


Cooperative Swarm Intelligence Algorithms for Adaptive Multilevel Thresholding Segmentation of COVID-19 CT-Scan Images


Muath Sabha

(Information Technology Department, Faculty of Engineering and Information Technology, Arab American University, Jenin, Palestine

 <https://orcid.org/0000-0002-0496-6502>, muath.sabha@aaup.edu)


Thaer Thaher

(Computer Systems Engineering Department, Arab American University, Jenin, Palestine

 <https://orcid.org/0000-0001-7047-5886>, Thaer.Thaher@aaup.edu)

Marwa M. Emam

(Faculty of Computers and Information, Minia University, Minia, Egypt

 <https://orcid.org/0000-0001-7399-6839>, marwa.khalef@mu.edu.eg)

Abstract: The Coronavirus Disease 2019 (COVID-19) is widespread throughout the world and poses a serious threat to public health and safety. A COVID-19 infection can be recognized using computed tomography (CT) scans. To enhance the categorization, some image segmentation techniques are presented to extract regions of interest from COVID-19 CT images. Multi-level thresholding (MLT) is one of the simplest and most effective image segmentation approaches, especially for grayscale images like CT scan images. Traditional image segmentation methods use histogram approaches; however, these approaches encounter some limitations. Now, swarm intelligence inspired meta-heuristic algorithms have been applied to resolve MLT, deemed an NP-hard optimization task. Despite the advantages of using meta-heuristics to solve global optimization tasks, each approach has its own drawbacks. However, the common flaw for most meta-heuristic algorithms is that they are unable to maintain the diversity of their population during the search, which means they might not always converge to the global optimum. This study proposes a cooperative swarm intelligence-based MLT image segmentation approach that hybridizes the advantages of parallel meta-heuristics and MLT for developing an efficient image segmentation method for COVID-19 CT images. An efficient cooperative model-based meta-heuristic called the CPGH is developed based on three practical algorithms: particle swarm optimization (PSO), grey wolf optimizer (GWO), and Harris hawks optimization (HHO). In the cooperative model, the applied algorithms are executed concurrently, and a number of potential solutions are moved across their populations through a procedure called migration after a set number of generations. The CPGH model can solve the image segmentation problem using MLT image segmentation. The proposed CPGH is evaluated using three objective functions, cross-entropy, Otsu's, and Tsallis, over the COVID-19 CT images selected from open-sourced datasets. Various evaluation metrics covering peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), and universal quality image index (UQI) were employed to quantify the segmentation quality. The overall ranking results of the segmentation quality metrics indicate that the performance of the proposed CPGH is better than conventional PSO, GWO, and HHO algorithms and other state-of-the-art methods for MLT image segmentation. On the tested COVID-19 CT images, the CPGH offered an average PSNR of 24.8062, SSIM of 0.8818, and UQI of 0.9097 using 20 thresholds.

Keywords: Swarm Intelligence algorithms, Image segmentation, Computer Vision, Multilevel thresholding, Metaheuristics

1 Introduction

The coronavirus disease 2019 (COVID-19) is a severe infectious disease that has expanded worldwide and is growing at an exponential rate [Ferrer, 2020]. COVID-19 has affected many lives; it contributed to millions of deaths and left many with respiratory tract complications and fatalities (i.e., COVID-19 pneumonia, Acute Respiratory Distress Syndrome (ARDS), and acute respiratory failure). On March 11, 2020, World Health Organization (WHO) declared the novel COVID-19 outbreak a global pandemic. It emphasized the importance of a global union to provide approaches to reduce morbidity, mortality, and burden on health care systems [Sohrabi et al., 2020]. Accordingly, there is a need for technologies to diagnose this lethal disease quickly. Utilizing a variety of tests to detect COVID-19 is one of the severe issues. The most often utilized test is the real-time polymerase chain reaction (real-time PCR) is commonly used to measure gene expression. The diagnosis is false-negative, takes a long time, and is invasive. Another test that is crucial to the diagnosis of COVID-19 is chest computed tomography (CT). The pathophysiology described by CT can be used as a guide for the diagnosis and progression of particular disease stages. It has developed to become a helpful diagnostic technique for treating COVID-19-related lung disease [Harmon et al., 2020]. Chest CT shows a high detection sensitivity for a lung disease associated with COVID-19, according to preliminary studies.

In computer vision-related applications such as medical imaging, geographical imaging, autonomous target recognition, robotic vision, and many others, image segmentation is deemed the initial and essential process for analyzing and interpreting the captured image [Baby Resma and Nair, 2021, Houssein et al., 2022d, Houssein et al., 2021d]. Medical imaging approaches are vital in diagnosing complex diseases and in the patient's healthcare. They help doctors diagnose, treat, and early detect life-threatening diseases. Much detailed information can be extracted from chest CT images; however, manual processing of these images is inaccurate. For this reason, several algorithms were developed to assist in identifying and diagnosing COVID-19. Multilevel thresholding image segmentation (MIS) is one of the practical approaches that have been applied to improve the process of COVID-19 detection [Houssein et al., 2021a, Rodríguez et al., 2022, Houssein et al., 2022c, Houssein et al., 2022b].

Image segmentation is one of the critical stages for processing medical images, and it has been widely employed in several medical applications. Multi-level thresholding (MLT) is a simple and powerful image segmentation technique. Some conventional methods use histogram thresholding approaches; however, these approaches encounter some difficulties. Meta-heuristic algorithms have been leveraged in MLT, which is considered an NP-hard problem [Abualigah et al., 2021b, Elaziz et al., 2020]. In common, image segmentation can be defined as partitioning an image into multiple image segments, also known as image regions or objects, based on various criteria such as gray level values, color, shape, or textures [Baby Resma and Nair, 2021, Rodríguez-Esparza et al., 2020]. More precisely, image segmentation is assigning a label to each pixel in an image so that pixels with the same label share specific properties. Most image segmentation approaches are based on similarity and discontinuity, two critical features of intensity values. The similarity technique, which relies on similarity among image objects with pre-determined criteria for partitioning, is widely employed [Baby Resma and Nair, 2021].

In general, there are four popular types of image segmentation techniques. They can be categorized into clustering-based approaches, merging and region-based split approaches, histogram thresholding-based approaches, and texture analysis-based approaches [He and Huang, 2017]. The prime similarity-based techniques include thresholding, region expanding, region splitting, and merging. Considering all existing segmentation ap-

proaches, the thresholding approach is the best because they are accurate, simple, and resilient [Merzban and Elbayoumi, 2019]. A threshold-based segmentation technique subdivides an image into smaller segments by determining their boundaries using at least one gray-level value. In addition, bi-level and MLT are other types of thresholding approaches. Bi-level techniques utilize a single threshold value for splitting an image into two sections (e.g., homogeneous foreground and background). In contrast, MLT will find multiple gray-level threshold values to differentiate the objects of interest from the image's background.

In recent years, several thresholding-based segmentation approaches have been developed. For instance, M. Sezgin et al. [Sezgin and Sankur, 2004] conducted a survey and discovered that global histogram-based techniques are frequently employed to estimate the threshold values in MLT. Otsu's approach is based on the idea of between-class variance [Otsu, 1979a], and Kapur's approach relies on the concept of entropy [Kapur et al., 1985a], which are two of the best image thresholding methods. They are used to identify the best thresholds for dividing the region of gray-level values in an image, depending on some pre-defined criteria. In Otsu's technique, maximizing the between-class variance of gray levels of the histogram is employed to determine the most appropriate threshold values. In contrast, in Kapur's method, histogram entropy is maximized. Otsu's and Kapur's techniques have been proposed to deal with the bi-level thresholding problem. However, both exploit an exhaustive search algorithmic paradigm to optimize the objective function, and thus the computational time grows exponentially with the number of threshold values. For this reason, they can not be used to resolve MLT problems [Baby Resma and Nair, 2021]. Previous studies reveal that several techniques have been developed to boost efficiency and decrease the time complexity of MLT approaches. For example, a faster form of Otsu's method that integrates a recursive algorithm with a look-up table has been proposed by Liao et al. [Liao et al., 2001] to lessen the long processing time needed for obtaining the ideal threshold values. However, when the number of thresholds is increased, the approach will again take a long time to compute [Song et al., 2017].

The presence of a large number of image thresholds makes the selection of thresholding values a crucial step when performing image segmentation. Therefore, MLT is viewed as a challenging optimization problem that must be resolved. In general, research findings have demonstrated the ability of meta-heuristic algorithms to solve various challenging optimization problems in different areas such as bio-informatics, engineering, communications, drug design, and feature selection [Thaher and Arman, 2020]. In contrast to deterministic approaches, meta-heuristic techniques employ search agents to explore a problem landscape. Each search agent in the population acts as a candidate solution that is adjusted repeatedly in an iterative manner using heuristic operators. These operators provide diverse search behaviors when used in different combinations and provide different search mechanisms. Different natural and artificial processes influence search strategies in meta-heuristics. Genetic algorithm (GA) [Mirjalili, 2019], differential evolution (DE) [Lampinen and Storn, 2004], and particle swarm optimization (PSO) [Poli et al., 2007] are instances of the most successful meta-heuristic algorithms. In addition to these classic algorithms, many recently introduced meta-heuristic techniques such as gravitational search algorithm (GSA) [Rashedi et al., 2009], gray wolf optimizer (GWO) [Mirjalili et al., 2014a], arithmetic optimization algorithm (AOA) [Abualigah et al., 2021a], Aquila optimizer (AO) [Abualigah et al., 2021c], Harris hawks optimization (HHO) [Heidari et al., 2019a], Moth-flame optimization (MFO) [Mirjalili, 2015], and bee swarm optimization (BSO) [Akbari et al., 2010] have also earned special consideration from researchers where they are extensively investigated in different domains [Houssein

et al., 2021b]. However, most currently available meta-heuristics are considered sequential optimization algorithms. Despite their success in dealing with various optimization problems, well-performance is not always ensured, mainly when dealing with large-scale optimization problems. In this regard, the probability of being stuck into local optima grows, and the computation time for obtaining a satisfactory solution becomes high [Alba et al., 2013].

In the parallel design of meta-heuristic, three parallel models, namely algorithmic level, iteration level, and solution level, can be employed. At the iteration level, a meta-heuristic algorithm is parallelized at each iteration, and the algorithm's behavior does not change. The main goal of this type is to reduce the search time. However, for real-life problems, the iteration level model requires a large size of computational resources [Talbi, 2009]. A single solution from the search space is handled at the solution level. As in the iteration level, the heuristic's behavior stays unchanged. In the algorithmic model, two types of parallelization, including independent or cooperating self-contained meta-heuristics, can be utilized. In the case of independent meta-heuristics, the different meta-heuristics are run without any cooperation. In this way, the search is performed similarly to the sequential run of the meta-heuristics. However, the cooperative model will change the search behaviors of the meta-heuristics and enhance the quality of solutions. In the cooperative mode of parallel meta-heuristics, various algorithms exchange information belonging to the search to improve the quality and robustness of the obtained solutions [Talbi, 2009]. Cooperative meta-heuristics are an excellent optimization algorithm that benefits from parallel computing techniques and meta-heuristics to tackle complex optimization problems with less numerical and computational costs [Crainic, 2016]. In this work, the cooperative model is adopted to deal with the MLT of image segmentation due to its advantages over its peers.

1.1 Motivations and Contributions

Over the years, several meta-heuristic methods have been implemented to address medical image segmentation issues. However, each algorithm has merits and demerits when solving a specific problem. Furthermore, the exploration and exploitation capabilities of the method under consideration significantly impact performance. Therefore, new ideas are typically integrated to address the shortcomings of the algorithm's basic version. For instance, a technique that improves the exploration capability can be added to an existing algorithm with a strong exploitation capability to balance both capacities. Besides, referring to the No Free Lunch (NFL) theorem, [Wolpert, 1996], no universal optimization algorithm can be considered to solve all types of optimization problems efficiently. In alternative words, the door is still open to investigate and develop new variants of meta-heuristics-based approaches. These reasons have motivated researchers to build upon previous work and integrate new techniques into existing ones. In this paper, we have introduced an efficient MIS approach by adapting a cooperative swarm intelligence paradigm. To be specific, the contributions of this research can be summarized as follows:

- A cooperative swarm intelligence model has been proposed by embedding three well-known metaheuristics called PSO, GWO, and HHO for handling image segmentation problems.
- The proposed segmentation model is verified over a set of COVID-19 CT images.
- Various objective functions, including Otsu, cross-entropy, and Tsallis entropy, are examined.

- Segmentation quality is quantified by utilizing three well-established metrics, namely Peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), and universal quality index (UQI).
- The stability and effectiveness of the suggested technique are evaluated using various segmentation levels.
- Comparative results and analysis revealed that the proposed cooperative model achieved better performance than other considered algorithms.

The remainder of the paper is structured as follows: Section 2 explores a set of recent state-of-the-art research efforts on metaheuristics-based image segmentation. Section 3 confers an overview of the employed methods to develop the suggested approach. Section 4 introduces the proposed cooperative optimization model. Section 5 presents the experimental results and analysis. Lastly, Section 6 presents the conclusion of this paper and future work.

2 Related Works

2.1 Meta-heuristic based multi-level thresholding of image segmentation

Image thresholding convincingly proves the usefulness and efficiency of meta-heuristic algorithms in the relevant paradigm in the literature [Houssein et al., 2021b, Elaziz et al., 2020, Rodríguez-Esparza et al., 2020]. There are several successful examples of meta-heuristic applications in this field. However, a few remarkable state-of-the-art research efforts are presented here. For instance, to resolve the problem of MLT, Kapur's entropy-based Crow Search Algorithm (CSA) was introduced by [Upadhyay and Chhabra, 2020] for estimating optimal values of multilevel thresholds. Several experiments were conducted on benchmarked images for various threshold values. The performance of the proposed approach was compared against several algorithms, including PSO, DE, GWO, MFO, and cuckoo search (CS). Experimental results reveal that the proposed method achieved better performance than other algorithms using well-established performance measures such as PSNR, SSIM, and feature similarity index (FSIM).

[Khairuzzaman and Chaudhury, 2017] applied GWO to solve the MIS problem. Otsu's and Kapur's functions were used to identify the best set of thresholds. The proposed approach was tested on a collection of standard evaluation images, and its performance was compared against enhanced forms of bacterial foraging optimization (BFO) and PSO algorithms. The mean structural similarity (MSSIM) index was applied as an evaluation measure. As a result, GWO converged better to optimum solutions than BFO and PSO. Experimental results revealed that the proposed approach is more stable and yielded higher quality solutions than PSO and BFO-based approaches. In addition, the GWO-based method has been shown to be faster than the BFO-based approach and slower than the PSO-based approach.

An MLT approach based on a modified version of the grasshopper optimization algorithm (GOA) boosted with Lévy flight was developed by [Liang et al., 2019] for color image segmentation. Tsallis cross-entropy was used as an objective (evaluation) function to optimize threshold values for multilevel image thresholding. The performance of the proposed approach was examined on six real-life benchmark images and two lant stomata images. In addition, the performance of Tsallis cross-entropy was compared with two thresholding techniques, including between-class variance (Otsu) and Renyi's

entropy. Experiments were conducted using five well-known meta-heuristic algorithms and the proposed grasshopper-based algorithm. Qualitative experimental results proved that the proposed image segmentation approach recorded a higher segmentation accuracy with fewer iterations compared to other considered algorithms.

[Abdel-Basset et al., 2021] proposed an MLT selection method for grayscale images based on a novel meta-heuristic called Equilibrium Optimizer (EO). Kapur's entropy was applied as an objective function to find optimal threshold values. The proposed algorithm was compared with seven other algorithms, including the whale optimization algorithm (WOA), salp swarm algorithm (SSA), HHO, bat algorithm (BA), sine-cosine algorithm (SCA), CSA as well as PSO. Using a set of well-known test images obtained from the Berkeley segmentation dataset, the proposed algorithm is superior to all tested algorithms in terms of various solution quality metrics such as PSNR, SSIM, some accuracy measures including maximum absolute error, and computation time for resource complexity. However, the proposed EO-based algorithm recorded worse standard deviation values and computational time than others.

As presented in [Rodríguez-Esparza et al., 2020], an efficient approach for MIS was proposed using the recently appeared HHO algorithm and the minimum cross-entropy as an evaluation function. To assess the effectiveness of the HHO-based approach, it was tested over a benchmark set of images obtained from the Berkeley segmentation database and medical images of digital mammography. Three matrices, including the PSNR, SSIM, and FSIM, were applied to verify the quality of segmented images. Comparing the suggested method to PSO, harmony search (HS), DE, artificial bee colony (ABC), and SCA, it can be concluded that the proposed approach can deliver efficient results in terms of quality, consistency, and accuracy. The results of HHO were also compared against two machine learning techniques, K-means and fuzzy IterAg. Experimental results revealed that the HHO-based approach improved over other segmentation approaches currently applied in the literature. Meanwhile, using the dragonfly algorithm (DA) [Díaz-Cortés et al., 2018] proposed an approach for solving the problem of unclear regional borders in low-resolution thermography images in health care. DA algorithm was applied to determine the optimal threshold values of the energy curve for breast cancer diagnosis in thermal imaging. Using a set of eight images obtained from the DA-Breast Thermography database and Otsu and Kapur's as objective functions for evaluating the quality of the obtained solution by the DA algorithm, it was observed that the DA algorithm outperformed PSO, GA, and krill herd (KH) algorithm, and runner-root algorithm (RRA).

Most meta-heuristic-based approaches employed in the literature for solving a wide range of complex optimization tasks suffer from main drawbacks, such as trapping in local optima, early convergence, and a lack of global search ability. These limitations motivate researchers to introduce modified and hybrid versions of the basic meta-heuristic algorithms to overcome the abovementioned limitations. Opposition-based learning (OBL) is one of the most successful approaches for improving the search efficiency of many meta-heuristic algorithms [Rojas-Morales et al., 2017]. OBL has been integrated with many meta-heuristic techniques in several ways to boost the exploration and searchability of MLT problems. For instance, OBL and dynamic Cauchy mutation (DCM) was applied to improve elephant herding optimization (EHO) in [Chakraborty et al., 2019]. OBL was utilized to resolve delayed convergence, while DCM was used to handle premature convergence of EHO. In addition, the proposed algorithm was examined on the MLT problem. The authors used Otsu and Kapur's approaches to estimate the thresholding values for image segmentation. When comparing the performance of the enhanced EHO algorithm with the results of four other popular meta-heuristic algorithms comprising

PSO, ABC, CS, BA, and a traditional dynamic programming method, the EHO-based approach outperformed all its competitors.

Another novel OBL-based enhancement technique was proposed by [Houssein et al., 2021d]. This study proposed a modified form of the recent meta-heuristic algorithm called the marine predators algorithm (MPA) for handling the MIS. The OBL was employed to enhance the searchability and convergence behaviors of the original MPA. The proposed MPA-OBL. Otsu and Kapur's methods were applied as objective functions. The performance of three regularly used evaluation matrices, namely Structural similarity (SSIM), Peak signal-to-noise ratio (PSNR), and Feature similarity (FSIM), was used to assess the qualitatively and quantitatively performance of the MPA-OBL-based approach over a variety of benchmark images at varying levels of thresholds. In addition, statistical post-hoc analysis showed that the MPA-OBL algorithm provided highly efficient and accurate outcomes compared to other meta-heuristic algorithms.

Moreover, Houssein and Emam et al. [Houssein et al., 2021c] used the Levy flight mechanism and the OBL strategy to enhance the performance of the chimp optimization algorithm (ChOA), the modified version called IChOA. The IChOA algorithm has been used to solve image segmentation problems and performs better than ChOA and other algorithms.

Emam et al. [Emam et al., 2022] proposed a modified reptile search algorithm (mRSA) for global optimization and image segmentation problems. The mRSA algorithm incorporates the RSA algorithm with the (RUNge Kutta optimizer) RUN algorithm. mRSA was applied as multilevel image segmentation for MRI brain images based on Otsu's method as an activation function. The experimental results have revealed the superiority of the proposed mRSA.

Recently, to solve the image segmentation problem, [Baby Resma and Nair, 2021] proposed a novel MLT approach using the Krill herd (KH) meta-heuristic algorithm. The ideal threshold values can be reached by applying the KH technique to maximize Kapur's or Otsu's objective function. Several benchmark images were utilized to prove the applicability and computational efficiency of the KH-based MLT approach. In addition, detailed comparisons with other existing bio-inspired techniques, including GA, PSO, MFO, and Bacterial Foraging (BF), were done to demonstrate the superior performance of the proposed approach. The comparative analysis revealed that KH-based MLT decreases the computational time needed for computing the optimal thresholds.

Houssein et al. [Houssein et al., 2022a] proposed a novel image segmentation method based on Otsu's fitness function, which used an improved golden jackal optimization algorithm to select the best threshold values for skin cancer imaging.

Chen et al. [Chen et al., 2022] proposed an ensemble multi-strategy-driven shuffled frog leaping algorithm with horizontal and vertical crossover search for multi-threshold image segmentation.

Helong et al. [Yu et al., 2022] presented an enhanced multi-stage grey wolf optimizer for multi-threshold segmentation of Leaf Spot Diseases on Maize.

2.2 Meta-heuristic based multi-level thresholding for Covid-19 images segmentation

Recently, many meta-heuristic-based MLT approaches have been applied to segment CT scans to discover COVID-19 infections. For instance, [Abualigah et al., 2021b] proposed an MLT technique using an evolutionary AOA algorithm. The DE algorithm was employed to enhance the local search of the AOA algorithm. Using between class variance (Kapur's) measure functions, the performance of the proposed algorithm was

examined on nature and CT scan COVID-19 images. The accuracy of segmented images was determined based on two standard measures comprising SSIM and PSNR. The proposed DAOA algorithm recorded higher quality solutions compared to other comparative algorithms.

To enhance the diagnostic level of COVID-19, [Liu et al., 2021] proposed a new technique called CLACO-MIS based on ant colony optimization (ACO) algorithm in conjunction with each of the Cauchy mutation and the greedy Levy mutation for multilevel COVID-19 X-ray image segmentation. In addition, 2D Kapur's entropy was employed as an evaluation function. CLACO algorithm was compared with some other variants as well as other efficient algorithms on 30 benchmark functions gathered from IEEE CEC2014 to prove its superiority in terms of convergence speed, search power, and the ability to avoid the local optimum. Moreover, the experimental results show that CLACO-MIS yielded better segmentation influence at various threshold levels than its peers in achieving segmentation of COVID-19 CT scan images.

As presented in [Nama, 2022], Sukanta Nama developed an improved version of the slime mold algorithm (SMA) with the quasi-reflection operator (QRSMA) for Covid-19 chest X-ray image segmentation. Quasi-reflection-based jumping was applied to improve the convergence, avoid local optimum, and control the balance between exploitation and exploration. The experimental findings show that the proposed QRSMA has higher robust searchability than the original SMA and different search approaches regarding various measures such as convergence and diversity. The author concluded that QRSMA could be used as an effective MLT approach for image segmentation superior to other available approaches.

A hybrid approach that exploits the properties of two swarm intelligence algorithms, MPA and MFO, was proposed by [Elaziz et al., 2020]. In the proposed approach (MPAMFO), the MFO is employed as a local search strategy for MPA to avoid the local optima trap. The MPAMFO was introduced as an MLT approach for image segmentation, presenting outstanding performance in all conducted experiments. To examine the efficiency of MPAMFO, two different tests were performed. In the first one, ten natural gray-scale images were segmented using the MPAMFO method, while in the second test, MPAMFO was tested for a real-world application (e.g., CT images of COVID-19) where thirteen CT images were utilized to examine the performance of MPAMFO. In addition, extensive comparisons with several swarm intelligence algorithms were performed to evaluate the quality of the segmentation and the performance of the MPAMFO. Overall experimental outcomes confirmed that the MPAMFO is a robust MLT approach that demonstrated its superiority compared to existing methods.

The authors in [Kaya et al., 2022] proposed a new method to detect COVID-19 from x-ray images using angle transformation with GoogleNet and LSTM. In transformation, the angle information is created by each pixel on the image with the surrounding pixels. The images are trained with a hybrid deep learning model combining GoogleNet and long short-term memory. Mendeley database is used to evaluate the proposed method achieving an accuracy of 98.97%.

In [Eken, 2020], the researchers proposed a method to detect COVID-19. COVID-19 is detected using a novel capsule network-based model from chest X-ray images.

Furthermore, in [Yilmaz, 2021], Yilmaz et al. proposed a multichannel Convolutional Neural Network (CNN) method to detect COVID-19 from X-ray images. The multichannel CNN architecture consists of five convolution channels. The proposed multi-channel CNN was trained on the three different datasets obtained from the Kaggle data repository.

In [Nair et al., 2021], the authors presented a fully automatic model that uses chest

CT scans to predict COVID-19. The model is called CORNet, which is based on the ResNet architecture. The proposed approach has been evaluated by classifying the CT images of community-acquired pneumonia (CAP) and other non-pneumonia.

In [Panwar et al., 2020b], the authors presented a deep-learning neural network method named nCOVnet. This alternative fast screening method can detect COVID-19 by using X-rays images. The Proposed method detects COVID-19-positive patients with an accuracy of 97%, whereas the overall accuracy of the proposed model is 88%.

The authors of [Panwar et al., 2020a] proposed a deep transfer learning method that revs the detection of COVID-19 using X-ray and CT- images. They used three datasets: COVID-chest X-ray, SARS-COV-2 CT scan, and Chest X-Ray Images (Pneumonia). The results show that the proposed deep learning model can detect COVID-19-positive cases faster than RT-PCR tests in less than 2 seconds.

In [Siddiqui et al., 2020], Siddiqui et al. proposed a k-means clustering-based machine learning method on the data set from different regions of China.

Qi, Ailiang, et al. [Qi et al., 2022] proposed a multilevel image segmentation technique based on an improved ant colony optimization algorithm to enhance the image segmentation of COVID-19 X-rays.

In [Houssein et al., 2022c], Houssein et al. proposed an improved version of the EO that incorporates the standard operators with the dimension learning hunting (DLH) to prevent the algorithm from being trapped in local optima. The proposed approach, called I-EO, was employed as an MIS technique. The proposed method was tested over the CEC'2020 benchmark functions, and quantitative and qualitative results demonstrated the superiority and robustness of the proposed algorithm compared to other well-known optimizers. In addition, I-EO was also applied as MLT segmentation for segmenting a set of CT images of COVID-19 via maximizing the fuzzy entropy. The obtained segmentation results demonstrated the excellent performance of I-EO in all performed experiments. They approved that the proposed I-EO approach can be applied as an accurate tool for image segmentation. Finally, [Cohen et al., 2020a] [Houssein et al., 2021a] [Su et al., 2022] represent successful examples of applying meta-heuristic-based MLT for Covid-19 images segmentation.

The above literature review points to the various shortcomings of segmented COVID-19 CT images, such as: (i) working on only one activation function that is based on image histogram, they don't use the entropy-based methods; (ii) most of the meta-heuristic techniques applied to a diversified range of optimization problems in literature often come with specific shortcomings, such as trapping in local regions, early conversion and needing more global search ability. The overall outcome of the reviewed previous studies related to the MIS problem shows that the MIS method based on the meta-heuristic optimization algorithm is widely adopted. It is also observed that the ideal threshold set in the MIS approach highly relies on the applied meta-heuristic algorithm. In other words, a high-performance meta-heuristic algorithm can notably boost the results of MIS. This has motivated the authors of this work to enhance the diagnosis of Covid-19 by exploiting the merits of parallel computing techniques and MHs to develop a cooperative meta-heuristic algorithms-based MIS model for segmenting COVID-19 CT scan images.

3 Background

This section covers the techniques involved in developing the suggested segmentation approach. The fundamentals of image segmentation thresholding techniques are dis-

cussed, and the structure and operation of the basic PSO, GWO, and HHO algorithms are clarified.

3.1 Thresholding Techniques for Image Segmentation

Thresholding methods are most frequently utilized for segmenting images. Considering grayscale images, these thresholds determine the intensity values for categorizing the image into several groups. According to thresholding levels calculated on the gray level range, thresholding schemes or techniques can be divided into two primary categories: bi-level thresholding and MLT [Suresh and Lal, 2016].

3.1.1 Bi-level Thresholding

The bi-level thresholding method divides an image into two regions dependent on the threshold level. For example, given a grayscale image X with intensity values ranging from 0 to $N - 1$, N represents the highest possible intensity value. This image's bi-level thresholding identifies an intensity value that allows the foreground and background objects to be distinguished as in Eq.(1).

$$\begin{aligned} C_0 &= \{I(x, y) \in X \mid 0 \leq I(x, y) \leq k - 1\} \\ C_1 &= \{I(x, y) \in X \mid k \leq I(x, y) \leq N - 1\} \end{aligned} \quad (1)$$

where X refers to the image being processed, $I(x, y)$ stands for the pixel's corresponding intensity value as indicated by the coordinate values given by x and y , k is the threshold value, and C denotes the class.

Several approaches are used to determine the threshold value that binarizes the image in this way, including computing the variance of pixel values, and entropy values [Sezgin and Sankur, 2004].

3.1.2 Multilevel Thresholding (MLT)

Bi-level thresholding technique is ineffective in images where it is difficult to discern between the background and objects of interest. Unfortunately, in real-world applications, images contain more than two classes, necessitating the usage of multilevel thresholding. As a result, we must switch to a multilevel image thresholding approach [Arora et al., 2008, Gao et al., 2010]. MLT will find multiple gray-level threshold values to differentiate the objects of interest from the image's background.

MLT divides a grayscale image into several regions by selecting more than one threshold value. Accordingly, the image is divided into distinct, homogeneous regions representing the background and other objects. This is beneficial for objects with colored or intricate backgrounds on which binary thresholding fails to yield good results [Bandyopadhyay et al., 2021].

The following definition of the MLT problem: The number of classes in a grayscale image I is $m + 1$, and m thresholds are needed to divide the image into segments. This problem can be mathematically formulated as in Eq. (2).

$$\begin{aligned} C_0 &= \{I(x, y) \in X \mid 0 \leq I(x, y) \leq k_1 - 1\} \\ C_1 &= \{I(x, y) \in X \mid k_1 \leq I(x, y) \leq k_2 - 1\} \\ C_i &= \{I(x, y) \in X \mid k_i \leq I(x, y) \leq k_{i+1} - 1\} \\ C_m &= \{I(x, y) \in X \mid k_m \leq I(x, y) \leq N - 1\} \end{aligned} \quad (2)$$

where C_i is the i^{th} class of the image I , k_i denotes the threshold value for $k \in 1, 2, 3, \dots, m$, m is the total number of distinct threshold values, and N the maximum intensity value.

Consequently, the primary objective of MLT is to identify the threshold values that divide the pixels in an image into groups or segments. Several segmentation-based thresholding techniques have been widely used in numerous multilevel threshold segmentation applications. Examples of these methods are Otsu's method, cross-entropy thresholding, and Tsallis entropy [Suresh and Lal, 2016, Bandyopadhyay et al., 2021].

3.1.3 Otsu's method

This subsection describes the mathematical model of the Otsu method [Otsu, 1979b], which is a thresholding method based on the maximum variance between classes. The segmentation process depends on the image histogram [Glasbey, 1993]. The histogram is passed to otsu's method, selecting the best thresholding values to divide the image into various classes. This technique assumes the L_v intensity levels of the image, and the probability is obtained by Eq. 3. It can be used for RGB images where Otsu is applied to each layer individually.

$$h_i = \frac{h_i}{N_p}, \sum_{i=1}^{N_p} Ph_i = 1 \quad (3)$$

where i indicates intensity level in ($0 \leq i \leq L_v - 1$) and N_p denotes the total number of pixels. h_i is the number of intensity frequencies i in the image denoted by the histogram. In a probability distribution Ph_i , the histogram is normalized. The classes for bilevel segmentation are computed based on the probability distribution as follows:

$$C_1 = \frac{Ph_1}{\omega_0(th)}, \dots, \frac{Ph_{th}}{\omega_0(th)} \text{ and } C_2 = \frac{Ph_{th+1}^c}{\omega_1(th)}, \dots, \frac{Ph_{L_v}}{\omega_1(th)} \quad (4)$$

where $\omega_0(th)$ and $\omega_1(th)$ are probabilities distributions for C_1 and C_2 that is defined by Eq. 5.

$$\omega_0(th) = \sum_{i=1}^{th} Ph_i \text{ and } \omega_1(th) = \sum_{i=th+1}^{L_v} Ph_i \quad (5)$$

It is essential to calculate the mean levels μ_0 and μ_1 that describe the classes by Eq.6. After those operators have been determined, the Otsu between-class σ_B^2 is computed by Eq. 7.

$$\mu_0 = \sum_{i=1}^{th} \frac{iPh_i}{\omega_0(th)} \text{ and } \mu_1 = \sum_{i=th+1}^{L_v} \frac{iPh_i}{\omega_1(th)} \quad (6)$$

$$\sigma_B^2 = \sigma_1 + \sigma_2 \quad (7)$$

Notice that σ_1 and σ_2 in Eq. 7 are the variances of C_1 and C_2 that identified as follows:

$$\sigma_1 = \omega_0(\mu_0 + \mu_y)^2 \text{ and } \sigma_2 = \omega_1(\mu_1 + \mu_y)^2 \quad (8)$$

where $\mu_y = \omega_0\mu_0 + \omega_1\mu_1$. Based on the values σ_1 and σ_2 , Eq. 9 illustrates the objective function. As a result, the optimization problem can be simplified for determining the intensity level, which maximizes Eq. 9.

$$F_{otsu}(th) = Max(\sigma_B^2(th)), 0 \leq th \leq L_v - 1 \quad (9)$$

where $\sigma_B^2(th)$ indicates the Otsu of a provided th value. Otsu's approach is used for a single layer from an image, which means applying it to the three layers for color images is required. The initial idea of a bilevel technique can be changed to accommodate multiple thresholds. The fitness function $F_{otsu}(th)$ in Eq. 9 can be changed for multiple thresholds based on the following:

$$F_{otsu}(TH) = Max(\sigma_B^2(th)), 0 \leq th_i \leq L_v - 1, i = [1, 2, 3, \dots, K] \quad (10)$$

where $TH = [th_1, th_2, \dots, th_k - 1]$, denotes a vector involving multiple thresholding and the variances can be calculated by Eq. 11.

$$\sigma_B^2 = \sum_{i=1}^k \sigma_i = \sum_{i=1}^k \omega_1(\mu_1 - \mu_y)^2 \quad (11)$$

where i denotes the specific class, ω_i and μ_j indicate the occurrence probability and the class mean, respectively. For multiple thresholds, values are taken as:

$$\omega_{k-1}(th) = \sum_{i=th_k+1}^{L_v} Ph_i \quad (12)$$

for mean values:

$$\mu_{k-1} = \sum_{i=th_k+1}^{L_v} \frac{iPh_i}{\omega_1(th_k)} \quad (13)$$

In this paper, we use Otsu's method as a fitness function that will be maximized. The reason for using otsu's method is that it is one of the most widely used methods in multi-level thresholding image segmentation problems.

3.1.4 Entropy based thresholding

Entropy creates a statistical variety score based on the intensity levels of an input image. Entropy-based thresholding is a reliable technique since the chosen threshold is independent of slight variations in the input image and is decided based on an overall and objective quality of the image histogram [Kapur et al., 1985a]. Furthermore, the cross-entropy-based objective function has recently provided promising results for the brain MRI segmentation problem [Bandyopadhyay et al., 2021]. Accordingly, to solve the COVID-19 CT image segmentation problem, the entropy-based thresholding method is preferred in the current study. For this purpose, cross-entropy is employed as an objective function for the fitness assignment of the generated solutions (i.e., threshold values) in the proposed cooperative optimization model.

3.1.5 Cross-entropy's method

The Cross-entropy method, named Kapur [Kapur et al., 1985b] technique, is a thresholding technique used to develop the image segmentation process. Kapur's method chooses the best thresholding based on maximizing entropy. The mathematical representation has outlined as follows:

$$F_{kapur}(th) = H_1 + H_2 \quad (14)$$

where the entropies H_1 and H_2 are computed as:

$$H_1 = \sum_{i=1}^{th} \frac{Ph_i}{\omega_0} \ln\left(\frac{Ph_i}{\omega_0}\right) \text{ and } H_2 = \sum_{i=th+1}^L \frac{Ph_i}{\omega_1} \ln\left(\frac{Ph_i}{\omega_1}\right) \quad (15)$$

where Ph_i is the probability distribution of the intensity levels achieved by Eq. (3), $\omega_0(th)$ and $\omega_1(th)$ are probabilities distributions for the classes C_1 and C_2 . Like the Otsu technique, the Kapur method has been modified for multi-threshold values. It is crucial to divide the image into N classes by $N - 1$ thresholds. The following formula represents the new fitness function:

$$F_{kapur}(TH) = \sum_{i=1}^N H_i \quad (16)$$

$TH = [th_1, th_2, \dots, th_{N-1}]$ is a vector contains different thresholds. The entropy is calculated independently and given its associated value (th), as seen in Eq. (16). The entropy is represented by:

$$H_N^c = \sum_{i=th_{N-1}+1}^L \frac{Ph_i}{\omega_{N-1}} \ln\left(\frac{Ph_i}{\omega_{N-1}}\right) \quad (17)$$

The probability values occurrence ($\omega_0^c, \omega_1, \dots, \omega_{N-1}$) of the N classes are obtained by Eq. (12) and the probability distribution Ph_i with Eq. (3).

3.1.6 Tsallis entropy

The entropy of a discrete source is frequently determined from the probability distribution, where $Pr = Pr_j$ represents the probability of finding the system in each possible state j [Agrawal et al., 2013]. $0 \leq Pr \leq 1$ and $\sum_{j=0}^m Pr_j = 1$. The Shannon entropy can be defined as:

$$Sh = - \sum_{j=1}^k Pr_j \ln(Pr_j) \quad (18)$$

where k is the total number of states. Shannon entropy has the extensive property:

$$S(A + B) = S(A) + S(B) \quad (19)$$

Based on multi-fractal theory, Tsallis entropy can be generalized to non-extensive systems using a familiar entropic formula [Tsai, 1985]:

$$S_q = \frac{1 - \sum_{j=1}^k (Pr_j)^q}{q - 1} \quad (20)$$

where q is the measure of the degree of non-extensivity of the system known as the Tsallis parameter or entropic index. Tsallis entropy can be described by a pseudo-additivity entropic rule using Eq. 21

$$S_q(A + B) = S_q(A) + S_q(B) + (1 - q) \times S_q(A) \times S_q(B) \quad (21)$$

This kind of technique is exploited for thresholding the image. Assume that Gr gray levels are presented in the input image having a range of $1, 2, \dots, Gr$, and $Pr_i = Pr_1, Pr_2, \dots, Pr_{Gr}$ is the probability distribution of gray intensity points. Based on the above two probability distributions, two classes named class A and class B can be obtained for background and for the object of interest, which is given by Eq. 22

$$Pr_A = \frac{Pr_1}{Pr^A}, \frac{Pr_2}{Pr^A}, \dots, \frac{Pr_t}{Pr^A} \quad \text{and} \quad Pr_B = \frac{Pr_{t+1}}{Pr^B}, \frac{Pr_{t+2}}{Pr^B}, \dots, \frac{Pr_{Gr}}{Pr^B} \quad (22)$$

$$\text{where } Pr^A = \sum_{i=1}^t Pr_i \text{ and } Pr^B = \sum_{i=t+1}^{Gr} Pr_i.$$

For each class, Tsallis entropy can be classified as:

$$S_q^A(t) = \frac{1 - \sum_{i=1}^t (Pr_i/Pr^A)^q}{q - 1}, \quad S_q^B(t) = \frac{1 - \sum_{i=t+1}^G r(Pr_i/Pr^B)^q}{q - 1} \quad (23)$$

Maximizing the computation of information between classes A and B is possible. An ideal threshold value is referred to as the corresponding grey level. The ideal threshold value for bi-level thresholding is the grey level for which the information measure between two classes (foreground and background) is maximized; this can be accomplished by utilizing the objective function with little computational effort:

$$T = \arg \max [S_q^A(t) + S_q^B(t) + (1 - q) \times S_q^A(t) \times S_q^B(t)] \quad (24)$$

According to the following constraint: $|Pr^A + Pr^B| - 1 < S < 1 - |Pr^A + Pr^B|$
The above process can be simply extended for multi-level thresholding using Eq.

$$[T_1, T_2, \dots, T_m] = \arg \max [S_q^1(t) + S_q^2(t) + \dots + S_q^M(t) + (1 - q) \times S_q^1(t) \times S_q^2(t) \dots S_q^M(t)] \quad (25)$$

where,

$$S_q^1(t) = \frac{1 - \sum_{i=1}^{t_1} (Pr_i/Pr^1)^q}{q-1}, \quad S_q^2(t) = \frac{1 - \sum_{i=t_1+1}^{t_2} (Pr_i/Pr^2)^q}{q-1}$$

$$\text{and, } S_q^M(t) = \frac{1 - \sum_{i=(t_m+1)}^{Gr} (Pr_i/Pr^M)^q}{q-1}, \quad M = m + 1$$

3.2 Swarm intelligence inspired meta-heuristics

Meta-heuristics are general-purposed approximate algorithms designed to tackle a wide range of optimization problems, particularly combinatorial ones. Meta-heuristics are a prominent type of stochastic optimization that uses some degree of randomness to obtain high-quality (i.e., near-optimal) solutions in a reasonable amount of time. [Glover and Kochenberger, 2003] states that Meta-heuristics are search algorithms that coordinate the interaction between top-level strategies and local improvement methods to create an effective search process capable of performing a reliable global search. As a result, these methods decrease the size of the search space by efficiently examining the promising regions [Talbi, 2009]. In Meta-heuristics, exploitation and exploration are frequently opposing processes. Meta-heuristics use and promote their randomized operators in the exploration process to try to explore different search space regions. Contrarily, exploitation focuses on identifying the vicinity of prospective solutions to acquire higher-quality solutions. Any reliable Meta-heuristic algorithm should be able to achieve a balance between exploitation and exploration potentials, lowering the risk of getting stuck in local optima [Boussaid et al., 2013].

Meta-heuristics can generally be categorized based on a variety of factors. However, classifying them into trajectory- and population-based algorithms is one of the most popular classifications in the literature [Eshay et al., 2019]. When compared to trajectory algorithms, population-based algorithms offer more significant exploratory potential. Specifically, a population of possible solutions is created and iteratively evolved during the search process until satisfactory results are obtained. Swarm intelligence Meta-heuristics are population-based, cooperative natural-inspired algorithms that imitate the group behavior (e.g., self-organized and decentralized systems) that occurs in swarming animals, including wasps, fish, bees, ants, whales, birds, termites, and others [Ab Wahab et al., 2015]. The essential features of the Swarm intelligence algorithms, according to [Talbi, 2009], are that the particles are simple agents that cooperate by moving in the decision space via an indirect operation. The most used algorithm in this group is PSO, which was first introduced by [Eberhart and Kennedy, 1995]. GWO [Gao et al., 2017], and HHO [Heidari et al., 2019b] are other examples of swarm intelligence techniques. This study introduces a cooperative swarm intelligence paradigm by embedding the three well-regarded algorithms (PSO, GWO, and HHO). These algorithms were chosen because they have been successfully exploited to tackle various challenging real-world problems in different domains. The following subsections are concerned with providing an overview of these algorithms.

3.2.1 Particle Swarm Optimization (PSO)

One of the most effective swarm optimization algorithms was presented by Kennedy and Eberhart and is known as PSO [Eberhart and Kennedy, 1995]. The fundamental idea is to

imitate the behavior of flocking birds using a collection of randomly initialized particles that search across the search space to find the best solution. PSO is a general iterative method that directs each particle toward the optimal location. For its optimal placement and the best location for all particles, each particle evolves in two processes: exploration and exploitation. To accomplish these two operations, each particle modifies its position in the search space depending on two variables: velocity and position updating rules. Figure 1 shows the schematic operation of basic PSO. The updating criteria used by the PSO algorithm are shown in equations (26) and (27).

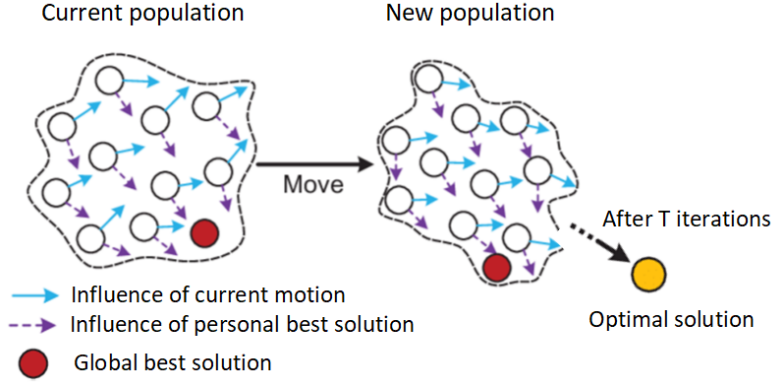


Figure 1: Schematic of basic particle swarm optimization [Chen et al., 2020]

$$v_i^j(t+1) = \omega_1 v_i^j(t) + c_1 r_1 (pbest_i^j - x_i^j(t)) + c_2 r_2 (gbest_i^j - x_i^j(t)) \quad (26)$$

$$x_i^j(t+1) = x_i^j(t) + v_i^j(t+1) \quad (27)$$

where t refers to iteration number, the initial weight ω_1 is used to control the searching tendencies either for global or local solutions. $v_i^j(t)$ presents the current velocity at iteration t for j -th dimension in i -th agent, $x_i^j(t)$ is j -th dimension in i -th particle. Two random numbers between $(0,1)$ are expressed in terms r_1 and r_2 , and c_1 and c_2 present the individual and social factors, respectively. The $pbest$ and $gbest$ refer to the personal and global best solutions (i.e., agent), respectively. Since each solution will follow $pbest$ and $gbest$ solutions, PSO shows a fast convergence behavior while performing the exploration and exploitation processes. However, PSO, in some cases, may be trapped in local optima problems when handling complex and multi-modal optimization problems [Thaher et al., 2022b].

3.2.2 Grey Wolf Optimizer (GWO)

The GWO is a well-regarded optimization algorithm proposed by [Mirjalili et al., 2014b]. The basic idea of this algorithm is inspired by the social structure and hunting strategy of grey wolves in the wild. The leadership hierarchy is simulated using four different breeds

of grey wolves, including alpha, beta, delta, and omega. Additionally, three fundamental hunting techniques, looking for prey, surrounding prey, and attacking prey, are used to carry out optimization.

In the wild, grey wolves begin their hunt by tracking and pursuing target prey. They then pursue and surround the prey until it stops. Attacking and killing the target is the final step. The following mathematical model could be used to describe encircling behavior:

$$\vec{D} = |(\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t))| \tag{28}$$

$$\vec{X}(t+1) = \vec{X}_p(t) + \vec{A} \cdot \vec{D} \tag{29}$$

$$\vec{A} = 2 \vec{a} \cdot \vec{r}_1 - \vec{a} \tag{30}$$

$$\vec{C} = 2 \vec{r}_2 \tag{31}$$

where t is the number of iterations, $||$ is the absolute value, and $(.)$ is a component-by-component multiplication, and D is the distance vector specified in Eq. (28). \vec{A} and \vec{C} are random coefficient vectors (are calculated as in Eq.s (30), and (31)), $\vec{X}_p(t)$ is the prey position vector, $\vec{X}(t)$ is the gray wolf position vector. The detailed mathematical model of GWO updating rules is presented in the original paper [Mirjalili et al., 2014b]. Figure 2 exhibits the potential next position of a search agent in a 2D search space.

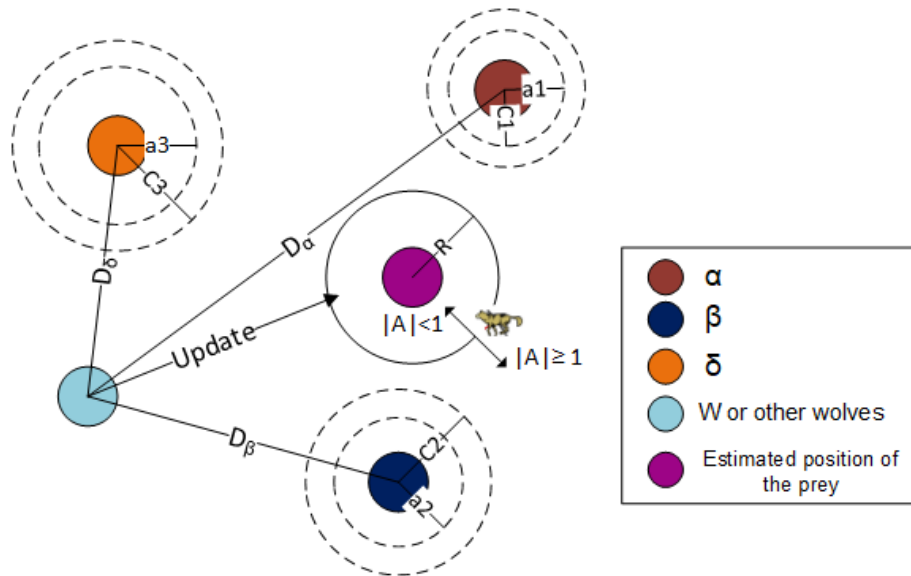


Figure 2: updating strategy for each search agent in GWO [Thaher et al., 2022a]

3.2.3 Harris hawks optimization (HHO)

HHO is a popular, well-regarded swarm-based, gradient-free optimization algorithm introduced by [Heidari et al., 2019b] in 2019. The main logic of the HHO method is designed based on the chain of action and reactions of hawks and rabbits during their hunting process. The core mathematical foundation of this optimizer, as revealed in the original paper of HHO, makes it an effective optimizer in dealing with various constrained and unconstrained problems. The search agents in this optimizer are updated through two phases of exploration and four phases of exploitation. In addition, it employs several time-varying mechanisms and a greedy scheme to improve the quality of results. The stages of HHO are demonstrated in Figure 3. The detailed phases and the complete mathematical model of HHO are revealed in the original paper [Heidari et al., 2019b].

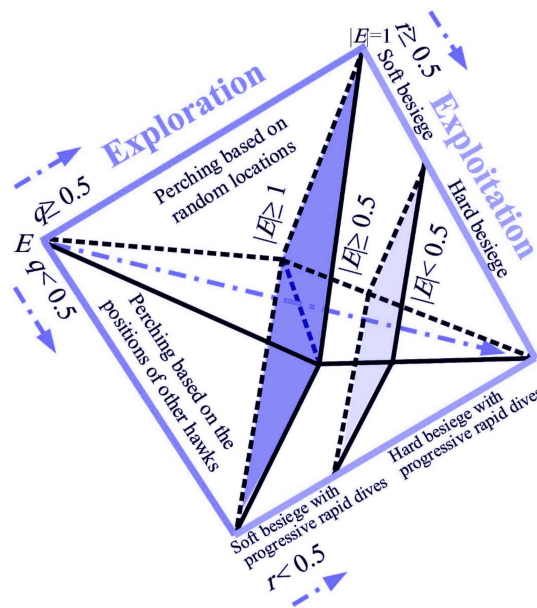


Figure 3: The stages of HHO [Thaher et al., 2020]

4 Proposed cooperative algorithmic-level model

The heterogeneous cooperative model in this work involves three swarm intelligence algorithms (PSO, GWO, and HHO). In the cooperative model, the applied algorithms cooperate with each other by exchanging search-based information. In general, the cooperative model of meta-heuristic algorithms depends on some factors, comprising the communication mechanism (topology), the type of exchanged information, and the criteria of exchange decision (when) [Talbi, 2009]. Furthermore, the exchanged information help in adapting the search behaviors of algorithms and provides a better exploration of the search space, hence determining optimal or near to the optimal solution.

Considering these factors, the proposed model in this work begins the search process for the optimal solution by running three algorithms, including PSO, GWO, and HHO, concurrently, where each algorithm has its own configuration and search mechanism. All algorithms run for a specific number of iterations. At a pre-specified number of iterations (e.g., 100,200,300,400, and 500), the algorithms exchange search-based information, including a set of N best solutions where each algorithm replaces the worst N solutions in its population with N best solutions obtained from one of the cooperated algorithms. Figure 4 depicts an example of the migration process using ring topology.

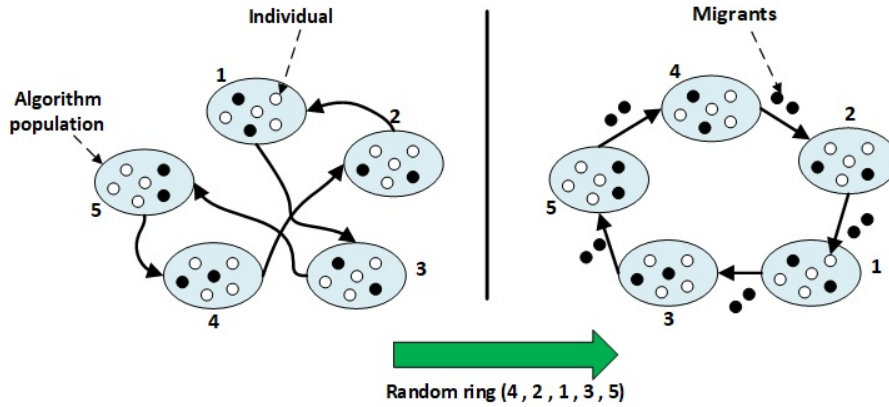


Figure 4: Migration process among algorithms using random ring topology

Simply, selected solutions will migrate between some algorithms after a pre-specified number of generations. The migration operation is a unique procedure that algorithms perform to enhance the diversity of their searches [Al-Betar and Awadallah, 2018]. In addition, a number of parameters such as the number of algorithms, topology, migration rate, replacement mechanism, and migration frequency usually control the migration process. The following subsections present the parameters of the migration process and their importance [Abed-alguni et al., 2019].

4.1 Interconnection topology

Cooperated algorithms are logically interconnected to establish a communication network where each node represents an algorithm, and the network's edges represent the pairs of algorithms that could exchange their search information [Talbi, 2009]. The connection topology of the algorithms significantly impacts the quality of the final solution. Various forms of connection topology have been introduced (e.g., ring topology, star topology, grid topology, and complete graph topology). For parallel meta-heuristics, based on the overall performance, ring topology is considered the most effective and practical one [Al-Betar and Awadallah, 2018] [Abed-alguni et al., 2019].

4.2 The migration frequency

The migration process means the exchange of solutions between algorithms. Determine when to exchange the search-based information must be accurately decided. Hence, the

migration operation is often controlled by a pre-determined number of iterations which is named migration frequency (M_f). This number is employed to control the number of iterations between each successful swap of information between neighbor algorithms. The value of the migration frequency has a significant impact on the overall performance of the cooperative model [Crainic, 2016][Alba et al., 2013].

4.3 The migration rate

The migration rate (M_r) presents another significant factor in the algorithmic-level model. This ratio determines the number of solutions that will migrate between the neighbor algorithm. It could be fixed or a percentage of the population. The selection of the value of M_r is essential where a small value may affect the efficiency of the cooperation between algorithms, While a considerable value may decline the diversity of the search process. Hence, the chance of premature convergence will increase [Crainic, 2016][Alba et al., 2013].

4.4 The selection and replacement mechanisms

Generally, high-quality solutions are the most proper type of information exchanged [Crainic, 2016]. Therefore, a selection mechanism must be applied to decide which solutions will be migrated between neighbor algorithms. The selection mechanisms can be divided into two categories: deterministic or stochastic [Talbi, 2009]. The fittest solutions are chosen using different selection approaches such as tournament and roulette wheel. In addition, a deterministic mechanism can also be applied to replace the received solutions. In the stochastic mechanism, solutions are chosen in a random way without considering their fitness values. In the literature, the best-worst migration approach is the most applied selection method [Al-Betar and Awadallah, 2018] [Abed-alguni et al., 2019].

4.5 CPGH-based image segmentation

This section provides further information on the proposed CPGH-based MIS model for segmenting the COVID-19 CT chest images. Figure 5 displays the flowchart for the segmentation framework. To sum up the development of the proposed CPGH algorithm for choosing the optimal thresholds in COVID-19 CT images using the Otsu, cross-entropy, and Tsallis entropy fitness functions, the following steps can be used to implement the suggested CPGH-based image segmentation:

– Step 1: Initialize the Problem and Adjustable Parameters

At this step, it is necessary to initialize the problem, the *CPGH* parameters, and the migration parameters that were stated before.

- **Step 1a:** Read the original CT image I_R and store it as a grayscale image I_{GRY} .
- **Step 1b:** Obtain the histogram for the input image and compute the probability distribution.
- **Step 1c:** Initialize the problem parameters, including the number of thresholds (i.e., problem dimension), the possible range of threshold values (1–256), and the objective function (Otsu’s method, cross-entropy, and Tsallis entropy).

- **Step 1d:** Setup the common and internal parameters of each integrated algorithm in the CPGH model. The parameters that are shared are the population size N and the maximum number of iterations (t_{max}). In this stage, the internal parameters of *PSO*, *GWO*, and *HHO*, as described in Table 1, must also be configured.
- **Step 1e:** Setup the parameters of the migration process, including migration frequency (M_f), migration rate (M_r), communication topology, and selection-replacement strategy (see Table 1).

– **Step 2: Generate the initial population for each algorithm**

In this step, a population of candidate solutions $X = (x_1, x_2, x_3, \dots, x_n)$ are usually generated randomly, where each candidate solution (x_j , where $j \in (1, 2, 3, 4, \dots, N)$) represents a vector of threshold values computed using Eq. (32).

$$\vec{X}_j = \vec{X}_L + r(\vec{X}_U - \vec{X}_L) \quad (32)$$

where r is a random number within $[0, 1]$, \vec{X}_L and \vec{X}_U are the lower and upper bound of the threshold values.

– **Step 3: Optimization process**

This process includes three steps (Step 3a-Step 3c) as follows:

- **Step 3a:** Evaluate the quality of each candidate solution X using the cross-entropy fitness function as in Eq. 16.

- **Step 3b: Improvement process**

In this phase, the search process for the optimal solution begins by running cooperative algorithms, including *PSO*, *WOA*, and *HHO*, concurrently, where each algorithm has its own configuration, population, and search mechanism. In this step, each algorithm evolves its population independently using the mathematical operators of the original algorithm as discussed in Sections 3.2.1, 3.2.2, and 3.2.3.

- **Step 3c: Migration process**

This process is regularly started after a predetermined number of iterations according to the (M_f) value: i) A commutation topology (e.g., ring topology) is constructed randomly to connect the cooperative algorithms. ii) A portion of each population is exchanged among the connected algorithms according to the (M_r) parameter. iii) Based on the selection-replacement approach, the swapping process is carried out between every pair of neighboring algorithms (e.g., the best-worst strategy). The best-worst migration policy selects the best $M_r \times N$ individuals from the source population to replace the worst $M_r \times N$ individuals in the target population.

– **Step 4: Memorize the best solution obtained by each algorithm.**

In this step, the best solution X^* obtained so far in each evolved population is memorized.

– **Step 5: Check stop criterion**

The iteration counter is increased by 1, and if the stop condition (i.e., the maximum number of iterations) is satisfied, go to step 14. Otherwise, jump to Step 3.

– **Step 6: Find the best threshold values**

The best solution that contains the best thresholds among the memorized solutions in step 4 is returned and applied to the input COVID-19 CT image to provide the segmented image.

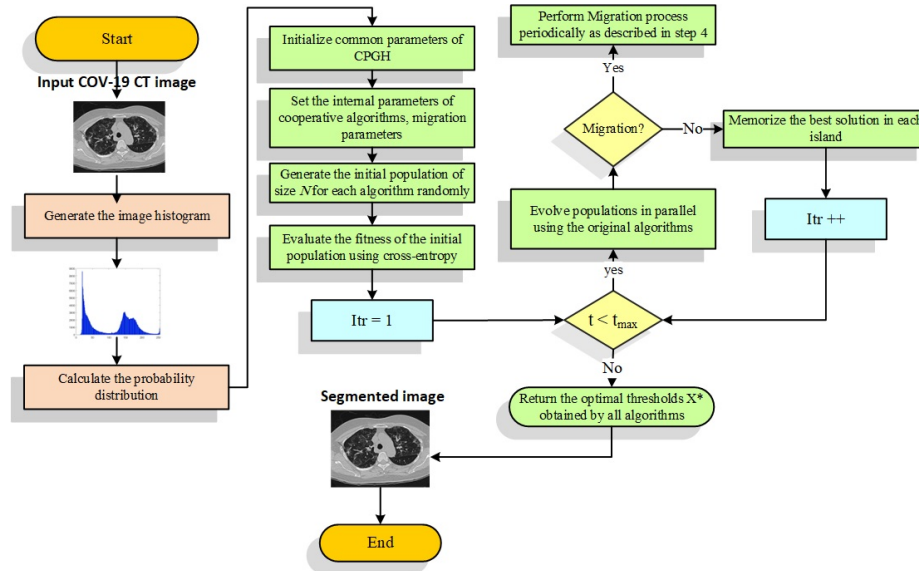


Figure 5: Flowchart of the proposed CPGH-based multi-thresholding image segmentation

5 Experimental Results and Discussion

This section presents and discusses the experimental results of segmenting COVID-19 CT images. The experiments have been conducted to validate the performance of the suggested CPGH method, and its results have been compared with those of some state-of-the-art algorithms. Mainly, the experimental work was conducted in two phases as follows:

- Firstly, various objective functions were adopted to guarantee better segmentation quality of COVID-19 CT images.
- Secondly, a novel CPGH model was introduced as an MLT technique to handle the segmentation problem. Besides, the proposed approach was compared with other state-of-the-art methods. These methods include the original PSO, GWO, and HHO in addition to the other three metaheuristic algorithms: self-adaptive DE (SADE), GA, and improved harmony search (IHS).

The following is how the rest of this section is structured: The experimental setup is described in Section 5.1. Section 5.2 describes the COVID-19 images dataset utilized in the experiments. The image quality metrics used in this study are detailed in Section 5.3. The experimental setting is described in Section 5.4. Results of various objective functions are displayed in Section 5.5. The performance analysis of the proposed CPGH and other algorithms is presented in Section 5.6.

5.1 Experimental setup

The efficiency of the proposed cooperative approach is compared against PSO [Eberhart and Kennedy, 1995], GWO [Mirjalili et al., 2014b], and HHO [Heidari et al., 2019b] algorithms. The COVID-19 CT scan images are segmented with 2, 4, 6, 10, 15, 20, and 25 thresholds. Such thresholds were selected based on previous studies to assess the quality of the images and evaluate the algorithms in multidimensional real-world problems [Houssein et al., 2022c, Su et al., 2022, Elsayed Abd Elaziz et al., 2021]. Three objective functions were employed: Otsu, cross-entropy, and Tsallis entropy. In addition, PSNR, SSIM, and UQI measures are adopted to assess the proposed model. All algorithms have been implemented and run using Python on a computer running Ubuntu 20.04 LTS with Intel(R) Core(TM) i7-1165G7 CPU @ 2.80GHz (8 CPUs) and 16 GB RAM.

Because of the non-deterministic nature of meta-heuristic algorithms, each algorithm is run 10 times for each test image at each threshold level. Accordingly, we provided the average results from all method runs to lessen the impact of random components. The top values are highlighted in **boldface** in the reported results. Furthermore, the Wilcoxon rank-sum and Friedman mean rank tests are also utilized to demonstrate the performance difference between the proposed methods and statistically assess the suggested method's importance. We used Wilcoxon rank-sum test with a degree of 5% significant level. Wilcoxon rank-sum test is a non-parametric test employed to compare the outcomes of each pair of algorithms [Liao et al., 2007]. The Friedman mean rank test is another non-parametric test used to compare three or more matched groups and to evaluate how well competitive algorithms perform. The total ranking of the competing algorithms is specified using Friedman's mean rank [Richardson, 2015].

5.2 COVID-19 CT images dataset

Computed tomography (CT) is a helpful tool for diagnosing COVID-19 patients during the COVID-19 outbreak. Open-sourced datasets from [Zhao et al., 2020a, Cohen et al., 2020b] were employed in this study to evaluate the proposed methodology. The CT COVID-19 dataset consists of 349 CT scans with clinical COVID-19 findings from 216 patients. An experienced radiologist who has been identifying and treating COVID-19 patients ever since the pandemic's emergence attests to the usefulness of this dataset. The COVID-19 images were collected from individuals of both sexes aged 40 to 84. In this paper, ten images for ten distinct patients are selected from this dataset to assess the performance of the proposed methods. The chosen test images are named COVID-CT1, COVID-CT2,..., and COVID-CT10. Figure 6 demonstrates these images and the corresponding histograms.

5.3 Image quality metrics

For multilevel segmentation, it is critical to examine the pixel classification's accuracy by using a quantitative measure of the quality of a segmented image. In this study, three

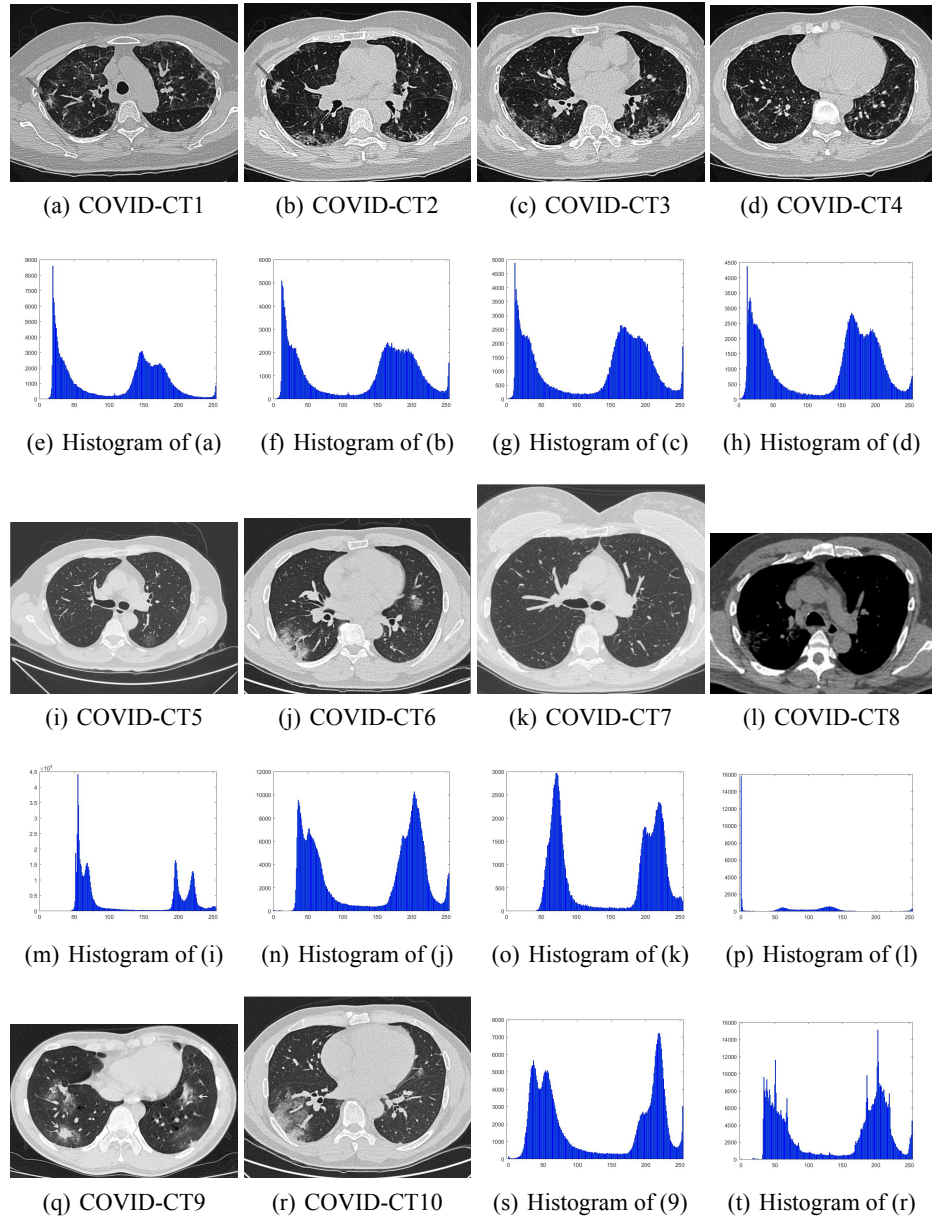


Figure 6: Set of tested COVID-19 CT images and their histograms

popularly used metrics, namely PSNR, SSIM, and UQI, are employed to evaluate the quality of the segmented images. The following subsections will briefly define these measures.

5.3.1 Peak Signal to Noise Ration (PSNR)

PSNR [Avcibas et al., 2002] is used to verify the similarity between the original and segmented images using the Root Mean Square Error (RMSE) of each pixel. The mathematical formulas for PSNR and RMSE are given in Eqs (33) and (34), respectively.

$$RMSE = \sqrt{\frac{1}{mn} \sum_{i=1}^n \sum_{j=1}^m (I(i, j) - I_s(i, j))^2} \quad (33)$$

where I and I_s represent the original image and the thresholded image, respectively, while $m.n$ denotes the image size.

$$PSNR = 20 \log_{10} \left(\frac{255^2}{RMSE} \right) \quad (34)$$

5.3.2 Structural Similarity Index (SSIM)

The SSIM [Wang et al., 2004] is a perceptual metric that quantifies image quality degradation caused by processing the reference image. This measure determines the similarity between the original and the segmented image. SSIM is calculated using Eq. (35)

$$SSIM = \frac{(2\mu_I\mu_{I_s} + c_1)(2\sigma_{I,I_s}) + c_2}{(\mu_I^2 + \mu_{I_s}^2 + c_1)(\sigma_I^2 + \sigma_{I_s}^2 + c_2)} \quad (35)$$

where μ_I, μ_{I_s} denote the mean of intensity values of the original and the segmented images, respectively, σ_I, σ_{I_s} represent the standard deviations of I and I_s , σ_{I,I_s} is the covariance of I and I_s , c_1 and c_2 are constants for the weak denominator stabilization.

5.3.3 Universal Quality Image Index (UQI)

The correlation loss, contrast distortion, and luminance distortion that make up an image's distortion are combined to form the UQI metric [Wang and Bovik, 2002]. The UQI can be calculated using Eq.(36)

$$UQI = \frac{4\sigma_{I,I_s}\mu_I\mu_{I_s}}{(\sigma_I^2 + \sigma_{I_s}^2)(\mu_I^2 + \mu_{I_s}^2)} \quad (36)$$

where, μ and σ denote the mean and standard deviations of the reference and processed images, and σ_{I,I_s} denotes the covariance of the images.

5.4 Parameter settings

To guarantee that the investigations are as objective as feasible, all algorithm comparison tests in this study are carried out under the same conditions, using the parameters in

Table 1. The same common parameters (population size = 30, maximum iterations = 500, and the number of runs = 10) were used to examine all the optimizers for each test image. The lower and upper bounds are set at 1 and 256, respectively. The internal parameters of the implemented algorithms were chosen based on the suggested default values from the literature [Heidari et al., 2019b, Mirjalili et al., 2014b, Thaher et al., 2022b]. As stated in [Arcuri and Fraser, 2013], a fair parametrization is to use default parameter values. Furthermore, using default settings lessens the possibility of comparison bias since no algorithm would benefit from better parametrization. The migration parameters of the proposed CPGH were selected according to the experimental work conducted in [Thaher and Sartawi, 2020].

Table 1: Predefined parameter settings for the tested algorithms

Common parameters		
population size		30
Maximum number of iterations		500
Number of runs		10
Dimension		No. thresholds
Lower and upper bounds		1 , 256
significance level (α)		0.05
Specific parameters		
Algorithm	parameter	value
HHO [Heidari et al., 2019b]	convergence constant (E)	[2 0]
GWO [Mirjalili et al., 2014b]	convergence constant (a)	[2 0]
PSO [Thaher et al., 2022b]	inertai weight (w)	[0.9 0.2]
	cognitive constant (c_1)	2
	social constant (c_2)	2
CPGH [Thaher and Sartawi, 2020]	communication topology	Ring
	selection replacement policy	best-worst
	migration frequency (m_f)	5
	migration rate (m_r)	0.3

5.5 Assessment of various objective functions

In the first round of experiments, three objective functions, including Otsu, cross-entropy, and Tsallis entropy, were employed for fitness assignment to the PSO population. These experiments aim to identify the most effective objective function for the segmentation of COVID-19 CT images tested in this study. The PSO algorithm has been used in this phase to handle the image segmentation problem over ten CT COVID-19 images. Each experiment is executed with 10 independent runs per tested image, each run with 300 iterations, and the swarm size is set to 30. The experiments were performed using 10 threshold values.

The quantitative outcomes of the various objective functions utilized for fitness assignment in the PSO algorithm are presented in Table 2. When inspecting the results, we noticed that PSNR, SSIM, and UQI performances perceived by entropy-based functions

Table 2: Comparison of mean PSNR, SSIM, and UQI values obtained by PSO using different objective functions for fitness assignment.

Image	PSNR			SSIM			UQI		
	Cross_Entropy	Otsu	Tsallis	Cross_Entropy	Otsu	Tsallis	Cross_Entropy	Otsu	Tsallis
COVID-CT1	22.788	22.599	23.211	0.8128	0.7784	0.7480	0.8766	0.8095	0.7544
COVID-CT2	25.288	23.541	25.783	0.8619	0.8371	0.8213	0.9153	0.8717	0.8267
COVID-CT3	24.911	22.662	26.047	0.8740	0.8491	0.8453	0.9314	0.8821	0.8569
COVID-CT4	24.996	21.670	26.427	0.8634	0.8343	0.8445	0.9357	0.8781	0.8729
COVID-CT5	18.096	17.361	18.521	0.6072	0.5911	0.7109	0.7089	0.6695	0.7548
COVID-CT6	24.514	20.026	25.909	0.7931	0.7356	0.8474	0.9078	0.8110	0.9596
COVID-CT7	20.000	17.834	17.906	0.7515	0.7168	0.7407	0.8818	0.8193	0.7676
COVID-CT8	18.355	24.906	28.047	0.8884	0.9093	0.9294	0.8353	0.8397	0.8489
COVID-CT9	23.877	21.012	28.922	0.7855	0.7347	0.8369	0.9357	0.8251	0.9804
COVID-CT10	23.229	21.641	22.094	0.7798	0.7614	0.7691	0.8714	0.8330	0.8382
Rank (F-Test)	2.00	2.80	1.20	1.50	2.60	1.90	1.50	2.40	2.10

(i.e., cross-entropy and Tsallis) are better than those perceived by Otsu's function in most of the cases. The cross-entropy objective function outperforms the others in terms of SSIM and UQI in 60% of test cases. The values provided by this function differ from those of the other objective functions. As per F-test, the first rank is obtained by cross-entropy, followed by Tsallis entropy and Otsu's method, respectively.

A boxplot analysis can be used to help in understanding the characteristics of the data distribution. Boxplots are the ideal graphs to display data distributions in quartiles. The algorithm's lowest and highest data points, which form the edges of the whiskers, are the minimum and maximum. The rectangles' ends serve as the boundaries between the lower and upper quartiles. High data agreement is indicated by a boxplot that is narrow. Examining the boxplot outcomes of the SSIM metric in Figure 7, the cross-entropy-based objective function typically had the greatest median value on most test images. Besides, for most images, the boxplots of the cross-entropy function are quite narrow compared to other distributions. Indeed, results clearly demonstrate the superior performance of the cross-entropy-based thresholding method.

Moreover, we illustrate a convergence analysis of the proposed CPGH algorithm compared with counterpart algorithms. Figure 8 show the convergence curves for CPGH, PSO, GWO, and HHO algorithms when dealing with tested images regarding 25 thresholds. Figure 8 (b, d, e, f, h, i) illustrates the convergence curves for the COVID-CT2, COVID-CT4, COVID-CT5, COVID-CT6, COVID-CT8, and COVID-CT9 test images, respectively. Accordingly, the CPGH algorithm demonstrates an early exploration compared to all other counterparts. Over the test images COVID-CT1 COVID-CT3, and COVID-CT10 as illustrated in Figure 8(a, c, j) the PSO explicit a significant performance than other algorithms. For the test image, COVID-CT7, as presented in Figure 8 (g), the CPGH, PSO, and HHO algorithms have similar performance. It is noticed that the CPGH algorithm gets a stable point in most test images. This explains that the proposed CPGH converges correctly towards and nearer the optimal solution.

Our findings prove that the proper selection of objective function often guarantees a better segmentation quality. Entropy creates an index of statistical variability based on the intensity levels in an input image. Entropy-based thresholding is a reliable technique since the chosen threshold is independent of slight variations in the input image and is decided based on an overall and objective characteristic of the image histogram. Thus the cross-entropy function is preferred in the present research for the COVID-19 CT

segmentation problem. It is adopted in the subsequent experiments for fitness assignment in all tested algorithms.

5.6 Segmentation results of CPGH, PSO, GWO, and HHO

In this phase, we intensely performed a set of experiments to investigate the performance of the proposed CPGH model by comparing it with the basic PSO, GWO, and HHO algorithms. It is worth noting that cross-entropy is used as the objective function because it has delivered the best performance from the previous analysis. Besides, fitness values, PSNR, SSIM, and UQI are used to assess the performance of the proposed methods. Tables 3, 5, 6 and 7 outline the mean results of fitness, PSNR, SSIM, and UQI evaluation matrices, respectively. A higher value, which is highlighted with **boldface**, indicates a more trustworthy and efficient algorithm.

Table 3: Mean values of the objective function over 10 runs obtained by cross entropy-based CPGH, PSO, GWO, and HHO algorithms.

image	Thr	PSO	GWO	HHO	CPGH	image	Thr	PSO	GWO	HHO	CPGH
COVID-CT1	2	146617703.2	146617703.2	146617703.2	146617703.2	COVID-CT6	2	490828113.2	490828113.2	490828113.2	490828113.2
	4	146884970.3	146889387.1	146883543	146889484.8		4	491257620.1	491257343.6	491250845.4	491257620.1
	6	146974234.6	146974227.6	146961336.9	146974234.6		6	491395952.7	491406148.8	491364208.3	491406672.8
	10	147028065.2	147027643.1	147016447	147028101.3		10	491507645.8	491503376.5	491472262.1	491508566.7
	15	147048028.3	147046576.9	147037756.2	147048288.3		15	491553351.2	491549721.3	491531031.9	491555488.2
	20	147055395.2	147055580.4	147046212.2	147055809		20	491570806.8	491570301.7	491550513	491571788.2
COVID-CT2	25	147058677.3	147058426	147050517.4	147059250.6	25	491578732.1	491577795.4	491563834.4	491579004.9	
	2	171455440.9	171455440.9	171455440.9	171455440.9	COVID-CT7	2	121720500.8	121720500.8	121720500.4	121720500.8
	4	171794560.4	171794271.3	171775830.5	171794560.4		4	121777915.3	121777915.3	121775242.3	121777915.3
	6	171886943.8	171886932	171876748.3	171886945.2		6	121793617.9	121794364.1	121789561.7	121794526.6
	10	171949985.2	171949935.7	171936455.4	171950071.4		10	121807890.3	121807056.9	121801315.8	121807905.5
	15	171973273.7	171971438.2	171959610.4	171973404.2		15	121813025	121812315.3	121807270.5	121812981.7
20	171981956	171981634.7	171972951.4	171982387.5	20		121814845.7	121788745.5	121811089	121814972.5	
COVID-CT3	25	171985257.8	171984685.1	171979344.6	171986260.5	25	121813312.4	121796610.2	121812069.4	121815141	
	2	177519053.7	177519053.7	177519053.7	177519053.7	COVID-CT8	2	18963872.77	18963872.77	18963872.77	18963872.77
	4	177855531.9	177855531.9	177840635	177855531.9		4	19055652.9	19055645.02	19054024.18	19055652.9
	6	177949250.5	177949245.5	177934534.1	177949250.5		6	19075389.37	19075364.82	19073612.28	19075389.37
	10	178015083.5	178014038.1	178001783.7	178015154.5		10	19085885.1	19085861.86	19083206.08	19085893.43
	15	178038939.8	178037822.3	178027228.9	178039111.5		15	19089993.99	19090335.56	19087881.01	19090264.09
20	178046808.8	178047643.1	178037845.6	178048318.5	20		19091590.31	19091871.57	19090289.6	19091908.98	
COVID-CT4	25	178051616.6	178051388.5	178045023.3	178052201.9	25	19092274.84	19092607.24	19091340.43	19092674.59	
	2	173768421.9	173768421.9	173768421.9	173768421.9	COVID-CT9	2	312673280.2	312673280.2	312673280.2	312673280.2
	4	174076778	174076778	174071046.5	174076778		4	313030502	313034381.1	313027987	313034381.1
	6	174160555.6	174159859.7	174150478.4	174160554.7		6	313151309.5	313152612	313133242.9	313152614.7
	10	174222041.7	174220482.2	174211650.7	174222050.8		10	313235712.6	313235839.8	313217940.1	313236302.1
	15	174245235.4	174244584.7	174234039.1	174245602.3		15	313271968.8	313271747.4	313255743.8	313272889.7
20	174253382.4	174253506.8	174245383.5	174254709.5	20		313286812.7	313287058.9	313272765.2	313287797	
COVID-CT5	25	174258161.4	174257483.5	174251127.1	174258711.2	25	313293051.1	313293660.2	313283977.7	313294945.6	
	2	487542519.8	487542519.8	487542519.8	487542519.8	COVID-CT10	2	499033021.5	499033021.5	499033021.5	499033021.5
	4	487831329.3	487839000.2	487826982.5	487839222.2		4	499493685.8	499493426	499474477.5	499493685.8
	6	487905742.1	487906519.2	487882528.9	487908488.7		6	499644239	499644362.8	499607341.4	499646670.9
	10	487961074.9	487961767.5	487935612.9	487963312		10	499752321.4	499752065.4	499718151.5	499752446.7
	15	487982447.4	487982138.5	487959476.9	487982753.4		15	499790798.8	499790973.4	499763841.1	499791777.5
20	487990166.4	487990376.5	487971672.4	487990766.2	20		499804586.3	499806166.4	499782453.4	499805902	
Overall Mean rank		PSO	2.71	GWO	2.41	HHO	1.24	CPGH	3.64	Sig.	8.77E-32

Tables 3 reports the mean fitness values obtained from the CPGH, PSO, GWO, and HHO with different numbers of thresholds Thr = 2, 4, 6, 10, 15, 20, 25. Successively, the CPGH has perceived the best ranking of 3.64. This is because it provided the highest fitness value in most test images at most threshold levels. More precisely, it is clear that the comparative algorithms achieved similar results at the threshold level of 2. In

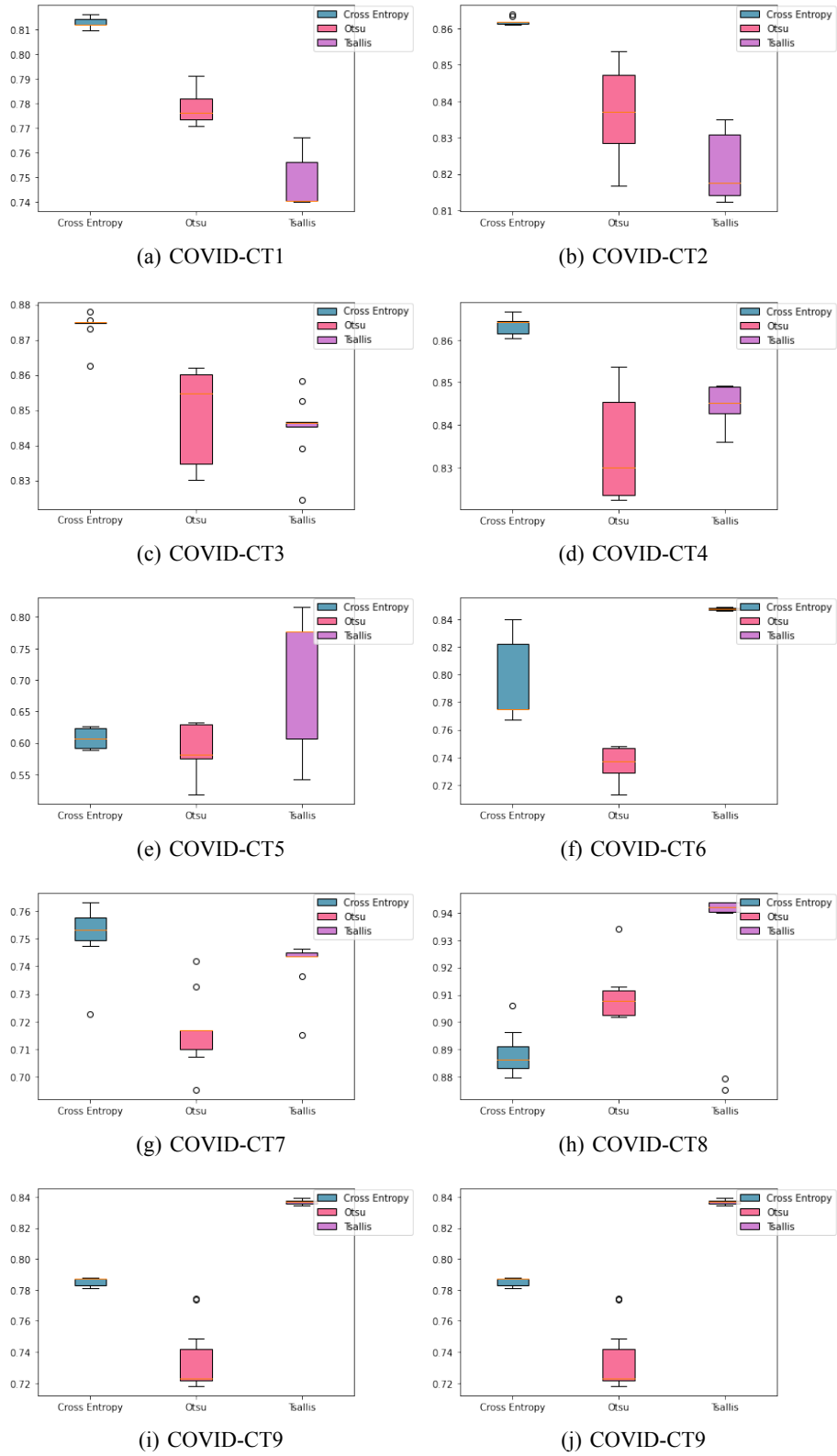


Figure 7: Box-Plots of the SSIM results obtained by PSO using cross entropy, Otsu, and Tsallis based objective functions

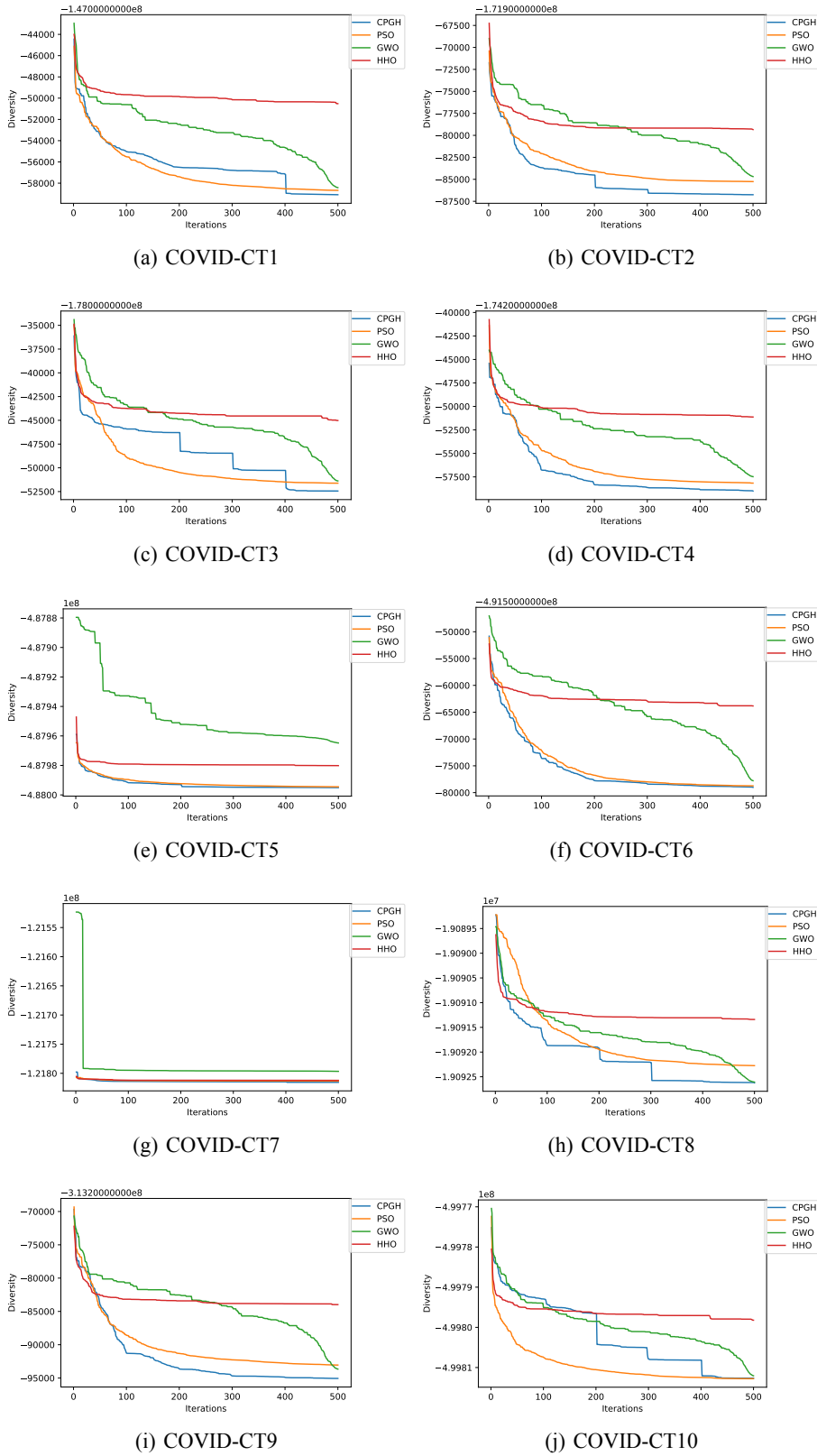


Figure 8: Convergence curves of CPGH, PSO, GWO, and HHO when dealing with tested images using 25 thresholds

comparison, the superiority of the proposed CPGH became evident with the increase in the number of threshold values.

Table 4 presents the P-values obtained by the Wilcoxon Rank-Sum test comparing the proposed CPGH versus other algorithms based on the fitness results reported in Table 3. According to the P-values, and based on a 5% significant level, when ($P\text{-value} \leq 0.05$) the difference between the results' ranks produced by a pair of algorithms is significant. While NaN means that CPGH performance is similar to the compared algorithm. According to Table 4, the numbers of ($P\text{-value} \leq 0.05$) are 17 (CPGH vs. PSO), 20 (CPGH vs. GWO), and 60 (CPGH vs HHO), respectively. The findings indicate the superiority of the CPGH; The proposed CPGH and the competing algorithms differ significantly from one another in terms of fitness results in most cases.

Table 4: P-values produced by the Wilcoxon Rank-Sum test comparing CPGH versus other algorithms as reported in Table 3 ($P\text{-values} \leq 0.05$ are significant and shown in boldface, NaN: Not Applicable)

image	Thr	CPGH vs PSO	CPGH vsGWO	CPGH vs HHO	image	Thr	CPGH vs PSO	CPGH vsGWO	CPGH vs HHO
COVID-CT1	2	NaN	NaN	NaN	COVID-CT6	2	NaN	NaN	NaN
	4	4.04E-05	3.32E-05	6.39E-05		4	NaN	3.68E-01	2.31E-04
	6	NaN	1.68E-01	6.39E-05		6	4.42E-04	7.20E-04	8.74E-05
	10	2.54E-01	1.03E-01	1.72E-04		10	8.83E-02	5.78E-03	1.82E-04
	15	2.12E-01	8.50E-01	1.83E-04		15	6.23E-01	2.57E-02	1.83E-04
	20	6.40E-02	6.23E-01	1.83E-04		20	2.11E-02	1.40E-01	1.83E-04
COVID-CT2	25	4.39E-02	7.91E-01	1.83E-04	25	7.91E-01	3.76E-02	1.83E-04	
	2	NaN	NaN	NaN	COVID-CT7	2	NaN	NaN	3.68E-01
	4	NaN	1.68E-01	6.39E-05		4	NaN	NaN	6.39E-05
	6	1.34E-03	1.20E-03	8.74E-05		6	4.97E-04	1.08E-02	1.11E-04
	10	1.19E-01	1.35E-02	1.68E-04		10	3.05E-01	4.35E-04	1.81E-04
	15	2.41E-01	7.57E-02	1.83E-04		15	3.85E-01	1.21E-01	1.83E-04
20	4.60E-02	1.73E-02	1.83E-04	20		4.27E-01	2.83E-03	1.83E-04	
COVID-CT3	25	4.27E-01	2.11E-02	1.83E-04	25	9.10E-01	1.04E-01	5.83E-04	
	2	NaN	NaN	NaN	COVID-CT8	2	NaN	NaN	NaN
	4	NaN	NaN	7.51E-04		4	NaN	3.68E-01	7.51E-04
	6	NaN	1.67E-01	6.39E-05		6	NaN	1.68E-01	6.39E-05
	10	3.05E-01	1.03E-01	1.73E-04		10	1.59E-01	3.59E-02	1.59E-04
	15	9.10E-01	1.21E-01	1.83E-04		15	1.73E-02	3.07E-01	1.83E-04
20	1.40E-02	3.76E-02	1.83E-04	20		9.11E-03	5.21E-01	1.83E-04	
COVID-CT4	25	1.62E-01	7.57E-02	1.83E-04	25	4.40E-04	7.91E-01	1.83E-04	
	2	NaN	NaN	NaN	COVID-CT9	2	NaN	NaN	NaN
	4	NaN	NaN	6.34E-05		4	7.67E-02	NaN	6.39E-05
	6	4.34E-01	1.74E-01	1.41E-04		6	1.83E-02	7.26E-01	1.31E-04
	10	6.23E-01	1.13E-02	1.81E-04		10	1.19E-01	2.51E-02	1.73E-04
	15	9.10E-01	7.34E-01	1.83E-04		15	5.90E-02	4.27E-01	1.83E-04
20	2.83E-03	1.13E-02	1.83E-04	20		3.07E-01	2.73E-01	1.83E-04	
COVID-CT5	25	5.21E-01	2.73E-01	1.83E-04	25	1.40E-01	2.12E-01	1.83E-04	
	2	NaN	NaN	3.68E-01	COVID-CT10	2	NaN	NaN	NaN
	4	7.56E-04	7.56E-04	6.39E-05		4	NaN	3.68E-01	6.39E-05
	6	1.26E-03	1.02E-01	1.72E-04		6	2.43E-05	4.88E-05	6.39E-05
	10	1.00E+00	3.82E-01	1.69E-04		10	3.42E-01	4.43E-02	1.72E-04
	15	1.86E-01	9.70E-01	1.83E-04		15	9.61E-02	1.40E-01	1.83E-04
20	4.27E-01	5.71E-01	1.83E-04	20		2.12E-01	3.07E-01	1.83E-04	
25	2.12E-01	8.90E-02	1.83E-04	25	7.91E-01	1.04E-01	1.83E-04		

Considering the segmentation quality metrics. Tables 5 presents the average PSNR scored by the CPGH and other approaches. As previously indicated, it provides an estimate of how similar the segmented image and the original are, with a more significant value indicating better segmentation quality. Inspecting the results, the PSO algorithm achieved higher PSNR scores in most images and threshold levels (overall rank of 2.82), followed by CPGH (overall rank of 2.71). On the other side, the worst method is found to be HHO, with the lowest overall rank of 2.35.

Table 5: Comparison results of the proposed CPGH and other competitors in terms of mean PSNR values.

image	Thr	PSO	GWO	HHO	CPGH		Thr	PSO	GWO	HHO	CPGH
COVID-CT1	2	10.5992	10.5992	10.5992	10.5992	COVID-CT6	2	13.7276	13.7276	13.7276	13.7276
	4	16.9341	15.3254	16.3993	15.3019		4	18.6942	18.6708	18.6088	18.6942
	6	19.4290	19.4418	19.4882	19.4290		6	20.9475	22.0148	20.2480	21.9989
	10	22.4868	22.3442	21.6659	22.3520		10	24.3580	24.7469	23.6510	26.1199
	15	23.3937	21.7203	21.7794	22.5193		15	26.1323	25.4225	23.9804	26.3324
	20	25.1163	24.0136	24.9006	24.6191		20	25.5364	26.8652	26.4685	26.2794
25	24.9831	21.6106	23.5981	24.5407	25	26.6263	24.2376	27.0337	26.0861		
COVID-CT2	2	12.0974	12.0974	12.0974	12.0974	COVID-CT7	2	13.1547	13.1547	13.1550	13.1547
	4	17.6907	17.6910	17.3987	17.6907		4	17.1775	17.1775	16.8688	17.1775
	6	21.8371	21.9303	21.9504	21.8092		6	18.5492	17.9882	17.2754	18.0755
	10	24.6374	24.7412	23.1868	24.9822		10	19.8461	20.5609	18.5002	19.9749
	15	25.2949	23.7693	22.3488	25.3906		15	20.9437	22.5667	19.3145	20.3285
	20	27.6297	24.9886	25.8564	28.5991		20	21.2299	16.8664	20.2226	20.2471
25	27.6962	20.8100	25.6887	25.2344	25	20.6540	21.3924	19.2999	20.9915		
COVID-CT3	2	11.9983	11.9983	11.9983	11.9983	COVID-CT8	2	10.3183	10.3183	10.3183	10.3183
	4	17.5386	17.5386	17.5341	17.5386		4	17.2914	17.3060	16.3700	17.2914
	6	22.1277	22.1338	22.2218	22.1277		6	17.1337	17.1059	16.5823	17.1337
	10	25.2650	24.4780	23.2585	24.9848		10	18.1071	17.8210	18.8789	17.9556
	15	25.5747	21.6613	22.8554	25.9693		15	21.4431	17.1459	16.5964	18.2818
	20	26.9708	24.6994	27.9155	28.8043		20	25.8294	19.4855	20.2345	21.3894
25	26.6418	22.2719	24.5160	26.0664	25	28.7620	17.2467	20.3597	19.5646		
COVID-CT4	2	11.9437	11.9437	11.9437	11.9437	COVID-CT9	2	14.8041	14.8041	14.8041	14.8041
	4	17.3229	17.3229	17.4434	17.3229		4	18.7576	19.5153	18.9474	19.5153
	6	22.0019	21.9104	22.2845	22.0466		6	19.9009	19.6719	19.4966	19.6549
	10	24.6259	23.9118	24.1622	24.5898		10	24.4571	24.2294	23.8030	23.5421
	15	24.8654	22.0091	23.8136	24.3899		15	25.5787	22.1686	23.2462	23.2523
	20	27.8941	22.3847	26.8718	28.4383		20	26.4452	22.8418	24.6148	25.6936
25	27.7839	22.7131	24.6049	24.9256	25	26.4195	21.1100	24.6618	24.5275		
COVID-CT5	2	13.3262	13.3262	13.3267	13.3262	COVID-CT10	2	13.8247	13.8247	13.8247	13.8247
	4	16.7286	16.8779	16.8048	16.8845		4	18.4180	18.4428	18.1964	18.4180
	6	16.9323	16.8618	17.2323	16.7530		6	21.8243	21.9804	19.8968	22.0811
	10	17.9349	18.6262	18.3177	18.0232		10	23.2876	22.9771	22.2985	23.2488
	15	18.8720	19.6680	19.2413	19.1863		15	23.5914	24.7258	23.0163	23.9560
	20	19.5308	20.3787	19.8898	18.9936		20	24.4338	25.6506	22.6981	24.9979
25	19.0411	21.2905	18.9560	20.2165	25	25.3144	26.2168	24.0219	26.6761		
Overall Mean rank		PSO	2.82	GWO	2.35	HHO	2.11	CPGH	2.71	Sig.	1.18E-03

According to the SSIM values presented in 6, Both the CPGH and PSO algorithms usually produced higher SSIM values. However, one performed better than the other for a particular set of test images. For example, CPGH provides better SSIM in the COVID-CT2, COVID-CT3, COVID-CT6, and COVID-CT10 images. As per F-test, CPGH achieves the first rank (2.90), followed by PSO (2.74), GWO (2.52), and HHO (1.84), respectively.

Table 7 tabulates the average UQI values obtained by comparative algorithms. As can be observed, the best algorithm is found to be CPGH (rank of 2.81), followed by PSO (rank of 2.69).

The Overall ranking results of the proposed approaches in dealing with all images based on segmentation quality metrics are summarized in Table 8. As illustrated in the results, we can see that the performance of the proposed CPGH is better than conventional PSO, GWO, and HHO algorithms. In specific, CPGH offered the best average rank of 3.02. Furthermore, it perceived the optimal F-test score regarding fitness, SSIM, and UQI. Hence, it can be inferred that the CPGH was the best variant in the current work. In

Table 6: Comparison results of the proposed CPGH and other competitors in terms of mean SSIM values.

image	Thr	PSO	GWO	HHO	CPGH		Thr	PSO	GWO	HHO	CPGH	
COVID-CT1	2	0.33222	0.33222	0.33222	0.33222	COVID-CT6	2	0.36703	0.36703	0.36703	0.36703	
	4	0.61554	0.60999	0.61476	0.61032		4	0.55075	0.55046	0.54659	0.55075	
	6	0.72449	0.72485	0.71888	0.72449		6	0.66533	0.67489	0.65184	0.67482	
	10	0.81202	0.81269	0.78733	0.81292		10	0.78904	0.80315	0.79345	0.82137	
	15	0.84920	0.85088	0.82698	0.84839		15	0.86444	0.86687	0.85433	0.87115	
	20	0.87463	0.88613	0.88641	0.87693		20	0.88977	0.91193	0.89510	0.90181	
COVID-CT2	25	0.87875	0.88388	0.87221	0.88179	25	0.90478	0.90661	0.90533	0.90025		
	2	0.33527	0.33527	0.33527	0.33527	COVID-CT7	2	0.39856	0.39856	0.39857	0.39856	
	4	0.66611	0.66555	0.65646	0.66611		4	0.55594	0.55594	0.54760	0.55594	
	6	0.77757	0.77773	0.76966	0.77757		6	0.65405	0.63652	0.62368	0.64078	
	10	0.85959	0.85978	0.84254	0.86081		10	0.75098	0.75701	0.71360	0.75367	
	15	0.89351	0.89116	0.86870	0.89439		15	0.79608	0.81750	0.77418	0.78895	
20	0.91551	0.90725	0.89658	0.92124	20		0.83966	0.69495	0.80722	0.83212		
COVID-CT3	25	0.92122	0.88923	0.90501	0.91662	25	0.83092	0.79478	0.79894	0.84739		
	2	0.34609	0.34609	0.34609	0.34609	COVID-CT8	2	0.58498	0.58498	0.58498	0.58498	
	4	0.67877	0.67877	0.67795	0.67877		4	0.81631	0.81646	0.79308	0.81631	
	6	0.79315	0.79323	0.79365	0.79315		6	0.83813	0.83744	0.82561	0.83813	
	10	0.87502	0.87244	0.85588	0.87505		10	0.88450	0.88163	0.88612	0.88209	
	15	0.90805	0.89335	0.89492	0.91006		15	0.92690	0.87815	0.87093	0.89510	
20	0.92622	0.91791	0.91783	0.93180	20		0.95574	0.89928	0.91483	0.93090		
COVID-CT4	25	0.93406	0.90957	0.90776	0.93135	25	0.96335	0.88142	0.91376	0.91894		
	2	0.35886	0.35886	0.35886	0.35886	COVID-CT9	2	0.40579	0.40579	0.40579	0.40579	
	4	0.65665	0.65665	0.65988	0.65665		4	0.55087	0.56507	0.55339	0.56507	
	6	0.77175	0.76999	0.77313	0.77150		6	0.64550	0.64537	0.63365	0.64516	
	10	0.86217	0.85676	0.85620	0.86165		10	0.78750	0.78733	0.77492	0.78779	
	15	0.90041	0.88531	0.87804	0.89850		15	0.86859	0.85013	0.84172	0.85795	
20	0.92739	0.89573	0.91208	0.93168	20		0.91152	0.88616	0.87141	0.90520		
COVID-CT5	25	0.93653	0.90758	0.91264	0.92445	25	0.91580	0.88166	0.88971	0.90515		
	2	0.39078	0.39078	0.39080	0.39078	COVID-CT10	2	0.36424	0.36424	0.36424	0.36424	
	4	0.50353	0.50453	0.50111	0.50443		4	0.53770	0.53767	0.53433	0.53770	
	6	0.53954	0.55027	0.55710	0.55478		6	0.67651	0.67621	0.64454	0.67858	
	10	0.60113	0.62701	0.63915	0.60175		10	0.77925	0.77524	0.77046	0.77982	
	15	0.65685	0.67234	0.68006	0.66280		15	0.83320	0.84390	0.80184	0.82918	
20	0.71422	0.74151	0.75262	0.70640	20		0.87039	0.89203	0.84377	0.87978		
Overall Mean rank		PSO	2.74	GWO	2.52		HHO	1.84	CPGH	2.90	Sig.	2.79E-07

summary, the performance of the algorithms can be ranked from best to worst as follows: CPGH, PSO, GWO, and HHO.

Taken together, the experiments and comparison findings showed the benefits of the suggested CPGH-based model when dealing with the multi-thresholding image segmentation problem. The excellent behavior of CPGH can be attributed to two reasons. Firstly, the embedded algorithms (i.e., PSO, GWO, and HHO) are practical optimizers in dealing with various optimization problems. As a result, the population evolved using diverse exploration and exploitation operators. Secondly, the diversity in exploring and exploiting the search space during the optimization process was improved using the migration process. This is due to exchanging selected solutions between the three algorithms, which allows the proposed method to show diverse exploratory behaviors.

Table 9 reports the average running time of the proposed CPGH and other compared algorithms for all test images. It is observed that the PSO algorithm has a better running time among all test algorithms for most test images at all levels. The GWO algorithm performs better for the low thresholding levels only. The proposed algorithm takes

Table 7: Comparison results of the proposed CPGH and other competitors in terms of mean UQI values.

image	Thr	PSO	GWO	HHO	CPGH	Thr	PSO	GWO	HHO	CPGH	
COVID-CT1	2	0.66474	0.66474	0.66474	0.66474	COVID-CT6	2	0.74539	0.74539	0.74539	0.74539
	4	0.79386	0.79508	0.79308	0.79529		4	0.83331	0.83235	0.82768	0.83331
	6	0.84542	0.84562	0.84218	0.84542		6	0.86314	0.85994	0.84244	0.85876
	10	0.87656	0.87832	0.85739	0.87899		10	0.90234	0.92860	0.92357	0.95305
	15	0.88978	0.89380	0.87516	0.89067		15	0.95527	0.96979	0.95161	0.96724
	20	0.89299	0.92699	0.92317	0.90113		20	0.95063	0.97240	0.96190	0.95005
COVID-CT2	25	0.88489	0.91109	0.89673	0.89756	25	0.95767	0.96420	0.96845	0.94934	
	2	0.74419	0.74419	0.74419	0.74419	COVID-CT7	2	0.62363	0.62363	0.62362	0.62363
	4	0.85556	0.85534	0.84519	0.85556		4	0.81591	0.81591	0.79545	0.81591
	6	0.89059	0.89104	0.88975	0.89046		6	0.86802	0.82999	0.79613	0.83441
	10	0.91334	0.91350	0.90386	0.91447		10	0.87789	0.89178	0.80710	0.88047
	15	0.92528	0.92466	0.91969	0.92600		15	0.89904	0.93787	0.84147	0.88432
20	0.93447	0.93443	0.92662	0.94143	20		0.89348	0.85262	0.85834	0.89009	
COVID-CT3	25	0.93511	0.90440	0.93444	0.93747	25	0.90024	0.91644	0.82411	0.88829	
	2	0.75514	0.75514	0.75514	0.75514	COVID-CT8	2	0.65097	0.65097	0.65097	0.65097
	4	0.87216	0.87216	0.86902	0.87216		4	0.82580	0.82606	0.79986	0.82580
	6	0.91100	0.91106	0.91040	0.91100		6	0.81633	0.81596	0.80908	0.81633
	10	0.93212	0.92831	0.92103	0.93249		10	0.83155	0.82850	0.84000	0.82881
	15	0.94287	0.92263	0.93739	0.94531		15	0.86030	0.80429	0.80712	0.82441
20	0.94721	0.94113	0.95095	0.95318	20		0.89144	0.82710	0.85290	0.86175	
COVID-CT4	25	0.94875	0.91735	0.94614	0.95065	25	0.90217	0.79987	0.84837	0.84384	
	2	0.76142	0.76142	0.76142	0.76142	COVID-CT9	2	0.71213	0.71213	0.71213	0.71213
	4	0.86843	0.86843	0.87193	0.86843		4	0.82527	0.82800	0.82372	0.82800
	6	0.91306	0.91114	0.91396	0.91243		6	0.86395	0.86663	0.84806	0.86640
	10	0.93454	0.92800	0.93525	0.93395		10	0.93458	0.93642	0.93249	0.94251
	15	0.94529	0.92612	0.93874	0.94374		15	0.97022	0.95104	0.95600	0.96054
20	0.95542	0.91628	0.95196	0.96162	20		0.97672	0.95027	0.95517	0.97513	
COVID-CT5	25	0.96098	0.92002	0.94405	0.94563	25	0.96618	0.93071	0.96271	0.96152	
	2	0.49195	0.49195	0.49195	0.49195	COVID-CT10	2	0.73040	0.73040	0.73040	0.73040
	4	0.68821	0.68922	0.68556	0.68922		4	0.80086	0.80092	0.79541	0.80086
	6	0.68566	0.67432	0.70131	0.66133		6	0.86221	0.86151	0.83955	0.86116
	10	0.70841	0.73207	0.70748	0.70776		10	0.87124	0.86303	0.88419	0.87160
	15	0.73865	0.76384	0.74034	0.75283		15	0.89516	0.91725	0.87794	0.88626
20	0.75908	0.80062	0.77298	0.74499	20		0.90089	0.94891	0.89645	0.91716	
25	0.73086	0.85375	0.73155	0.78063	25	0.91861	0.96477	0.91976	0.94072		
Overall Mean rank		PSO	2.69	GWO	2.59	HHO	1.91	CPGH	2.81	Sig.	1.79E-05

Table 8: Overall rank by the Friedman test for all algorithms based on fitness, PSNR, SSIM, and UQI results reported in Tables 3, 5, 6, and 7.

Measure	PSO	GWO	HHO	CPGH
Fitness	2.71	2.41	1.24	3.64
PSNR	2.82	2.35	2.11	2.71
SSIM	2.74	2.52	1.84	2.90
UQI	2.69	2.59	1.91	2.81
Mean rank	2.74	2.47	1.78	3.02
Final rank	2	3	4	1

Table 9: Average running time of the proposed CPGH and other individual algorithms

image	Thr	PSO	GWO	HHO	CPGH		Thr	PSO	GWO	HHO	CPGH
COVID-CT1	2	3.07	3.03	5.05	8.99	COVID-CT6	2	3.09	3.09	5.07	9.73
	4	4.36	6.00	9.88	10.08		4	5.95	5.95	9.89	11.72
	6	8.37	8.11	13.09	16.44		6	7.95	7.90	11.26	16.98
	10	12.68	11.47	16.48	17.05		10	11.42	11.45	13.25	24.28
	15	17.62	16.68	26.60	28.60		15	16.03	16.49	19.12	23.81
	20	17.28	17.98	29.78	41.78		20	18.77	19.33	29.28	28.85
25	21.89	23.17	35.01	35.44	25	21.65	22.56	41.78	49.32		
COVID-CT2	2	3.14	3.04	5.05	8.77	COVID-CT7	2	3.08	3.12	5.21	10.07
	4	6.06	5.96	9.88	12.59		4	5.96	6.00	9.91	11.71
	6	8.90	8.00	12.99	17.84		6	7.95	7.85	11.37	20.19
	10	12.41	11.42	16.07	17.08		10	11.61	11.34	13.41	23.80
	15	16.30	16.74	26.22	32.75		15	13.07	16.50	19.45	23.34
	20	18.54	18.11	29.24	41.97		20	18.00	18.33	29.61	28.68
25	21.89	23.28	34.97	36.45	25	21.26	20.67	34.48	48.69		
COVID-CT3	2	3.18	3.03	5.11	9.62	COVID-CT8	2	3.06	3.11	5.18	9.12
	4	6.72	6.05	9.91	13.83		4	5.97	6.11	9.85	11.70
	6	8.40	8.02	13.02	15.89		6	7.98	7.89	11.18	15.66
	10	11.37	11.39	16.13	17.78		10	12.02	11.44	13.37	23.77
	15	16.93	17.23	26.34	33.53		15	11.67	16.01	19.42	23.93
	20	18.08	18.99	28.91	42.23		20	17.88	19.97	29.58	28.85
25	21.48	23.56	35.07	51.55	25	21.42	21.57	31.91	34.41		
COVID-CT4	2	3.20	3.06	5.08	9.16	COVID-CT9	2	3.13	3.10	5.22	9.66
	4	6.79	6.51	9.98	14.05		4	6.00	5.96	9.98	15.13
	6	7.92	7.95	13.11	13.69		6	7.91	7.87	11.38	11.83
	10	11.44	11.44	15.92	21.84		10	11.45	11.44	13.77	23.56
	15	16.04	16.37	26.16	32.98		15	11.97	14.64	20.31	23.68
	20	18.03	19.32	29.03	40.92		20	18.08	18.90	33.35	28.91
25	21.71	23.86	35.99	49.11	25	21.43	21.58	32.22	33.94		
COVID-CT5	2	3.15	3.04	5.09	9.40	COVID-CT10	2	3.13	3.08	5.23	10.37
	4	6.08	6.02	9.92	10.84		4	6.02	5.94	9.94	16.36
	6	7.91	7.92	13.23	15.44		6	8.33	7.87	11.21	11.65
	10	11.44	11.37	13.35	24.90		10	11.42	11.39	13.56	24.87
	15	16.88	16.06	21.55	24.33		15	13.13	15.03	19.06	22.95
	20	17.97	19.23	28.83	29.95		20	17.75	18.33	33.33	28.80
25	22.66	23.11	35.15	48.07	25	21.38	21.54	32.05	33.87		
Overall Mean rank		PSO	1.48	GWO	1.52	HHO	3.07	CPGH	3.93	Sig.	0.00

Table 10: Threshold values and the corresponding best quality metrics obtained by the proposed CPGH using cross entropy function.

image	Thr	PSNR	SSIM	UQI	Threshold values																									
COVID-CT1	25	26.34	0.894	0.898	23	27	32	38	44	52	61	71	82	94	106	115	126	132	140	148	155	163	171	179	189	198	209	224	237	
COVID-CT2	20	30.96	0.929	0.942	17	23	30	37	46	57	70	83	96	114	130	145	155	165	174	184	194	205	218	237						
COVID-CT3	25	29.02	0.947	0.967	15	19	24	30	36	43	51	61	71	82	95	110	125	138	147	155	163	171	179	187	195	203	212	225	242	
COVID-CT4	25	25.38	0.946	0.967	12	16	21	26	32	38	44	53	60	65	70	78	95	119	134	147	156	166	175	185	193	203	212	227	242	
COVID-CT5	20	22.33	0.789	0.870	51	55	58	61	65	69	73	79	88	100	114	131	150	169	187	199	207	217	227	240						
COVID-CT6	20	28.00	0.922	0.987	5	22	39	45	51	58	65	73	83	97	115	135	157	175	186	196	205	214	225	241						
COVID-CT7	15	22.38	0.819	0.930	58	64	69	75	81	89	100	114	133	158	184	202	213	224	238											
COVID-CT8	20	28.26	0.966	0.897	2	6	16	32	49	62	72	83	95	106	117	127	135	144	153	166	183	202	222	242						
COVID-CT9	25	27.20	0.931	0.984	11	26	33	39	45	51	57	63	69	76	84	92	102	113	122	133	144	159	174	188	198	208	217	226	240	
COVID-CT10	25	29.10	0.938	0.985	1	27	38	43	49	54	60	67	73	78	84	95	105	116	129	142	158	170	183	192	201	210	219	230	245	

more time to reach the best solution than the other counterparts. The increase in time is attributable to the excessive communication overhead due to the migration process in the cooperative model. Even though the suggested CPGH was a bit slower than other methods, it could still locate better solution quality most of the time, which helped it get an exceptional score in the overall ranking.

Table 10 presents the best threshold values and the corresponding quality metrics provided by the proposed CPGH using the cross-entropy function. In contrast, the qualitative analysis of the segmentation results are given in figures inside Table 11 and 12. The figures provide the segmented images and the corresponding histograms for four levels of thresholds 6, 15, and 25. The original image is positioned beside the thresholded images to view how the thresholded images and original images differ from one another. The segmented images by the CPGH-based model are better using higher thresholds.

5.7 Comparison of CPGH with Well-known Algorithms

In this part, the effectiveness of the CPGH is further compared with the other three state-of-the-art methods. The comparison algorithms are Self-adaptive DE (SADE) [Brest et al., 2006], simple GA [Biscani and Izzo, 2020], and improved harmony search (iHS) [Mahdavi et al., 2007]. These algorithms were chosen because they fall under different categories of meta-heuristic algorithms and have a variety of exploration and exploitation capabilities. Comparative analysis in terms of cross-entropy, PSNR, and SSIM are presented in Tables 13, 14, 15, respectively. Each experiment is executed 10 independent runs per tested image, each run with 500 iterations, and the swarm size is set to 30. The experiments were performed using 25 threshold values.

As per cross_entropy results in Table 13, it can be seen that the proposed CPGH has outperformed other peers on 90% of test images. F-test shows that the CPGH is ranked first (rank of 3.9), followed by SADE (rank of 2.9), GA (rank of 2.10), and iHS (rank of 1.10).

Based on the average PSNR in Table 14, the CPGH has attained the best rates on 40% of cases. Both CPGH and GA ranked first and showed competitive results based on F-test results. Inspecting the SSIM results in Table 15, CPGH yielded the highest SSIM rates in 50% of cases (optimal rank of 3.10).

To be more precise, it can be concluded that the proposed CPGH was a valuable and potential MLT tool based on the results obtained. In most test images, it performs better than all competing techniques regarding fitness values and other segmentation quality metrics. The cooperative model of meta-heuristic algorithms helps adapt algorithms' search behaviors and provides a better exploration of the search space. Although the proposed CPGH has shown well competitive performance with other well-known algorithms, we believe that the CPGH, like the other meta-heuristics methods, follows the same No Free Lunch (NFL) theory, which asserts that no single optimization algorithm is capable of solving all optimization problems. Therefore, we still need to develop a more efficient variant of the CPGH-based MTL model.

5.8 Comparison the proposed CPGH with other state-of-the-art methods

This subsection provides a comparison between the proposed CPGH-based image segmentation and other methods. The compared methods are applied based on the same objective function as the proposed method and also used the same COVID-19 CT images dataset. Table 16 presented a comparison with the proposed CPGH algorithm, and

Table 11: Comparison of the original and segmented images at different threshold levels attained by the proposed CPGH model [Part 1]





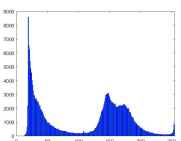
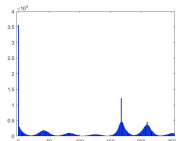
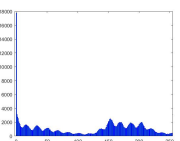
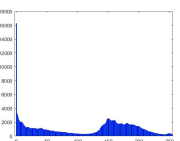
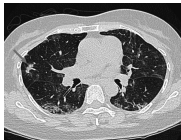



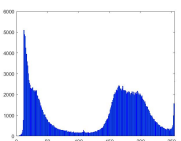
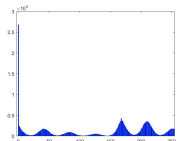
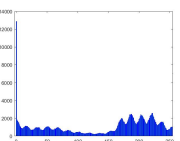
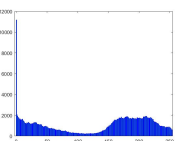
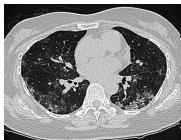

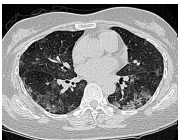
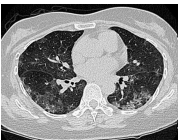
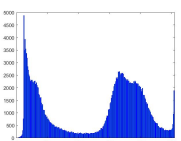
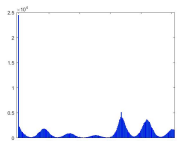
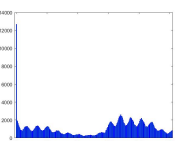
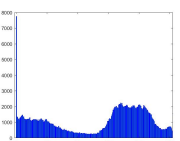




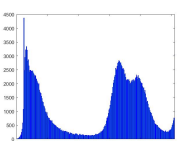
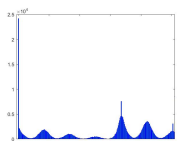
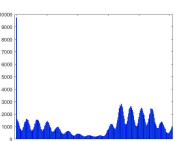
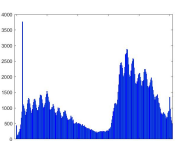




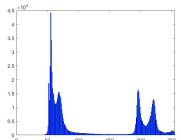
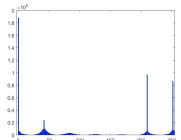
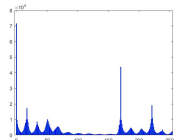
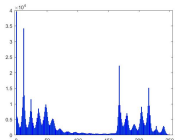
image	original	Thr=6	Thr=15	Thr=25
COVID-CT1				
				
COVID-CT2				
				
COVID-CT3				
				
COVID-CT4				
				
COVID-CT5				
				

Table 12: Comparison of the original and segmented images at different threshold levels attained by the proposed CPGH model [Part 2]





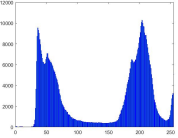
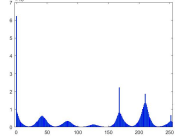
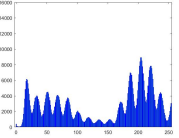
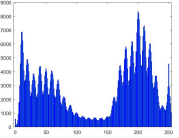




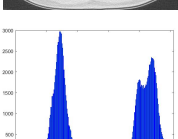
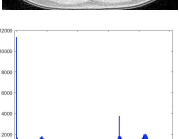
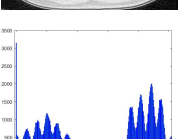
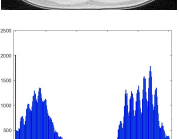
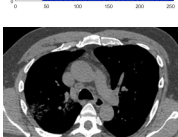











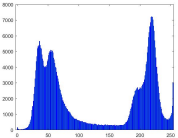
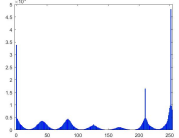
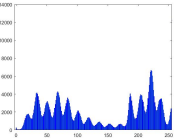
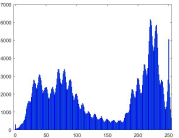




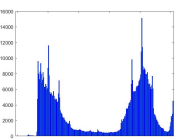
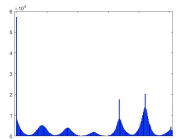
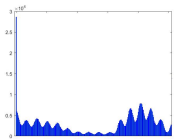
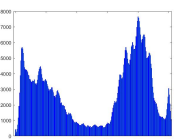
image	original	Thr=6	Thr=15	Thr=25
COVID-CT6				
				
COVID-CT7				
				
COVID-CT8				
				
COVID-CT9				
				
COVID-CT10				
				

Table 13: The cross-entropy comparison results of CPGH and other methods

image	CPGH	GA	SADE	iHS
COVID-CT1	147059250.64	147058015.72	147059045.15	147051989.90
COVID-CT2	171986260.50	171984835.44	171986065.85	171979118.83
COVID-CT3	178052201.90	178051037.10	178052141.51	178044688.30
COVID-CT4	174258711.18	174257659.41	174258849.84	174251169.15
COVID-CT5	487994823.26	487994520.61	487994640.61	487984936.35
COVID-CT6	491579004.92	491576863.55	491578736.40	491564741.60
COVID-CT7	121815141.02	121812431.15	121812223.15	121812320.15
COVID-CT8	19092674.59	19092451.03	19092657.27	19091321.54
COVID-CT9	313294945.65	313292705.13	313294412.69	313282980.43
COVID-CT10	499813424.02	499811205.96	499813200.75	499798592.98
Mesan Rank	3.90	2.10	2.90	1.10
P-value (F-test)		0.000012		

Table 14: The PSNR comparison results of CPGH and other methods

image	CPGH	GA	SADE	iHS
COVID-CT1	24.540738	22.892101	23.515427	24.409441
COVID-CT2	25.234364	26.203852	25.755971	23.736575
COVID-CT3	26.066412	24.775471	25.348622	24.992676
COVID-CT4	24.925591	24.172807	23.951069	25.025762
COVID-CT5	20.216525	21.10168	19.980076	20.494638
COVID-CT6	26.086126	27.013936	27.702032	26.55932
COVID-CT7	20.991495	22.718546	22.366848	20.12687
COVID-CT8	19.564581	19.142716	18.869984	18.822153
COVID-CT9	24.527463	24.248867	23.107309	24.020063
COVID-CT10	26.676126	26.745418	27.421983	24.896069
Mesan Rank	2.80	2.80	2.40	2.00
P-value (F-test)		0.45052		

Table 15: The SSIM comparison results of CPGH and other methods

image	CPGH	GA	SADE	iHS
COVIDCT1	0.881786	0.884758	0.883523	0.869272
COVIDCT2	0.916622	0.9203	0.918981	0.894355
COVIDCT3	0.93135	0.924004	0.925818	0.90764
COVIDCT4	0.924448	0.919439	0.922418	0.905329
COVIDCT5	0.741772	0.770027	0.730966	0.748153
COVIDCT6	0.900253	0.896457	0.901581	0.895992
COVIDCT7	0.847385	0.853812	0.847865	0.829616
COVIDCT8	0.918943	0.915127	0.914956	0.896714
COVIDCT9	0.905147	0.900615	0.898046	0.883364
COVIDCT10	0.899654	0.895912	0.898901	0.87037
Mesan Rank	3.10	3.00	2.70	1.20
P-value (F-test)		0.002851		

the MRFO-OBL [Houssein et al., 2021 a], the MRFO [Zhao et al., 2020b], MFO [Mirjalili, 2015], SCA [Mirjalili, 2016], WOA [Mirjalili and Lewis, 2016], SSA [Mirjalili et al., 2017], and EO [Faramarzi et al., 2020] methods regarding the PSNR values for COVID-19 CT images. A higher value means a more reliable and more effective algorithm. As reported in Table 16, higher PSNR values were generally achieved using the proposed CPGH algorithm in most test images, especially COVID-CT2, COVID-CT3, COVID-CT4, COVID-CT6, COVID-CT9, and COVID-CT10 images.

Table 16: Comparison between CPGH and all other methods according to the PSNR mean values.

Test Image	Th	MFO	WOA	SCA	SSA	EO	MRFO	MRFO-OBL	CPGH
COVID-CT1	10	22.785	22.7217	20.4506	22.7112	22.8869	22.7883	22.8366	22.3520
	15	23.605	23.6016	21.4088	23.5316	23.6355	23.5724	23.6096	22.5193
	20	24.383	24.3157	22.0108	24.3711	24.5068	24.4058	24.4777	24.6191
	25	24.956	24.9986	22.7393	24.9413	25.2453	25.0747	25.0750	24.5407
COVID-CT2	10	23.340	23.2898	21.0569	23.3548	23.3451	23.3543	23.3691	24.9822
	15	24.252	24.2280	22.2538	24.2437	24.2915	24.2500	24.2945	25.3906
	20	25.1918	25.1816	22.4196	25.1842	25.1998	25.2425	25.2675	28.5991
	25	25.9299	25.9645	22.6362	25.9269	26.0340	25.9906	26.0447	28.5991
COVID-CT3	10	23.3127	23.3402	20.7045	23.3244	23.3664	23.3779	23.3835	24.9848
	15	24.2557	24.2363	21.5903	24.2289	24.2585	24.2597	24.2920	25.9693
	20	25.2191	25.0960	22.4194	25.1901	25.1978	25.2401	25.2815	28.8043
	25	25.9385	25.9699	23.0266	25.9578	26.0504	26.0143	26.0406	26.0664
COVID-CT4	10	23.1153	22.9441	21.0348	23.0054	23.0594	23.0455	23.1315	24.5898
	15	24.1451	24.1102	21.4647	24.1141	24.1073	24.1247	24.1701	24.3899
	20	25.2197	25.1061	22.2991	25.2064	25.1943	25.2826	25.3379	28.4383
	25	26.0147	25.9803	22.7990	25.9808	26.1555	26.0265	26.0860	26.9256
COVID-CT5	10	17.7297	18.0911	17.3997	17.8317	17.7346	17.6816	17.6923	18.0232
	15	18.0694	18.6874	17.1534	18.3007	18.1689	18.0973	18.1808	19.1863
	20	18.4704	19.4253	18.0159	19.0553	18.6530	18.4984	18.6158	18.9936
	25	18.8423	20.3143	18.2669	19.1845	19.3208	18.9574	19.0474	20.2165
COVID-CT6	10	21.1169	21.0827	19.8469	21.0500	21.1999	21.0786	21.1912	26.1199
	15	22.0606	22.1863	20.3460	22.1110	22.2209	22.2276	22.2106	26.3324
	20	22.9744	22.7802	20.8418	23.0122	23.0780	22.9386	23.0090	26.2794
	25	23.4674	23.4204	22.0136	23.7771	23.8257	23.5627	23.6860	26.0861
COVID-CT7	10	19.6031	19.3921	18.3899	19.4554	19.7201	19.4133	19.6895	19.9749
	15	20.6718	20.3370	18.5243	20.6694	20.8207	20.7551	20.7020	20.3285
	20	21.5069	21.4917	19.9200	21.5961	21.8813	21.4188	21.8512	20.2471
	25	22.4764	22.9517	20.4139	22.1888	22.8390	22.6882	22.9905	20.9915
COVID-CT8	10	24.1907	24.1384	22.3768	24.1360	24.3791	24.0661	24.1512	17.9556
	15	25.6295	25.5060	23.0455	25.3830	25.8864	25.4969	25.6688	18.2818
	20	27.0029	27.0489	23.5396	27.1684	26.9507	27.0617	27.0500	21.3894
	25	27.8852	27.8808	24.2742	27.9904	27.9053	27.9826	27.9897	28.5646
COVID-CT9	10	21.7801	21.6394	20.3303	21.7302	21.8315	21.7791	21.7893	23.5421
	15	22.6114	22.4253	21.0667	22.4741	22.7819	22.5405	22.6872	23.2523
	20	23.6409	23.5144	21.5845	23.4062	24.0351	23.6402	23.7957	25.6936
	25	24.6765	24.1344	21.9324	24.5486	25.0857	24.7814	25.0439	24.5275
COVID-CT10	10	20.5813	20.3124	19.4163	20.4473	20.6924	20.4594	20.6069	23.2488
	15	21.9369	21.6074	20.4364	21.9044	22.1685	22.0541	22.1712	23.9560
	20	22.9325	22.9381	20.8123	23.0184	23.0016	23.0701	23.0850	24.9979
	25	23.4817	23.8029	22.0695	23.6455	23.7193	23.5813	23.6204	26.6761

6 Conclusion and future work

COVID-19 has been spreading over the world since December 2019, everyone has been working to develop methods for telling infected people from healthy ones. CT scans might effectively determine whether the suspected patients have been infected after extensive testing and verification by medical experts. In order to improve the classification methods, a variety of segmentation techniques are used to extract regions of interest from CT scans. Medical image segmentation is regarded as an essential stage for many medical applications that must be carried out properly for useful picture analysis. Thresholding is one of the most fundamental and important techniques for image segmentation. Finding the best thresholding values for multilevel thresholding image segmentation was treated as an optimization issue in this study, we use three methods: Otsu's, cross-entropy, and Tsaliis's approach as the objective function. So, a cooperative swarm intelligence-based MLT for developing an efficient image segmentation approach for COVID-19 CT images has been proposed. Three algorithms comprising PSO, GWO, and HHO were used to develop an algorithmic level-based cooperative optimization model called CPGH for finding the optimum thresholding values. Moreover, the proposed approach tested cross-entropy, Otsu, and Tsallis as objective functions. Real-life CT scan images of the COVID-19 dataset were employed to evaluate the proposed meta-heuristic cooperative model. Several metrics covering PSNR, SSIM, and UQI were utilized for evaluation purposes. The experimental results proved that the proper objective function selection often guarantees a better segmentation quality. It was noticed that entropy-based thresholding has a more significant impact on the performance of the MLT-based image segmentation than other methods used in the comparisons. Furthermore, the proposed CPGH performed better than conventional PSO, GWO, and HHO and the other algorithms: SADE, GA, and iHS, in dealing with the MLT image segmentation problem.

According to the proposed algorithm's efficiency results, we can use it in the future for various real-world applications, such as calculating solar cell parameters, object tracking, electrical applications, hyperparameter optimization, and color image segmentation. Also, one possible work is to use other meta-heuristic algorithms such as WOA, SCA, and MFO as alternatives to PSO, GWO, and HHO algorithms in the cooperative model since these algorithms have shown promising results in MLT image segmentation problems. Moreover, we will extend the proposed method to test in other datasets, such as COVID-19 X-ray images and other medical imaging applications. We will also examine the proposed method with different migration settings to improve the segmentation results. Other challenging real-world problems like parameter identification, traffic lights schedule, and feature selection can be addressed in the future using the suggested CPGH model. Another approach is to create a binary version of CPGH and apply it to binary problems, including feature selection problems.

Acknowledgements

References

- [Ab Wahab et al., 2015] Ab Wahab, M. N., Nefti-meziati, S., and Atyabi, A. (2015). A comprehensive review of swarm optimization algorithms. *PLoS one*, 10:e0122827.
- [Abdel-Basset et al., 2021] Abdel-Basset, M., Chang, V., and Mohamed, R. (2021). A novel equilibrium optimization algorithm for multi-thresholding image segmentation problems. *Neural Computing and Applications*, 33:10685–10718.

- [Abed-alguni et al., 2019] Abed-alguni, B., Klaib, A., and Nahar, K. (2019). Island-based whale optimization algorithm for continuous optimization problems. *International Journal of Reasoning-based Intelligent Systems*, 11:319–329.
- [Abualigah et al., 2021a] Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M., and Gandomi, A. H. (2021a). The arithmetic optimization algorithm. *Computer methods in applied mechanics and engineering*, 376:113609.
- [Abualigah et al., 2021b] Abualigah, L., Diabat, A., Sumari, P., and Gandomi, A. H. (2021b). A novel evolutionary arithmetic optimization algorithm for multilevel thresholding segmentation of covid-19 ct images. *Processes*, 9(7).
- [Abualigah et al., 2021c] Abualigah, L., Yousri, D., Abd Elaziz, M., Ewees, A. A., Al-Qaness, M. A., and Gandomi, A. H. (2021c). Aquila optimizer: a novel meta-heuristic optimization algorithm. *Computers & Industrial Engineering*, 157:107250.
- [Agrawal et al., 2013] Agrawal, S., Panda, R., Bhuyan, S., and Panigrahi, B. K. (2013). Tsallis entropy based optimal multilevel thresholding using cuckoo search algorithm. *Swarm and Evolutionary Computation*, 11:16–30.
- [Akbari et al., 2010] Akbari, R., Mohammadi, A., and Ziarati, K. (2010). A novel bee swarm optimization algorithm for numerical function optimization. *Communications in Nonlinear Science and Numerical Simulation*, 15(10):3142–3155.
- [Al-Betar and Awadallah, 2018] Al-Betar, M. A. and Awadallah, M. A. (2018). Island bat algorithm for optimization. *Expert Systems with Applications*, 107:126–145.
- [Alba et al., 2013] Alba, E., Luque, G., and Nesmachnow, S. (2013). Parallel metaheuristics: recent advances and new trends. *International Transactions in Operational Research*, 20(1):1–48.
- [Arcuri and Fraser, 2013] Arcuri, A. and Fraser, G. (2013). Parameter tuning or default values? an empirical investigation in search-based software engineering. *Empirical Software Engineering*, 18(3).
- [Arora et al., 2008] Arora, S., Acharya, J., Verma, A., and Panigrahi, P. K. (2008). Multilevel thresholding for image segmentation through a fast statistical recursive algorithm. *Pattern Recognition Letters*, 29(2):119–125.
- [Avcibas et al., 2002] Avcibas, I., Sankur, B., and Sayood, K. (2002). Statistical evaluation of image quality measures. *Journal of Electronic Imaging*, 11(2):206 – 223.
- [Baby Resma and Nair, 2021] Baby Resma, K. and Nair, M. S. (2021). Multilevel thresholding for image segmentation using krill herd optimization algorithm. *Journal of King Saud University - Computer and Information Sciences*, 33(5):528–541.
- [Bandyopadhyay et al., 2021] Bandyopadhyay, R., Kundu, R., Oliva, D., and Sarkar, R. (2021). Segmentation of brain mri using an altruistic harris hawks’ optimization algorithm. *Knowledge-Based Systems*, 232:107468.
- [Biscani and Izzo, 2020] Biscani, F. and Izzo, D. (2020). A parallel global multiobjective framework for optimization: pagmo. *Journal of Open Source Software*, 5(53):2338.
- [Boussaid et al., 2013] Boussaid, I., Lepagnot, J., and Siarry, P. (2013). A survey on optimization metaheuristics. *Information Sciences*, 237:82 – 117. Prediction, Control and Diagnosis using Advanced Neural Computations.
- [Brest et al., 2006] Brest, J., Greiner, S., Boskovic, B., Mernik, M., and Zumer, V. (2006). Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation*, 10(6):646–657.
- [Chakraborty et al., 2019] Chakraborty, F., Roy, P., and Nandi, D. (2019). Oppositional elephant herding optimization with dynamic cauchy mutation for multilevel image thresholding. *Evolutionary Intelligence*, 12:445–467.

- [Chen et al., 2022] Chen, Y., Wang, M., Heidari, A. A., Shi, B., Hu, Z., Zhang, Q., Chen, H., Mafarja, M., and Turabieh, H. (2022). Multi-threshold image segmentation using a multi-strategy shuffled frog leaping algorithm. *Expert Systems with Applications*, 194:116511.
- [Chen et al., 2020] Chen, Z., Li, X., Zhu, Z., Zhao, Z., Wang, L., Jiang, S., and Rong, Y. (2020). The optimization of accuracy and efficiency for multistage precision grinding process with an improved particle swarm optimization algorithm. *International Journal of Advanced Robotic Systems*, 17:172988141989350.
- [Cohen et al., 2020a] Cohen, J., Morrison, P., Dao, L., Roth, K., Duong, T., and Ghassemi, M. (2020a). Covid-19 image data collection: Prospective predictions are the future.
- [Cohen et al., 2020b] Cohen, J. P., Morrison, P., Dao, L., Roth, K., Duong, T. Q., and Ghassemi, M. (2020b). Covid-19 image data collection: Prospective predictions are the future. *arXiv 2006.11988*.
- [Crainic, 2016] Crainic, T. G. (2016). *Parallel Meta-heuristic Search*.
- [Díaz-Cortés et al., 2018] Díaz-Cortés, M.-A., Ortega-Sánchez, N., Hinojosa, S., Oliva, D., Cuevas, E., Rojas, R., and Demin, A. (2018). A multi-level thresholding method for breast thermograms analysis using dragonfly algorithm. *Infrared Physics & Technology*, 93:346–361.
- [Eberhart and Kennedy, 1995] Eberhart, R. and Kennedy, J. (1995). Particle swarm optimization. In *Proceedings of the IEEE international conference on neural networks*, volume 4, pages 1942–1948. Citeseer.
- [Eken, 2020] Eken, S. (2020). A topic-based hierarchical publish/subscribe messaging middleware for covid-19 detection in x-ray image and its metadata. *Soft Computing*, pages 1–11.
- [Elaziz et al., 2020] Elaziz, M. A., Ewees, A. A., Yousri, D., Alwerfali, H. S. N., Awad, Q. A., Lu, S., and 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.
- [Elsayed Abd Elaziz et al., 2021] Elsayed Abd Elaziz, M., Neggaz, N., Moghdani, R., Ewees, A., Cuevas, E., and Lu, S. (2021). Multilevel thresholding image segmentation based on improved volleyball premier league algorithm using whale optimization algorithm. *Multimedia Tools and Applications*, 80.
- [Emam et al., 2022] Emam, M. M., Houssein, E. H., and Ghoniem, R. M. (2022). A modified reptile search algorithm for global optimization and image segmentation: Case study brain mri images. *Computers in Biology and Medicine*, page 106404.
- [Eshtay et al., 2019] Eshtay, M., Faris, H., and Obeid, N. (2019). Metaheuristic-based extreme learning machines: a review of design formulations and applications. *International Journal of Machine Learning and Cybernetics*, 10:1543–1561.
- [Faramarzi et al., 2020] Faramarzi, A., Heidarnejad, M., Stephens, B., and Mirjalili, S. (2020). Equilibrium optimizer: A novel optimization algorithm. *Knowledge-Based Systems*, 191:105190.
- [Ferrer, 2020] Ferrer, R. (2020). Covid-19 pandemic: the greatest challenge in the history of critical care. *Medicina intensiva*.
- [Gao et al., 2017] Gao, H., Fu, Z., Pun, C.-M., Hu, H., and Lan, R. (2017). A multi-level thresholding image segmentation based on an improved artificial bee colony algorithm. *Computers & Electrical Engineering*, 70.
- [Gao et al., 2010] Gao, H., Xu, W., Sun, J., and Tang, Y. (2010). Multilevel thresholding for image segmentation through an improved quantum-behaved particle swarm algorithm. *IEEE Transactions on Instrumentation and Measurement*, 59(4):934–946.
- [Glasbey, 1993] Glasbey, C. A. (1993). An analysis of histogram-based thresholding algorithms. *CVGIP: Graphical models and image processing*, 55(6):532–537.

- [Glover and Kochenberger, 2003] Glover, F. and Kochenberger, G. (2003). Handbook of metaheuristics. *Kluwer Academic Publishers, Boston, 2003*, 57.
- [Harmon et al., 2020] Harmon, S., Sanford, T., Xu, S., Turkbey, E., Roth, H., Xu, Z., Yang, D., Myronenko, A., Anderson, V., Amalou, A., Blain, M., Kassin, M., Long, D., Varble, N., Walker, S., Ierardi, A., Stellato, E., Plensich, G., and Turkbey, B. (2020). Artificial intelligence for the detection of covid-19 pneumonia on chest ct using multinational datasets. *Nature Communications*, 11:4080.
- [He and Huang, 2017] He, L. and Huang, S. (2017). Modified firefly algorithm based multilevel thresholding for color image segmentation. *Neurocomputing*, 240:152–174.
- [Heidari et al., 2019a] Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., and Chen, H. (2019a). Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems*, 97:849–872.
- [Heidari et al., 2019b] Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., and Chen, H. (2019b). Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems*, 97:849–872.
- [Houssein et al., 2021a] Houssein, E., Emam, M., and Ali, A. (2021a). Improved manta ray foraging optimization for multi-level thresholding using covid-19 ct images. *Neural Computing and Applications*, 33:1–21.
- [Houssein et al., 2022a] Houssein, E. H., Abdelkareem, D. A., Emam, M. M., Hameed, M. A., and Younan, M. (2022a). An efficient image segmentation method for skin cancer imaging using improved golden jackal optimization algorithm. *Computers in Biology and Medicine*, page 106075.
- [Houssein et al., 2022b] Houssein, E. H., Abohashima, Z., Elhoseny, M., and Mohamed, W. M. (2022b). Hybrid quantum-classical convolutional neural network model for covid-19 prediction using chest x-ray images. *Journal of Computational Design and Engineering*, 9(2):343–363.
- [Houssein et al., 2021b] Houssein, E. H., din Helmy, B. E., Oliva, D., Elngar, A. A., and Shaban, H. (2021b). A novel black widow optimization algorithm for multilevel thresholding image segmentation. *Expert Systems with Applications*, 167:114159.
- [Houssein et al., 2022c] Houssein, E. H., din Helmy, B. E., Oliva, D., Jangir, P., Premkumar, M., Elngar, A. A., and Shaban, H. (2022c). An efficient multi-thresholding based covid-19 ct images segmentation approach using an improved equilibrium optimizer. *Biomedical Signal Processing and Control*, 73:103401.
- [Houssein et al., 2021c] Houssein, E. H., Emam, M. M., and Ali, A. A. (2021c). An efficient multilevel thresholding segmentation method for thermography breast cancer imaging based on improved chimp optimization algorithm. *Expert Systems with Applications*, 185:115651.
- [Houssein et al., 2022d] Houssein, E. H., Emam, M. M., and Ali, A. A. (2022d). An optimized deep learning architecture for breast cancer diagnosis based on improved marine predators algorithm. *Neural Computing and Applications*, pages 1–19.
- [Houssein et al., 2021d] Houssein, E. H., Hussain, K., Abualigah, L., Elaziz, M. A., Alomoush, W., Dhiman, G., Djenouri, Y., and Cuevas, E. (2021d). An improved opposition-based marine predators algorithm for global optimization and multilevel thresholding image segmentation. *Knowledge-Based Systems*, 229:107348.
- [Kapur et al., 1985a] Kapur, J., Sahoo, P., and Wong, A. (1985a). A new method for gray-level picture thresholding using the entropy of the histogram. *Computer Vision, Graphics, and Image Processing*, 29(3):273–285.
- [Kapur et al., 1985b] Kapur, J. N., Sahoo, P. K., and Wong, A. K. (1985b). A new method for gray-level picture thresholding using the entropy of the histogram. *Computer vision, graphics, and image processing*, 29(3):273–285.
- [Kaya et al., 2022] Kaya, Y., Yiner, Z., Kaya, M., and Kuncan, F. (2022). A new approach to covid-19 detection from x-ray images using angle transformation with googlenet and lstm.

Measurement Science and Technology, 33(12):124011.

[Khairuzzaman and Chaudhury, 2017] Khairuzzaman, A. K. and Chaudhury, S. (2017). Multilevel thresholding using grey wolf optimizer for image segmentation. *Expert Systems with Applications*, 86.

[Lampinen and Storn, 2004] Lampinen, J. and Storn, R. (2004). Differential evolution. In *New optimization techniques in engineering*, pages 123–166. Springer.

[Liang et al., 2019] Liang, H., Jia, H., Xing, Z., Ma, J., and Peng, X. (2019). Modified grasshopper algorithm-based multilevel thresholding for color image segmentation. *IEEE Access*, 7:11258–11295.

[Liao et al., 2007] Liao, C., Li, S., and Luo, Z. (2007). Gene selection using wilcoxon rank sum test and support vector machine for cancer classification. volume 4456, pages 57–66.

[Liu et al., 2021] Liu, L., Zhao, D., Yu, F., Heidari, A. A., Li, C., Ouyang, J., Chen, H., Mafarja, M., Turabieh, H., and 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.

[Mahdavi et al., 2007] Mahdavi, M., Fesanghary, M., and Damangir, E. (2007). An improved harmony search algorithm for solving optimization problems. *Applied Mathematics and Computation*, 188(2):1567–1579.

[Merzban and Elbayoumi, 2019] Merzban, M. H. and Elbayoumi, M. (2019). Efficient solution of otsu multilevel image thresholding: A comparative study. *Expert Systems with Applications*, 116:299–309.

[Mirjalili, 2015] Mirjalili, S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-based systems*, 89:228–249.

[Mirjalili, 2016] Mirjalili, S. (2016). Sca: a sine cosine algorithm for solving optimization problems. *Knowledge-based systems*, 96:120–133.

[Mirjalili, 2019] Mirjalili, S. (2019). Genetic algorithm. In *Evolutionary algorithms and neural networks*, pages 43–55. Springer.

[Mirjalili et al., 2017] Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., and Mirjalili, S. M. (2017). Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software*, 114:163–191.

[Mirjalili and Lewis, 2016] Mirjalili, S. and Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software*, 95:51 – 67.

[Mirjalili et al., 2014a] Mirjalili, S., Mirjalili, S. M., and Lewis, A. (2014a). Grey wolf optimizer. *Advances in Engineering Software*, 69:46 – 61.

[Mirjalili et al., 2014b] Mirjalili, S., Mirjalili, S. M., and Lewis, A. (2014b). Grey wolf optimizer. *Advances in engineering software*, 69:46–61.

[Nair et al., 2021] Nair, R., Alhudhaif, A., Koundal, D., Doewes, R. I., and Sharma, P. (2021). Deep learning-based covid-19 detection system using pulmonary ct scans. *Turkish Journal of Electrical Engineering and Computer Sciences*, 29(8):2716–2727.

[Nama, 2022] Nama, S. (2022). A novel improved sma with quasi reflection operator: Performance analysis, application to the image segmentation problem of covid-19 chest x-ray images. *Applied Soft Computing*, 118:108483.

[Otsu, 1979a] Otsu, N. (1979a). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66.

[Otsu, 1979b] Otsu, N. (1979b). A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66.

- [Panwar et al., 2020a] Panwar, H., Gupta, P., Siddiqui, M. K., Morales-Menendez, R., Bhardwaj, P., and Singh, V. (2020a). A deep learning and grad-cam based color visualization approach for fast detection of covid-19 cases using chest x-ray and ct-scan images. *Chaos, Solitons & Fractals*, 140:110190.
- [Panwar et al., 2020b] Panwar, H., Gupta, P., Siddiqui, M. K., Morales-Menendez, R., and Singh, V. (2020b). Application of deep learning for fast detection of covid-19 in x-rays using ncovnet. *Chaos, Solitons & Fractals*, 138:109944.
- [Poli et al., 2007] Poli, R., Kennedy, J., and Blackwell, T. (2007). Particle swarm optimization. *Swarm intelligence*, 1(1):33–57.
- [Qi et al., 2022] Qi, A., Zhao, D., Yu, F., Heidari, A. A., Wu, Z., Cai, Z., Alenezi, F., Mansour, R. F., Chen, H., and Chen, M. (2022). Directional mutation and crossover boosted ant colony optimization with application to covid-19 x-ray image segmentation. *Computers in biology and medicine*, 148:105810.
- [Rashedi et al., 2009] Rashedi, E., Nezamabadi-Pour, H., and Saryazdi, S. (2009). Gsa: a gravitational search algorithm. *Information sciences*, 179(13):2232–2248.
- [Richardson, 2015] Richardson, A. (2015). Nonparametric statistics: A step-by-step approach. *International Statistical Review*, 83.
- [Rodríguez et al., 2022] Rodríguez, A., Cuevas, E., Zaldivar, D., Morales-Castañeda, B., Sarkar, R., and Houssein, E. H. (2022). An agent-based transmission model of covid-19 for re-opening policy design. *Computers in Biology and Medicine*, page 105847.
- [Rodríguez-Esparza et al., 2020] Rodríguez-Esparza, E., Zanella-Calzada, L. A., Oliva, D., Heidari, A. A., Zaldivar, D., Pérez-Cisneros, M., and Foong, L. K. (2020). An efficient harris hawks-inspired image segmentation method. *Expert Systems with Applications*, 155:113428.
- [Rojas-Morales et al., 2017] Rojas-Morales, N., Riff Rojas, M.-C., and Montero Ureta, E. (2017). A survey and classification of opposition-based metaheuristics. *Computers & Industrial Engineering*, 110:424–435.
- [Sezgin and Sankur, 2004] Sezgin, M. and Sankur, B. (2004). Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1):146–165.
- [Siddiqui et al., 2020] Siddiqui, M. K., Morales-Menendez, R., Gupta, P. K., Iqbal, H., Hussain, F., Khatoon, K., and Ahmad, S. (2020). Correlation between temperature and covid-19 (suspected, confirmed and death) cases based on machine learning analysis. *J Pure Appl Microbiol*, 14(suppl 1):1017–1024.
- [Sohrabi et al., 2020] Sohrabi, C., Alsafi, Z., O’Neill, N., Khan, M., Kerwan, A., Al-Jabir, A., Iosifidis, C., and Agha, R. (2020). World health organization declares global emergency: A review of the 2019 novel coronavirus (covid-19). *International journal of surgery*, 76:71–76.
- [Song et al., 2017] Song, J.-H., Cong, W., and Li, J. (2017). A fuzzy c-means clustering algorithm for image segmentation using nonlinear weighted local information. *Journal of Information Hiding and Multimedia Signal Processing*, 8:578–588.
- [Su et al., 2022] Su, H., Zhao, D., Elmannai, H., Heidari, A. A., Bourouis, S., Wu, Z., Cai, Z.-N., Gui, W., and Chen, M. (2022). Multilevel threshold image segmentation for covid-19 chest radiography: A framework using horizontal and vertical multiverse optimization. *Computers in Biology and Medicine*, 146:105618.
- [sung Liao et al., 2001] sung Liao, P., sheng Chen, T., and choo Chung, P. (2001). A fast algorithm for multilevel thresholding. *Journal of Information Science and Engineering*, 17:713–727.
- [Suresh and Lal, 2016] Suresh, S. and Lal, S. (2016). An efficient cuckoo search algorithm based multilevel thresholding for segmentation of satellite images using different objective functions. *Expert Systems with Applications*, 58:184–209.

- [Talbi, 2009] Talbi, E. (2009). *Metaheuristics From design to implementation*. Wiley Online Library.
- [Thaher and Arman, 2020] Thaher, T. and Arman, N. (2020). Efficient multi-swarm binary harris hawks optimization as a feature selection approach for software fault prediction. In *2020 11th International Conference on Information and Communication Systems (ICICS)*, pages 249–254.
- [Thaher et al., 2022a] Thaher, T., Awad, M., Aldasht, M., Sheta, A., Turabieh, H., and Chantar, H. (2022a). An enhanced evolutionary based feature selection approach using grey wolf optimizer for the classification of high-dimensional biological data. *JUCS - Journal of Universal Computer Science*, 28(5):499–539.
- [Thaher et al., 2022b] Thaher, T., Chantar, H., Too, J., Mafarja, M., Turabieh, H., and Houssein, E. H. (2022b). Boolean particle swarm optimization with various evolutionary population dynamics approaches for feature selection problems. *Expert Systems with Applications*, 195:116550.
- [Thaher et al., 2020] Thaher, T., Heidari, A. A., Mafarja, M., Dong, J. S., and Mirjalili, S. (2020). *Binary Harris Hawks Optimizer for High-Dimensional, Low Sample Size Feature Selection*, pages 251–272. Springer Singapore, Singapore.
- [Thaher and Sartawi, 2020] Thaher, T. and Sartawi, B. (2020). An experimental design approach to analyse the performance of island-based parallel artificial bee colony algorithm. In *2020 IEEE 14th International Conference on Application of Information and Communication Technologies (AICT)*, pages 1–7.
- [Tsai, 1985] Tsai, W.-H. (1985). Moment-preserving thresholding: A new approach. *Computer vision, graphics, and image processing*, 29(3):377–393.
- [Upadhyay and Chhabra, 2020] Upadhyay, P. and Chhabra, J. K. (2020). Kapur’s entropy based optimal multilevel image segmentation using crow search algorithm. *Applied Soft Computing*, 97:105522.
- [Wang and Bovik, 2002] Wang, Z. and Bovik, A. (2002). A universal image quality index. *IEEE Signal Processing Letters*, 9(3):81–84.
- [Wang et al., 2004] Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612.
- [Wolpert, 1996] Wolpert, D. H. (1996). The lack of a priori distinctions between learning algorithms. *Neural Computation*, 8(7):1341–1390.
- [Yılmaz, 2021] Yılmaz, A. (2021). Diagnosing covid-19 from x-ray images with using multi-channel cnn architecture. *Journal of the Faculty of Engineering and Architecture of Gazi University*, 36(4):1761–1774.
- [Yu et al., 2022] Yu, H., Song, J., Chen, C., Heidari, A. A., Liu, J., Chen, H., Zaguia, A., and Mafarja, M. (2022). Image segmentation of leaf spot diseases on maize using multi-stage cauchy-enabled grey wolf algorithm. *Engineering Applications of Artificial Intelligence*, 109:104653.
- [Zhao et al., 2020a] Zhao, J., Zhang, Y., He, X., and Xie, P. (2020a). Covid-ct-dataset: a ct scan dataset about covid-19. *arXiv preprint arXiv:2003.13865*.
- [Zhao et al., 2020b] Zhao, W., Zhang, Z., and Wang, L. (2020b). Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications. *Engineering Applications of Artificial Intelligence*, 87:103300.