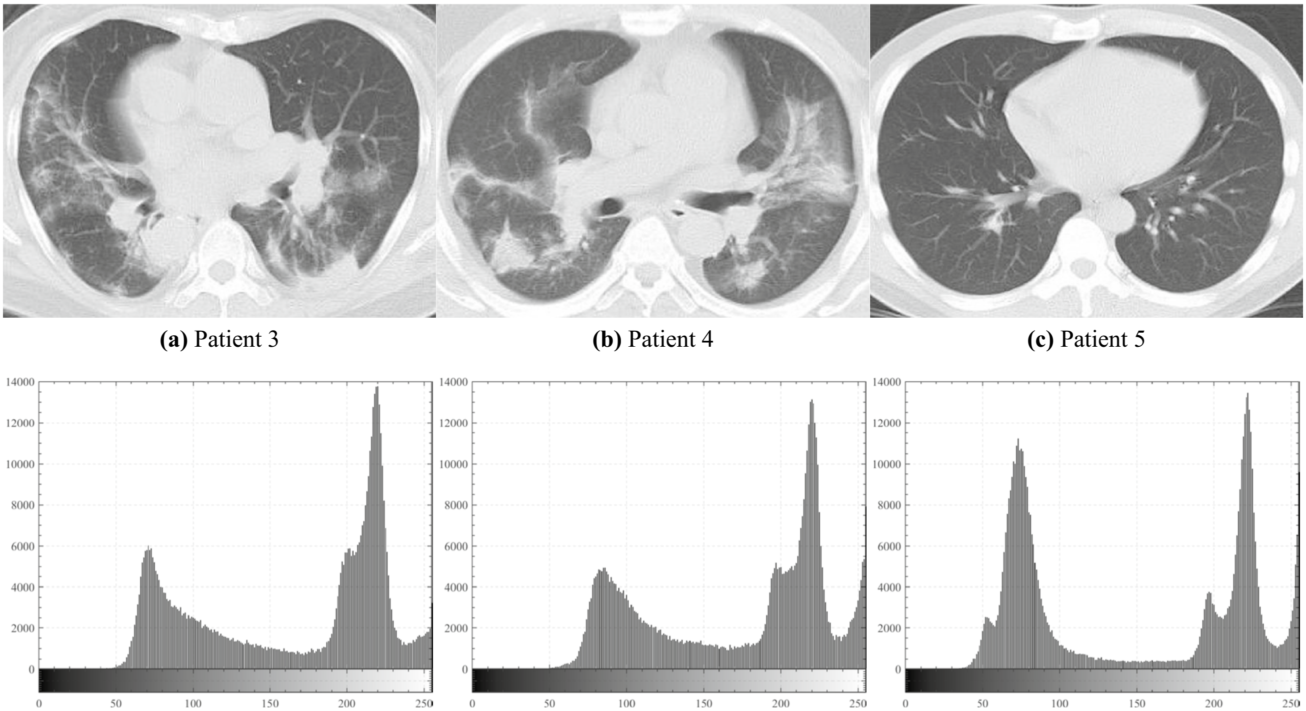
Further analysis of previous subsection reveals that the time complexity of the QLGJO algorithm is mainly influenced by three components. The first is the calculation of the fitness, the second is the update of the golden jackal population and the mutation of the golden jackal individuals, and the last is the update of the reinforcement learning component (Q-table). In which the calculation of the fitness will costs $O\left(T \times N \times O_{Otsu}\right)$ time to calculate, where T indicates the maximum number of iterations, N denotes the size of the population, and O_{Otsu} is the cost of Otsu method. And the population needs $O(T \times N \times M)$ time to be updated, where M represents the dimensions of the object function, which is generally the same as the threshold number in the Otsu method. In addition, the mutation strategy also costs $O(T \times N \times M)$ time complexity. Finally, the Q-table spends $O(T \times N)$ time to be updated. Therefore, the total time complexity of QLGJO is $O(T \times N \times (O_{Otsu} + M))$ after simplification.

3.3.2 Space Complexity

According to Algorithm 3, we can observe that the proposed algorithm additionally adds a Q-table for each individual. In addition, the space complexity of the population is $O(M \times N)$ and the space complexity of the Q-table is $O(9 \times N) \rightarrow O(N)$. Therefore, the space complexity of the algorithm is the sum of the space occupied by the population and the space required by the Q-table, which is $O(M \times N)$ with simplification.

4 Experimental Results and Analysis

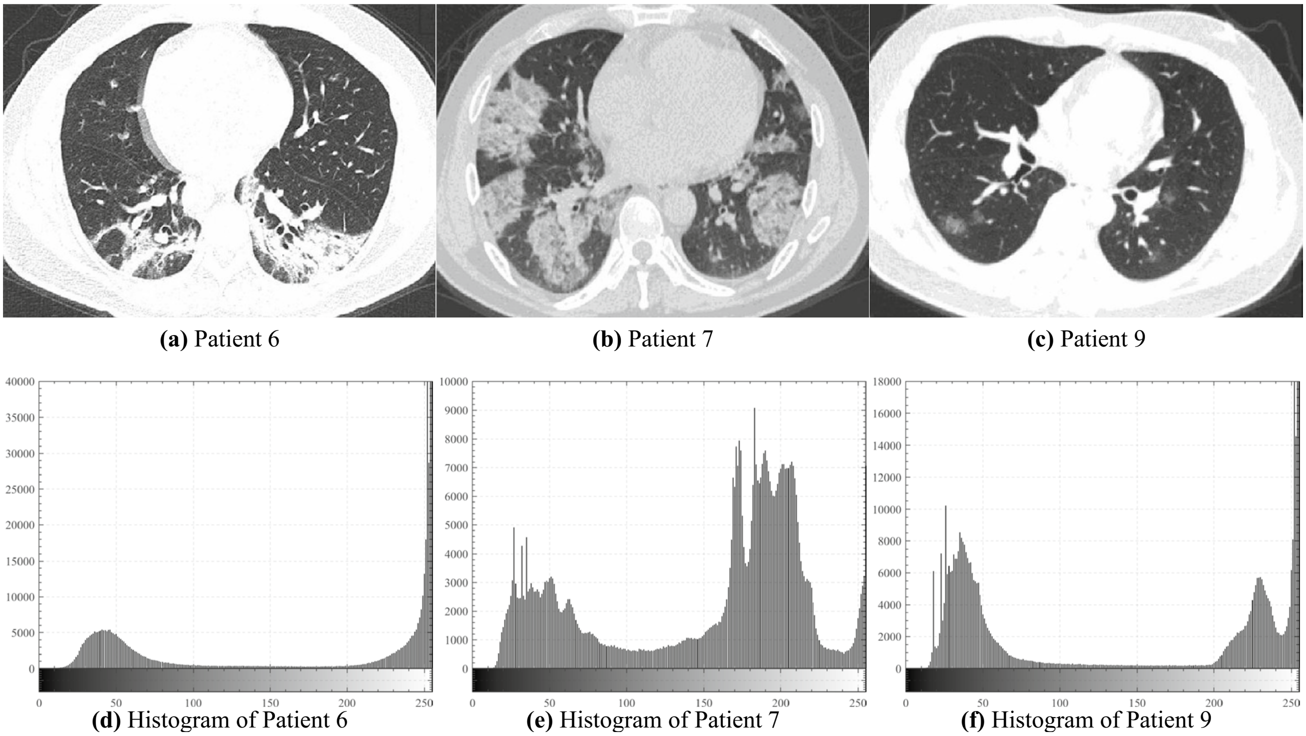
In this section, we demonstrate the performance of the proposed algorithm through two different sets of experiments. First, we use IEEE CEC2022 as a benchmark test function to test the performance of proposed method. In this experiment, we selected six advanced meta-heuristics for comparison, including: the original algorithm of the proposed method, GJO. The first medical image segmentation variant of the GJO algorithm, IGJO. One of the latest meta-heuristics, INFO. One of the commonly used meta-heuristics in recent years, MVO. The best old algorithm, DE, and the most popular algorithm, PSO. Then we further evaluate the practical performance of QLGJO by presenting the experimental results of the COVID-19 image segmentation. As mentioned in Sect. 2, we use 9 random chosen images from Yang et al. [79] as experimental data to compare the performance of the proposed algorithm with other meta-heuristics. The twelve images were named as Patient 3, Patient 4, Patient 5, Patient 6, Patient 7, Patient 9, Patient 13, Patient 24, Patient 30, Patient 37, Patient 80, and Patient 121. Figures 3, 4, 5, and 6 show each test image and their histogram information. In the comparison experiments, the algorithms involved in the comparison were kept the same as the previous experiment and the Otsu method was used as the objective function for the segmentation. Moreover, Peak



(a) Patient 3 **(b)** Patient 4 **(c)** Patient 5

(d) Histogram of Patient 3 **(e)** Histogram of Patient 4 **(f)** Histogram of Patient 5

Fig. 3 COVID-19 CT test images and their histograms


(**a**) Patient 6 (**b**) Patient 7 (**c**) Patient 9

(**d**) Histogram of Patient 6 (**e**) Histogram of Patient 7 (**f**) Histogram of Patient 9

Fig. 4 COVID-19 CT test images and their histograms

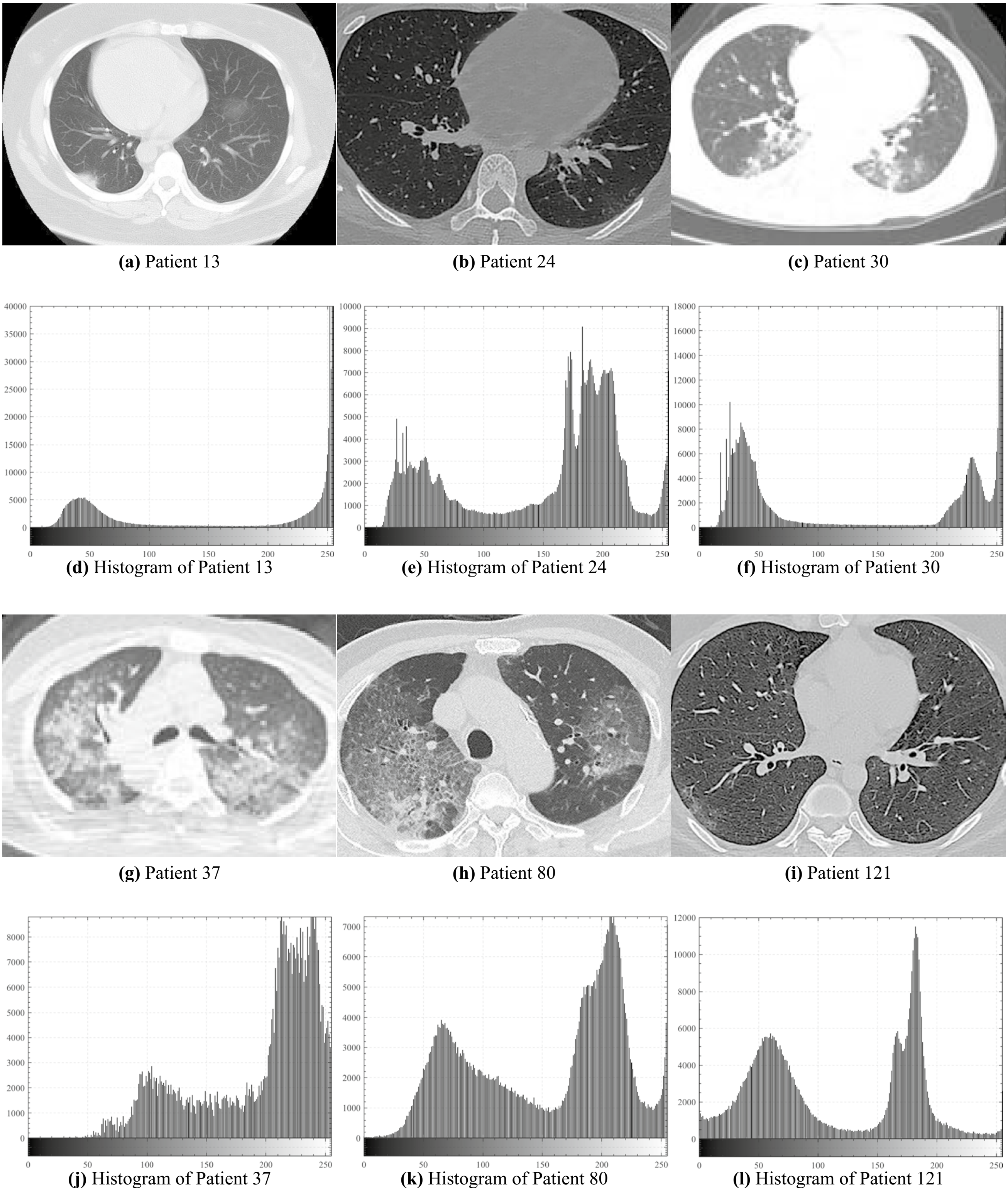
Signal to Noise Ratio ($PSNR$) [70], Structural Similarity Index ($SSIM$) [71], and Feature Similarity Index ($FSIM$) [72] were used as evaluation metrics to assess the performance of all algorithms.

4.1 Environment Settings

To ensure the fairness of the experiments, all algorithms were run 21 times independently in the same environment. For IEEE CEC2022 benchmark functions, the number of iterations was set to 5000 and the population size is fixed to 120. On the other hand, for segmentation experiment, the number of iterations was set to 200, and the population size was fixed at 60. The parameters of all algorithms were kept at default parameters to ensure that they were in a relatively optimal state, considering the suggestion by Arcuri et al. [80]. In addition, the specific configuration information of the running environment is given in Table 1, and the parameter settings for each algorithm are listed in Table 2.

4.2 Experiment on IEEE CEC2022

The CEC2022 benchmark test function is the most recent test function set which contains 12 different functions, of which F1 belongs to the Unimodal Functions, F2-F5 belongs to the Multi-modal Functions, F6-F8 belongs to the Hybrid Functions, and F9-12 belongs to the Composition



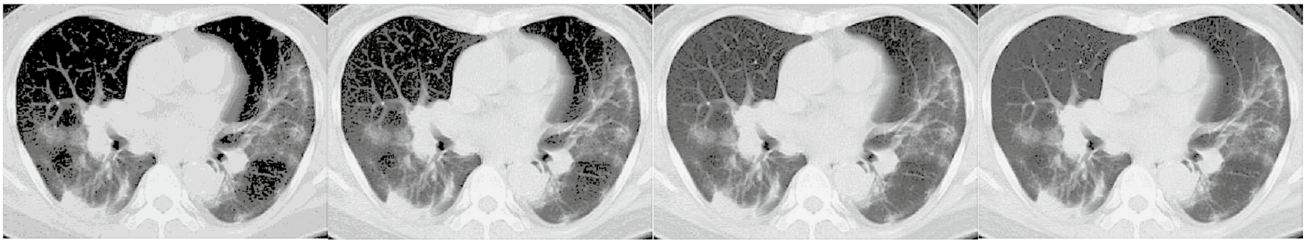
(a) Patient 13 **(b)** Patient 24 **(c)** Patient 30

(d) Histogram of Patient 13 **(e)** Histogram of Patient 24 **(f)** Histogram of Patient 30

(g) Patient 37 **(h)** Patient 80 **(i)** Patient 121

(j) Histogram of Patient 37 **(k)** Histogram of Patient 80 **(l)** Histogram of Patient 121

Fig. 5 COVID-19 CT test images and their histograms



(a) The 8 level result. **(b)** The 12 level result. **(c)** The 16 level result. **(d)** The 20 level result.

Fig. 6 The Segmented result for Patient 3

Table 1 Runtime Environment

Configurations	
Hardware	
CPU	Intel(R) Core(TM) i9-10980HK CPU @ 2.40 GHz
GPU	NVIDIA GeForce RTX 3080 Laptop GPU
RAM	64.0 GB
Software	
OS	Microsoft Windows [Version 10.0.19043.2130]
Interpreter	MATLAB R2021a (9.10.0.1602886)

Table 2 The parameters setting of all algorithms

Algorithm	Parameters
QLGJO	$C_1 = 1.5, \gamma = 0.5, \lambda_{initial} = 0.9, \lambda_{final} = 0.1$
GJO	$C_1 = 1.5$
IGJO	$C_1 = 1.5$
INFO	$c = 2, d = 4$
MVO	$\text{WEP}_{max} = 1, \text{WEP}_{min} = 0.2$
DE	$P_{Cr} = 0.8, F = 0.85$
PSO	$C_1 = 1.49445, C_2 = 1.49445$

Functions. In addition, Table 3 gives the details of the CEC2022 benchmark test function. In this experiment, we analyze the performance of the proposed method through quantitative and qualitative indicators. Among them, quantitative indicator includes the mean, median and Standard Deviations (std) [81] obtained in each benchmark function, while qualitative indicator is obtained by analyzing box-plot and convergence curve. In addition, the comprehensive performance of all algorithms was ranked by the Friedman mean rank test [82].

Table 4 shows a general overview of the performance of each algorithm on the CEC2022 benchmarking function, the optimal values are highlighted in this table. With the data in this table, we can see that for the accuracy, QLGJO achieves an impressive improvement over all functions compared to the original GJO, IGJO, and PSO. Compared to the latest meta-heuristic INFO, QLGJO achieves a great advantage on F2, F4, F5, F6, F7, F8, F10, F11, and F12, and shows similar performance F1, F3, and F9. Compared to the common newer meta-heuristic MVO, QLGJO also achieves a great success over F2, F3, F4, F6, F7, F8, F9, and F11, except for F10, which is surpassed by MVO, and shows similar performance F1, F5, and F12. Finally, compared with the best classical algorithm DE, QLGJO shows

Table 3 The CEC2022 benchmark test function

Function no	Function type	Function name	Dim	Range	Optimal
F1	Unimodal	Zakharov Function	10	[– 100, 100]	300
F2	Multi-modal	Rosenbrock's Function	10	[– 100, 100]	400
F3	Multi-modal	Schaffer's F7	10	[– 100, 100]	600
F4	Multi-modal	Rastrigin's Function	10	[– 100, 100]	800
F5	Multi-modal	Levy Function	10	[– 100, 100]	900
F6	Hybrid	Hybrid Function 1	10	[– 100, 100]	1800
F7	Hybrid	Hybrid Function 2	10	[– 100, 100]	2000
F8	Hybrid	Hybrid Function 3	10	[– 100, 100]	2200
F9	Composition	Composition Function 1	10	[– 100, 100]	2300
F10	Composition	Composition Function 2	10	[– 100, 100]	2400
F11	Composition	Composition Function 3	10	[– 100, 100]	2600
F12	Composition	Composition Function 4	10	[– 100, 100]	2700

Table 4 The CEC2022 benchmark test function

Function		QLGJO	GJO	IGJO	INFO	MVO	DE	PSO
F1	Mean	**300.000**	398.217	404.755	**300.000**	**300.000**	**300.000**	892.979
	Median	**300.000**	401.919	403.550	**300.000**	**300.000**	**300.000**	894.713
	STD	**0.000E + 00**	4.095E + 01	4.292E + 01	1.798E-14	5.076E-05	4.403E-14	1.857E + 01
	Cost	7.745	3.466	3.502	12.724	5.298	6.076	**3.457**
F2	Mean	**401.519**	415.278	412.458	411.467	404.243	404.501	423.496
	Median	**400.000**	409.203	409.177	403.987	404.216	403.987	419.326
	STD	1.984E + 00	1.744E + 01	1.524E + 01	2.001E + 01	**1.382E + 00**	2.040E + 00	7.362E + 00
	Cost	7.706	3.693	3.645	11.919	5.508	5.735	**3.379**
F3	Mean	**600.000**	602.587	602.394	**600.000**	600.041	**600.000**	611.545
	Median	**600.000**	602.383	602.018	**600.000**	600.026	**600.000**	610.907
	STD	9.229E-06	2.247E + 00	2.213E + 00	3.607E-04	4.178E-02	**0.000E + 00**	1.586E + 00
	Cost	9.653	4.795	4.607	14.082	7.447	6.660	**4.509**
F4	Mean	**804.454**	817.601	819.222	819.249	806.633	846.362	835.649
	Median	**803.980**	817.242	818.057	815.919	805.970	846.561	835.653
	STD	1.429E + 00	7.367E + 00	5.829E + 00	9.323E + 00	2.047E + 00	1.645E + 00	**1.275E + 00**
	Cost	8.454	**3.868**	3.992	13.294	6.137	6.583	4.008
F5	Mean	**900.000**	919.579	917.289	901.381	**900.000**	**900.000**	958.368
	Median	**900.000**	913.693	902.298	900.454	**900.000**	**900.000**	959.277
	STD	**0.000E + 00**	2.225E + 01	2.539E + 01	2.406E + 00	2.601E-05	**0.000E + 00**	3.520E + 00
	Cost	9.296	4.237	**4.013**	12.589	6.315	6.357	4.104
F6	Mean	**1800.096**	7333.616	6867.159	1803.759	2369.383	1800.732	1,519,252.339
	Median	**1800.093**	8075.227	8073.260	1802.387	2067.923	1800.741	1,558,007.090
	STD	5.382E-02	1.535E + 03	1.896E + 03	4.192E + 00	6.263E + 02	**3.238E-02**	2.457E + 05
	Cost	7.946	3.639	3.588	12.385	5.083	5.591	**3.463**
F7	Mean	2002.354	2027.980	2026.404	2014.468	2003.524	**2000.969**	2052.058
	Median	2001.133	2024.277	2024.166	2020.618	2000.784	**2000.000**	2051.357
	STD	4.298E + 00	7.250E + 00	5.967E + 00	9.738E + 00	7.133E + 00	4.371E + 00	**2.785E + 00**
	Cost	11.314	5.376	5.257	16.250	8.994	7.662	**5.076**
F8	Mean	2200.565	2221.089	2222.912	2218.737	2210.658	**2200.558**	2230.354
	Median	**2200.312**	2222.844	2222.450	2220.659	2203.924	2200.537	2230.769
	STD	6.373E-01	5.728E + 00	1.223E + 00	6.169E + 00	9.702E + 00	**2.973E-01**	2.010E + 00
	Cost	12.375	6.120	6.105	15.733	10.139	8.194	**5.724**
F9	Mean	**2529.284**	2536.896	2534.651	**2529.284**	2529.285	**2529.284**	2533.201
	Median	**2529.284**	2530.469	2529.315	**2529.284**	2529.285	**2529.284**	2533.198
	STD	**0.000E + 00**	1.394E + 01	1.619E + 01	**0.000E + 00**	2.361E-04	**0.000E + 00**	1.780E-01
	Cost	11.219	5.226	5.198	14.947	8.720	7.732	**5.115**
F10	Mean	2505.244	2528.519	2516.929	2528.417	**2500.153**	2500.515	2528.155
	Median	2500.227	2500.376	2500.311	2500.320	**2500.166**	2500.519	2501.164
	STD	2.302E + 01	5.179E + 01	4.179E + 01	5.157E + 01	4.214E-02	**2.017E-02**	5.705E + 01
	Cost	11.064	5.178	5.124	15.047	7.941	7.235	**4.790**
F11	Mean	**2600.000**	2722.870	2664.239	2720.706	2600.067	**2600.000**	2873.177
	Median	**2600.000**	2730.033	2605.279	**2600.000**	2600.067	**2600.000**	2752.069
	STD	1.134E-11	8.387E + 01	6.753E + 01	1.834E + 02	9.118E-03	**0.000E + 00**	1.804E + 02
	Cost	13.062	6.132	6.022	15.956	9.909	8.220	**5.914**
F12	Mean	2859.186	2862.573	2862.445	2863.082	**2858.620**	2860.199	2865.250
	Median	**2858.618**	2863.495	2863.669	2863.495	2858.619	2861.405	2865.191
	STD	8.321E-01	2.182E + 00	2.712E + 00	1.213E + 00	**2.648E-03**	1.326E + 00	6.227E-01
	Cost	13.354	6.106	6.054	16.680	10.511	8.659	**6.006**
Friedman mean rank		**1.7917**	5.6667	4.9167	4.0000	2.8333	2.2083	6.5833

Table 4 (continued)

Function	QLGJO	GJO	IGJO	INFO	MVO	DE	PSO
Rank	**1**	6	5	4	3	2	7

some advantages on F2, F4, F6, and F12, and is comparable to DE on the rest of the benchmark functions. On the other hand, as far as the cost time of the algorithm is concerned, the proposed method is somewhat lacking and fails to obtain the best score among all the benchmark functions. However, compared to INFO, the proposed method still has some advantages. Finally, according to the Friedman mean rank test, we can further conclude that QLGJO ranks first in terms of overall performance among all the compared algorithms, followed by DE, MVO ranks third, while INFO, IGJO, GJO, and PSO rank fourth, fifth, sixth, and seventh, respectively.

Figure 7 shows the boxplot of all the algorithms on the CEC2022 benchmark functions. Through the boxplot we can intuitively comprehend the data distribution of the algorithm after multiple runs. The maximum value is marked at the highest point and the minimum value is marked at the lowest point. Therefore, in general, we consider that the more stable the performance of an algorithm, the lower the height of the corresponding box plot. By analyzing Fig. 7, we can see that the proposed method has the lowest boxplot in the overall view. Therefore, we can tentatively conclude that QLGJO has the most stable performance among all the compared algorithms.

Figure 8 records the convergence curve plots of each algorithm. One point should be pointed out that, in order to compare the accuracy among the algorithms more intuitively, the convergence curves of all algorithms are operated with the difference between the optimal value of the benchmark function. Therefore, the closer the curve is to 0, the higher the accuracy of the algorithm is. In Fig. 8, the proposed algorithm reaches the lowest point except for F7 and F10. In other words, in terms of generalization, QLGJO has the highest accuracy among all the compared algorithms. However, one point also needs to be pointed out that although the convergence speed of QLGJO is substantially improved compared to the original algorithm, it still needs further enhancement compared to INFO and DE. In summary, through the experimental analysis in this subsection, we have a preliminary impression of the performance of QLGJO, and the proposed method is extremely competitive in numerical experiments compared with existing advanced algorithms. Therefore, in the next subsection, we will further validate the effectiveness of the proposed method in real-world problems with the image segmentation experiments on COVID-19.

4.3 COVID-19 CT Image Segmentation Experiment

4.3.1 Performance Metrics

As mentioned above, in this experiment, three evaluation parameters, $PSNR$, $SSIM$, and $FSIM$ are used to evaluate the performance of the proposed algorithm. In this subsection, a brief review of these three metrics is presented.

$PSNR$ is a common metric in digital image processing, which is used in multilevel threshold segmentation as a very critical performance metric [83]. $PSNR$ could represent the difference between the original image and the segmented image, and is calculated using Eq. (28).

$$PSNR = 20\log_{10}\left(\frac{255}{RMSE}\right) \tag{28}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(Img_{org}(i,j) - Img_{seg}(i,j))^2}{M \times N}} \tag{29}$$

where $RMES$ is the root mean square error calculated by Eq. (29) [84], Img_{org} and Img_{seg} represent the original image and the segmented image, respectively, M and N denote the size of the image. In addition, the closer the value of $PSNR$ is to 0, the greater the difference between the two images.

Structural similarity index is another common metric used to compare the similarity of two images [85]. The $SSIM$ is between 0 and 1, which is similar to the $PSNR$. The greater the difference between the two images, the closer the $SSIM$ is to 0. It is calculated as follows:

$$SSIM(Img_{org}, Img_{seg}) = \frac{(2\mu_{org}\mu_{seg} + c_1)(2\delta_{org,seg} + c_2)}{(\mu_{org}^2 + \mu_{seg}^2 + c_1)(\delta_{org}^2 + \delta_{seg}^2 + c_2)} \tag{30}$$

where μ_{org} and μ_{seg} denote the mean intensities for the original image and segmented image, respectively, σ_{org} and σ_{seg} represent the standard deviations of Img_{org} and Img_{seg}, respectively. In addition, the covariance of Img_{org} and Img_{seg} was represented by $\sigma_{org,seg}$. Finally, C_1 and C_2 are two constant values.

Feature similarity index is a new metric used to compare the degree of feature difference between two images [86]. $FSIM$ obtains the result by compounding the Phase Congruency (PC) [87] and Gradient Magnitude (GM) [88]. The higher

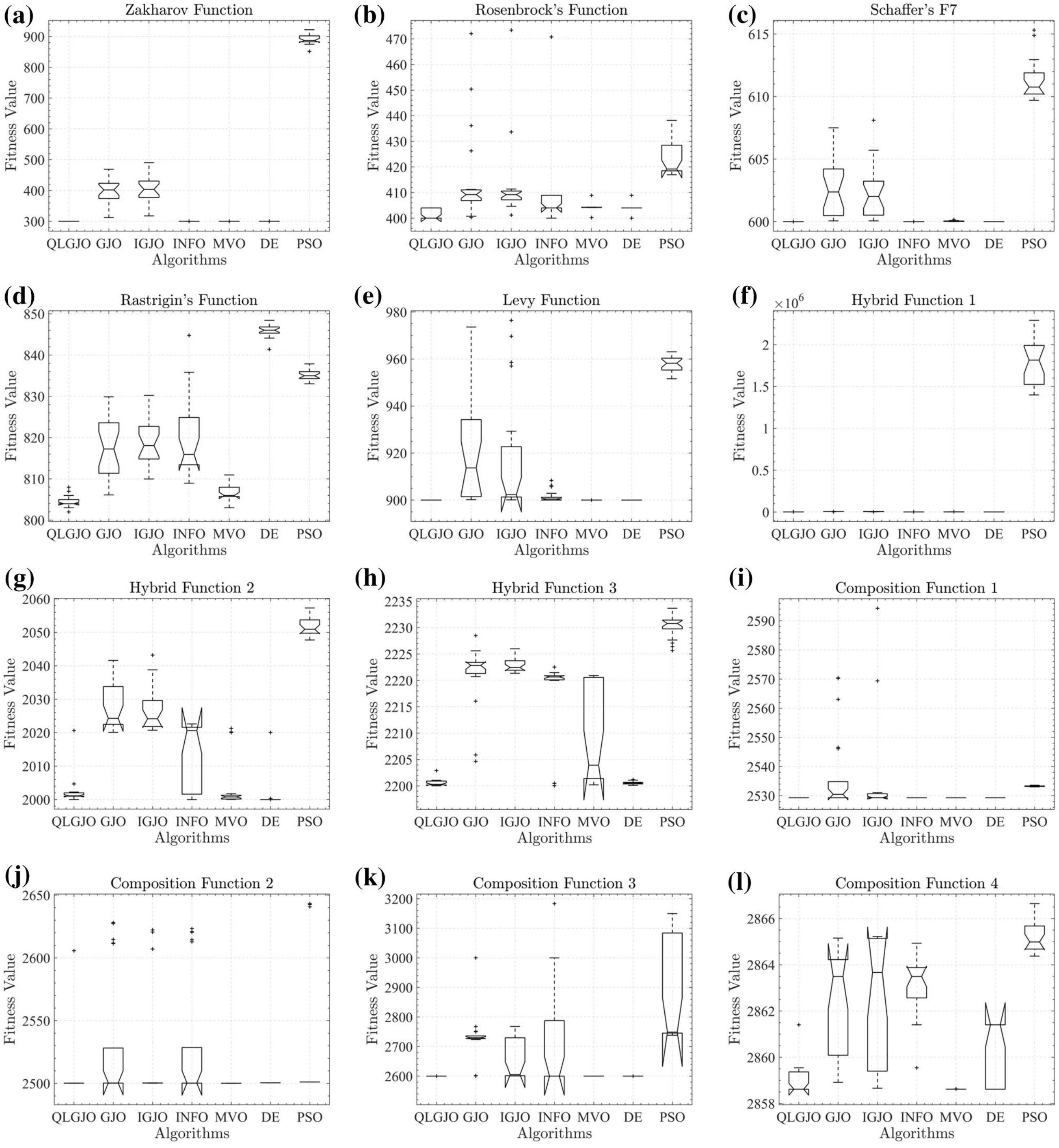


Fig. 7 The boxplot for CEC2022

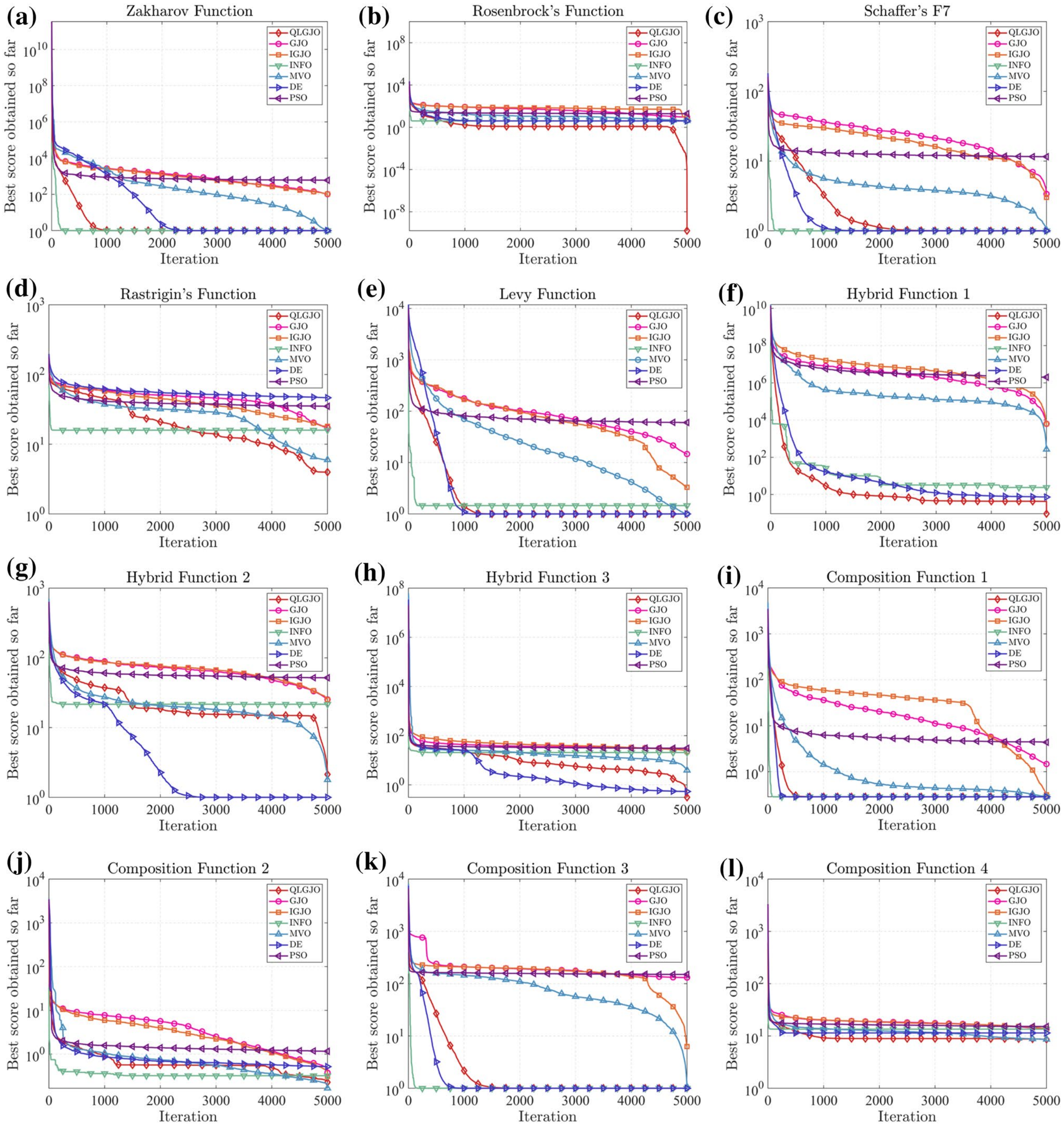


Fig. 8 The convergence curves for CEC2022

FSIM value indicates the better performance of the thresholding method. It can be described as follows:

$$FSIM = \frac{\sum_{x\in\Omega} S_L(x)PC_m(x)}{\sum_{x\in\Omega} PC_m(x)} \tag{31}$$

$$S_L(x) = [S_{PC}(x)]^{\alpha}[S_G(x)]^{\beta} \tag{32}$$

$$S_{PC}(x) = \frac{2PC_{org}(x)PC_{seg}(x) + T_1}{PC_{org}^2(x) + PC_{seg}^2(x) + T_1} \tag{33}$$

$$S_G(x) = \frac{2G_{org}(x)G_{seg}(x) + T_2}{G_{org}^2(x) + G_{seg}^2(x) + T_2} \tag{34}$$

where the phase congruency of Img_{org} and Img_{seg} were denoted by PC_{org} and PC_{seg}, respectively. And the gradient magnitude of Img_{org} and Img_{seg} were represented by G_{org} and G_{seg}. T_1 and T_2 are two positive constants, respectively. α and β are two constants, respectively.

4.3.2 Experimental Results and Analysis

In this subsection, we will discuss and analyze the results of the proposed algorithm for multilevel thresholding segmentation on COVID-19 CT images. In this experiment, we use the Otsu method, which is mentioned in Sect. 2.3 as the objective function. And the selected images are segmented with threshold levels of 8, 12, 16, and 20, respectively. Figures 4, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19 shows the segmented images of the proposed algorithm for all the tested images at different threshold levels. Table 5, Table 6, and Table 7 show the mean value and std of all algorithms on *PSNR*, *SSIM*, and *FSIM* metrics, respectively. In addition, the Friedman mean rank test was also again used to rank the comprehensive performance of all algorithms. It is noteworthy that an accurate and effective multilevel thresholding technique should have higher mean values on *PSNR*, *SSIM*, and *FSIM*, while the STD should be as low as possible. Therefore, the maximum mean value and the minimum STD value are highlighted in the tables.

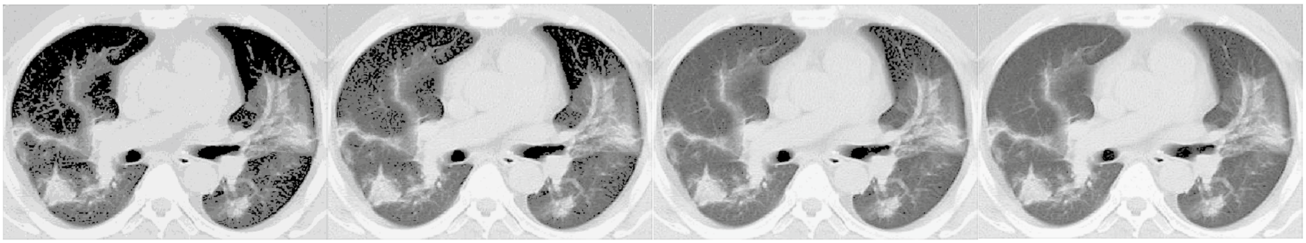
Table 5 shows the *PSNR* values for each segmented image. Through the analysis of the data in the table, the proposed method has achieved the best experimental results, except that it was surpassed by the INFO algorithm in the Patient 37 segmentation experiment with the threshold value of 8. With the Friedman mean rank test, the ranking results of all algorithms are shown as follows: QLGJO is ranked first, MVO is ranked second, INFO is ranked third, DE is ranked fourth, IGJO is ranked fifth, and GJO and PSO are ranked sixth and seventh, respectively.

As shown in Table 6, where the *SSIM* values of the segmentation results are recorded. We can observe that in the segmentation experiment of Patient 30 with a threshold of 8, DE achieves the best results, and the proposed algorithm is also slightly inferior compared to MVO. In the segmentation experiment on Patient 37 with a threshold of 8, DE again achieves the best results, but only slightly outperforms the proposed algorithm. According to the Friedman mean rank test, the top three algorithms are QLGJO, MVO, and IGJO.

Table 7 lists the mean *FSIM* values after all images were segmented. With the comparison of the data in this table, the original GJO, IGJO, and MVO did not obtain satisfactory results. While INFO, DE and PSO show a large improvement in *FSIM* metrics, in contrast, the proposed QLGJO outperforms the other algorithms in the majority of experimental results. Moreover, the proposed algorithm still maintains the best performance as measured with the Friedman mean rank test.

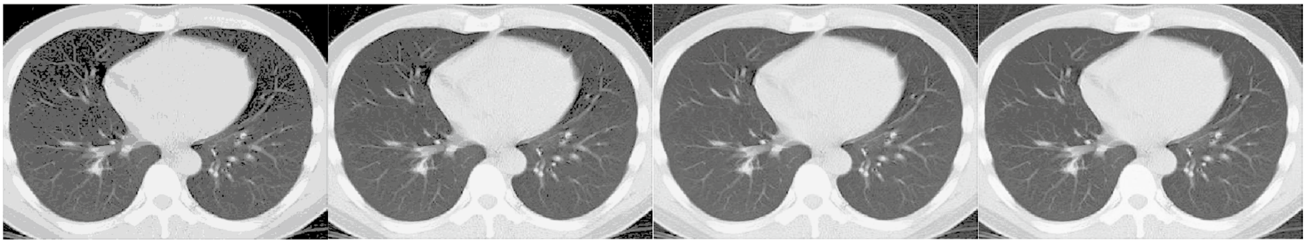
Furthermore, Fig. 20 shows the average time slots achieved by the different algorithms for each image segmentation experiment in terms of time cost. Each row in the figure represents the percentage of time consumed by the different algorithms in the same segmentation experiment. We can clearly observe that the percentage of time slots occupied by the proposed algorithms does not vary significantly across all experiments. This phenomenon indicates that the overhead of the QLGJO algorithm is not restricted to a specific image or threshold. Besides this, it should be noted that the proposed algorithm has a slightly higher time cost than the original algorithm, owing to the additional time cost of introducing reinforcement learning. Overall, the proposed algorithm does not perform particularly well at the time complexity level, but it can complete the specified task in a reasonable amount of time.

In addition, Table 8 presents the results of all algorithms tested for fitness values on the Otsu method according to the Wilcoxon rank sum test. The Wilcoxon rank-sum test was used to verify whether there was a significant



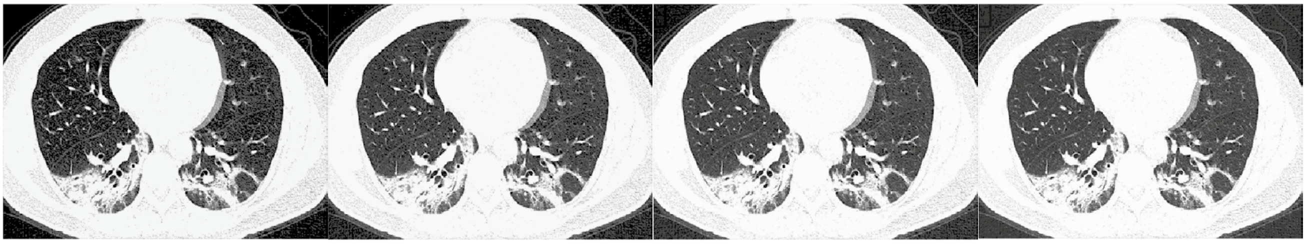
(a) The 8 level result. **(b)** The 12 level result. **(c)** The 16 level result. **(d)** The 20 level result.

Fig. 9 The Segmented result for Patient 4



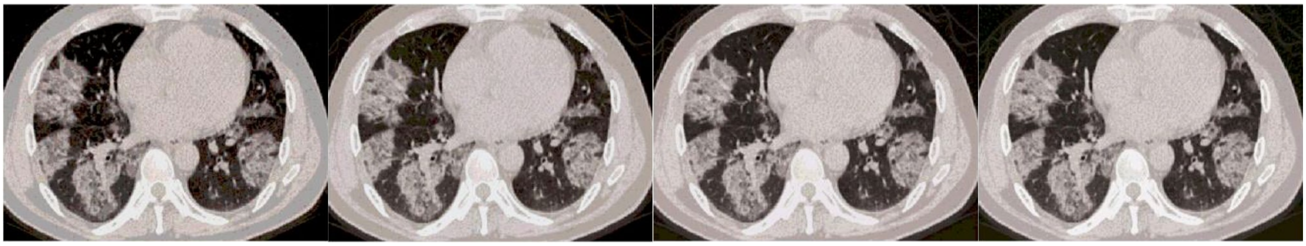
(a) The 8 level result. **(b)** The 12 level result. **(c)** The 16 level result. **(d)** The 20 level result.

Fig. 10 The Segmented result for Patient 5



(a) The 8 level result. **(b)** The 12 level result. **(c)** The 16 level result. **(d)** The 20 level result.

Fig. 11 The Segmented result for Patient 6



(a) The 8 level result. **(b)** The 12 level result. **(c)** The 16 level result. **(d)** The 20 level result.

Fig. 12 The Segmented result for Patient 7

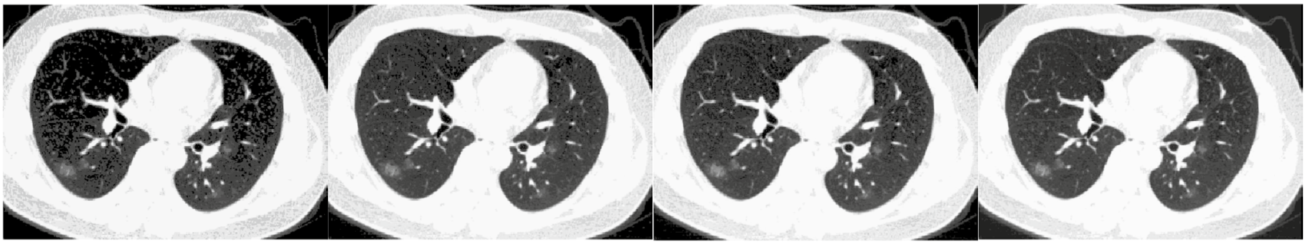

(a) The 8 level result. **(b)** The 12 level result. **(c)** The 16 level result. **(d)** The 20 level result.

Fig. 13 The Segmented result for Patient 9

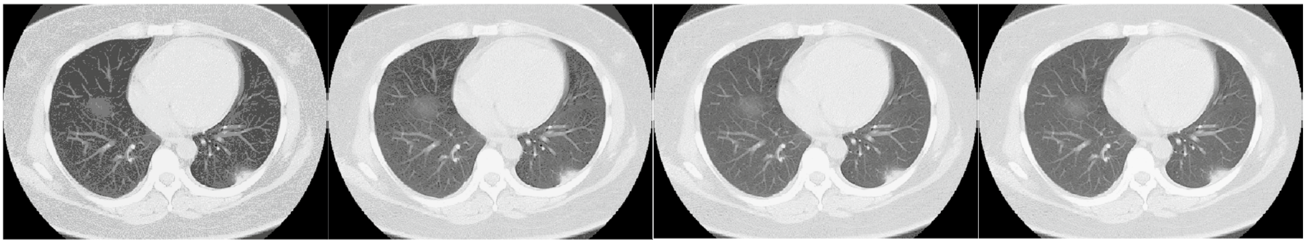

(a) The 8 level result. **(b)** The 12 level result. **(c)** The 16 level result. **(d)** The 20 level result.

Fig. 14 The Segmented result for Patient 13

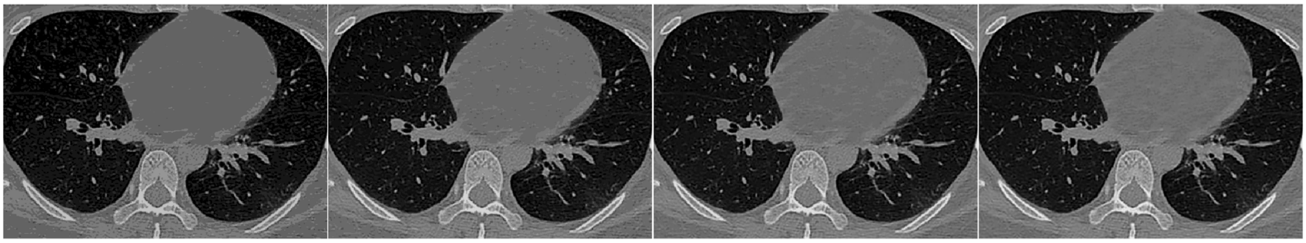
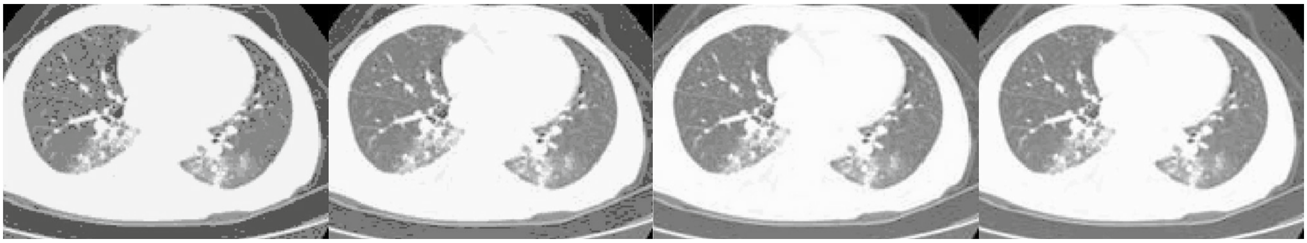

(a) The 8 level result. **(b)** The 12 level result. **(c)** The 16 level result. **(d)** The 20 level result.

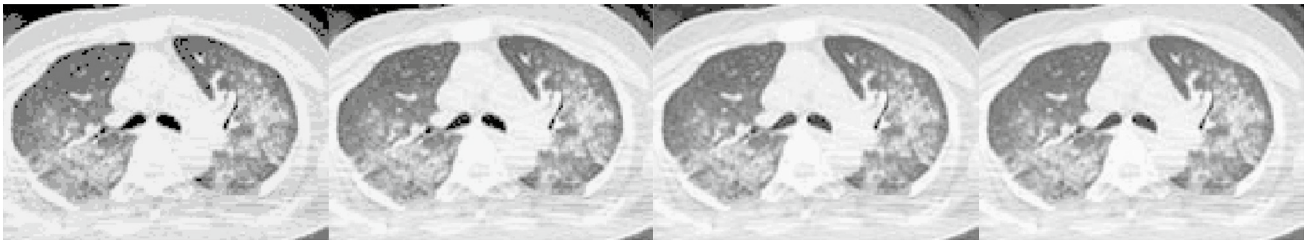
Fig. 15 The Segmented result for Patient 24

difference between the algorithms. When the p-value is less than 0.05, it can be considered that there is a significant difference between the algorithms. In other words, it means that there is a significant improvement in the proposed algorithm. Conversely, this means that the performance of the proposed algorithm is similar to or worse than the algorithm being compared. To better represent the analysis of the values, we use the symbols "$++$" and "$-$" to represent the cases where the p-value is less than 0.05 and the cases where the p-value is greater than 0.05, respectively. From Table 8, we can observe that there is a significant difference in the fitness values of the proposed algorithm compared to the original GJO, IGJO, INFO, DE, and PSO. Although it did not completely beat MVO in comparison, it only showed no significant improvement in the case of Patient 7 with a threshold of 20 and Patient 80 with a threshold of 16 and 20. Therefore, based on the results of the Wilcoxon rank sum test, we can conclude that the proposed QLGJO algorithm has higher performance in multilevel threshold segmentation using the Otsu method.



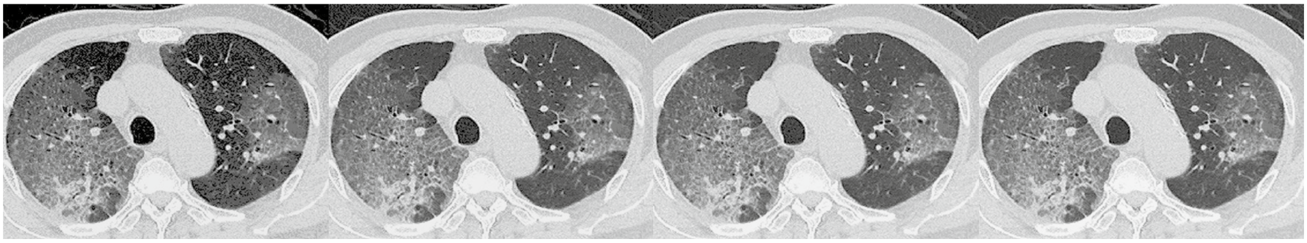
(a) The 8 level result. **(b)** The 12 level result. **(c)** The 16 level result. **(d)** The 20 level result.

Fig. 16 The Segmented result for Patient 30



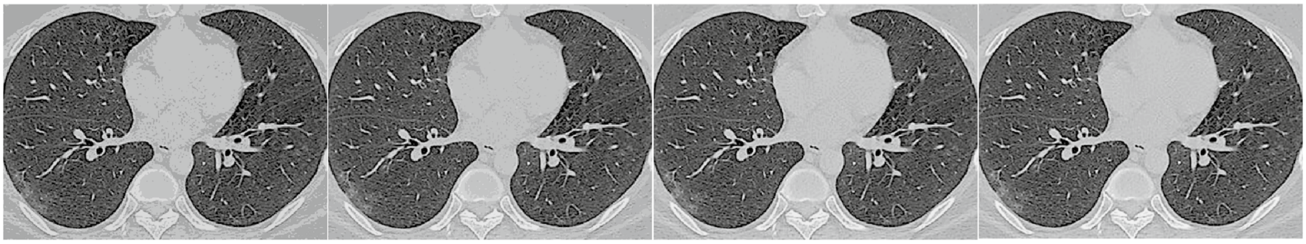
(a) The 8 level result. **(b)** The 12 level result. **(c)** The 16 level result. **(d)** The 20 level result.

Fig. 17 The Segmented result for Patient 37



(a) The 8 level result. **(b)** The 12 level result. **(c)** The 16 level result. **(d)** The 20 level result.

Fig. 18 The Segmented result for Patient 80



(a) The 8 level result. **(b)** The 12 level result. **(c)** The 16 level result. **(d)** The 20 level result.

Fig. 19 The Segmented result for Patient 121

Further inspection of the data in Tables 5–7, we can observe that although the QLGJO obtains the best results in terms of mean value, it still has some deficiencies in terms of STD, none of the proposed methods obtains a STD close to 0. Therefore, we further compare with the original GJO and IGJO to analyze the reasons for the poor performance of the proposed method in terms of STD. To analyze the performance differences between different

Table 5 Comparison of *PSNR*

Image Name	Level	QLGJO		GJO		IGJO		INFO		MVO		DE		PSO	
		Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
Patient 3	8	**18.49**	4.85E− 02	17.56	2.78E− 01	17.43	3.33E− 01	17.71	1.82E− 01	17.87	1.10E− 01	18.35	**0.00E+00**	14.01	9.72E− 01
	12	**21.29**	**2.39E− 01**	18.75	1.21E+00	19.19	1.31E+00	18.77	1.06E+00	19.40	4.35E− 01	19.17	6.30E− 01	17.49	1.09E+00
	16	**25.48**	**7.60E− 01**	20.21	1.92E+00	20.51	2.49E+00	19.56	1.99E+00	20.99	1.23E+00	20.90	1.40E+00	18.89	1.48E+00
	20	**29.61**	1.77E+00	21.27	2.16E+00	22.06	2.08E+00	21.78	2.63E+00	23.39	**1.61E+00**	22.96	1.91E+00	20.41	1.71E+00
Patient 4	8	**18.28**	4.50E− 02	15.90	4.93E− 01	16.97	3.79E− 01	17.57	1.67E− 01	17.14	3.96E− 01	18.13	**3.64E− 15**	14.76	9.49E− 01
	12	**22.00**	**3.21E− 01**	18.61	7.63E− 01	19.04	9.68E− 01	19.49	9.43E− 01	20.58	3.56E− 01	18.94	8.76E− 01	14.96	1.57E+00
	16	**27.32**	**8.79E− 01**	19.02	2.48E+00	22.36	2.09E+00	20.87	2.36E+00	21.74	1.56E+00	21.91	1.57E+00	18.03	1.95E+00
	20	**31.00**	**1.47E+00**	23.58	1.89E+00	22.13	2.38E+00	21.43	2.71E+00	25.10	**1.47E+00**	22.77	2.31E+00	18.00	2.96E+00
Patient 5	8	**21.64**	7.31E− 02	17.38	1.29E+00	18.56	1.03E+00	20.88	2.35E− 01	19.04	9.40E− 01	21.39	**1.09E− 14**	16.61	1.19E+00
	12	**25.00**	**2.09E− 01**	23.49	7.04E− 01	22.30	7.80E− 01	23.09	4.57E− 01	23.83	3.09E− 01	22.58	6.23E− 01	18.39	1.78E+00
	16	**28.72**	**6.93E− 01**	25.33	9.93E− 01	24.92	1.03E+00	23.92	1.47E+00	25.32	8.84E− 01	23.76	7.26E− 01	21.41	1.84E+00
	20	**30.58**	1.23E+00	27.17	8.69E− 01	27.54	**7.77E− 01**	25.50	1.33E+00	26.10	1.26E+00	25.72	1.62E+00	22.66	1.68E+00
Patient 6	8	**22.36**	2.78E− 02	21.33	3.98E− 01	21.38	2.61E− 01	21.88	1.12E− 01	22.15	3.44E− 02	22.28	**1.50E− 03**	19.05	7.42E− 01
	12	**25.58**	**1.41E− 01**	23.74	5.35E− 01	24.33	3.59E− 01	24.30	3.71E− 01	24.51	2.41E− 01	24.43	2.83E− 01	21.72	1.09E+00
	16	**29.24**	**6.11E− 01**	26.37	8.78E− 01	26.41	8.12E− 01	25.99	8.73E− 01	26.25	6.81E− 01	25.58	8.02E− 01	21.50	1.58E+00
	20	**32.97**	1.06E+00	28.87	1.10E+00	27.43	1.63E+00	26.74	1.12E+00	26.83	1.65E+00	26.33	1.17E+00	23.73	1.28E+00
Patient 7	8	**21.90**	7.09E− 02	19.87	4.84E− 01	20.62	2.91E− 01	21.59	5.07E− 02	21.62	**3.91E− 02**	21.63	0.00E+00	19.75	4.94E− 01
	12	**25.42**	1.48E− 01	24.23	3.54E− 01	23.64	4.23E− 01	24.13	2.64E− 01	24.50	**2.43E− 01**	24.08	3.10E− 01	20.66	1.02E+00
	16	**27.96**	**3.90E− 01**	25.10	8.42E− 01	24.92	7.86E− 01	25.44	6.17E− 01	25.96	6.61E− 01	25.29	5.20E− 01	23.10	8.85E− 01
	20	**29.71**	7.02E− 01	26.41	9.03E− 01	26.53	9.08E− 01	27.04	**5.80E− 01**	27.05	6.73E− 01	26.29	7.11E− 01	24.19	1.02E+00
Patient 9	8	**22.64**	8.72E− 02	21.06	4.81E− 01	21.13	3.10E− 01	21.89	1.22E− 01	21.20	2.78E− 01	22.28	**3.64E− 15**	19.33	8.19E− 01
	12	**26.66**	**2.67E− 01**	23.86	7.76E− 01	23.58	9.09E− 01	24.21	4.62E− 01	23.57	1.05E+00	25.01	4.03E− 01	21.79	8.64E− 01
	16	**27.94**	**2.87E− 01**	25.83	5.91E− 01	26.12	4.06E− 01	26.25	4.45E− 01	26.43	3.38E− 01	26.15	4.38E− 01	22.93	1.05E+00
	20	**29.08**	4.82E− 01	26.72	9.02E− 01	27.37	1.09E+00	27.54	9.35E− 01	27.40	**3.87E− 01**	26.86	6.54E− 01	24.16	1.29E+00
Patient 13	8	**21.88**	8.22E− 02	18.92	9.62E− 01	18.98	8.07E− 01	20.87	2.74E− 01	20.60	3.84E− 01	21.55	**4.26E− 02**	17.51	1.13E+00
	12	**28.13**	**2.84E− 01**	23.19	1.38E+00	23.45	1.24E+00	25.42	6.51E− 01	25.05	7.23E− 01	25.74	5.64E− 01	19.32	1.55E+00
	16	**31.56**	**4.50E− 01**	26.54	1.03E+00	27.54	1.10E+00	27.50	7.91E− 01	28.08	1.00E+00	27.47	9.42E− 01	23.24	1.68E+00
	20	**33.89**	5.26E− 01	28.95	1.01E+00	30.09	8.44E− 01	30.98	**5.22E− 01**	29.52	9.90E− 01	29.56	6.58E− 01	24.34	1.44E+00
Patient 24	8	**25.02**	3.58E− 02	22.36	8.58E− 01	22.74	6.60E− 01	24.09	2.55E− 01	22.95	7.24E− 01	24.89	**2.61E− 02**	18.65	1.43E+00
	12	**29.28**	**9.87E− 02**	25.09	1.03E+00	26.80	6.38E− 01	27.99	3.75E− 01	27.44	5.32E− 01	27.60	3.87E− 01	21.93	1.26E+00
	16	**31.87**	**2.24E− 01**	29.25	6.87E− 01	28.93	6.56E− 01	29.85	4.79E− 01	29.78	4.72E− 01	29.89	3.00E− 01	23.82	1.31E+00
	20	**33.86**	3.64E− 01	30.94	6.40E− 01	30.79	7.33E− 01	31.88	3.41E− 01	32.65	**3.38E− 01**	31.14	3.47E− 01	27.52	8.47E− 01
Patient 30	8	**21.54**	1.67E− 01	18.06	1.29E+00	17.86	1.26E+00	20.33	2.44E− 01	19.68	4.77E− 01	21.07	**1.22E− 01**	15.04	2.10E+00
	12	**26.67**	**5.74E− 01**	22.07	1.48E+00	22.68	1.33E+00	25.61	7.20E− 01	22.16	1.21E+00	24.41	8.59E− 01	17.56	1.95E+00
	16	**32.02**	**8.91E− 01**	24.35	1.88E+00	24.84	2.08E+00	27.77	1.07E+00	25.99	1.60E+00	26.36	1.17E+00	22.21	1.73E+00
	20	**34.10**	9.56E− 01	28.06	1.54E+00	26.67	1.74E+00	30.06	**8.85E− 01**	27.10	2.31E+00	29.13	9.35E− 01	24.35	1.57E+00

Table 5 (continued)

Image Name	Level	QLGJO		GJO		IGJO		INFO		MVO		DE		PSO	
		Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
Patient 37	8	**23.09**	6.57E−02	19.53	1.46E+00	19.96	1.22E+00	22.66	1.81E−01	19.79	1.13E+00	22.98	**1.09E−14**	17.51	1.43E+00
	12	**25.91**	**2.76E−01**	24.61	1.03E+00	24.28	1.08E+00	25.45	6.54E−01	24.79	9.74E−01	24.29	4.52E−01	20.04	1.43E+00
	16	**29.43**	7.71E−01	27.29	8.31E−01	27.04	9.33E−01	26.06	9.64E−01	26.09	1.50E+00	25.71	**7.65E−01**	23.12	1.43E+00
	20	**32.91**	1.07E+00	29.88	5.91E−01	30.34	**5.54E−01**	26.96	1.18E+00	26.48	1.90E+00	27.04	1.27E+00	21.85	2.26E+00
Patient 80	8	21.27	7.08E−02	20.23	8.88E−01	20.40	6.81E−01	**21.33**	3.30E−01	20.93	4.00E−01	21.07	**9.89E−03**	18.05	1.09E+00
	12	**26.45**	3.13E−01	23.33	1.02E+00	23.68	8.86E−01	25.94	**2.78E−01**	25.28	3.90E−01	24.36	4.65E−01	20.43	1.32E+00
	16	**29.50**	3.59E−01	26.44	8.14E−01	26.96	7.04E−01	28.09	**3.37E−01**	28.08	4.91E−01	27.28	4.67E−01	23.66	1.10E+00
	20	**31.70**	**2.19E−01**	28.73	6.37E−01	29.27	6.47E−01	30.32	2.45E−01	30.74	3.80E−01	28.12	5.47E−01	25.15	8.26E−01
Patient 121	8	**23.96**	4.05E−02	22.56	3.49E−01	23.25	1.82E−01	23.78	5.35E−02	23.03	4.22E−01	23.87	**3.71E−02**	18.46	1.08E+00
	12	**27.84**	**1.22E−01**	25.34	7.03E−01	25.36	6.00E−01	27.19	1.63E−01	26.64	3.10E−01	26.71	2.07E−01	19.81	1.44E+00
	16	**30.37**	**1.85E−01**	27.08	7.62E−01	27.88	6.26E−01	29.13	2.70E−01	29.59	2.38E−01	28.71	2.49E−01	25.16	8.09E−01
	20	**32.21**	2.76E−01	29.30	6.19E−01	29.42	6.19E−01	30.92	**2.71E−01**	30.92	4.18E−01	29.97	3.01E−01	26.84	5.28E−01
Friedman mean rank		**1.0208**		5.0833		4.5000		3.4271		3.3646		3.6042		7.0000	
Rank		**1**		6		5		3		2		4		7	

versions of the GJO algorithm, radar plots [89] for the original GJO, IGJO, and proposed QLGJO are plotted separately in Fig. 21. It should be noted that the Otsu method used in this paper is a maximum optimization problem. Thus, the closer an algorithm is to the outer layer, the stronger its performance. From the images, we notice that the original GJO only performs better on Patient 3, and Patient 9. On the other hand, IGJO improves on the original GJO by overcoming the issue that the GJO tends to fall into a local optimum in most experiments. However, these improvements are also very limited in front of QLGJO, since both algorithms mentioned earlier are inside QLGJO, which indicates that the performance of the proposed algorithm in this paper surpasses all other versions of GJO algorithms.

By analyzing the radar plots of the three different versions of the GJO, we did not obtain a directly reason which affects the stability of the QLGJO. Therefore, we further analyzed the population diversity of different versions of the GJO. Figures 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69 show the convergence plots of the population diversity analysis. In which, the abscissa indicates the number of iterations and the ordinate represents the mean Euclidean distance between population individuals. These figures can objectively reflect the distribution among individuals of the population and further report the diversity characteristics of the population. Through the observation and analysis of these figures, we can see that the reinforcement learning strategy can help the population keep its diversity characteristics unaffected by the problem. Combined with reinforcement learning, the proposed QLGJO algorithm converges to stability earlier than the original GJO as well as the improved GJO, and the convergence curve roughly obeys the distribution of the $f(t) = 1/t$. In addition, it is worth noting that the populations of both the original GJO and the IGJO eventually converge and cluster together, forcing the populations to fail to escape from the local optima. However, the populations of QLGJO still maintain a certain degree of diversity, which allows the algorithm to have the potential to step away from the local optimum to further improve the quality of the solution in the late iteration. Therefore, by analyzing the proposed method from multiple perspectives, we finally determined that the factor which affected the stability of the QLGJO algorithm was that the RL strategy would lead to a certain oscillation effect in the later Iteration. Overall, combining the results of CEC2022 benchmark test function and COVID-19 image segmentation, the proposed method can obtain satisfactory results in terms of convergence accuracy and convergence speed.

Table 6 Comparison of *SSIM*

Image Name	Level	QLGJO		GJO		IGJO		INFO		MVO		DE		PSO	
		Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
Patient 3	8	**0.680**	8.60E–04	0.637	1.63E–02	0.636	1.75E–02	0.646	9.24E–03	0.667	2.44E–03	0.678	**1.14E–16**	0.284	9.83E–02
	12	**0.782**	**3.56E–03**	0.743	2.57E–02	0.751	2.21E–02	0.729	1.98E–02	0.734	1.04E–02	0.734	1.01E–02	0.406	7.99E–02
	16	**0.844**	**1.11E–02**	0.742	3.13E–02	0.777	3.54E–02	0.763	2.42E–02	0.799	1.37E–02	0.782	1.75E–02	0.416	8.35E–02
	20	**0.908**	2.38E–02	0.781	3.02E–02	0.826	2.09E–02	0.800	2.90E–02	0.852	**1.56E–02**	0.810	2.33E–02	0.521	8.20E–02
Patient 4	8	**0.696**	5.06E–04	0.609	1.90E–02	0.654	1.61E–02	0.666	7.72E–03	0.656	1.52E–02	0.694	**1.14E–16**	0.469	8.87E–02
	12	**0.807**	**4.87E–03**	0.720	1.82E–02	0.756	1.81E–02	0.740	1.91E–02	0.789	6.74E–03	0.743	1.52E–02	0.249	1.28E–01
	16	**0.891**	**1.18E–02**	0.739	3.71E–02	0.829	3.04E–02	0.797	2.54E–02	0.824	1.96E–02	0.785	2.29E–02	0.653	6.07E–02
	20	**0.935**	2.02E–02	0.817	2.65E–02	0.813	2.77E–02	0.807	3.06E–02	0.879	**1.31E–02**	0.816	2.86E–02	0.441	1.00E–01
Patient 5	8	**0.723**	1.44E–03	0.573	4.68E–02	0.613	3.73E–02	0.689	8.53E–03	0.637	3.19E–02	0.717	**1.14E–16**	0.511	4.91E–02
	12	**0.841**	**9.39E–03**	0.790	2.23E–02	0.747	2.83E–02	0.759	1.92E–02	0.805	1.17E–02	0.755	2.30E–02	0.413	8.66E–02
	16	**0.899**	1.24E–02	0.861	1.50E–02	0.839	1.44E–02	0.778	2.69E–02	0.865	**1.06E–02**	0.792	2.03E–02	0.674	5.71E–02
	20	**0.906**	1.19E–02	0.871	1.23E–02	0.908	**1.10E–02**	0.832	1.92E–02	0.879	1.34E–02	0.864	1.83E–02	0.694	4.33E–02
Patient 6	8	**0.800**	1.76E–03	0.767	1.08E–02	0.770	7.90E–03	0.791	6.78E–03	0.791	9.63E–04	0.794	**5.90E–05**	0.511	6.53E–02
	12	**0.863**	**4.48E–03**	0.829	1.12E–02	0.848	7.42E–03	0.845	1.44E–02	0.846	4.94E–03	0.828	9.66E–03	0.733	6.11E–02
	16	**0.928**	**1.25E–02**	0.887	1.69E–02	0.879	1.53E–02	0.875	1.63E–02	0.876	**1.25E–02**	0.869	1.52E–02	0.634	5.43E–02
	20	**0.965**	1.58E–02	0.920	**1.40E–02**	0.898	2.11E–02	0.889	1.62E–02	0.885	2.10E–02	0.866	1.73E–02	0.837	3.00E–02
Patient 7	8	**0.710**	2.16E–03	0.627	2.03E–02	0.670	1.20E–02	0.703	4.91E–03	0.702	8.22E–04	0.702	**2.28E–16**	0.592	2.58E–02
	12	**0.833**	**5.89E–03**	0.793	1.33E–02	0.792	1.41E–02	0.791	1.03E–02	0.810	1.01E–02	0.777	1.13E–02	0.642	4.02E–02
	16	**0.894**	**9.60E–03**	0.813	2.82E–02	0.825	2.12E–02	0.820	1.52E–02	0.838	1.65E–02	0.827	1.15E–02	0.719	3.61E–02
	20	**0.912**	**1.37E–02**	0.859	2.33E–02	0.862	1.88E–02	0.866	1.07E–02	0.873	1.07E–02	0.841	1.57E–02	0.788	2.92E–02
Patient 9	8	**0.699**	3.76E–03	0.672	1.18E–02	0.669	6.75E–03	0.687	6.20E–03	0.671	3.83E–03	0.686	**2.28E–16**	0.497	4.58E–02
	12	**0.828**	**6.86E–03**	0.746	2.71E–02	0.740	3.15E–02	0.744	2.18E–02	0.730	3.61E–02	0.781	1.84E–02	0.645	5.65E–02
	16	**0.853**	**9.09E–03**	0.813	2.41E–02	0.818	1.06E–02	0.805	1.35E–02	0.834	6.50E–03	0.804	1.67E–02	0.675	4.26E–02
	20	**0.874**	1.15E–02	0.833	2.52E–02	0.844	3.12E–02	0.848	2.81E–02	0.844	**6.85E–03**	0.830	1.79E–02	0.683	5.15E–02
Patient 13	8	**0.747**	3.54E–03	0.695	2.01E–02	0.697	1.73E–02	0.713	1.20E–02	0.724	9.70E–03	0.742	**7.03E–04**	0.650	5.82E–02
	12	**0.883**	**6.14E–03**	0.797	2.61E–02	0.776	2.57E–02	0.818	1.79E–02	0.822	1.42E–02	0.812	1.58E–02	0.667	5.36E–02
	16	**0.941**	**8.31E–03**	0.883	1.59E–02	0.877	2.03E–02	0.847	1.72E–02	0.879	1.73E–02	0.870	1.61E–02	0.802	4.96E–02
	20	**0.961**	**7.66E–03**	0.907	1.44E–02	0.913	1.18E–02	0.921	9.76E–03	0.925	9.53E–03	0.895	1.13E–02	0.790	3.43E–02
Patient 24	8	**0.790**	5.11E–03	0.743	3.71E–02	0.725	3.00E–02	0.736	2.81E–02	0.716	3.18E–02	0.784	**3.56E–03**	0.428	8.52E–02
	12	**0.904**	**4.97E–03**	0.855	2.15E–02	0.866	1.42E–02	0.853	1.89E–02	0.881	1.14E–02	0.823	1.92E–02	0.592	6.91E–02
	16	**0.936**	6.73E–03	0.898	1.36E–02	0.903	9.51E–03	0.891	1.03E–02	0.924	**6.33E–03**	0.877	1.22E–02	0.671	4.54E–02
	20	**0.956**	7.07E–03	0.942	7.20E–03	0.933	7.17E–03	0.922	6.38E–03	0.938	**4.92E–03**	0.896	1.18E–02	0.800	2.89E–02
Patient 30	8	0.823	1.38E–02	0.814	2.47E–02	0.801	2.44E–02	0.843	7.90E–03	0.846	**1.16E–02**	**0.861**	1.42E–02	0.378	1.08E–01
	12	**0.896**	1.64E–02	0.883	1.63E–02	0.893	1.19E–02	0.886	3.08E–02	0.885	**1.17E–02**	0.860	3.21E–02	0.504	7.97E–02
	16	**0.943**	1.41E–02	0.904	1.47E–02	0.910	1.53E–02	0.820	3.48E–02	0.914	**1.16E–02**	0.791	3.22E–02	0.655	8.37E–02
	20	**0.967**	1.25E–02	**0.967**	8.32E–03	0.930	**1.10E–02**	0.919	1.35E–02	0.929	1.31E–02	0.904	1.89E–02	0.742	7.09E–02

Table 6 (continued)

Image Name	Level	QLGJO		GJO		IGJO		INFO		MVO		DE		PSO	
		Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
Patient 37	8	0.834	2.48E− 03	0.784	2.19E− 02	0.788	2.07E− 02	0.823	7.17E− 03	0.786	1.90E− 02	**0.839**	**0.00E + 00**	0.620	6.54E− 02
	12	**0.897**	7.62E− 03	0.868	1.05E− 02	0.890	9.87E− 03	0.875	1.22E− 02	0.892	**4.82E− 03**	0.862	7.37E− 03	0.752	5.45E− 02
	16	**0.926**	**4.35E− 03**	0.921	7.34E− 03	0.923	1.12E− 02	0.910	9.63E− 03	0.914	8.73E− 03	0.883	7.17E− 03	0.737	3.77E− 02
	20	**0.950**	**5.11E− 03**	0.939	7.90E− 03	0.936	5.33E− 03	0.921	9.09E− 03	0.925	8.38E− 03	0.906	7.28E− 03	0.598	8.11E− 02
Patient 80	8	**0.776**	1.36E− 03	0.727	2.74E− 02	0.731	1.87E− 02	**0.776**	4.37E− 03	0.770	2.62E− 03	0.772	**4.99E− 04**	0.611	3.65E− 02
	12	**0.898**	**5.45E− 03**	0.814	2.33E− 02	0.838	1.80E− 02	0.885	6.16E− 03	0.883	6.17E− 03	0.854	1.01E− 02	0.755	4.96E− 02
	16	**0.944**	5.00E− 03	0.903	1.13E− 02	0.907	1.08E− 02	0.920	5.58E− 03	0.933	**4.69E− 03**	0.902	7.64E− 03	0.818	3.17E− 02
	20	**0.964**	**2.85E− 03**	0.932	7.21E− 03	0.940	6.90E− 03	0.950	3.40E− 03	0.957	3.15E− 03	0.919	7.27E− 03	0.865	1.63E− 02
Patient 121	8	**0.858**	7.88E− 04	0.827	7.72E− 03	0.845	4.08E− 03	0.857	1.40E− 03	0.839	8.49E− 03	0.857	**6.67E− 04**	0.606	4.83E− 02
	12	**0.924**	5.03E− 03	0.888	1.09E− 02	0.890	8.79E− 03	0.905	5.24E− 03	0.909	**4.40E− 03**	0.900	6.08E− 03	0.651	5.71E− 02
	16	**0.949**	4.43E− 03	0.914	9.83E− 03	0.923	8.55E− 03	0.927	6.84E− 03	0.941	**4.32E− 03**	0.922	6.80E− 03	0.856	2.49E− 02
	20	**0.966**	4.49E− 03	0.938	6.30E− 03	0.938	6.85E− 03	0.948	**4.75E− 03**	0.960	5.35E− 03	0.930	6.43E− 03	0.850	1.84E− 02
Friedman mean rank		**1.1250**		4.4792		3.9375		3.9688		3.0625		4.4271		7.0000	
Rank		**1**		6		3		4		2		5		7	

Consequently, we can consider QLGJO as one of the most competitive GJO variants at present.

5 Conclusions

With the global prevalence of COVID-19, the entire scientific community has been working on ways to mitigate its impact on society. Early screening and treatment of patients can effectively cut off the transmission of COVID-19. Through the efforts of multiple medical experts, CT images have been shown to be effective in identifying suspected patients who are already infected with the new coronavirus. The multilevel threshold segmentation of CT images can effectively reduce the difficulty of subsequent processing and save more precious time for patients and health care workers. Therefore, we proposed a reinforcement learning-based GJO algorithm, QLGJO, to solve the COVID-19 CT image segmentation. In this study, we use the Otsu method as an objective function to determine the optimal threshold for COVID-19 CT images and make improvements to the original GJO algorithm. First, reinforcement learning was introduced to the GJO algorithm for the first time to balance the exploration and exploitation of the algorithm. Second, a new iterative phase was extended to accelerate the convergence of the algorithm. Finally, three new variational mechanisms are introduced to assist the algorithm avoid local optima. The proposed algorithm is compared to six advanced meta-heuristics (GJO, IGJO, INFO, MVO, DE, and PSO). Firstly, the CEC2022 benchmark test function was used to verify the performance of QLGJO. Then, Peak signal-to-noise ratio, structural similarity index, and feature similarity index are used as evaluation metrics to measure the performance differences between the algorithms in the segmentation experiments. The experimental results demonstrated that the proposed algorithm produced the most satisfactory results when compared to other algorithms and achieved efficient COVID-19 CT image segmentation.

In future work, we intend to introduce more objective functions into the proposed algorithm, such as Kapur Entropy, Tsallis Entropy, and Fuzzy Entropy. Meanwhile, we will extend the proposed algorithm to be applied to a wider range of fields, including feature selection, image classification, and drone path planning. Furthermore, it would also be a remarkable contribution to further reduce

Table 7 Comparison of *FSIM*

Image Name	Level	QLGJO		GJO		IGJO		INFO		MVO		DE		PSO	
		Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
Patient 3	8	0.803	3.06E−03	0.769	1.32E−02	0.770	1.28E−02	0.832	1.20E−02	0.793	8.45E−04	0.797	**1.14E−16**	**0.835**	2.61E−02
	12	0.861	5.49E−03	0.843	1.51E−02	0.851	1.25E−02	**0.881**	1.11E−02	0.831	**4.07E−03**	0.864	9.85E−03	0.842	3.12E−02
	16	**0.929**	**1.09E−02**	0.830	2.17E−02	0.851	2.75E−02	0.881	2.20E−02	0.857	1.01E−02	0.879	1.86E−02	0.878	2.59E−02
	20	**0.955**	1.99E−02	0.847	2.36E−02	0.869	2.02E−02	0.886	2.63E−02	0.883	**1.59E−02**	0.931	2.48E−02	0.887	1.97E−02
Patient 4	8	0.802	2.43E−03	0.739	1.41E−02	0.772	1.27E−02	**0.824**	8.96E−03	0.771	9.99E−03	0.796	**1.14E−16**	0.814	2.73E−02
	12	**0.875**	4.37E−03	0.816	1.15E−02	0.843	1.08E−02	0.873	1.56E−02	0.857	**3.93E−03**	0.861	1.26E−02	0.820	2.44E−02
	16	**0.945**	**1.19E−02**	0.829	2.47E−02	0.881	2.31E−02	0.884	2.67E−02	0.879	1.40E−02	0.895	2.05E−02	0.858	2.12E−02
	20	**0.970**	1.36E−02	0.876	1.91E−02	0.867	2.12E−02	0.914	1.95E−02	0.909	**1.10E−02**	0.936	1.90E−02	0.905	2.31E−02
Patient 5	8	**0.819**	4.70E−03	0.740	2.33E−02	0.759	1.90E−02	0.815	6.36E−03	0.777	1.27E−02	0.808	**3.41E−16**	0.808	3.51E−02
	12	**0.915**	**6.86E−03**	0.865	1.61E−02	0.839	1.87E−02	0.866	1.48E−02	0.879	8.84E−03	0.878	1.87E−02	0.828	2.73E−02
	16	**0.957**	7.33E−03	0.913	9.23E−03	0.898	9.32E−03	0.934	9.71E−03	0.916	**5.70E−03**	0.887	1.54E−02	0.867	3.57E−02
	20	**0.974**	8.05E−03	0.922	7.88E−03	0.939	**6.62E−03**	0.948	7.45E−03	0.922	7.91E−03	0.931	1.40E−02	0.868	2.99E−02
Patient 6	8	0.918	9.19E−04	0.915	2.64E−03	0.916	2.33E−03	0.923	2.74E−03	0.919	6.23E−04	0.921	**1.89E−04**	**0.931**	1.61E−02
	12	**0.953**	2.90E−03	0.930	5.20E−03	0.938	3.79E−03	0.936	5.90E−03	0.938	**2.24E−03**	0.932	6.32E−03	0.911	1.93E−02
	16	**0.970**	**5.25E−03**	0.961	7.98E−03	0.954	7.13E−03	0.950	9.65E−03	0.950	6.32E−03	0.946	9.56E−03	0.908	1.93E−02
	20	**0.987**	**5.95E−03**	0.970	6.54E−03	0.959	1.04E−02	0.959	8.80E−03	0.952	1.06E−02	0.944	1.08E−02	0.923	1.39E−02
Patient 7	8	0.896	1.39E−03	0.860	9.15E−03	0.872	5.94E−03	0.895	3.91E−03	0.896	6.91E−04	**0.897**	**0.00E+00**	0.853	1.92E−02
	12	**0.949**	**3.80E−03**	0.931	4.86E−03	0.933	5.48E−03	0.948	5.34E−03	0.936	4.65E−03	0.937	6.90E−03	0.903	1.68E−02
	16	**0.959**	**4.04E−03**	0.938	1.04E−02	0.938	1.00E−02	0.950	6.40E−03	0.952	7.66E−03	0.952	7.46E−03	0.932	1.20E−02
	20	**0.976**	4.43E−03	0.959	6.48E−03	0.953	5.68E−03	0.955	5.92E−03	0.960	**4.42E−03**	0.953	7.42E−03	0.936	9.12E−03
Patient 9	8	**0.913**	1.04E−03	0.893	6.17E−03	0.895	4.29E−03	0.915	4.07E−03	0.896	3.43E−03	0.910	**1.14E−16**	0.905	1.70E−02
	12	**0.959**	**3.80E−03**	0.932	7.74E−03	0.927	7.90E−03	0.944	3.03E−03	0.929	7.23E−03	0.940	6.09E−03	0.924	1.91E−02
	16	**0.954**	4.69E−03	0.945	8.06E−03	0.941	6.34E−03	0.952	7.04E−03	0.949	**1.96E−03**	0.945	6.48E−03	0.926	1.41E−02
	20	**0.982**	7.41E−03	0.950	9.84E−03	0.953	1.06E−02	0.951	1.08E−02	0.951	**3.68E−03**	0.949	8.13E−03	0.959	1.33E−02
Patient 13	8	0.867	5.27E−03	0.819	1.26E−02	0.821	9.79E−03	0.847	1.68E−02	0.844	**4.13E−03**	0.858	4.79E−04	**0.877**	2.21E−02
	12	**0.945**	**4.67E−03**	0.870	1.94E−02	0.871	1.77E−02	0.920	1.06E−02	0.900	9.98E−03	0.930	9.59E−03	0.888	2.14E−02
	16	**0.974**	**5.90E−03**	0.917	1.12E−02	0.929	9.36E−03	0.962	6.26E−03	0.935	8.75E−03	0.928	1.36E−02	0.927	1.55E−02
	20	**0.982**	**2.12E−03**	0.943	7.97E−03	0.959	5.06E−03	0.971	3.93E−03	0.948	6.82E−03	0.969	3.42E−03	0.916	1.56E−02
Patient 24	8	0.905	**1.40E−03**	0.875	1.12E−02	0.875	8.86E−03	**0.920**	4.59E−03	0.890	5.22E−03	0.902	1.52E−03	0.909	1.61E−02
	12	0.959	**2.70E−03**	0.919	9.26E−03	0.938	5.44E−03	**0.973**	3.97E−03	0.952	2.93E−03	0.963	4.21E−03	0.931	1.14E−02
	16	**0.980**	2.31E−03	0.956	4.13E−03	0.956	5.09E−03	0.978	**2.29E−03**	0.968	2.36E−03	0.968	4.23E−03	0.960	1.27E−02
	20	**0.991**	1.54E−03	0.970	4.42E−03	0.964	4.62E−03	0.989	2.79E−03	0.982	**1.14E−03**	0.986	2.46E−03	0.967	6.75E−03
Patient 30	8	**0.898**	6.66E−03	0.854	1.21E−02	0.860	9.38E−03	0.886	**4.13E−03**	0.864	5.23E−03	0.893	6.08E−03	0.867	1.99E−02
	12	**0.935**	7.14E−03	0.905	9.83E−03	0.907	8.08E−03	0.922	1.04E−02	0.905	**6.77E−03**	0.931	8.10E−03	0.915	1.38E−02
	16	**0.974**	**6.60E−03**	0.914	1.16E−02	0.925	1.19E−02	0.965	8.21E−03	0.925	8.46E−03	0.952	9.59E−03	0.913	1.43E−02
	20	**0.984**	**3.48E−03**	0.959	6.80E−03	0.939	8.26E−03	0.973	4.01E−03	0.942	9.40E−03	0.959	7.00E−03	0.931	1.01E−02

Table 7 (continued)

Image Name	Level	QLGJO		GJO		IGJO		INFO		MVO		DE		PSO	
		Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
Patient 37	8	**0.879**	3.55E−03	0.839	1.36E−02	0.841	1.46E−02	0.881	4.98E−03	0.840	1.06E−02	0.869	**2.28E−16**	0.849	1.71E−02
	12	**0.943**	6.25E−03	0.898	7.54E−03	0.919	9.24E−03	0.940	7.16E−03	0.915	**3.84E−03**	0.939	6.41E−03	0.899	1.36E−02
	16	**0.974**	4.86E−03	0.935	5.77E−03	0.938	8.87E−03	0.961	4.69E−03	0.933	6.82E−03	0.962	**4.17E−03**	0.914	1.28E−02
	20	**0.980**	**3.09E−03**	0.949	5.52E−03	0.942	4.52E−03	0.969	4.13E−03	0.938	6.19E−03	0.962	3.71E−03	0.897	1.82E−02
Patient 80	8	**0.872**	3.50E−03	0.842	1.71E−02	0.844	1.29E−02	**0.872**	1.04E−02	0.863	2.73E−03	0.866	**2.50E−04**	0.858	1.95E−02
	12	0.948	5.79E−03	0.900	1.57E−02	0.909	1.17E−02	**0.953**	6.46E−03	0.928	**5.59E−03**	0.941	8.55E−03	0.906	2.01E−02
	16	**0.973**	5.51E−03	0.946	8.67E−03	0.944	7.51E−03	0.971	**4.15E−03**	0.954	6.79E−03	0.967	6.40E−03	0.924	1.79E−02
	20	**0.989**	2.29E−03	0.962	4.76E−03	0.965	6.10E−03	0.987	**1.91E−03**	0.973	3.11E−03	0.968	7.10E−03	0.949	1.16E−02
Patient 121	8	0.919	9.28E−04	0.903	4.62E−03	0.910	2.56E−03	0.920	1.36E−03	0.924	2.99E−03	0.918	**3.19E−04**	**0.944**	1.55E−02
	12	0.966	3.44E−03	0.948	7.18E−03	0.946	7.83E−03	**0.971**	3.76E−03	0.965	**2.05E−03**	0.968	3.75E−03	0.933	1.40E−02
	16	0.986	**2.32E−03**	0.957	5.31E−03	0.966	4.63E−03	**0.987**	3.17E−03	0.979	2.87E−03	0.980	3.46E−03	0.961	7.57E−03
	20	0.990	2.53E−03	0.972	3.91E−03	0.967	5.08E−03	**0.993**	3.21E−03	0.985	**2.44E−03**	0.987	3.43E−03	0.978	7.55E−03
Friedman mean rank		**1.5625**		5.8646		5.2188		2.3542		4.3229		3.3542		5.3229	
Rank		**1**		7		5		2		4		3		6	

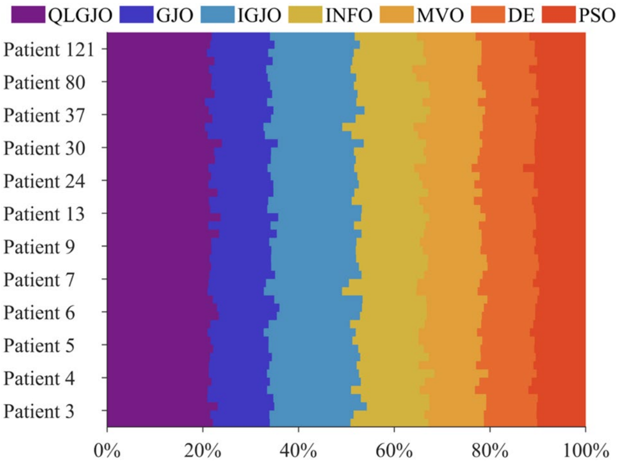


Fig. 20 The average time slots achieved by the different algorithms for each image segmentation experiment

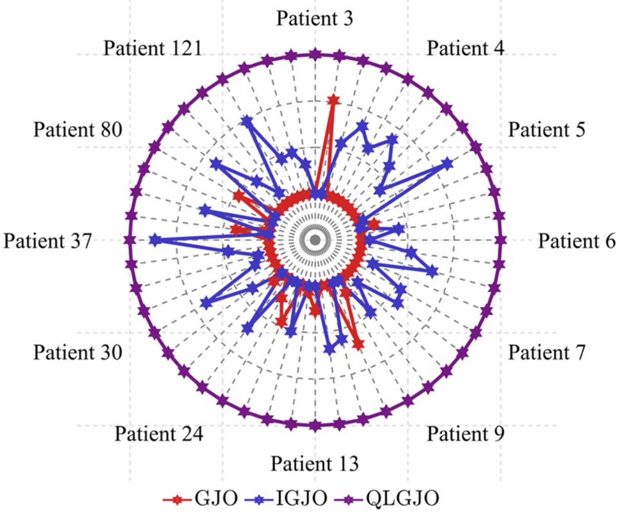


Fig. 21 Comparison of radar plots for GJO, IGJO, and QLGJO

Table 8 Comparison of the Wilcoxon signed-rank test for Otsu method

Image Name	Level	QLGJO VS. GJO	QLGJO VS. IGJO	QLGJO VS. INFO	QLGJO VS. MVO	QLGJO VS. DE	QLGJO VS. PSO
Patient 3	8	3.090E–05++	3.090E–05++	3.086E–05++	4.571E–05++	2.374E–04++	3.090E–05++
	12	3.090E–05++	3.090E–05++	3.090E–05++	9.660E–05++	3.090E–05++	3.090E–05++
	16	3.090E–05++	3.090E–05++	3.090E–05++	3.622E–04++	3.090E–05++	3.090E–05++
	20	3.090E–05++	3.090E–05++	3.090E–05++	1.270E–04++	3.090E–05++	3.090E–05++
Patient 4	8	3.090E–05++	3.090E–05++	3.043E–05++	1.217E–04++	8.654E–05++	3.090E–05++
	12	3.090E–05++	3.090E–05++	3.090E–05++	1.904E–03++	3.090E–05++	3.090E–05++
	16	3.090E–05++	3.090E–05++	3.090E–05++	3.578E–05++	3.090E–05++	3.090E–05++
	20	3.090E–05++	3.090E–05++	3.090E–05++	1.523E–03++	3.090E–05++	3.090E–05++
Patient 5	8	3.090E–05++	3.090E–05++	3.090E–05++	4.603E–05++	1.534E–04++	3.090E–05++
	12	3.090E–05++	3.090E–05++	3.090E–05++	4.779E–05++	3.090E–05++	3.090E–05++
	16	3.090E–05++	3.090E–05++	3.090E–05++	1.897E–04++	3.090E–05++	3.090E–05++
	20	3.090E–05++	3.090E–05++	3.090E–05++	6.015E–03++	3.090E–05++	3.090E–05++
Patient 6	8	3.090E–05++	3.090E–05++	3.090E–05++	3.090E–05++	5.982E–04++	3.090E–05++
	12	3.090E–05++	3.090E–05++	3.090E–05++	4.779E–05++	3.090E–05++	3.090E–05++
	16	3.090E–05++	3.090E–05++	3.090E–05++	1.080E–03++	3.090E–05++	3.090E–05++
	20	3.090E–05++	3.090E–05++	3.090E–05++	3.743E–02++	3.090E–05++	3.090E–05++
Patient 7	8	3.090E–05++	3.090E–05++	3.090E–05++	1.442E–04++	4.603E–05++	3.090E–05++
	12	3.090E–05++	3.090E–05++	3.090E–05++	4.137E–05++	3.090E–05++	3.090E–05++
	16	3.090E–05++	3.090E–05++	3.090E–05++	7.579E–04++	3.090E–05++	3.090E–05++
	20	3.090E–05++	3.090E–05++	3.090E–05++	5.026E–02–	3.090E–05++	3.090E–05++
Patient 9	8	3.090E–05++	3.090E–05++	3.090E–05++	4.779E–05++	8.725E–05++	3.090E–05++
	12	3.090E–05++	3.090E–05++	3.090E–05++	3.090E–05++	3.090E–05++	3.090E–05++
	16	3.090E–05++	3.090E–05++	3.090E–05++	9.660E–05++	3.090E–05++	3.090E–05++
	20	3.090E–05++	3.090E–05++	3.090E–05++	7.309E–03++	3.090E–05++	3.090E–05++
Patient 13	8	3.090E–05++	3.090E–05++	3.090E–05++	3.090E–05++	4.603E–05++	3.090E–05++
	12	3.090E–05++	3.090E–05++	3.090E–05++	4.779E–05++	3.578E–05++	3.090E–05++
	16	3.090E–05++	3.090E–05++	3.090E–05++	5.514E–05++	3.090E–05++	3.090E–05++
	20	3.090E–05++	3.090E–05++	3.090E–05++	9.660E–05++	3.090E–05++	3.090E–05++
Patient 24	8	3.090E–05++	3.090E–05++	3.090E–05++	4.603E–05++	3.090E–05++	3.090E–05++
	12	3.090E–05++	3.090E–05++	3.090E–05++	3.578E–05++	3.090E–05++	3.090E–05++
	16	3.090E–05++	3.090E–05++	3.090E–05++	3.090E–05++	3.090E–05++	3.090E–05++
	20	3.090E–05++	3.090E–05++	3.090E–05++	1.661E–04++	3.090E–05++	3.090E–05++
Patient 30	8	3.090E–05++	3.090E–05++	3.090E–05++	2.806E–04++	3.622E–04++	3.090E–05++
	12	3.090E–05++	3.090E–05++	3.090E–05++	4.108E–04++	3.578E–05++	3.090E–05++
	16	3.090E–05++	3.090E–05++	3.090E–05++	1.212E–03++	3.090E–05++	3.090E–05++
	20	3.090E–05++	3.090E–05++	3.090E–05++	3.578E–05++	3.090E–05++	3.090E–05++
Patient 37	8	3.090E–05++	3.090E–05++	3.086E–05++	4.489E–04++	1.221E–04++	3.090E–05++
	12	3.090E–05++	3.090E–05++	3.090E–05++	3.090E–05++	3.090E–05++	3.090E–05++
	16	3.090E–05++	3.090E–05++	3.090E–05++	3.578E–05++	3.090E–05++	3.090E–05++
	20	3.090E–05++	3.090E–05++	3.090E–05++	7.315E–05++	3.090E–05++	3.090E–05++
Patient 80	8	3.090E–05++	3.090E–05++	3.090E–05++	5.266E–04++	1.025E–04++	3.090E–05++
	12	3.090E–05++	3.090E–05++	3.090E–05++	4.276E–01–	3.090E–05++	3.090E–05++
	16	3.090E–05++	3.090E–05++	3.090E–05++	2.301E–01–	3.090E–05++	3.090E–05++
	20	3.090E–05++	3.090E–05++	3.090E–05++	8.091E–02++	3.090E–05++	3.090E–05++
Patient 121	8	3.090E–05++	3.090E–05++	3.090E–05++	3.090E–05++	1.103E–04++	3.090E–05++
	12	3.090E–05++	3.090E–05++	3.090E–05++	6.015E–03++	3.090E–05++	3.090E–05++
	16	3.090E–05++	3.090E–05++	3.090E–05++	8.411E–05++	3.090E–05++	3.090E–05++
	20	3.090E–05++	3.090E–05++	3.090E–05++	4.677E–02++	3.090E–05++	3.090E–05++

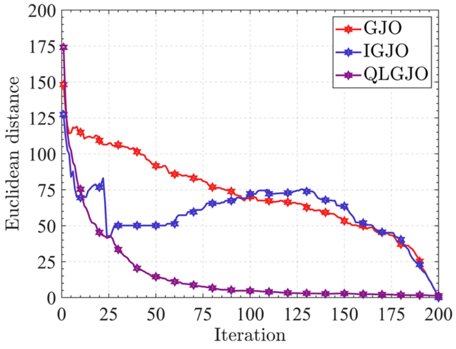


Fig. 22 Patient 3 with *th* 8

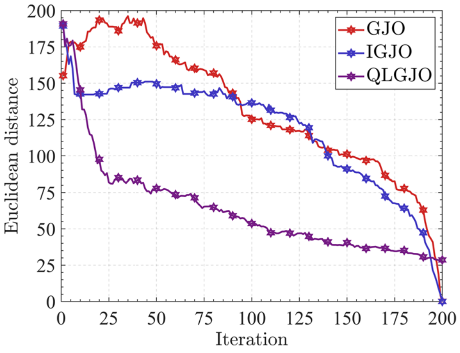


Fig. 25 Patient 3 with *th* 20

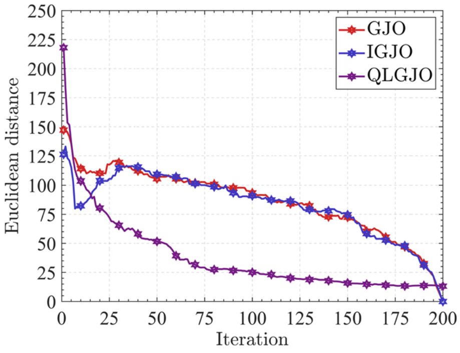


Fig. 23 Patient 3 with *th* 12

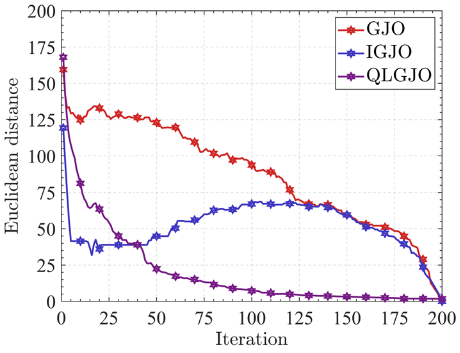


Fig. 26 Patient 4 with *th* 8

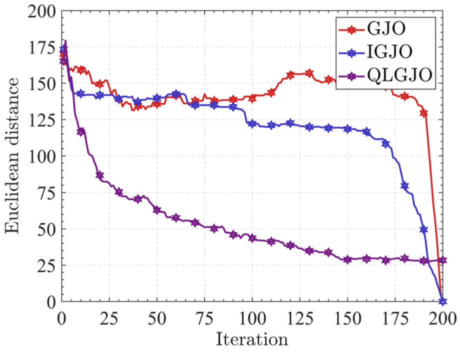


Fig. 24 Patient 3 with *th* 16

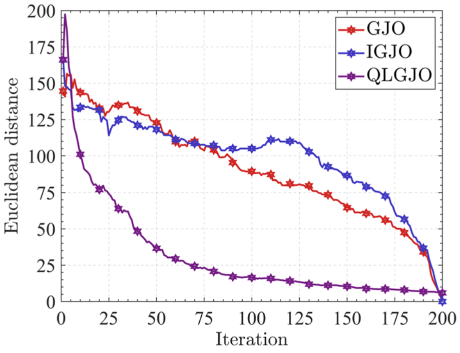


Fig. 27 Patient 4 with *th* 12

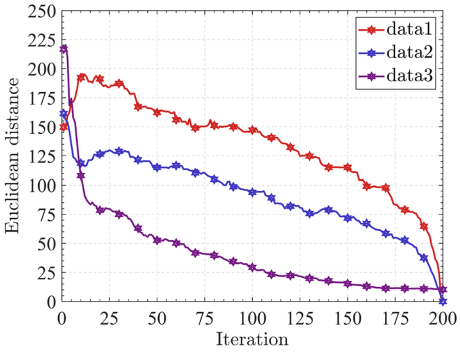


Fig. 28 Patient 4 with *th* 16

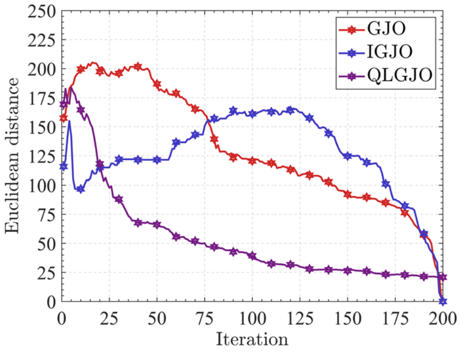


Fig. 29 Patient 4 with *th* 20

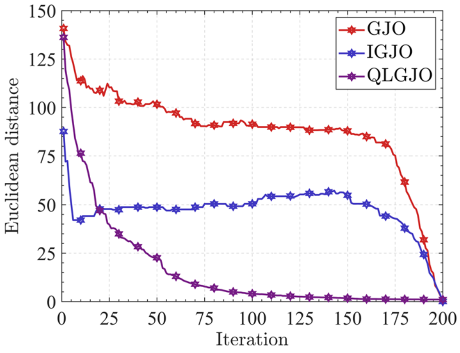


Fig. 30 Patient 5 with *th* 8

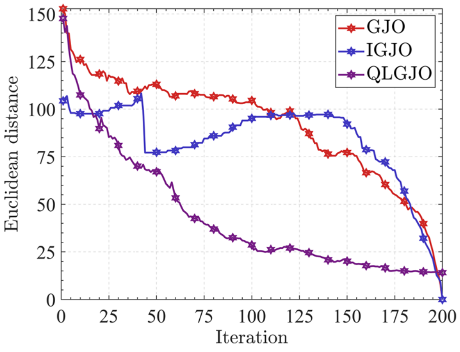


Fig. 31 Patient 5 with *th* 12

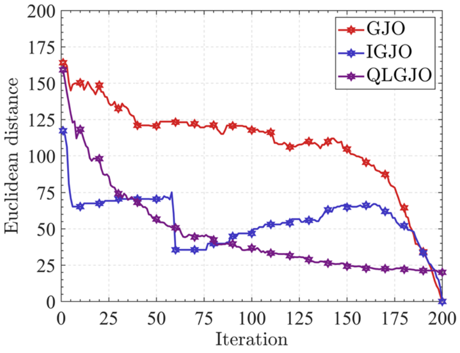


Fig. 32 Patient 5 with *th* 16

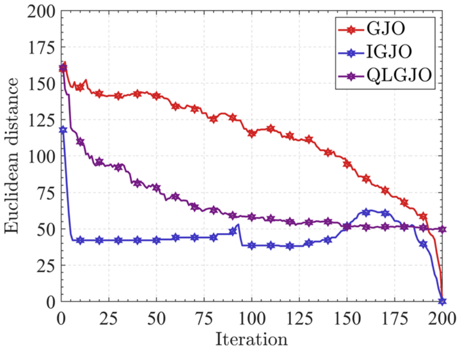


Fig. 33 Patient 5 with *th* 20

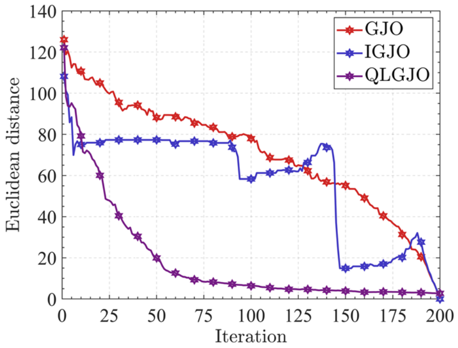


Fig. 34 Patient 6 with *th* 8

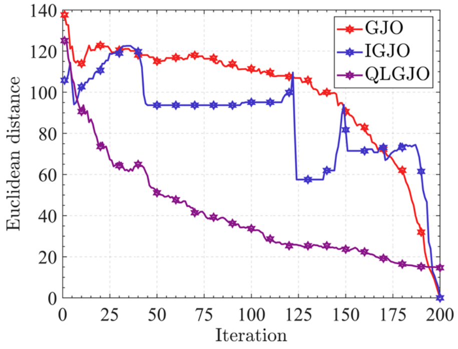


Fig. 35 Patient 6 with *th* 12

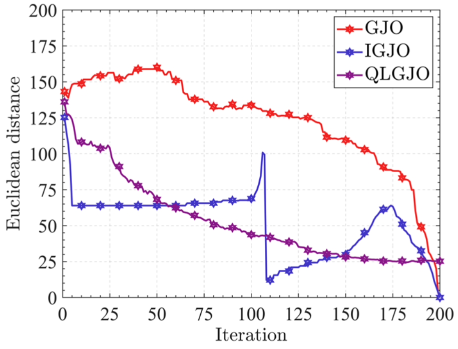


Fig. 36 Patient 6 with *th* 16

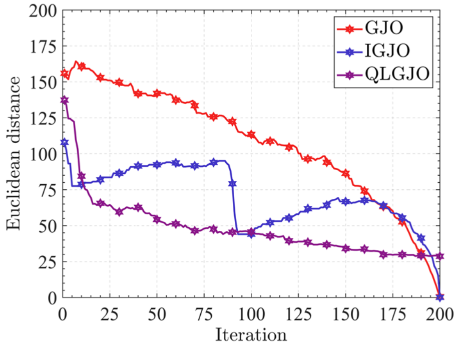


Fig. 37 Patient 6 with *th* 20

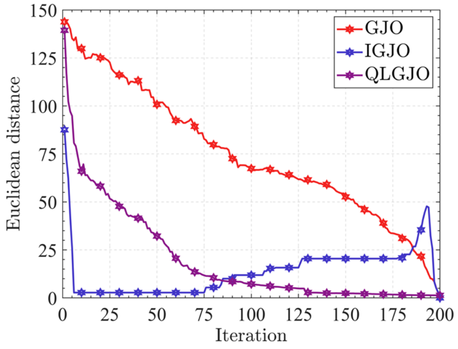


Fig. 38 Patient 7 with *th* 8

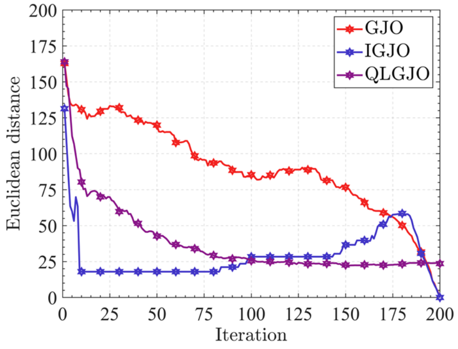


Fig. 39 Patient 7 with *th* 12

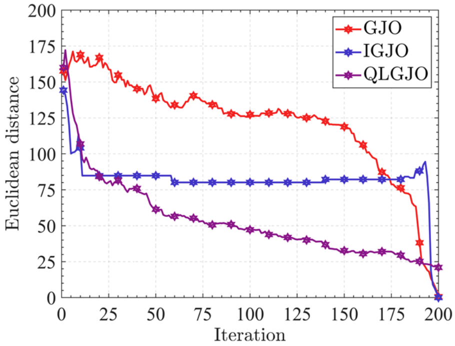


Fig. 40 Patient 7 with *th* 16

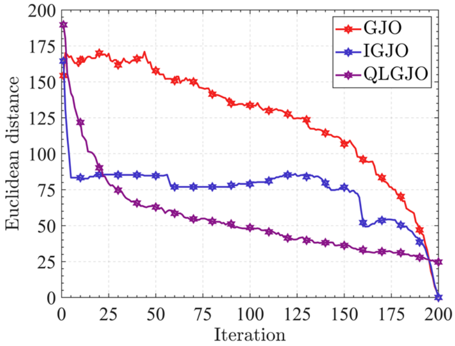


Fig. 41 Patient 7 with *th* 20

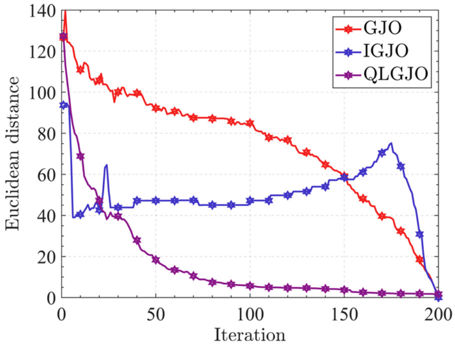


Fig. 42 Patient 9 with *th* 8

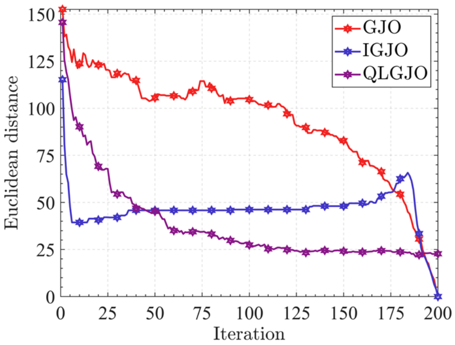


Fig. 43 Patient 9 with *th* 12

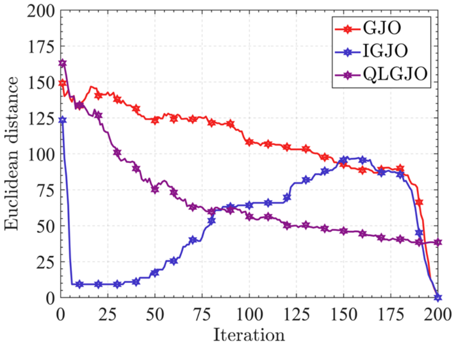


Fig. 44 Patient 9 with *th* 16

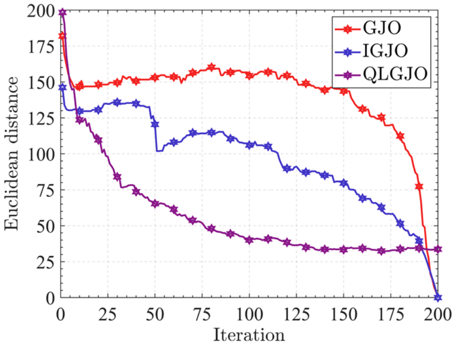


Fig. 45 Patient 9 with *th* 20

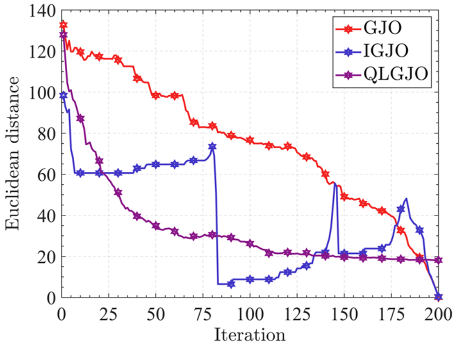


Fig. 46 Patient 13 with *th* 8

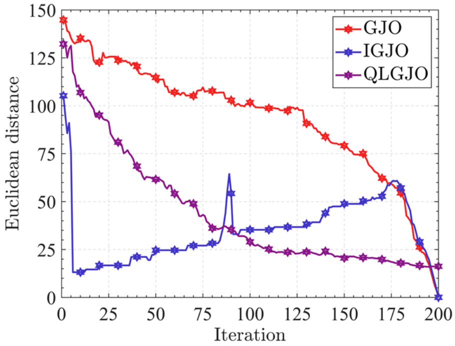


Fig. 47 Patient 13 with *th* 12

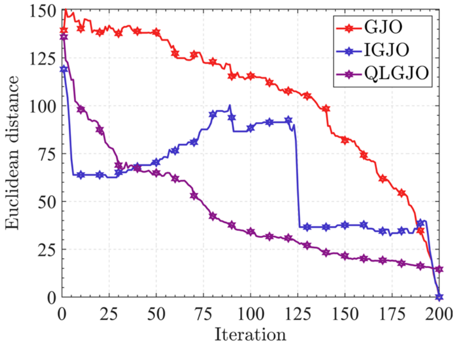


Fig. 48 Patient 13 with *th* 16

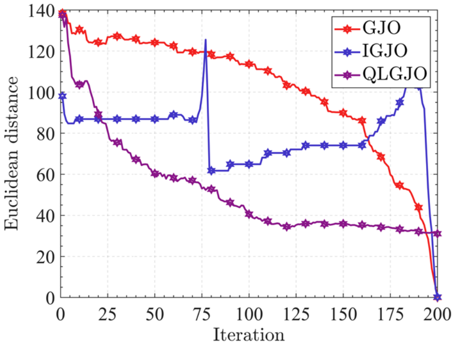


Fig. 49 Patient 13 with *th* 20

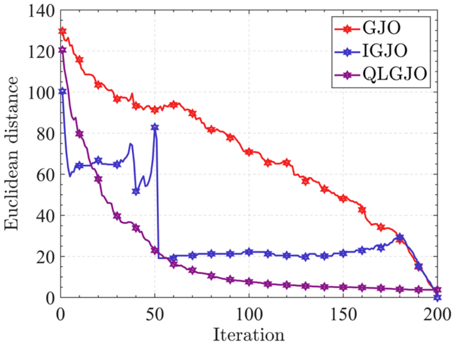


Fig. 50 Patient 24 with *th* 8

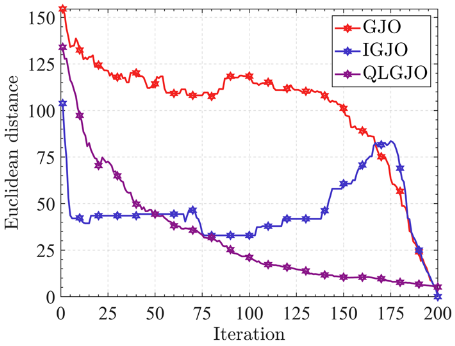


Fig. 51 Patient 24 with *th* 12

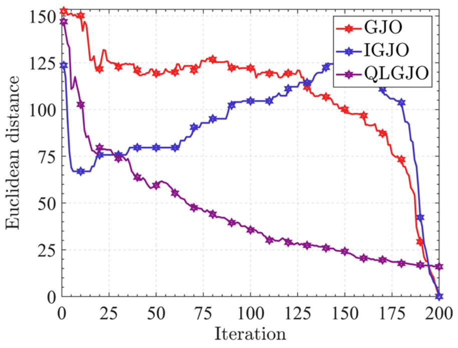


Fig. 52 Patient 24 with *th* 16

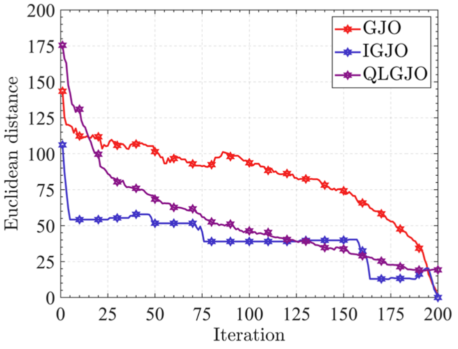


Fig. 55 Patient 30 with *th* 12

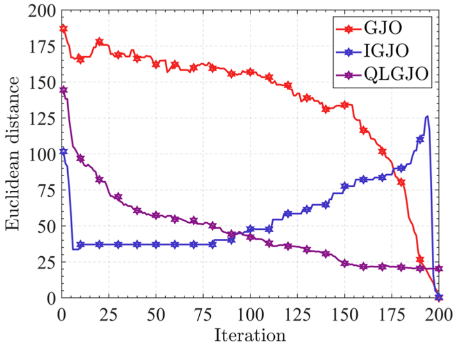


Fig. 53 Patient 24 with *th* 20

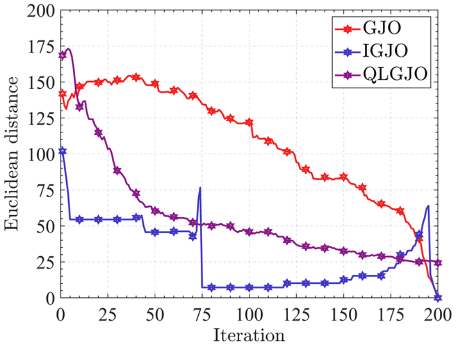


Fig. 56 Patient 30 with *th* 16

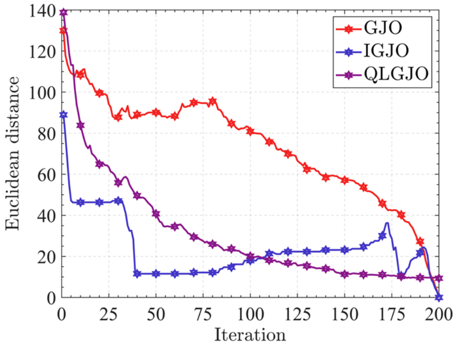


Fig. 54 Patient 30 with *th* 8

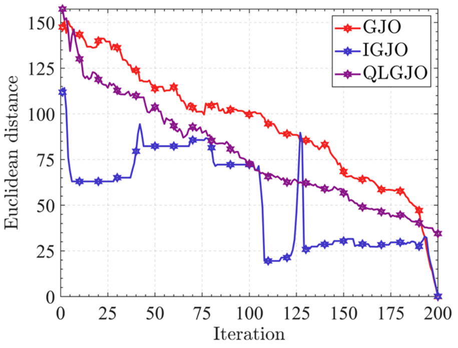


Fig. 57 Patient 30 with *th* 20

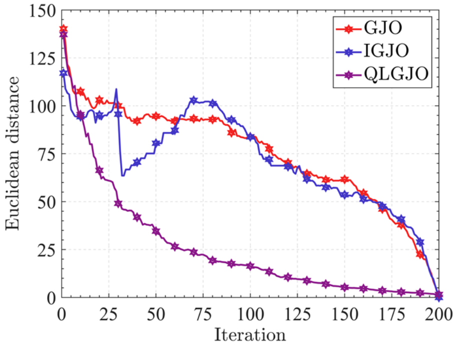


Fig. 58 Patient 37 with *th* 8

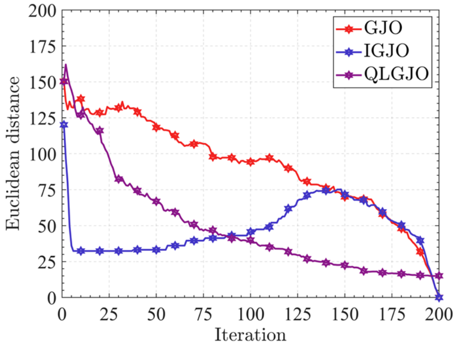


Fig. 59 Patient 37 with *th* 12

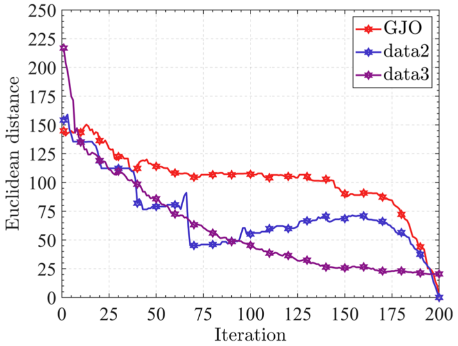


Fig. 60 Patient 37 with *th* 16

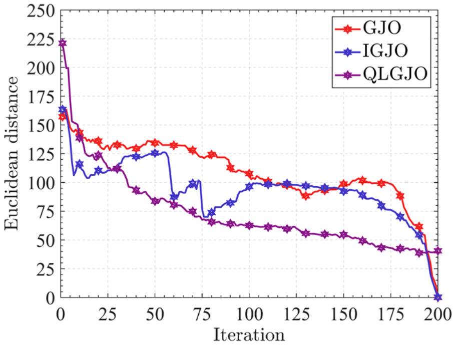


Fig. 61 Patient 37 with *th* 20

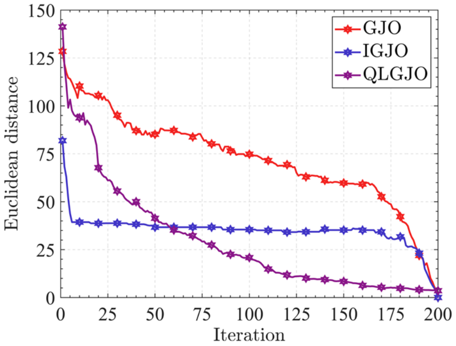


Fig. 62 Patient 80 with *th* 8

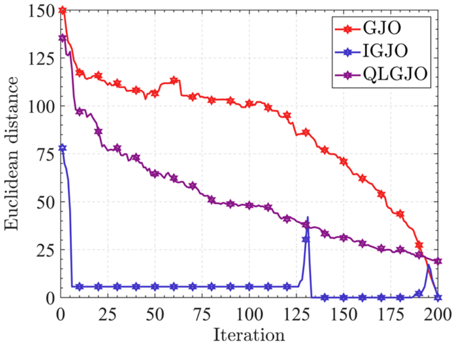


Fig. 63 Patient 80 with *th* 12

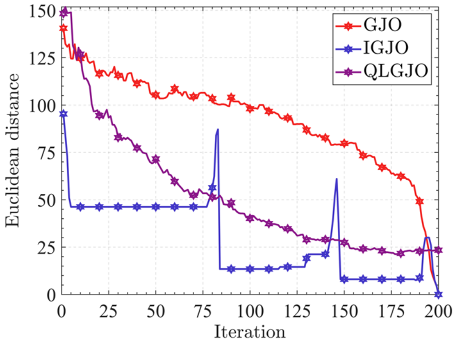


Fig. 64 Patient 80 with *th* 16

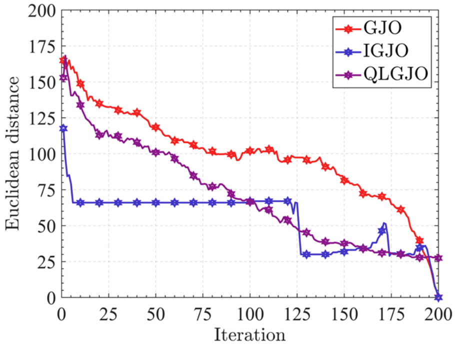


Fig. 65 Patient 80 with *th* 20

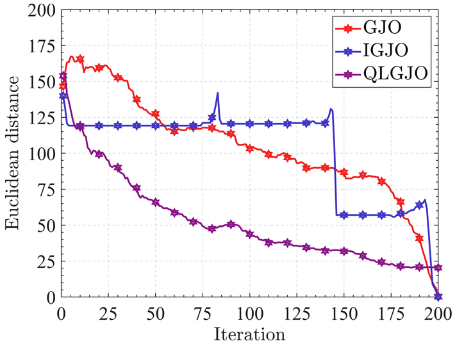


Fig. 66 Patient 121 with *th* 8

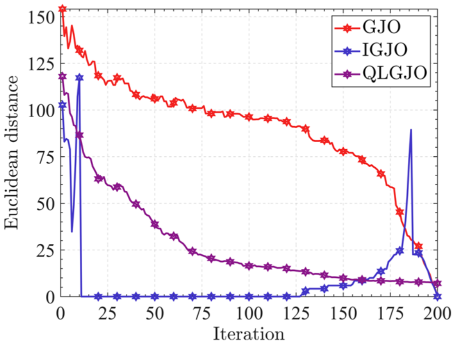


Fig. 67 Patient 121 with *th* 12

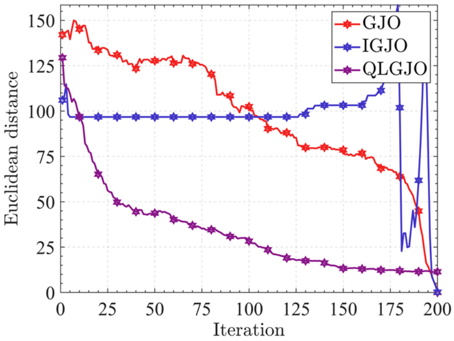


Fig. 68 Patient 121 with *th* 16

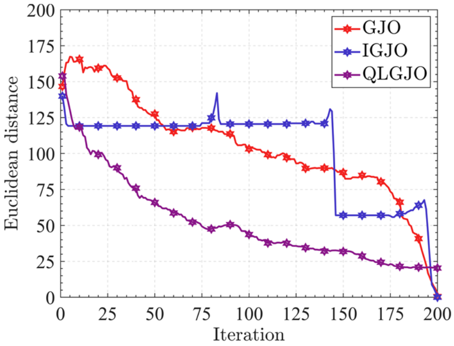


Fig. 69 Patient 121 with *th* 20

the additional overhead which is added by introducing reinforcement learning as well as to enhance the stability of the algorithm in the latter iterations, as mentioned in subSect. 4.3.2.

Acknowledgements The authors acknowledge support by the National Natural Science Foundation of China [grant numbers 21466008]; the Guangxi Natural Science Foundation, China [grant numbers 2019GXNSFAA185017]; the Scientific Research Project of Guangxi Minzu University [grant numbers 2021MDKJ004]; and the Innovation Project of Guangxi Graduate Education [grant numbers YCSW2022255]. In addition, the authors would like to thank the anonymous reviewers for providing their valuable insights that could improve the quality of this study.

Author Contributions All authors contributed to the study conception and design. Conceptualization, methodology, validation, formal analysis and coding were performed by ZW. Review, editing and supervision were performed by YM. Writing and editing were performed by MC. The first draft of the manuscript was written by ZW and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding This work was supported by the National Natural Science Foundation of China [grant numbers 21466008]; the Guangxi Natural Science Foundation, China [grant numbers 2019GXNSFAA185017]; the Scientific Research Project of Guangxi Minzu University [grant numbers 2021MDKJ004]; and the Innovation Project of Guangxi Graduate Education [grant numbers YCSW2022255].

Data Availability The COVID-19 datasets analyzed during the current study are available in the [COVID-CT-Dataset] repository, [https://arxiv.org/abs/2003.13865].

Declarations

Conflict of Interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Mathieu, E., Ritchie, H., Ortiz-Ospina, E., Roser, M., Hasell, J., Appel, C., Giattino, C., & Rodés-Guirao, L. (2021). A global database of COVID-19 vaccinations. *Nature Human Behaviour, 5*, 947–953. https://doi.org/10.1038/s41562-021-01122-8
2. Ndwandwe, D., & Wiysonge, C. S. (2021). COVID-19 vaccines. *Current Opinion in Immunology, 71*, 111–116. https://doi.org/10.1016/j.coi.2021.07.003
3. Yuki, K., Fujiogi, M., & Koutsogiannaki, S. (2020). COVID-19 pathophysiology: A review. *Clinical Immunology, 215*, 108427. https://doi.org/10.1016/j.clim.2020.108427
4. Cruz-Cárdenas, J., Zabelina, E., Guadalupe-Lanas, J., Palacio-Fierro, A., & Ramos-Galarza, C. (2021). COVID-19, consumer behavior, technology, and society: A literature review and bibliometric analysis. *Technological Forecasting and Social Change., 173*, 121179. https://doi.org/10.1016/j.techfore.2021.121179
5. Lee, D. Y. W., Li, Q. Y., Liu, J., & Efferth, T. (2021). Traditional Chinese herbal medicine at the forefront battle against COVID-19: Clinical experience and scientific basis. *Phytomedicine, 80*, 153337. https://doi.org/10.1016/j.phymed.2020.153337
6. Grodecki, K., Lin, A., Razipour, A., Cadet, S., McElhinney, P. A., Chan, C., Pressman, B. D., Julien, P., Maurovich-Horvat, P., Gaibazzi, N., Thakur, U., Mancini, E., Agalbato, C., Menè, R., Parati, G., Cernigliaro, F., Nerlekar, N., Torlasco, C., Pontone, G., … Dey, D. (2021). Epicardial adipose tissue is associated with extent of pneumonia and adverse outcomes in patients with COVID-19. *Metabolism, 115*, 154436. https://doi.org/10.1016/j.metabol.2020.154436
7. Caramello, V., Macciotta, A., De Salve, A. V., Mussa, A., De Leo, A. M., Bar, F., Panno, D., Nota, F., Ling, C. Y. G., Solitro, F., Ricceri, F., Sacerdote, C., & Boccuzzi, A. (2021). False-negative real-time polymerase chain reaction tests in COVID-19 patients: An epidemiological analysis of 302 patients. *Public Health, 200*, 84–90. https://doi.org/10.1016/j.puhe.2021.09.010
8. Jalaber, C., Lapotre, T., Morcet-Delattre, T., Ribet, F., Jouneau, S., & Lederlin, M. (2020). Chest CT in COVID-19 pneumonia: A review of current knowledge. *Diagnostic and Interventional Imaging, 101*, 431–437. https://doi.org/10.1016/j.diii.2020.06.001
9. Elaziz, M. A., Ewees, A. A., Yousri, D., Alwerfali, H. S. N., Awad, Q. A., Lu, S., & Al-Qaness, M. A. A. (2020). An improved marine predators algorithm with Fuzzy entropy for multi-level thresholding: Real world example of COVID-19 CT image segmentation. *IEEE Access, 8*, 125306–125330. https://doi.org/10.1109/ACCESS.2020.3007928
10. Gao, Y., Li, Q., Wang, S., & Gao, J. (2018). Adaptive neural network based on segmented particle swarm optimization for remote-sensing estimations of vegetation biomass. *Remote Sensing of Environment, 211*, 248–260. https://doi.org/10.1016/j.rse.2018.04.026
11. Houssein, E. H., Saber, E., Ali, A. A., & Wazery, Y. M. (2022). Centroid mutation-based search and rescue optimization algorithm for feature selection and classification. *Expert Systems with Applications, 191*, 116235. https://doi.org/10.1016/j.eswa.2021.116235
12. Houssein, E. H., Helmy, B. E., Oliva, D., Elngar, A. A., & Shaban, H. (2021). A novel black widow optimization algorithm for multilevel thresholding image segmentation. *Expert Systems with Applications, 167*, 114159. https://doi.org/10.1016/j.eswa.2020.114159
13. Houssein, E. H., Emam, M. M., & Ali, A. A. (2021). An efficient multilevel thresholding segmentation method for thermography breast cancer imaging based on improved chimp optimization algorithm. *Expert Systems with Applications, 185*, 115651. https://doi.org/10.1016/j.eswa.2021.115651
14. Kumar, M., Aggarwal, J., Rani, A., Stephan, T., Shankar, A., & Mirjalili, S. (2022). Secure video communication using firefly optimization and visual cryptography. *Artificial Intelligence Review, 55*, 2997–3017. https://doi.org/10.1007/s10462-021-10070-8
15. Zhi, X. H., & Shen, H. B. (2018). Saliency driven region-edge-based top down level set evolution reveals the asynchronous focus in image segmentation. *Pattern Recognition, 80*, 241–255. https://doi.org/10.1016/j.patcog.2018.03.010
16. Fang, J., Liu, H., Zhang, L., Liu, J., & Liu, H. (2021). Region-edge-based active contours driven by hybrid and local fuzzy region-based energy for image segmentation. *Information Sciences, 546*, 397–419. https://doi.org/10.1016/j.ins.2020.08.078
17. Zhao, S., Wang, P., Heidari, A. A., Chen, H., Turabieh, H., Mafarja, M., & Li, C. (2021). Multilevel threshold image segmentation with diffusion association slime mould algorithm and Renyi's entropy for chronic obstructive pulmonary disease. *Computers in Biology and Medicine, 134*, 104427. https://doi.org/10.1016/j.compbiomed.2021.104427
18. Parida, P., & Bhoi, N. (2018). Fuzzy clustering based transition region extraction for image segmentation. *Engineering Science and Technology, an International Journal, 21*, 547–563. https://doi.org/10.1016/j.jestch.2018.05.012
19. Zhao, S., Wang, P., Heidari, A. A., Chen, H., He, W., & Xu, S. (2021). Performance optimization of salp swarm algorithm for multi-threshold image segmentation: Comprehensive study of breast cancer microscopy. *Computers in Biology and Medicine, 139*, 105015. https://doi.org/10.1016/j.compbiomed.2021.105015
20. Zhao, D., Liu, L., Yu, F., Heidari, A. A., Wang, M., Liang, G., Muhammad, K., & Chen, H. (2021). Chaotic random spare ant colony optimization for multi-threshold image segmentation of 2D Kapur entropy. *Knowledge-Based Systems, 216*, 106510. https://doi.org/10.1016/j.knosys.2020.106510
21. Zhao, D., Liu, L., Yu, F., Heidari, A. A., Wang, M., Oliva, D., Muhammad, K., & Chen, H. (2021). Ant colony optimization with horizontal and vertical crossover search: Fundamental visions for multi-threshold image segmentation. *Expert Systems with Applications, 167*, 114122. https://doi.org/10.1016/j.eswa.2020.114122
22. Yan, Z., Zhang, J., Yang, Z., & Tang, J. (2021). Kapur's entropy for underwater multilevel thresholding image segmentation based on whale optimization algorithm. *IEEE Access, 9*, 41294–41319. https://doi.org/10.1109/ACCESS.2020.3005452

23. Wang, Y., Zhang, G., & Zhang, X. (2019). Multilevel image thresholding using Tsallis entropy and cooperative pigeon-inspired optimization bionic algorithm. *Journal of Bionic Engineering, 16*, 954–964. https://doi.org/10.1007/s42235-019-0109-1
24. Sharma, A., Chaturvedi, R., Kumar, S., & Dwivedi, U. K. (2020). Multi-level image thresholding based on Kapur and Tsallis entropy using firefly algorithm. *Journal of Interdisciplinary Mathematics, 23*, 563–571. https://doi.org/10.1080/09720502.2020.1731976
25. Bansal, J. C. (2019). Particle swarm optimization. *Evolutionary and Swarm Intelligence Algorithms, 779*, 11–23. https://doi.org/10.1007/978-3-319-91341-4_2
26. Storn, R., & Price, K. (1997). Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization, 11*, 341–359. https://doi.org/10.1023/A:1008202821328
27. Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software, 69*, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007
28. Wang, G. G., Zhao, X., & Deb, S. (2015). A novel monarch butterfly optimization with greedy strategy and self-adaptive. *Second International Conference on Soft Computing and Machine Intelligence (ISCMI), 2015*, 45–50. https://doi.org/10.1109/ISCMI.2015.19
29. Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software, 95*, 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008
30. Mirjalili, S., Mirjalili, S. M., & Hatamlou, A. (2016). Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Computing and Applications, 27*, 495–513. https://doi.org/10.1007/s00521-015-1870-7
31. Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems, 97*, 849–872. https://doi.org/10.1016/j.future.2019.02.028
32. Hayyolalam, V., & Pourhaji Kazem, A. A. (2020). Black widow optimization algorithm: A novel meta-heuristic approach for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence, 87*, 103249. https://doi.org/10.1016/j.engappai.2019.103249
33. Li, S., Chen, H., Wang, M., Heidari, A. A., & Mirjalili, S. (2020). Slime mould algorithm: A new method for stochastic optimization. *Future Generation Computer Systems, 111*, 300–323. https://doi.org/10.1016/j.future.2020.03.055
34. Yang, Y., Chen, H., Heidari, A. A., & Gandomi, A. H. (2021). Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts. *Expert Systems with Applications, 177*, 114864. https://doi.org/10.1016/j.eswa.2021.114864
35. Ahmadianfar, I., Heidari, A. A., Gandomi, A. H., Chu, X., & Chen, H. (2021). RUN beyond the metaphor: An efficient optimization algorithm based on Runge Kutta method. *Expert Systems with Applications, 181*, 115079. https://doi.org/10.1016/j.eswa.2021.115079
36. Faramarzi, A., Heidarinejad, M., Mirjalili, S., & Gandomi, A. H. (2020). Marine predators algorithm: A nature-inspired metaheuristic. *Expert Systems with Applications, 152*, 113377. https://doi.org/10.1016/j.eswa.2020.113377
37. Ahmadianfar, I., Heidari, A. A., Noshadian, S., Chen, H., & Gandomi, A. H. (2022). INFO: An efficient optimization algorithm based on weighted mean of vectors. *Expert Systems with Applications, 195*, 116516. https://doi.org/10.1016/j.eswa.2022.116516
38. Su, H., Zhao, D., Heidari, A. A., Liu, L., Zhang, X., Mafarja, M., & Chen, H. (2023). RIME: A physics-based optimization. *Neurocomputing, 532*, 183–214. https://doi.org/10.1016/j.neucom.2023.02.010
39. Chopra, N., & Mohsin Ansari, M. (2022). Golden jackal optimization: A novel nature-inspired optimizer for engineering applications. *Expert Systems with Applications, 198*, 116924. https://doi.org/10.1016/j.eswa.2022.116924
40. Rezaie, M., Karamnejadiazar, K., Kardansani, A., Akbari, E., Ghadimi, N., Razmjooy, N., & Ghadamyari, M. (2022). Model parameters estimation of the proton exchange membrane fuel cell by a modified golden jackal optimization. *Sustainable Energy Technologies and Assessments, 53*, 102657. https://doi.org/10.1016/j.seta.2022.102657
41. Houssein, E. H., Abdelkareem, D. A., Emam, M. M., Hameed, M. A., & Younan, M. (2022). An efficient image segmentation method for skin cancer imaging using improved golden jackal optimization algorithm. *Computers in Biology and Medicine, 149*, 106075. https://doi.org/10.1016/j.compbiomed.2022.106075
42. Zhang, J., Zhang, G., Kong, M., & Zhang, T. (2023). Adaptive infinite impulse response system identification using an enhanced golden jackal optimization. *The Journal of Supercomputing*. https://doi.org/10.1007/s11227-023-05086-6
43. Huo, F., Sun, X., & Ren, W. (2020). Multilevel image threshold segmentation using an improved bloch quantum artificial bee colony algorithm. *Multimedia Tools and Applications, 79*, 2447–2471. https://doi.org/10.1007/s11042-019-08231-7
44. Xing, Z. (2020). An improved emperor penguin optimization based multilevel thresholding for color image segmentation. *Knowledge-Based Systems, 194*, 105570. https://doi.org/10.1016/j.knosys.2020.105570
45. Upadhyay, P., & Chhabra, J. K. (2020). Kapur's entropy based optimal multilevel image segmentation using crow search algorithm. *Applied Soft Computing, 97*, 105522. https://doi.org/10.1016/j.asoc.2019.105522
46. Abd Elaziz, M., Nabil, N., Moghdani, R., Ewees, A. A., Cuevas, E., & Lu, S. (2021). Multilevel thresholding image segmentation based on improved volleyball premier league algorithm using whale optimization algorithm. *Multimedia Tools and Applications, 80*, 12435–12468. https://doi.org/10.1007/s11042-020-10313-w
47. Abd Elaziz, M., Mohammadi, D., Oliva, D., & Salimifard, K. (2021). Quantum marine predators algorithm for addressing multilevel image segmentation. *Applied Soft Computing, 110*, 107598. https://doi.org/10.1016/j.asoc.2021.107598
48. Houssein, E. H., Helmy, B.E.-D., Elngar, A. A., Abdelminaam, D. S., & Shaban, H. (2021). An improved tunicate swarm algorithm for global optimization and image segmentation. *IEEE Access, 9*, 56066–56092. https://doi.org/10.1109/ACCESS.2021.3072336
49. Houssein, E. H., Hussain, K., Abualigah, L., Elaziz, M. A., Alomoush, W., Dhiman, G., Djenouri, Y., & Cuevas, E. (2021). An improved opposition-based marine predators algorithm for global optimization and multilevel thresholding image segmentation. *Knowledge-Based Systems, 229*, 107348. https://doi.org/10.1016/j.knosys.2021.107348
50. Houssein, E. H., Helmy, B. E., Oliva, D., Jangir, P., Premkumar, M., Elngar, A. A., & Shaban, H. (2022). An efficient multi-thresholding based COVID-19 CT images segmentation approach using an improved equilibrium optimizer. *Biomedical Signal Processing and Control, 73*, 103401. https://doi.org/10.1016/j.bspc.2021.103401
51. Abualigah, L., Al-Okbi, N. K., Elaziz, M. A., & Houssein, E. H. (2022). Boosting marine predators algorithm by salp swarm algorithm for multilevel thresholding image segmentation. *Multimedia Tools and Applications, 81*, 16707–16742. https://doi.org/10.1007/s11042-022-12001-3
52. Liu, L., Zhao, D., Yu, F., Heidari, A. A., Li, C., Ouyang, J., Chen, H., Mafarja, M., Turabieh, H., & Pan, J. (2021). Ant colony optimization with cauchy and greedy levy mutations for multilevel COVID 19 X-ray image segmentation. *Computers in Biology and*

Medicine, 136, 104609. https://doi.org/10.1016/j.compbiomed.2021.104609
53. Bhandari, A. K. (2020). A novel beta differential evolution algorithm-based fast multilevel thresholding for color image segmentation. *Neural Computing and Applications, 32*, 4583–4613. https://doi.org/10.1007/s00521-018-3771-z
54. Wu, B., Zhou, J., Ji, X., Yin, Y., & Shen, X. (2020). An ameliorated teaching–learning-based optimization algorithm based study of image segmentation for multilevel thresholding using Kapur's entropy and Otsu's between class variance. *Information Sciences, 533*, 72–107. https://doi.org/10.1016/j.ins.2020.05.033
55. He, L., & Huang, S. (2020). An efficient krill herd algorithm for color image multilevel thresholding segmentation problem. *Applied Soft Computing, 89*, 106063. https://doi.org/10.1016/j.asoc.2020.106063
56. Ren, L., Zhao, D., Zhao, X., Chen, W., Li, L., Wu, T., Liang, G., Cai, Z., & Xu, S. (2022). Multi-level thresholding segmentation for pathological images: Optimal performance design of a new modified differential evolution. *Computers in Biology and Medicine, 148*, 105910. https://doi.org/10.1016/j.compbiomed.2022.105910
57. Hosny, K. M., Khalid, A. M., Hamza, H. M., & Mirjalili, S. (2022). Multilevel segmentation of 2D and volumetric medical images using hybrid coronavirus optimization algorithm. *Computers in Biology and Medicine, 150*, 106003. https://doi.org/10.1016/j.compbiomed.2022.106003
58. Wei, Z., Liu, L., Kuang, F., Li, L., Xu, S., & Liang, Y. (2022). An efficient multi-threshold image segmentation for skin cancer using boosting whale optimizer. *Computers in Biology and Medicine, 151*, 106227. https://doi.org/10.1016/j.compbiomed.2022.106227
59. Emam, M. M., Houssein, E. H., & Ghoniem, R. M. (2023). A modified reptile search algorithm for global optimization and image segmentation: Case study brain MRI images. *Computers in Biology and Medicine, 152*, 106404. https://doi.org/10.1016/j.compbiomed.2022.106404
60. Han, Y., Chen, W., Heidari, A. A., & Chen, H. (2023). Multi-verse optimizer with rosenbrock and diffusion mechanisms for multilevel threshold image segmentation from COVID-19 chest X-Ray images. *Journal of Bionic Engineering*. https://doi.org/10.1007/s42235-022-00295-w
61. Xing, J., Zhao, H., Chen, H., Deng, R., & Xiao, L. (2023). Boosting whale optimizer with quasi-oppositional learning and gaussian barebone for feature selection and COVID-19 image segmentation. *Journal of Bionic Engineering, 20*, 797–818. https://doi.org/10.1007/s42235-022-00297-8
62. Zhou, J., Qiu, Y., Zhu, S., Armaghani, D. J., Li, C., Nguyen, H., & Yagiz, S. (2021). Optimization of support vector machine through the use of metaheuristic algorithms in forecasting TBM advance rate. *Engineering Applications of Artificial Intelligence, 97*, 104015. https://doi.org/10.1016/j.engappai.2020.104015
63. Majdoubi, E. L., & O., Abdoun F., Abdoun O. (2021). A new optimized approach to resolve a combinatorial problem: CoronaVirus optimization algorithm and self-organizing maps. *Digital Technologies and Applications, 211*, 947–957. https://doi.org/10.1007/978-3-030-73882-2_86
64. Seyyedabbasi, A., Aliyev, R., Kiani, F., Gulle, M. U., Basyildiz, H., & Shah, M. A. (2021). Hybrid algorithms based on combining reinforcement learning and metaheuristic methods to solve global optimization problems. *Knowledge-Based Systems, 223*, 107044. https://doi.org/10.1016/j.knosys.2021.107044
65. Yin, L., & Li, S. (2021). Hybrid metaheuristic multi-layer reinforcement learning approach for two-level energy management strategy framework of multi-microgrid systems. *Engineering Applications of Artificial Intelligence, 104*, 104326. https://doi.org/10.1016/j.engappai.2021.104326
66. Qu, C., Gai, W., Zhong, M., & Zhang, J. (2020). A novel reinforcement learning based grey wolf optimizer algorithm for unmanned aerial vehicles (UAVs) path planning. *Applied Soft Computing, 89*, 106099. https://doi.org/10.1016/j.asoc.2020.106099
67. Zamfirache, I. A., Precup, R.-E., Roman, R.-C., & Petriu, E. M. (2022). Policy iteration reinforcement learning-based control using a grey wolf optimizer algorithm. *Information Sciences, 585*, 162–175. https://doi.org/10.1016/j.ins.2021.11.051
68. Sadeg, S., Hamdad, L., Remache, A. R., Karech, M. N., Benatchba, K., & Habbas, Z. (2019). QBSO-FS: A reinforcement learning based bee swarm optimization metaheuristic for feature selection. *Advances in Computational Intelligence, 11507*, 785–796. https://doi.org/10.1007/978-3-030-20518-8_65
69. Chen, R., Yang, B., Li, S., & Wang, S. (2020). A self-learning genetic algorithm based on reinforcement learning for flexible job-shop scheduling problem. *Computers & Industrial Engineering, 149*, 106778. https://doi.org/10.1016/j.cie.2020.106778
70. Huynh-Thu, Q., & Ghanbari, M. (2008). Scope of validity of PSNR in image/video quality assessment. *Electronics Letters, 44*, 800–801.
71. Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing, 13*, 600–612. https://doi.org/10.1109/TIP.2003.819861
72. Zhang, L., Zhang, L., Mou, X., & Zhang, D. (2011). FSIM: A feature similarity index for image quality assessment. *IEEE Transactions on Image Processing, 20*, 2378–2386. https://doi.org/10.1109/TIP.2011.2109730
73. Huang, R. (2008). Some inequalities for the hadamard product and the fan product of matrices. *Linear Algebra and its Applications, 428*, 1551–1559. https://doi.org/10.1016/j.laa.2007.10.001
74. Mantegna, R. N. (1994). Fast, accurate algorithm for numerical simulation of lévy stable stochastic processes. *Physical Review E, 49*, 4677–4683. https://doi.org/10.1103/PhysRevE.49.4677
75. Sun, J., Gong, M., Zhao, Y., Han, C., Jing, L., & Yang, P. (2022). A hybrid deep reinforcement learning ensemble optimization model for heat load energy-saving prediction. *Journal of Building Engineering, 58*, 105031. https://doi.org/10.1016/j.jobe.2022.105031
76. Dolcetta, I. C., & Ishii, H. (1984). Approximate solutions of the bellman equation of deterministic control theory. *Applied Mathematics and Optimization, 11*, 161–181. https://doi.org/10.1007/BF01442176
77. Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE transactions on systems, 9*, 62–66.
78. Kapur, J. N., Sahoo, P. K., & Wong, A. K. C. (1985). A new method for gray-level picture thresholding using the entropy of the histogram. *Computer Vision, Graphics, and Image Processing, 29*, 273–285. https://doi.org/10.1016/0734-189X(85)90125-2
79. Yang, X., He, X., Zhao, J., Zhang, Y., Zhang, S., Xie, P. (2020). COVID-CT-dataset: A CT Scan dataset about COVID-19. arXiv preprint http://arxiv.org/abs/2003.13865
80. Arcuri, A., & Fraser, G. (2013). Parameter tuning or default values? An empirical investigation in search-based software engineering. *Empirical Software Engineering, 18*, 594–623. https://doi.org/10.1007/s10664-013-9249-9
81. Khorram, B., & Yazdi, M. (2019). A new optimized thresholding method using ant colony algorithm for MR brain image segmentation. *Journal of Digital Imaging, 32*, 162–174. https://doi.org/10.1007/s10278-018-0111-x
82. Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association, 32*, 675–701. https://doi.org/10.1080/01621459.1937.10503522
83. Abdel-Basset, M., Chang, V., & Mohamed, R. (2021). A novel equilibrium optimization algorithm for multi-thresholding image

segmentation problems. *Neural Computing and Applications, 33*, 10685–10718. https://doi.org/10.1007/s00521-020-04820-y

84. Karunasingha, D. S. K. (2022). Root mean square error or mean absolute error? Use their ratio as well. *Information Sciences, 585*, 609–629. https://doi.org/10.1016/j.ins.2021.11.036
85. Ewees, A. A., Abd Elaziz, M., Al-Qaness, M. A. A., Khalil, H. A., & Kim, S. (2020). Improved artificial bee colony using sine-cosine algorithm for multi-level thresholding image segmentation. *IEEE Access, 8*, 26304–26315. https://doi.org/10.1109/ACCESS.2020.2971249
86. Mousavirad, S. J., & Ebrahimpour-Komleh, H. (2020). Human mental search-based multilevel thresholding for image segmentation. *Applied Soft Computing, 97*, 105427. https://doi.org/10.1016/j.asoc.2019.04.002
87. Kovesi, P. (2000). Phase congruency: A low-level image invariant. *Psychological Research Psychologische Forschung, 64*, 136–148. https://doi.org/10.1007/s004260000024
88. Xue, W., Zhang, L., Mou, X., & Bovik, A. C. (2014). Gradient magnitude similarity deviation: A highly efficient perceptual image quality index. *IEEE Transactions on Image Processing, 23*, 684–695. https://doi.org/10.1109/TIP.2013.2293423
89. Saary, M. J. (2008). Radar plots: A useful way for presenting multivariate health care data. *Journal of Clinical Epidemiology, 61*, 311–317. https://doi.org/10.1016/j.jclinepi.2007.04.021

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