

A HYPER-HEURISTIC BASED STRATEGY FOR IMAGE SEGMENTATION USING MULTILEVEL THRESHOLDING

LUQMAN, FAKHRUD DIN, SHAH KHALID, KAMAL Z. ZAMLI* & AFTAB ALAM

ABSTRACT

Segmentation is a key process in image analysis as it involves dividing and separating a digital image into different regions of pixels for further detailed investigation. Among others, multilevel thresholding is a robust and most widely adopted image segmentation approach. To find the optimal multilevel threshold values, various heuristic and meta-heuristic algorithms have been applied to segment COVID-19 medical images. Although effective, these algorithms get stuck in local optima and need proper parameter tuning for solving optimisation problems. To address these issues, this work proposes a novel multilevel thresholding strategy using Exponential Monte Carlo with Counter (EMCQ) hyper-heuristic for the segmentation of Computed Tomography (CT) chest images. EMCQ uses four low-level heuristic sets adopted from the teaching learning-based optimisation (TLBO) algorithm, flower pollination algorithm (FPA), genetic algorithm (GA), and Jaya algorithm. One of these low-level heuristics is selected by EMCQ in each iteration based on the best performance. Apart from the best low-level heuristic, EMCQ also gives a chance to low-performing heuristics to search for the optimal threshold values using their probability density function. For performance evolution, common image processing performance metrics like the feature similarity index method (FSIM), mean square error (MSE), peak signal to noise ratio (PSNR), and structural similarity index measure (SSIM) are utilised. Experimental results of the proposed strategy are compared with other advanced meta-heuristic algorithms using the Otsu and Kapur fitness functions. For the CT images, the proposed strategy achieved mean best values of 83% for PSNR, 95% for MSE, and 66% for FSIM. On the classical image dataset, it outperformed an existing hyper-heuristic strategy by obtaining 70% best SSIM values compared to 50% from the existing approach. The validity of the proposed strategy is further confirmed through statistical analysis using the Wilcoxon rank-sum test. The experimental results, supported by statistical evidence, demonstrate the superiority of the new EMCQ-based image segmentation strategy in terms of attaining optimal threshold values with balanced exploitation and exploration operations.

Keywords: image segmentation; multilevel thresholding; hyper-heuristic; Exponential Monte Carlo with Counter; meta-heuristic algorithms; medical imaging

1. Introduction

Image segmentation breaks down a digital image into smaller parts or segments, typically referred to as "sets of pixels." The objective of image segmentation is to transform an image into a more manageable and meaningful form that can be more easily analysed. Through image segmentation, certain features of objects within the image are extracted, and the objects are then classified into specific groups or classes based on those extracted features. Image segmentation has been applied to a range of applications, including medical images, robotic vision, video surveillance, automatic traffic control system, and many more (Khalifa & Badr 2023; Minaee *et al.* 2022). Depending on the specific application, different methods such as clustering (Dubey *et al.* 2018), region growing (Szénási 2014), boundary-based (Shih 2017), hybrid techniques (El Aziz *et al.* 2016), and thresholding (Senthilkumaran & Vaithegi 2016) may be employed for image segmentation.

Thresholding-based segmentation is a widely used approach in image segmentation. It divides pixels into different classes, separating the image into background and object pixels. The goal is to identify areas of interest while disregarding irrelevant parts. Two main thresholding techniques used in related literature are bi-level and multilevel. Thresholding is bi-level when a single threshold value is used for separating objects in an image. Multiple thresholds are used in multilevel thresholding to divide images with varying shades into multiple regions. Multilevel thresholding accurately segments complex areas in satellite or medical images. However, computational cost increases with the number of thresholds. Standard approaches like Otsu and Kapur are effective but become computationally expensive with more thresholds. As a result, meta-heuristic algorithms are gaining attention as alternatives.

Heuristic and meta-heuristic algorithms have been used extensively to search for optimal multiple threshold values for effective image segmentation (Dhal *et al.* 2020). Although these algorithms are beneficial, they require fine-tuning of their algorithm-specific parameters to search for the optimum solution. Moreover, they not only heavily rely on domain-specific knowledge but also often become stuck in local optima. Therefore, this research proposes a new method employing Exponential Monte Carlo with Counter (EMCQ) hyper-heuristic as a high-level heuristic algorithm to segment large amounts of data related to the COVID-19 image dataset. The newly proposed strategy utilises multiple lightweight low-level heuristics, namely the crossover operator, peer learning operator, flower global pollination, and Jaya algorithm to search for the optimal threshold values for the segmentation of CT images of COVID-19 patients using the Otsu and Kapur methods as two different objective functions. The contributions of this work are summarised as follows:

- (1) The design and development of a new lightweight and parameter-free Exponential Monte Carlo with Counter (EMCQ) hyper-heuristic using four simple low-level heuristics, namely the crossover operator, Jaya algorithm, flower global pollination, and peer learning operator for multilevel thresholding segmentation of COVID-19 images.
- (2) The utilisation of the Otsu and Kapur methods as maximisation objective functions for finding optimal multilevel threshold values.
- (3) A comparison of the proposed strategy using various image processing metrics with other competent meta-heuristic algorithms on the COVID-19 image dataset and a hyper-heuristic algorithm on the classical image dataset.

The paper is laid out as follows: Section 2 discusses the literature review while Section 3 presents an overview of the multilevel thresholding problem. The proposed method, Exponential Monte Carlo with Counter (EMCQ) along with the descriptions of the low-level heuristics is discussed in Section 4. Section 5 elaborates on experimentations and image datasets, followed by Section 6, which presents the results and discusses them. Finally, Section 7 presents the conclusion and future direction.

2. Literature Review

Multilevel thresholding segmentation separates the pixels of an image into various groups or classes by defining the optimal threshold values. The maximisation functions, namely the Otsu method (Otsu 1979) and the Kapur method (Kapur *et al.* 1985), are used for the purpose of multilevel thresholding. In multi-level thresholding, these objective functions determine the optimal threshold values of an image (Satapathy *et al.* 2018; Kapur *et al.* 1985), which are often obtained by an employed optimisation method. The various optimisation methods used for image segmentation include heuristics (Suresh & Sakthi 2019), meta-heuristic methods (Abualigah *et al.* 2023), and hyper-heuristic methods (Abd Elaziz *et al.* 2020).

Heuristic methods can be defined as any practical problem-solving methods, such as rules of thumb or trial-and-error-based techniques, which are utilised for finding efficient and quick solutions when dealing with complex problems (Hertwig & Todd 2002). The main objective behind heuristics is to provide a good solution in a reasonable amount of time for the problem at hand. Heuristic techniques are applied in various fields, such as optimisation (Lee & El-Sharkawi 2002) and artificial intelligence (Glover & Greenberg 1989), and show efficiency compared to classical approaches that are very slow or fail in finding the exact solution. Commonly utilised heuristic methods for solving hard problems are Greedy search (Geng *et al.* 2011), Hill Climbing (HC) (Burke & Bykov 2017), Simulated Annealing (SA), and Tabu search (Youssef *et al.* 2001). Ohashi *et al.* (2003) utilised the HC-based approach for image segmentation, which generated coherent segments. Similarly, Salavati and Mozafari (2015) proposed a hybrid method that combined SA with Genetic Algorithm (GA) by utilising Kapur and Otsu entropies as objective functions for multi-level image segmentation. Nath *et al.* (2011) combined SA and HC for image segmentation, and as SA has the capability of accepting a worse solution for a certain probability, the method avoids getting trapped in local optimal and achieves a fast convergence to optimal threshold values.

Meta-heuristic algorithms are powerful optimisation methods and have been employed in numerous fields, such as engineering, finance, and computer science, to solve complex optimisation problems (Vasant 2012; Salih *et al.* 2023). These algorithms have mostly obtained inspiration from natural phenomena by imitating the working mechanisms of plants, hunting techniques of wild animals, human routines, the searching of food by insects and birds, and gaming rules (Khosrowshahli *et al.* 2025; Abdel-Basset *et al.* 2018). Meta-heuristic algorithms are specifically applied to solve hard optimisation problems where the search space is massive and challenging for finding better solutions (Desale *et al.* 2015). The working of these algorithms is based on randomisation, and the search space is explored randomly in an efficient manner in order to obtain the best solution. Meta-heuristic algorithms have become a prominent tool in tackling complex optimisation problems in various fields, and their impact on the development of innovative technologies and applications is expected to endure (Osaba *et al.* 2021; Nasser *et al.* 2024). Meta-heuristic algorithms are broadly classified based on their working into evolutionary algorithms, algorithms based on human behaviour, physical algorithms, and algorithms inspired by the intelligence of swarms (Abdel-Basset *et al.* 2018; Jia *et al.* 2024).

In the work of Vijn *et al.* (2023), a new hybrid meta-heuristic algorithm that integrates the Jellyfish Search Optimiser (JSO) with an effective improvement method was proposed, known as hybrid JSO (HJSO), for segmenting colour images of breast cancer. Extensive experimental results indicated the superior performance of the proposed algorithm. Similarly, Gharehchopogh and Ibrikci (2024) used an improved African Vultures Optimisation Algorithm (AVOA), which utilised the three binary thresholds (Kapur's entropy, Tsallis entropy, and Otsu's entropy) for multi-threshold image segmentation. The Quantum Rotation Gate (QRG) mechanism increased population diversity in optimisation stages and optimal local trap escapes to improve AVOA performance. The Association Strategy (AS) mechanism is used to support faster searches for optimal solutions. The experimental results from the proposed algorithm demonstrated a significant performance improvement over its counterparts. Additionally, Thapliyal and Kumar (2024) proposed an approach based on image partitioning (IP) with parameters that are optimised using Grey Wolf Optimiser (GWO). The proposed approach is named multilevel thresholding (IP-MH-MLT), which has been validated on a set of 15 benchmark images using various thresholds and compared quantitatively in terms of a number of various performance metrics. Mostafa *et al.* (2024) introduced the Improved Chameleon Swarm Algorithm (ICSA) for image segmentation and global optimisation tasks, aiming to improve the precision and efficiency of threshold selection in

medical image segmentation. Qiao *et al.* (2024) combined the Arithmetic Optimisation Algorithm (AOA) and Harris Hawks Optimiser (HHO) to improve AOA in the exploitation phase and achieve an optimal threshold vector for segmentation. Experiments show that the AOA-HHO method is better than the referenced algorithms. To address the challenges posed by inconsistent liver presence and unclear boundaries (Houssein *et al.* 2024), an enhanced Snake Optimisation (SO) algorithm was proposed that integrates with opposition-based learning (OBL), known as SO-OBL. The experimental results for multilevel image segmentation proved the effectiveness of the proposed SO-OBL.

Although heuristics and meta-heuristics are powerful and effective approaches, they still have some common drawbacks. Simple heuristic approaches are problem-specific designs that can get trapped in local optima and do not guarantee optimality. Similarly, meta-heuristic methods suffer from drawbacks such as proper tuning of specific parameters, no assurance of optimality of solutions, lack of reusability, and the balance between exploration and exploitation (Almufti *et al.* 2023). In meta-heuristic algorithms, there are several parameters, such as GA crossover probabilities and temperature schedules in SA, which play a vital role in finding the best solution. Without them, the algorithm returns unsatisfactory solutions.

In contrast to traditional heuristics, which are designed for problem instances, hyper-heuristics at a more comprehensive level aim to automate the selection or creation of heuristics that are applicable across a wide range of problem domains. A hyper-heuristic approach is a high-level, multipurpose search approach that is motivated by the automatic selection or generation of heuristics to solve a wide range of challenging optimisation problems (Drake *et al.* 2020; Din & Zamli 2018). The motivation behind hyper-heuristics is to increase the level of generality at which existing search approaches operate (Burke *et al.* 2003). According to this perception, hyper-heuristics can be assumed as a high-level general-purpose search approach that is given a specific problem instance and a set of low-level heuristics to be able to select and apply a suitable low-level heuristic at each decision point (Burke *et al.* 2003; Burke & Kendall 2014). The objective of hyper-heuristics is to utilise different solvers while solving a given problem instance.

Multi-level thresholding image segmentation is one of the most challenging and critical tasks, and procedural approaches, such as heuristics and meta-heuristics, require a high computation cost, as multi-level thresholding is a combinatorial optimisation problem. Therefore, hyper-heuristic-based approaches attract much attention from scholars in the present-day literature and have been utilised extensively in many problem domains, such as image processing, software engineering, networking, scheduling, etc. Abd Elaziz *et al.* (2020) proposed the only hyper-heuristic based strategy for image segmentation using multi-level thresholding. The suggested approach consists of two layers. In the first layer, GA is utilised to control the execution order of four meta-heuristic algorithms. In the second layer, a pool of four meta-heuristic methods is used, where each method is executed in a particular sequence to update the threshold population. Otsu's objective function is used to find the best threshold values. Although effective, this hyper-heuristic strategy employed full-strength meta-heuristic algorithms that might degrade its performance for challenging CT images. Moreover, this work extends the literature by introducing a second new simple yet efficient hyper-heuristic for both medical and classical images.

3. Multilevel Thresholding

In regard to image segmentation, multilevel thresholding has proven to be more effective and precise than bi-level thresholding (Dhal *et al.* 2020). The multilevel thresholding approach is very effective for dividing an image into various classes or groups and is efficient for complex

images where different parts of the image may have similar grey values. As the objective in bi-level thresholding is to find the optimum threshold value h , where intensities of pixels less than h represent background pixels, and pixel intensities greater than h represent the object's foreground. Eq. (1) provides a mathematical representation of this procedure:

$$\begin{aligned} C_0 &= \{ H(i, j) \in I \mid 0 \leq H(i, j) < h \} \\ C_1 &= \{ H(i, j) \in I \mid h \leq H(i, j) \leq L - 1 \} \end{aligned} \quad (1)$$

where L shows the maximum possible intensity value, I indicates the original input image, $H(i, j)$ is the pixel intensity value at coordinates i and j , and C_0 and C_1 are two classes into which $H(i, j)$ can be classified.

To expand bi-level thresholding into multilevel thresholding and to obtain multiple threshold values for segmenting the image into several parts, Eq. (2) is provided as follows:

$$\begin{aligned} C_0 &= \{ H(i, j) \in I \mid 0 \leq H(i, j) < h_1 \} \\ C_1 &= \{ H(i, j) \in I \mid h_1 \leq H(i, j) < h_2 \} \\ C_r &= \{ H(i, j) \in I \mid h_r \leq H(i, j) < h_{r+1} \} \\ C_n &= \{ H(i, j) \in I \mid h_n \leq H(i, j) \leq L - 1 \} \end{aligned} \quad (2)$$

where C_r represents r^{th} class, n is the total threshold values, and $\{h_1, h_2, \dots, h_n\}$ denotes a vector that contains a set of threshold values.

The challenge in multilevel image thresholding is to find the optimal threshold values h_r (for $r = 1, 2, \dots, n$) to precisely classify the image. To do this, methods such as Otsu and Kapur are employed. These are well-established methods that need to be maximised for finding optimum threshold values. Otsu's method provides a robust statistical criterion for segmentation by maximising inter-class variance. Kapur's method maximises entropy, which aligns with Shannon's entropy principles for optimal data representation. Both methods involve simple operations like variance or entropy computation, which makes them robust and accurate (Sathya *et al.* 2021). Moreover, these methods are non-parametric and do not require prior knowledge about the image. By defining clear objective functions (maximising variance or entropy), they can be optimised efficiently by the proposed EMCQ approach. These characteristics make the two methods reliable and widely accepted as objective functions for finding optimal multilevel threshold values using meta-heuristic algorithms (Hosny *et al.* 2022).

3.1. Otsu function

In Otsu's function (Otsu 1979), the aim is to find the threshold value that reduces the combined intra-class variance of the background and foreground. The multilevel Otsu method expands upon this strategy through its application to multi-level thresholding, adjusting the conventional method to suit more complicated circumstances.

Let $\mu_0, \mu_1, \dots, \mu_n$ indicate the mean of pixel values within segments 0, 1, ..., n respectively. Here, μ_T shows the global mean, and p_i represents the probability associated with the pixel intensity value being within the range i where $i \in (0, 1, \dots, 255)$ for various segmentation tasks. The between-class variance functions across n different classes can be written as in Eq. (3).

$$\sigma_0^2 = \omega_0(\mu_0 - \mu_T)^2, \sigma_1^2 = \omega_1(\mu_1 - \mu_T)^2, \dots, \sigma_n^2 = \omega_n(\mu_n - \mu_T)^2 \quad (3)$$

where ω_r is the cumulative probability of C_r and is computed using Eq. (4).

$$\omega_r = \sum_{i \in C_r} p_i \quad (4)$$

where p_i is the probability of the i^{th} intensity level and is denoted by Eq. (5).

$$p_i = \frac{frq_i}{\sum_{i=0}^{L-1} frq_i}, \quad (5)$$

where frq_i is the i^{th} intensity level frequency. The mean values for each group of pixels are expressed in Eq. (6).

$$\mu_0 = \frac{\sum_{i=0}^{h_1-1} i \cdot p_i}{\omega_0}, \quad \mu_1 = \frac{\sum_{i=h_1}^{h_2-1} i \cdot p_i}{\omega_1}, \quad \mu_r = \frac{\sum_{i=h_r}^{h_{r+1}-1} i \cdot p_i}{\omega_r}, \quad \mu_n = \frac{\sum_{i=h_n}^{L-1} i \cdot p_i}{\omega_n} \quad (6)$$

This segmentation method focuses on maximising $f(h)$, which is the sum of between-class variance functions so as to determine the optimal threshold values, and can be defined as shown in Eq. (7).

$$h^* = \arg \max(f(h)), \text{ where } f(h) = \sum_{i=0}^n (\sigma_i^2) \quad (7)$$

This segmentation method is applicable for processing both greyscale and colour image data sets.

3.2. Kapur's function

Similarly, Kapur *et al.* (1985) presented a non-parametric method for finding the best threshold result that is based on the entropy and probability distribution of the images. The objective of the method is to discover the appropriate threshold that exceeds the entire entropy. This optimal threshold value exceeds the overall entropy of the image. As a result, the entropy of an image measures its compactness and its capability to distinguish between different regions.

Kapur's approach can also be applied to multi-level thresholding, which could be stated mathematically as in Eq. (8).

$$\begin{aligned} E_0 &= - \sum_{i=0}^{h_1-1} \left(\frac{p_i}{\omega_0} \right) \log_2 \left(\frac{p_i}{\omega_0} \right); & E_1 &= - \sum_{i=h_1+1}^{h_2-1} \left(\frac{p_i}{\omega_1} \right) \log_2 \left(\frac{p_i}{\omega_1} \right); \\ E_r &= - \sum_{i=h_r}^{h_{r+1}-1} \left(\frac{p_i}{\omega_r} \right) \log_2 \left(\frac{p_i}{\omega_r} \right); & E_n &= - \sum_{i=h_n}^{L-1} \left(\frac{p_i}{\omega_n} \right) \log_2 \left(\frac{p_i}{\omega_n} \right); \end{aligned} \quad (8)$$

where

$$\begin{aligned} \omega_0 &= \sum_{i=0}^{h_1-1} p_i; & \omega_1 &= \sum_{i=h_1}^{h_2-1} p_i; \\ \omega_r &= \sum_{i=h_r}^{h_{r+1}-1} p_i; & \omega_n &= \sum_{i=h_n}^{L-1} p_i; \end{aligned}$$

Thus, $E_1, E_2, \dots, E_n$ represent the entropy values for classes $0, 1, \dots, n$ respectively, and p_i

represents the probability of a pixel value falling within the range $[0, 255]$ in a greyscale image. Furthermore, this method validates the usefulness of colour image segmentation by independently utilising the R, G, and B channels.

4. Proposed Work

In this work, EMCQ is used as a hyper-heuristic based strategy for the multi-level thresholding image segmentation of the COVID-19 image data set that comprises two levels, i.e., high-level and low-level. At the high-level, EMCQ is utilised as a manager that controls a set of four local low-level heuristic operators. These four low-level simple heuristic operators are utilised to search the problem space effectively. The selection of the four low-level heuristics—Jaya algorithm, peer-learning operator (learner phase of TLBO), crossover operator (Genetic Algorithm), and global pollination operator (Flower Pollination Algorithm)—in the design of the proposed work is to achieve a balance between exploration and exploitation. The Jaya algorithm and the TLBO peer-learning operator focus on solution refinement and convergence, while the crossover operator and global pollination operator introduce diversity and enable the algorithm to escape local optima. Collectively, these low-level operators attempt to achieve robust performance by leveraging their complementary strengths. The working mechanism of EMCQ is similar to most of the other local search methods. Like other heuristic approaches, EMCQ also generates an initial solution and then subsequently improves it. Acceptance of worse solutions in an intelligent manner separates EMCQ from its counterparts. Similarly, maintaining a proper balance between diversification and intensification is another characteristic of EMCQ. The general working of the EMCQ strategy is depicted in Figure 1. This novel strategy is also known as a selection hyper-heuristic based method, which improves the searching results as compared to using these search operators individually. Moreover, for the effective segmentation of the COVID-19 image dataset, two different entropies, Otsu and Kapur, are adopted as fitness functions.

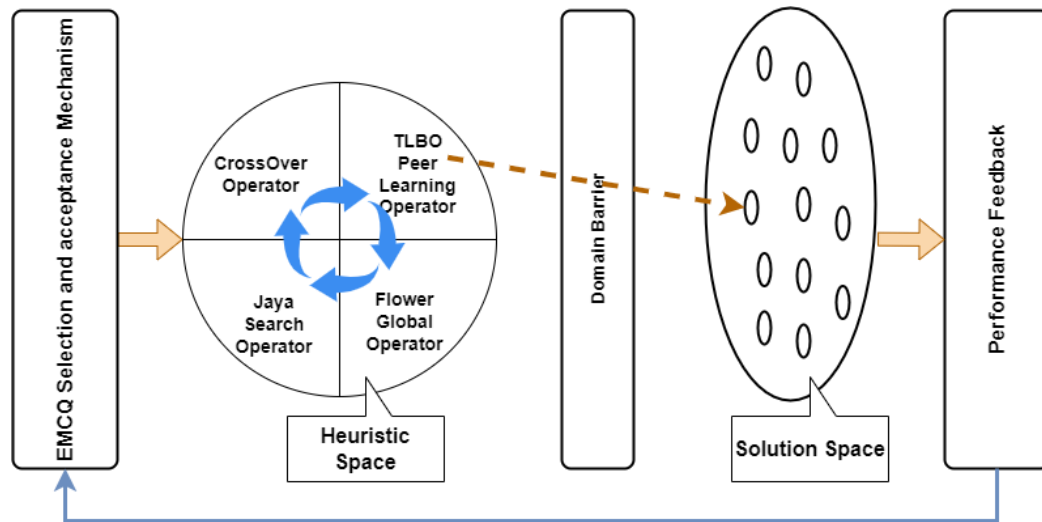


Figure 1: EMCQ hyper-heuristic for multilevel image thresholding

As shown in Figure 1, EMCQ operates on four low-level heuristics as its search space. EMCQ always favours a high-performing low-level heuristic. Apart from the best low-level heuristic,

EMCQ also gives a chance to low-performing heuristics during the search for optimal threshold values. The probability of accepting a poorer solution in the EMCQ method is represented mathematically, as shown in Eq. (9).

$$\varphi = e^{-\partial t/c} \quad (9)$$

where $\partial = f(X_{current}) - f(X_l)$ shows the difference between the fitness values of the current active low-level heuristic ($f(X_{current})$) and last active low-level heuristic ($f(X_l)$), t works as an iteration counter variable, and c is a control parameter that shows consecutive non-improving iterations.

The parameter c also manages the acceptance of an inferior heuristic. This parameter results in the proper adjustment of intensification and diversification of the search space, which leads to giving an opportunity to all four low-level heuristics for effective optimisation results. Initially c is assigned 1 (i.e., $c = 1$), and then its value is incremented by 1 (i.e., $c = c + 1$) for a specified number of consecutive non-improving iterations. When an inferior heuristic is accepted, c is reset to 1. In EMCQ, the accepting probability of heuristic operators generating poor threshold values decreases as the iteration number t increases. By avoiding inferior heuristics, EMCQ ensures faster convergence.

4.1. EMCQ strategy for segmentation of COVID-19 image data set

This section presents the implementation of the proposed EMCQ method, which searches for the optimum threshold to obtain quality segmentation results. Figure 2 outlines the pseudocode of the proposed method for multi-level thresholding in the COVID-19 image data set.

The following are details of the pseudocode. At the beginning, the algorithm performs the initialisation of the threshold dimension, where the dimension is set to 5, 7, 9, etc., and the maximum iteration (i.e., α_{max}) and population size are as given in line 1. In line 2, a random population X of 20 solutions is generated, which consists of an array of solutions. In line 3, EMCQ chooses a random low-level heuristic operator h_i from the set of four heuristics and applies it to generate an initial solution x_0 . After this, the variable x_{best} is set to the initial x_0 , and set h_i to the initial search operator h_0 . The main loop begins at line 5 and continues to repeat until the termination criteria are met. In line 6, the inner loop counter t is assigned the initial value of 1.

The inner while loop begins at line 7, with α_{max} representing the maximum number of iterations. Then, in lines 8-9, the selected search operator h_i is selected to generate the best solution from the population X , and the fitness of the chosen solution is calculated. The difference in the fitness between the new solution (i.e., x_i) produced and the previous best solution x_{best} is computed, as shown in the pseudocode. In lines 10-13, the algorithm retains the current heuristic based on variable ∂ . If this difference variable is greater than 0, it means the fitness improves, so the EMCQ gives another chance to the selected search operator h_i . Also, the c variable in line 12 is assigned value 1 as the algorithm is progressing by creating a better new solution.

The value of the variable φ for probability of accepting the worse solution is calculated in line 15 using the given equation. Similarly, in lines 16-19, if the fitness does not improve, the poor move might also be accepted based on the probability density variable value φ . If the poor solution is accepted, EMCQ offers another chance to the ongoing search operator, and the variable c is assigned the value 1. Otherwise, in lines 20-23, the search operator h_i is altered while the variable c is incremented by 1, as detailed in lines 21-23. Subsequently, lines 25-27 are responsible for updating the values of x_{best} and t in preparation for the upcoming iteration.

In case of nonfulfillment of the termination criteria, the aforementioned process is reiterated. Finally, EMCQ for multilevel thresholding returns the best threshold values and other computed performance metrics for image processing in line 28.

Algorithm 1: EMCQ pseudocode for multilevel thresholding of the COVID-19 image data set

Input: Greyscale or colour image (I), Threshold Dimension, No's of Iterations (α_{max}), set of heuristics $H = \{h_0, h_1, h_2, \dots, h_n\}$
Output: Best thresholds and other image processing related performance metrics

- 1 *Initialization: Initialize Maximum iterations (α_{max}), and population size (X), threshold dimension*
- 2 *Generating random population of solutions $X = \{x_0, x_1, x_2, \dots, x_n\}$*
- 3 *Choose random heuristic h_i from the set of heuristics $H = \{h_0, h_1, h_2, h_3\}$. For producing initial solution x_i by applying h_i*
- 4 *Initialize $x_{best} = x_0$, $h_i = h_0$*
- 5 **While** (termination criteria not satisfied)
- 6 Initialize $t = 1$
- 7 **while** ($t < \alpha_{max}$)
- 8 *apply h_j to produce the best solution x_j from the population (X)*
- 9 Calculate $\partial = f(x_i) - f(x_{best})$
- 10 **if** ($\partial > 0$)
- 11 // improving fitness
- 12 Retain h_j
- 13 $c = 1$
- 14 **End**
- 15 **Else**
- 16 Calculating probability for acceptance of worse solution: $\varphi = e^{-\partial t / c}$
- 17 **if** ($\text{random}(0,1) < \varphi$)
- 18 Retain h_i
- 19 $c = 1$
- 20 **End**
- 21 **else**
- 22 Randomly select a new search operator h_j , such that $j \neq i$
- 23 $c++$
- 24 **End**
- 25 **End**
- 26 $x_{best} = x_i$
- 27 $t++$
- 28 **End**
- 29 **Return** Best thresholds, compute and display other performance metrics

Figure 2: Pseudocode of the EMCQ method for segmentation of an image using multilevel thresholding

Four low-level heuristics for searching optimal threshold values are used under the control of the EMCQ as high-level heuristics, which are briefly discussed in the subsections below. These heuristics are simple operators with less extra overhead for finding optimal solutions. To balance the exploration and exploitation processes, this study adopted two global search operators and two local search operators. The Jaya and FPA global pollination operators are

responsible for global search, whereas the TLBO peer learning and the GA crossover operators are used for local search. The following subsections briefly explain these low-level search operators.

4.1.1 GA crossover operator

Crossover is the main search operator in the Genetic Algorithm (GA). In each generation, GA applies this search operator to the two best selected parents for the exchange of genetic material. At the start of the algorithm, x_{best} is set to a random solution x_0 . Then, the search operator enters a loop where a random new solution x_i is chosen from the existing solution during each iteration. This solution is updated using a crossover operation, which is performed between a randomly selected solution and an existing solution in the population over a randomised length, using Eq. (10).

$$\alpha = rand(0, length\ of\ x_j) \quad (10)$$

where α represents the crossover operator cut-off length and x_j represents a randomly selected solution. A new solution is then generated using Eq. (11).

$$x_{new} = exchange\ from\ position\ 0\ till\ \alpha\ from\ x_j\ to\ x_{current} \quad (11)$$

After modification of the new solution, if the newly modified solution has a better fitness value, i.e., x_{new} , that is better than the current solution $x_{current}$, then the current solution is updated. Moreover, the fitness of the updated solution is compared with the best solution, i.e., x_{best} , and if it is better, x_{best} is updated as well. This process continues until the termination condition is reached.

4.1.2 TLBO peer learning operator

The Teaching Learning-based Optimisation Algorithms (TLBO) has a peer learning operator that moves a poor learner towards its better peer learner. This local search operator of TLBO performs this step on the entire population of learners represented as $X = \{x_0, x_1, x_2, \dots, x_n\}$. Also, the algorithm initialises x_{best} with a random solution x_0 . After this step, during each iteration of the loop, a learner represented by x_i is randomly paired with another peer learner (x_j), in such a way that $i \neq j$ to improve his knowledge. Now if x_i 's fitness is superior then x_j 's fitness, then the new solution is updated using Eq. (12), and if x_i 's fitness is inferior, then the new solution is updated using Eq. (13).

$$x_{new} = x_i + \rho(x_i - x_j) \quad (12)$$

$$x_{new} = x_i + \rho(x_j - x_i) \quad (13)$$

where ρ is called a random vector and is taking values from the interval $[0,1]$. If there is no improvement in x_i fitness, it is not updated. The algorithm then checks the fitness of x_i against x_{best} , and if x_i has a better fitness, x_{best} is updated.

4.1.3 FPA global pollination operator

The FPA global pollination search operator is based on the Flower Pollination Algorithm (FPA), which is designed to explore all regions of the search space, making it a global search operator. The algorithm starts with a set of randomly generated threshold values as a population

of solutions. Initially a random solution x_0 is set to x_{best} . The operator then iteratively updates each threshold of the vector x_i using the Lévy Flight motion transformation, as shown in Eq. (14),

$$x_{new} = x_i + \rho \times L \times (x_{best} - x_i) \quad (14)$$

where x_{new} represents the updated solution, x_i represents the current solution, ρ represents a random vector, and L represents the step vector that obeys levy flight distribution. If the updated x_{new} solution has a better fitness value, it replaces the current solution, i.e., x_i , and x_{best} is also updated if the updated solution has a better fitness than x_{best} .

4.1.4 Jaya algorithm

The Jaya algorithm is a simple meta-heuristic algorithm. This search operator keeps track of both x_{best} and x_{poor} , unlike other search operators that only keep track of x_{best} . The transformation equation used by the Jaya search operator is shown in Eq. (15).

$$x_{new} = x_i + rand(0,1) \times (x_{best} - x_i) - rand(0,1) \times (x_{poor} - x_i) \quad (15)$$

The Jaya algorithm iteratively applies Eq. (15) to update all solutions. If the updated x_{new} has a better fitness value, it replaces the current solution x_i , and x_{best} is also updated if its fitness value is better than x_i . If the updated x_{new} has a poorer fitness value, this new solution is ignored and the search to find a new improved solution is repeated until completion of the search process.

5. Experimentation and Image Datasets

The results obtained from different runs are presented and evaluated using the Peak Signal to Noise Ratio (PSNR), Mean Squared Error (MSE), and Feature Similarity Index (FSIM) as performance metrics. The results of the proposed method are then compared with the full versions of the algorithms, namely Jaya algorithm, Teaching Learning-based Optimisation Algorithm (TLBO), Genetic Algorithm (GA), and Flower Pollination Algorithm (FPA). Moreover, boxplots are used for graphical depiction of the results.

5.1. Experimental setup

The hardware system consists of an Intel(R) Core (TM) i5 computer with a 2.5 GHz processor and 8.0 GB of main memory. The system is installed with the operating system Windows 11. Moreover, the approach under consideration is implemented utilising MATLAB R2018a, which has robust, refined, and useful properties for image handling.

The results are obtained and evaluated using performance matrices such as Mean Square Error (PSNR), Mean Square Error (MSE), and Feature Similarity Index (FSIM), which are computed using the best thresholds produced by the optimisation approaches. Threshold dimensions are assigned values of 5, 7, 9, and 11 in order to manifest the effectiveness of the suggested method. The population size and number of iterations are set to 20 and 100, respectively, for all included algorithms. Furthermore, standard values are used as a baseline for the algorithm specific parameters of GA and FPA. Subsequently, these parameters are fine-tuned by conducting a series of controlled experiments on benchmark image datasets used for multilevel thresholding. The tuning process involves varying one parameter at a time while keeping others constant (a one-factor-at-a-time approach) to observe its impact on performance

metrics such as segmentation accuracy and convergence speed. Parameters are adjusted within the ranges recommended in the literature (Chen *et al.* 2023; Jayaraman *et al.* 2022), guided by the problem-specific requirements of multilevel image segmentation. The final parameter values (crossover rate = 0.8, mutation rate = 0.2 for GA, and switch probability = 0.8 for FPA) are selected based on their ability to consistently produce high-quality segmentation results across different test cases.

5.2. Performance evaluation metrics

To measure the performance of the proposed strategy quantitatively against other search methods on segmenting the COVID-19 image data set, performance metrics such as Mean Square Error (MSE) (Horé and Ziou 2010), Peak Signal to Noise Ratio (PSNR) (Horé & Ziou 2010), and Feature Similarity Index (FSIM) (Zhang *et al.* 2011) are utilised as performance metric values.

MSE is the assessment of the true pixel values in the original image against those in the segmented image (Horé & Ziou 2010). In MSE, the difference between the original and degraded images is calculated by averaging the square errors. Similarly, PSNR is the procedure of dividing the maximum possible value by the power of the deteriorating noise in a signal, which affects representation quality (Horé & Ziou 2010). Also, FSIM is a metric by which computational models calculate image quality in alignment with subjective assessments (Zhang *et al.* 2011). Using a couple of alike images is imperative for conducting measurements. The approach mainly evaluates an image based on its low-level characteristics, with a specific emphasis on phase congruency as the prime feature within FSIM. On the classical image dataset, the Structured Similarity Index Measure (SSIM) is used for comparison of the proposed approach with the existing hyper-heuristic approach HHB. SSIM returns a similarity of two images in terms of brightness, contrast, and structural features. Mathematically, these metrics are given in Table 1.

Table 1: Mathematical forms of the metrics, i.e., MSE, PSNR, and FSIM

S#	Metric	Formulation
1	MSE	$MSE = \frac{1}{mn} \sum_{k=1}^m \sum_{l=1}^n [h_{k,l} - h'_{k,l}]$
2	PSNR	$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right)$
3	FSIM	$FSIM = \frac{\sum_{y \in Y} S_L(Y) PC_m(y)}{\sum_{y \in Y} PC_m(y)}$
4	SSIM	$SSIM = \frac{(2\mu_h \mu_{h'} + C_1)(2\sigma_{hh'} + C_2)}{(\mu_h^2 + \mu_{h'}^2 + C_1)(\sigma_h^2 + \sigma_{h'}^2 + C_2)}$

5.3. Image dataset

The proposed EMCQ is evaluated on 25 images of the COVID-19 image dataset to corroborate its segmentation results. The dataset is obtained from the Kaggle online repository (Ryalat *et al.* 2023). Three chest images from the COVID-19 image dataset, as shown in Figure 3, are selected to carry out the evaluation of the proposed segmentation method. One image named Img4 for reference is a 2-D image used for the analysis of the methods' performance via boxplot

and convergence curves. The proposed optimisation algorithm and other implemented algorithms have undergone 20 independent runs to determine the optimal threshold values.

Next, six images from the Berkeley image dataset repository are taken to validate the performance of the proposed hyper-heuristic strategy against the existing hyper-heuristic approach (Abd Elaziz *et al.* 2020). Figure 4 shows these six benchmark images that are used to check the performance of the EMCQ hyper-heuristic against the only hyper-heuristic strategy known as Hyper-heuristic Best (HHB) on the best obtained multilevel threshold values. To evaluate the best performance of the suggested approach, all the evaluation criteria and the results of other algorithms are adopted from (Abd Elaziz *et al.* 2020).

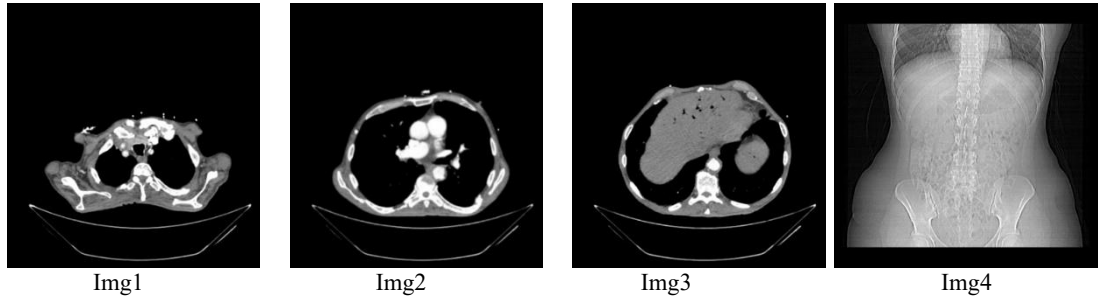


Figure 3: Selected images from COVID-19 dataset and a 2-D image for performance evaluation



Figure 4: Images for comparison of EMCQ with existing hyper-heuristic HHB

6. Results and Discussion

In this section, the various experiments implemented regarding the accuracy and consistency of the proposed hyper-heuristic method are presented. Firstly, the suggested method was applied and tested on the COVID-19 image dataset of 25 images. Also, one greyscale image was tested on the proposed approach and its block plot is shown. The results produced by the hyper-heuristic method based on EMCQ at the threshold values of 5, 7, 9, and 11 were checked on the Otsu and Kapur fitness functions. The segmented images obtained via all competing algorithms and the proposed algorithm are shown in Figures 5 to 8 for three images selected from the COVID-19 image dataset. From close observation of these images, it can be concluded that the EMCQ generated better visual segmentation results with better edge preservation of the COVID-19 images. The effectiveness of the proposed approach was evaluated and compared with the existing hyper-heuristic approach by Abd Elaziz *et al.* (2020) using the six benchmark images (i.e., Img5 represents Cameraman, Img6 represents Airplane Jet, Img7 represents Peppers, Img8 represents Lake, Img9 represents couple, and Img10 represents Bridge) that were taken from the Berkeley image dataset using a higher dimension of threshold values such as 10, 15, 20, and 30. By using the performance metrics, such as PSNR, SSIM, and the objective functions of Otsu and Kapur, the proposed EMCQ hyper-heuristic method outperforms existing hyper-heuristics as well as other meta-heuristic approaches included for comparison in this work.

Based on the thresholds, Tables 2 to 7 present different performance evaluation measurements such as PSNR, MSE, and FSIM for both objective functions.

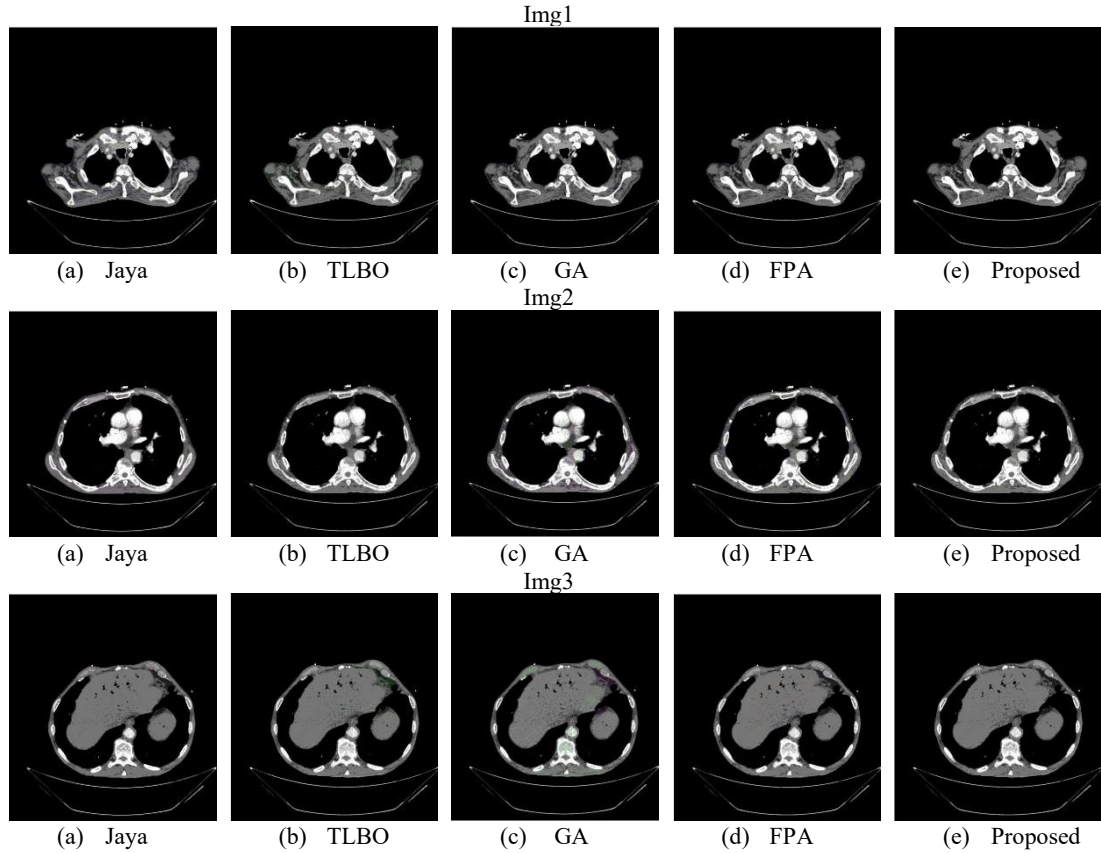


Figure 5: Comparison based on segmenting Img1, Img2, and Img3 using 5 threshold values

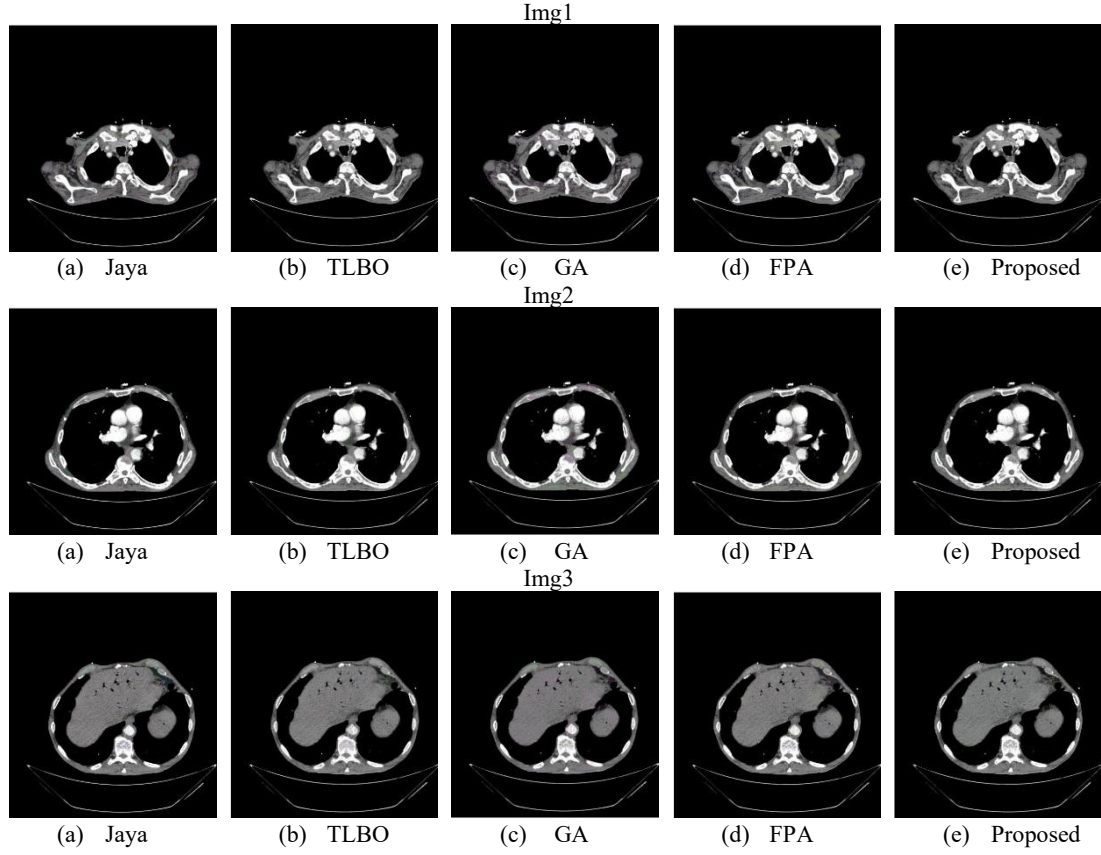


Figure 6: Comparison based on segmenting Img1, Img2, and Img3 using 7 threshold values

Table 2 shows that the PSNR values obtained by EMCQ for the three COVID-19 images using Otsu's objective function on all thresholds are superior to other algorithms. EMCQ performed even better than its competitors for higher threshold values. FPA offered the second-best mean segmentation results in two instances. Moreover, the proposed method shows superior performance in the case of generating maximum (Max) PSNR values for most of the problem instances. FPA again generated two max PSNR values, whereas TLBO only once obtained a max PSNR result. Table 3 shows the MSE values of the proposed and other algorithms, where the proposed approach shows superiority in MSE compared to other methods. As far as the minimum MSE values are concerned, the proposed method outperformed its counterparts. Here, again, only FPA generated two minimum MSE values and remained the runner-up. Lastly, the FSIM values presented in Table 4 determine that the proposed method is superior to all other competing methods. FPA also generated competing results, whereas TLBO only generated one best mean result.

Similarly, considering the PSNR values in Table 5 for the thresholds generated by EMCQ using Kapur's entropy for the COVID-19 images, it is evident that the proposed approach results are superior to all other algorithms. For mean best results, the hyper-heuristic method generated 9 out of 12 best results, which show the consistency of the method. GA remained second in generating the two best mean results. In the case of overall maximum results, GA remained on top by obtaining 7 out of 12 best results as compared to 5 best maximum PSNR values by the proposed method. When the MSE values obtained by EMCQ are considered in Table 6, the proposed method outperformed the remaining methods by generating 11 in total

12 best mean results. GA only once produced better results. In the case of minimum MSE values, GA again came first by obtaining 7 out of 12 best results. Lastly, Table 7 shows the FSIM values obtained over different thresholds by the proposed EMCQ. As far as mean best results are concerned, EMCQ obtained 7 best mean FSIM values, whereas GA obtained 5 best mean values and remained second. GA performance is on top when the maximum best results are observed in Table 7.

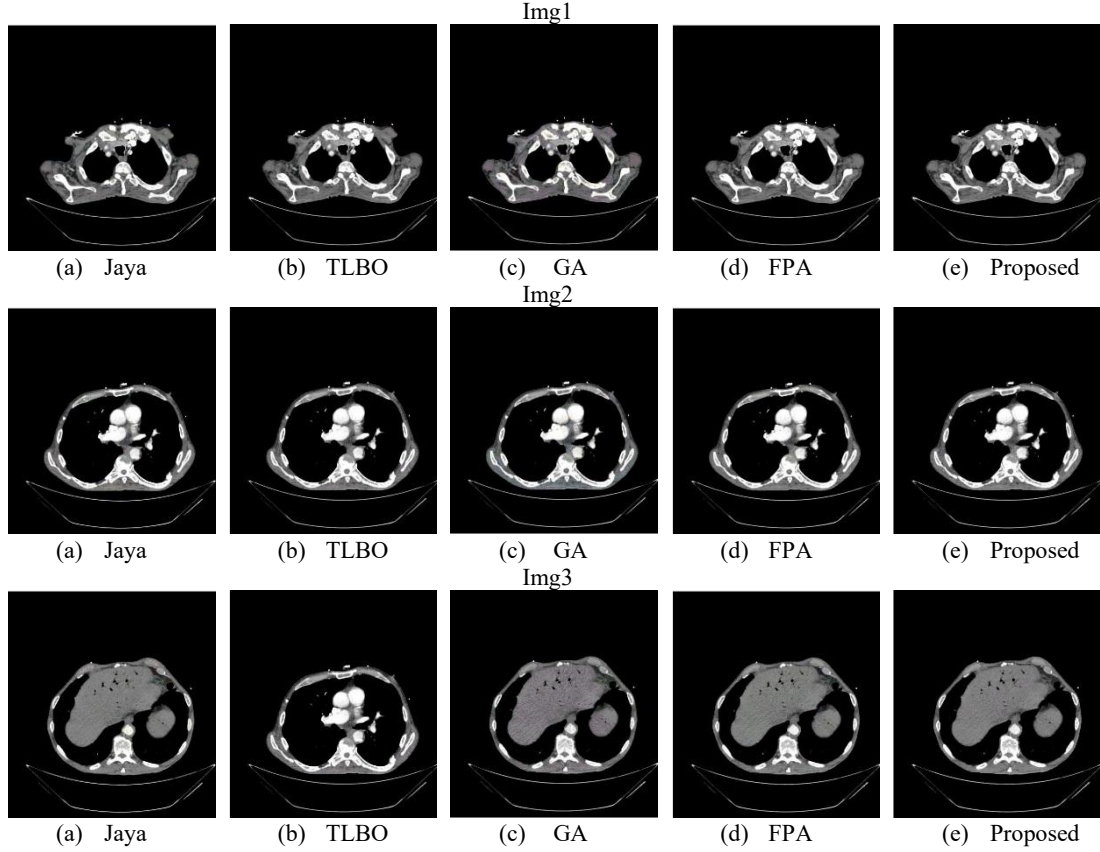


Figure 7: Comparison based on segmenting Img1, Img2, and Img3 using 9 threshold values

The elapsed time of the proposed method is shown in Table 8, where, due to the extra overhead of selection and acceptance mechanism, the EMCQ-based approach remains second-last after TLBO. The Jaya algorithm appeared as the winner owing to its simple operations for generating new solutions when elapsed time was considered for comparison. It is essential to note that the time complexity of the proposed EMCQ hyper-heuristic, FPA, GA, and Jaya algorithm is $O(T \cdot N \cdot L^m \cdot m)$, where T is the number of iterations, N is the population size, L is the number of gray levels and m is the number of thresholds. As TLBO evaluates the objective functions (Otsu and Kapur methods evaluation complexity is $O(L^m \cdot m)$) twice for each solution in every iteration, its complexity is $O(2 \cdot T \cdot N \cdot L^m \cdot m)$. While EMCQ might be slower in terms of execution time owing to random selection and evaluation process, its adaptive mechanism of overlooking multiple heuristics often provides optimal solutions. The trade-off between execution time and solution quality is a key strength of the EMCQ hyper-heuristic, especially in medical image segmentation, where precision is critical.

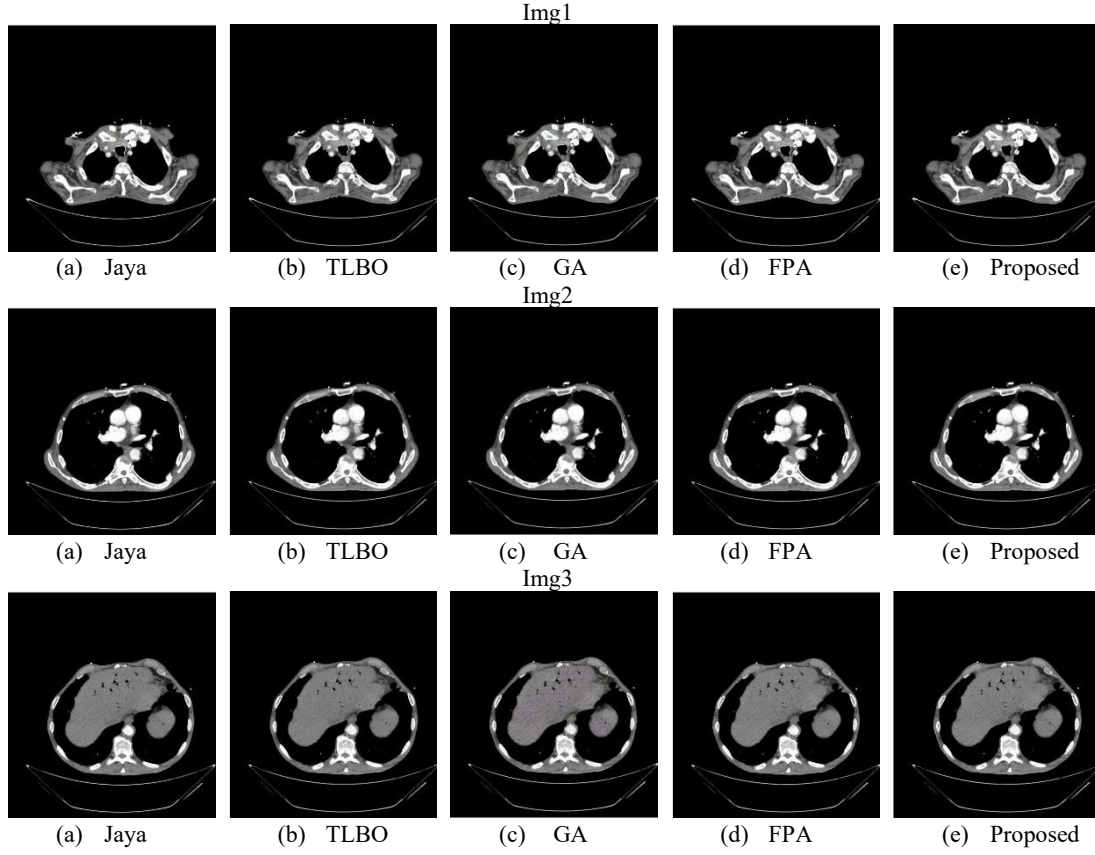


Figure 8: Comparison based on segmenting Img1, Img2, and Img3 using 11 threshold values

Furthermore, Tables 9 and 10 demonstrate the superior performance of the proposed EMCQ approach as compared to the hyper-heuristic approach and other meta-heuristic approaches that have been implemented and tested (Abd Elaziz *et al.* 2020), such as SCA (Mirjalili 2016), ABC (Karaboga & Ozturk 2011), SSO (Ouafeel & Taleb-Ahmed 2016), hybrid FA and SSO (El Aziz *et al.* 2016), hybrid FA and ABC, and hybrid ABC and SCA. All results reported in Tables 12 and 13, except the EMCQ, are adopted from (Abd Elaziz *et al.* 2020).

In Table 9, the PSNR best results obtained by EMCQ and other competing algorithms are presented. Overall, EMCQ showed comparable performance against the HHB. Both have generated almost the same number of best results on all six images at higher threshold values. Using threshold values of 15, the proposed approach finds the optimum result using Img5, Img7, and Img9, while in the case of Img6 and Img8, HHB achieved the best results. Similarly, the performance of both EMCQ and HHB is equivalent on Img10. All the other meta-heuristic approaches are runners. At a higher threshold value of 30 on all six images, EMCQ obtained equally good PSNR values compared to HHB. Based on the experimental evidence from Table 9, EMCQ is second in rank and is the best compared to all the other meta-heuristic algorithms.

Table 10 shows the results for SSIM of the proposed method along with the other hyper-heuristic and meta-heuristic approaches. Using a threshold level of 10, only two images, i.e., Img5 and Img7, Hyper-heuristic Union Best (HHUB), another version of HHB, outperformed all others, including HHB and EMCQ, which remained second. For the same threshold level, EMCQ obtained the best SSIM values compared to others on Img9 and Img10, whereas HHB

and HHUB are the second and third, respectively. Similarly, using the threshold level of 15, EMCQ and HHB are the winners by generating the two best SSIM results. HHB achieved first position by providing the best results for 20 threshold values on four out of six images. EMCQ won the competition in the case of 30 multilevel threshold values, as it has generated the best results in terms of SSIM values on Img5, Img9, and Img10. The runner-up in this case is HBB, as it has provided the two best SSIM values on Img6 and Img7. Moreover, as compared to EMCQ and HHB, the other method, such as HHUB, is third in rank. The SSO method is fourth, and FASSO is fifth, while SCA is ranked last with respect to the overall thresholds of the images.

The proposed method generated optimum threshold values and showed superiority in different evaluation metrics, which proves its effectiveness for the segmentation of images using multilevel thresholding. The boxplot of the results, such as peak signal to noise ratio (PSNR), feature similarity index (FSIM), and mean square error (MSE), obtained over different runs on different images using the threshold values of 5, 7, 9, and 11 is shown in Figure 9. This further illustrates the experimental results of the proposed method and other methods. When compared to the other methods, the boxplots for all threshold dimensions present better performance of the proposed methods compared to their counterparts. As evident from all boxplots, the proposed method spread is smaller with minimum mean values, whereas the variations of the threshold values generated by all competing meta-heuristic methods are larger. This indicates the consistency of the proposed method for efficiently addressing the problem of multilevel thresholding.

Figure 10 shows the best objective function values generated by each algorithm over 100 iterations on Img4 using the Otsu entropy. These plots show that the proposed method is efficient in obtaining better results in fewer iterations for the different threshold (i.e., 5, 7, 9, and 11) values. Moreover, the proposed method is able to obtain maximum fitness function values in all four cases compared to other methods. In essence, these illustrations suggest that the proposed method not only converges faster but can also reach the global optimum.

Figure 11 shows the segmented image results of the proposed method and other algorithms using the threshold value of 5. Similarly, Figures 12 and 13 show the results of segmented images obtained by the proposed method and other algorithms with 9 and 11 as the threshold values. The proposed method shows superior segmentation results compared to other algorithms, such as Jaya, TLBO, GA, and FPA algorithms.

6.1. Statistical analysis

The obtained PSNR results presented in Tables 2, 5, and 9 are subjected to statistical analysis. The comparisons are based on the $1 \times N$ pair with two confidence levels, i.e., 95% ($\alpha = 0.05$) and 90% ($\alpha = 0.1$). The Wilcoxon rank-sum test is adopted to showcase whether the proposed method is statistically significant than other competing methods. This nonparametric test is considered owing to the non-normal distribution of the obtained results.

The null hypothesis (H_0) suggests a significant difference between the results obtained by the proposed method and other methods, whereas the alternative hypothesis (H_1) suggests no statistical significance between the performance in terms of the obtained PSNR values. Tables 11 to 13 show complete statistical analyses. These results show that the proposed EMCQ hyper-heuristic is statistically better at both 95% and 90% confidence levels than the competing meta-heuristic algorithms. With regard to HHB, no significant improvement was achieved by the proposed method. Nevertheless, HHB was only tested on the classical image dataset, whereas EMCQ was mainly proposed for medical image segmentation.

Table 2: Comparing PSNR values produced by EMCQ with other algorithms using Otsu's inter-class variance

Otsu's inter-class variance: PSNR Comparison on CT Images																
Images	No	Jaya			GA			TLBO			FPA			Proposed		
		Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean
Img1	5	28.0132	28.3846	28.7687	25.4775	28.4628	27.3732	20.2897	29.2252	27.7729	27.8262	29.3406	28.9740	28.9075	29.2242	29.1051
	7	30.0665	31.6600	30.9455	26.5142	30.9911	29.3263	26.2311	31.8449	30.0936	30.5469	31.7676	31.4219	30.6052	31.9122	31.6957
	9	32.2560	33.2384	32.7350	29.7864	32.3369	31.3549	29.7535	33.9111	32.1925	32.5391	33.8221	33.2152	32.8996	33.7470	33.6024
Img2	11	33.3052	34.8467	34.0247	31.2384	34.2839	32.9378	32.6934	35.0085	33.9010	34.2263	34.9854	34.4948	34.2805	35.6121	35.0769
	5	28.4504	29.4827	28.9549	26.1809	28.7813	27.5886	22.6923	29.4435	27.8976	28.7924	29.5707	29.1841	29.1973	29.6468	26.4726
	7	30.8841	31.9574	31.3825	28.9432	31.2910	30.0484	23.9260	31.8998	30.2514	30.6405	32.0215	31.7634	31.5269	32.2200	32.0245
Img3	9	32.5147	33.9519	33.1051	31.0044	32.5118	31.8186	26.7497	33.6983	31.7696	33.1230	33.9226	33.5561	33.2004	34.3006	34.0209
	11	33.8351	35.0932	34.5122	31.9585	34.8069	33.5222	28.7402	35.5516	33.9728	34.1540	35.4277	35.0097	34.7783	35.9525	35.6358
	5	27.7928	28.8181	28.4437	24.3488	28.4611	26.9722	23.5065	28.9116	27.7116	28.6225	28.9524	28.7986	28.6294	28.8984	28.7839
	7	29.6423	31.1831	30.5732	26.5380	30.8637	29.1738	22.4284	31.6571	29.7739	30.5915	31.6041	31.1473	30.5362	31.7282	31.4798
	9	31.5785	33.3856	32.4047	26.9356	31.7989	30.7498	29.1607	33.3544	31.6161	31.7678	33.4468	32.7406	32.9300	33.8299	33.4544
	11	33.1355	34.6615	33.7222	30.7429	33.6512	32.4666	27.1184	35.0298	32.6541	33.5066	34.8031	34.1815	34.0052	35.4388	34.9681

Table 3: Comparing MSE values produced by EMCQ with other algorithms using Otsu's inter-class variance

Otsu's inter-class variance: MSE Comparison on CT Images																
Images	No.	Jaya			GA			TLBO			FPA			Proposed		
		Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean
Img1	5	74.9239	102.7452	86.6305	92.6403	184.2163	121.3586	77.7255	608.2896	126.6873	75.6864	107.2667	82.5987	77.7424	83.6242	79.9167
	7	44.3694	64.0370	52.5438	51.7573	145.0976	77.7544	42.5199	154.8727	69.1660	43.2837	57.3317	46.9711	41.8661	56.6571	44.1496
	9	30.8487	38.6796	34.7126	37.9657	68.3035	48.1989	26.4226	68.8222	40.9462	26.9695	36.2383	31.0982	26.8152	33.3517	28.4364
Img2	11	21.3015	30.3778	25.8154	24.2488	48.8927	33.6701	20.5223	34.9736	26.8082	20.6320	24.5724	23.1308	17.8593	24.2678	20.2853
	5	73.2504	92.9049	82.8893	86.0895	156.6727	114.9286	73.9151	349.8213	118.5227	71.7816	85.8701	78.5530	70.5339	78.2260	73.4527
	7	41.4326	53.4153	47.4153	48.3036	82.9400	65.0717	41.9856	263.3159	72.6316	40.8252	56.1084	43.4268	39.0015	45.7498	40.8423
Img3	9	26.1752	36.5263	31.9370	36.4667	51.5990	42.9784	27.7489	137.4391	47.6483	26.6524	61.6795	28.7069	24.1555	31.1199	25.8221
	11	20.1261	26.8890	23.0691	21.4976	41.4216	29.3018	18.1101	86.9077	28.4234	18.6342	24.9850	20.5602	16.5130	21.6396	17.8295
	5	85.3624	108.0946	93.2575	92.6769	238.8889	134.5894	83.5441	290.0209	115.9333	82.7538	89.2957	85.7655	83.7992	89.1544	86.0472
	7	49.5183	70.6075	57.2326	53.2974	144.3035	81.1712	44.3984	371.7449	83.0747	44.9442	56.7448	50.0468	43.6775	57.4731	46.3349
	9	29.8212	45.2094	37.5545	42.9729	131.6787	56.7820	30.0358	78.8885	47.0470	29.4037	43.2815	34.7177	26.9209	33.1194	29.4031
	11	22.2296	31.5888	27.6881	28.0517	54.8008	37.4804	20.4221	126.2538	41.1720	21.5165	29.0012	24.9040	18.5866	25.8560	20.7961

Table 4: Comparing FSIM values produced by EMCQ with other algorithms using Otsu's inter-class variance

Otsu's inter-class variance: FSIM Comparison on CT Images																
Images	No	Jaya			GA			TLBO			FPA			Proposed		
		Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean
Img1	5	0.9877	0.9892	0.9884	0.9825	0.9907	0.9877	0.9608	0.9893	0.9851	0.9881	0.9897	0.9887	0.9874	0.9915	0.9891
	7	0.9948	0.9934	0.9937	0.9901	0.9941	0.9922	0.9818	0.9938	0.9906	0.9932	0.9947	0.9939	0.9925	0.9951	0.9939
	9	0.9951	0.9967	0.9961	0.9926	0.9958	0.9949	0.9888	0.9960	0.9937	0.9954	0.9968	0.9962	0.9964	0.9957	0.9961
	11	0.9961	0.9972	0.9968	0.9957	0.9970	0.9963	0.9936	0.9971	0.9954	0.9967	0.9976	0.9972	0.9978	0.9966	0.9974
Img2	5	0.9858	0.9881	0.9866	0.9810	0.9887	0.9846	0.9609	0.9868	0.9804	0.9861	0.9890	0.9872	0.9837	0.9893	0.9874
	7	0.9908	0.9933	0.9924	0.9881	0.9930	0.9908	0.9682	0.9928	0.9868	0.9917	0.9936	0.9924	0.9917	0.9928	0.9920
	9	0.9937	0.9954	0.9946	0.9921	0.9949	0.9934	0.9774	0.9948	0.9898	0.9944	0.9956	0.9951	0.9941	0.9957	0.9953
	11	0.9945	0.9966	0.9959	0.9937	0.9962	0.9952	0.9833	0.9964	0.9935	0.9962	0.9970	0.9965	0.9959	0.9971	0.9966
Img3	5	0.9748	0.9774	0.9755	0.9704	0.9856	0.9769	0.9638	0.9823	0.9743	0.9747	0.9795	0.9761	0.9742	0.9825	0.9796
	7	0.9868	0.9919	0.9893	0.9819	0.9910	0.9873	0.9652	0.9908	0.9847	0.9894	0.9926	0.9911	0.9898	0.9919	0.9906
	9	0.9908	0.9940	0.9932	0.9877	0.9935	0.9915	0.9813	0.9934	0.9883	0.9923	0.9948	0.9939	0.9936	0.9945	0.9941
	11	0.9931	0.9960	0.9947	0.9922	0.9953	0.9939	0.9773	0.9955	0.9904	0.9943	0.9959	0.9952	0.9946	0.9967	0.9958

Table 5: Comparing PSNR values produced by EMCQ with other algorithms using Kapur's entropy

Kapur's entropy: PSNR Comparison on CT images																
Images	No.	Jaya			GA			TLBO			FPA			Proposed		
		Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean
Img1	5	24.2634	24.3865	24.3141	21.5820	27.0427	24.6916	20.6912	24.8376	23.9655	23.7883	24.9684	24.3547	23.2243	25.4757	24.7625
	7	25.5335	29.3709	28.4502	26.3263	29.6274	28.1335	23.4097	28.8405	26.9198	27.6830	29.0621	28.5941	28.5150	29.1807	28.8285
	9	27.6397	31.7960	30.7701	27.7782	31.6480	30.2714	25.5304	31.3156	29.4768	27.4439	32.0129	30.7300	30.7667	32.0985	31.4075
	11	31.0179	33.4972	32.1979	28.2089	33.8550	31.4813	28.1511	32.7361	31.5574	30.8007	33.5883	32.4387	33.3986	34.1439	33.8458
Img2	5	25.0188	26.9114	26.1945	25.0174	27.7809	26.4753	23.4461	26.4795	25.1414	25.6072	26.4426	26.0131	25.8025	25.8931	25.8429
	7	28.3865	30.1601	29.5025	26.3882	30.4531	28.8238	24.0853	29.9055	27.8300	29.0702	30.3884	29.6590	29.6570	29.9752	29.8098
	9	30.3356	32.3958	31.5907	29.5438	32.5338	30.8718	27.6470	32.4235	30.7346	30.4701	32.6357	31.8556	32.1638	32.6918	32.4269
	11	32.3492	33.9762	33.2855	30.9645	33.7029	32.7619	28.1069	34.0643	31.4631	32.2676	34.4236	33.6187	34.3553	34.8226	34.5255
Img3	5	22.7328	25.7392	24.0279	21.1881	26.5198	24.1113	19.9031	26.0006	23.6954	22.9320	24.1110	23.2945	22.7437	24.7114	23.5697
	7	23.4628	29.5715	27.0393	23.8858	29.3465	26.7390	19.7162	28.7377	26.7632	26.0756	29.0441	27.5622	26.7984	28.8649	27.9231
	9	25.8215	31.2715	29.4155	25.0651	31.1459	28.3785	23.8521	30.8278	28.1668	27.1711	30.8949	28.8808	29.6732	30.9353	30.4543
	11	25.9632	32.5889	30.1930	25.9697	33.1166	29.8900	25.5647	32.1261	30.0646	29.6563	33.5089	31.2976	31.9977	33.1725	32.4208

Table 6: Comparing MSE values produced by EMCQ with other algorithms using Kapur's function

Kapur's entropy: MSE Comparison on CT Images																
Images No.	Jaya			GA			TLBO			FPA			Proposed			
	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	
Img1	5	236.8241	243.6350	240.8116	128.4720	451.7315	231.2259	213.4615	521.1451	266.1564	207.1300	271.8021	239.1804	184.2938	309.4947	218.9127
	7	75.1599	181.8558	94.8344	70.8501	151.5148	102.6476	84.9236	296.5574	144.9903	80.6996	110.8614	90.1927	78.5256	91.5330	85.1942
	9	43.0014	111.9726	55.8207	44.4920	108.4567	62.3818	48.0304	181.9864	77.5292	40.9068	117.1369	56.4848	40.1077	54.5017	47.2013
	11	29.0641	51.4389	39.7997	26.7660	98.2186	48.8473	34.6310	99.5349	46.9603	28.4612	54.0762	37.6652	25.0432	29.7315	26.8501
Img2	5	132.4152	204.7390	157.0606	108.3902	204.8039	148.7812	146.2619	294.0830	203.9104	147.5090	178.7957	163.0173	167.4071	170.9363	169.3569
	7	62.6710	94.2819	73.3995	58.5831	149.3677	88.3988	66.4552	253.8364	115.0764	59.4626	80.5486	70.4844	65.3976	70.3693	67.9509
	9	37.4539	60.1895	45.6422	36.2825	72.2272	53.9767	37.2160	111.7833	57.0471	35.4411	58.3536	42.8171	34.9861	39.5090	37.1993
	11	26.0293	37.8579	30.6288	27.7198	52.0750	34.9842	25.5065	100.5526	50.4604	23.4811	38.5761	28.4935	21.4198	23.8532	22.9465
Img3	5	219.7579	345.7051	288.5853	144.9098	494.6224	269.8592	163.3124	664.9165	298.0693	252.3340	367.5318	306.3961	173.4429	346.5787	261.8562
	7	71.7679	292.957	136.4102	75.5843	265.7685	145.849	86.9581	694.1676	159.8155	81.0346	160.5168	115.8666	84.4481	135.9057	106.0444
	9	48.5213	170.1900	77.0124	49.9448	202.5672	101.7855	53.7401	267.8344	108.7456	52.9166	124.7288	85.8474	52.4269	70.1065	58.8525
	11	35.8250	164.7237	66.6627	31.7262	164.4804	71.7720	39.8536	180.5563	69.5397	28.9863	70.3800	49.3475	31.3206	41.0501	37.2989

Table 7: Comparing FSIM values produced by EMCQ with other algorithms using Kapur's entropy

Kapur's entropy: FSIM Comparison on CT Images																
Images	No.	Jaya			GA			TLBO			FPA			Proposed		
		Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean
Img1	5	0.9695	0.9794	0.9759	0.9569	0.9889	0.9785	0.9571	0.9809	0.9703	0.9668	0.9749	0.9708	0.9685	0.9694	0.9688
	7	0.9856	0.9885	0.9868	0.9797	0.9993	0.9891	0.9719	0.9906	0.9840	0.9864	0.9903	0.9888	0.9879	0.9931	0.9902
	9	0.9910	0.9957	0.9935	0.9917	0.9962	0.9940	0.9799	0.9942	0.9904	0.9912	0.9956	0.9941	0.9926	0.9954	0.9943
	11	0.9920	0.9966	0.9955	0.9920	0.9971	0.9952	0.9860	0.9961	0.9937	0.9944	0.9967	0.9959	0.9962	0.9972	0.9966
Img2	5	0.9724	0.9818	0.9772	0.9747	0.9856	0.9809	0.9629	0.9823	0.9717	0.9704	0.9794	0.9733	0.9700	0.9705	0.9701
	7	0.9864	0.9906	0.9883	0.9864	0.9923	0.9892	0.9684	0.9887	0.9821	0.9861	0.9901	0.9881	0.9855	0.9902	0.9876
	9	0.9914	0.9948	0.9932	0.9888	0.9942	0.9923	0.9800	0.9933	0.9900	0.9918	0.9939	0.9928	0.9921	0.9949	0.9934
	11	0.9940	0.9965	0.9953	0.9930	0.9957	0.9949	0.9807	0.9950	0.9905	0.9939	0.9966	0.9954	0.9948	0.9964	0.9955
Img3	5	0.9487	0.9689	0.9599	0.9473	0.9808	0.9653	0.9123	0.9773	0.9562	0.9461	0.9634	0.9522	0.9438	0.9606	0.9510
	7	0.9671	0.9848	0.9760	0.9664	0.9892	0.9798	0.9180	0.9860	0.9704	0.9686	0.9812	0.9760	0.9694	0.9769	0.9740
	9	0.9816	0.9850	0.9831	0.9724	0.9923	0.9849	0.9627	0.9920	0.9815	0.9801	0.9896	0.9835	0.9776	0.9910	0.9860
	11	0.9865	0.9913	0.9887	0.9763	0.9944	0.9877	0.9685	0.9941	0.9863	0.9858	0.9950	0.9911	0.9813	0.9955	0.9912

Table 8: Elapsed Time of the proposed and other approaches on Img4

Thresholds	Jaya			TLBO			GA			FPA			Proposed		
	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean
5	0.6600	0.8500	0.7300	0.8600	1.3600	1.1300	0.8200	1.4300	0.9600	0.6605	1.8864	0.8221	1.3672	1.8561	1.5376
7	0.8489	1.7896	1.0481	1.3308	1.8617	1.5011	0.8962	1.1664	0.9975	0.6169	2.4208	0.9766	1.5291	3.4369	1.9131
9	0.8843	1.1605	0.9939	1.6700	2.5000	2.0600	1.0900	1.6800	1.2100	0.9800	1.2100	1.1000	1.7839	2.1731	1.9512
11	0.9945	1.3968	1.0745	2.1061	2.7667	2.3790	1.6124	1.9726	1.7431	1.2530	1.6511	1.3555	2.1017	2.8195	2.3458

Table 9: Comparing PSNR values produced by EMCQ with other algorithms using high threshold values

Image	Threshold	HHUB	ABC	ABCSA	SCA	FAABC	SSO	HHB	EMCQ
Img5	10	26.842	17.908	20.357	18.423	19.815	24.386	26.682	27.496
	15	26.144	20.692	21.104	17.537	20.951	26.957	30.575	30.6568
	20	26.621	21.760	22.463	23.559	23.746	28.698	32.746	32.7275
	30	28.135	24.382	20.959	27.737	30.650	30.692	36.040	36.9841
Img6	10	22.667	19.897	19.712	21.376	23.666	23.498	27.388	26.7781
	15	24.710	20.590	19.715	22.908	24.227	25.111	30.683	29.6195
	20	26.194	23.610	21.054	25.607	25.749	27.114	33.067	30.6236
	30	27.487	26.485	27.764	28.536	29.878	30.404	33.181	32.0388
Img7	10	26.520	18.539	18.744	11.555	19.737	23.940	26.494	26.6677
	15	27.669	19.365	19.392	11.789	21.322	26.031	29.055	29.9254
	20	26.958	21.180	21.252	12.315	23.525	28.133	31.147	31.3863
	30	26.078	24.525	23.065	15.228	25.860	30.786	34.179	33.6468
Img8	10	24.874	18.332	19.018	12.060	19.395	23.145	27.516	25.3003
	15	24.678	19.927	19.277	14.784	21.149	25.760	29.516	28.3560
	20	25.415	22.586	19.080	16.162	24.619	27.235	32.033	30.3660
	30	27.532	24.486	20.991	14.352	28.278	29.233	35.235	32.9312
Img9	10	26.986	20.990	20.158	12.428	22.447	24.764	27.062	27.2575
	15	28.194	19.867	20.594	12.965	22.387	25.121	30.683	30.4528
	20	27.614	22.002	20.654	13.106	24.620	28.393	31.572	32.7269
	30	27.529	24.031	22.089	17.281	26.083	31.005	33.803	36.4163
Img10	10	25.659	19.361	19.246	19.108	19.313	23.064	25.836	26.4658
	15	26.384	20.136	20.412	11.554	22.455	25.926	27.898	26.1355
	20	26.521	21.974	20.733	13.164	24.887	27.864	30.713	30.5231
	30	26.897	24.357	21.790	21.455	26.887	30.418	34.163	35.2873

Table 10: Comparing SSIM values produced by EMCQ with other algorithms using high threshold values

Image	Thresholds	HHUB	ABC	ABCSA	SCA	FAABC	SSO	FASSO	HHB	EMCQ
Img5	10	0.8048	0.6737	0.7190	0.5757	0.7077	0.7472	0.7596	0.7992	0.7829
	15	0.8196	0.7708	0.7294	0.5486	0.7563	0.8054	0.8004	0.8394	0.8735
	20	0.8352	0.7797	0.7679	0.7254	0.7944	0.8330	0.8252	0.9319	0.9144
	30	0.8661	0.8229	0.8131	0.6219	0.8631	0.8704	0.8758	0.9636	0.9704
Img6	10	0.8143	0.7701	0.7638	0.7926	0.8200	0.8304	0.8306	0.8878	0.8947
	15	0.8535	0.7988	0.7899	0.7965	0.8574	0.8661	0.8585	0.9273	0.9275
	20	0.8664	0.8391	0.8040	0.8479	0.8655	0.8715	0.8824	0.9502	0.9206
	30	0.8976	0.8754	0.8950	0.8890	0.9145	0.9076	0.9129	0.9695	0.9507
Img7	10	0.8340	0.6883	0.7030	0.4245	0.7245	0.7879	0.7850	0.8313	0.8157
	15	0.8596	0.7192	0.7312	0.4645	0.7721	0.8336	0.8285	0.8833	0.8832
	20	0.8580	0.7648	0.7739	0.4775	0.8008	0.8651	0.8657	0.9125	0.9071
	30	0.8437	0.8194	0.8074	0.6494	0.8432	0.9068	0.9028	0.9426	0.9421
Img8	10	0.8348	0.6583	0.7004	0.4208	0.7003	0.7755	0.7915	0.8495	0.8757
	15	0.8290	0.7008	0.7304	0.5185	0.7570	0.8469	0.8467	0.8495	0.9201
	20	0.8450	0.7891	0.7346	0.5526	0.8253	0.8710	0.8777	0.9476	0.9401
	30	0.8805	0.8199	0.7966	0.4857	0.8941	0.8994	0.9090	0.9699	0.9659
Img9	10	0.8762	0.7239	0.7070	0.4129	0.7654	0.8099	0.8089	0.8844	0.8866
	15	0.8900	0.6996	0.7286	0.4262	0.7901	0.8652	0.8585	0.9044	0.9296
	20	0.8732	0.7655	0.7261	0.4450	0.8121	0.8984	0.8931	0.9076	0.9510
	30	0.8649	0.8104	0.7776	0.6232	0.8656	0.9312	0.9291	0.9468	0.9761
Img10	10	0.8996	0.7135	0.7102	0.2719	0.7678	0.8374	0.8337	0.8916	0.9220
	15	0.8951	0.7306	0.7497	0.3190	0.8103	0.8928	0.8958	0.9467	0.9227
	20	0.8907	0.7937	0.7616	0.4148	0.8268	0.9240	0.9240	0.9607	0.9606
	30	0.8950	0.8570	0.7933	0.8119	0.9001	0.9515	0.9512	0.9802	0.9875

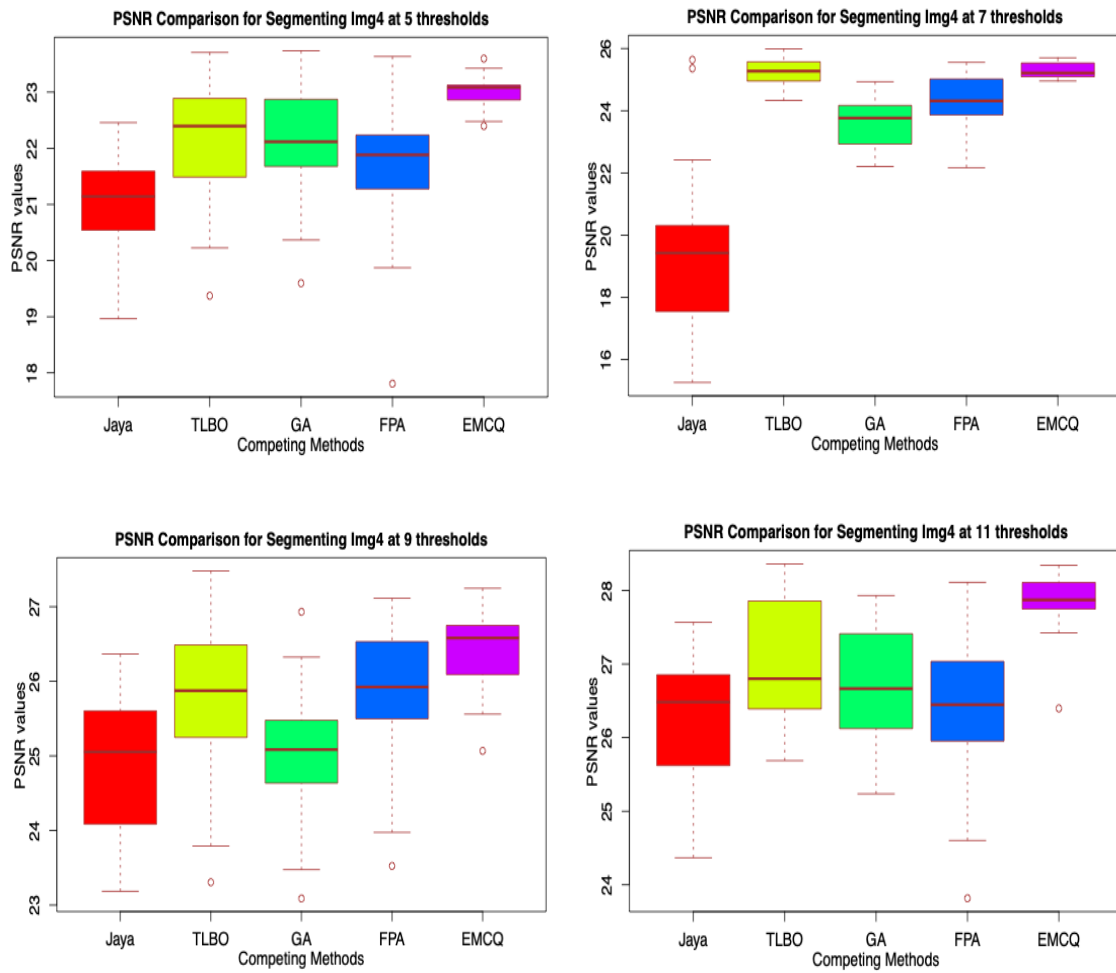


Figure 9: Boxplots for comparing PSNR values using different threshold levels

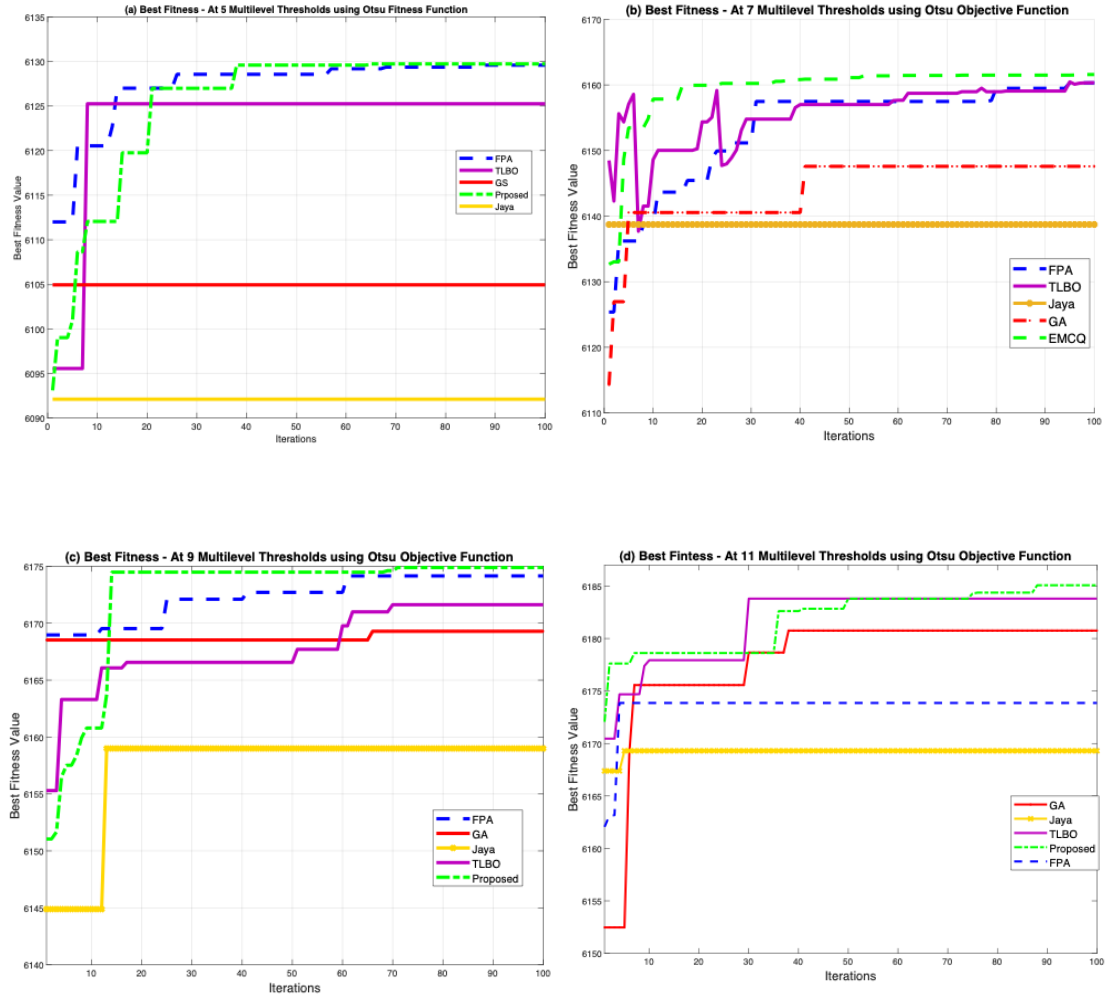


Figure 10: Depiction of objective function values over 100 iterations

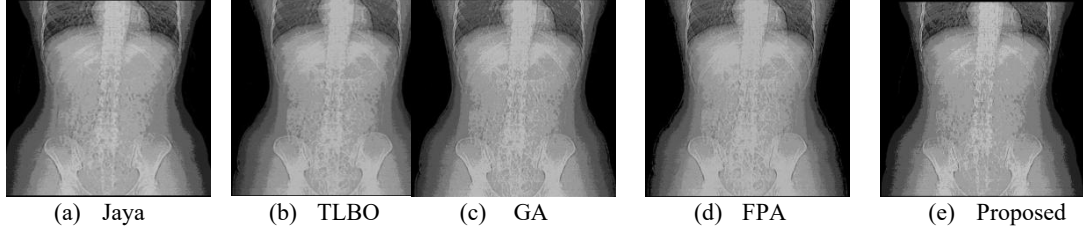


Figure 11: Comparison based on segmenting Img4 using 5 thresholds

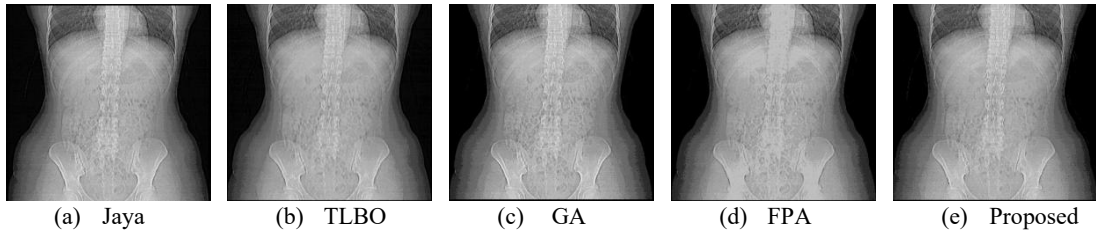


Figure 12: Comparison based on segmenting Img4 using 9 thresholds

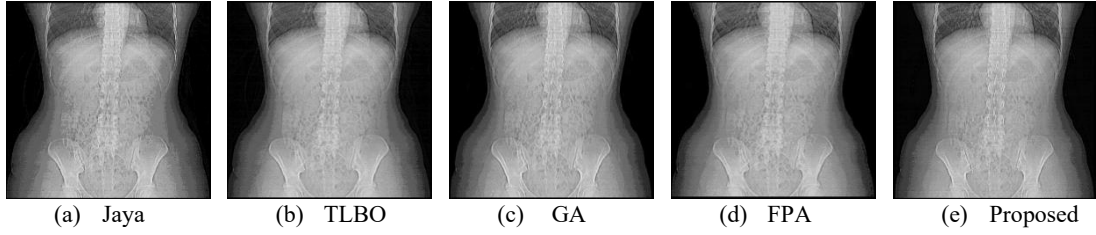


Figure 13: Comparison based on segmenting Img4 using 11 thresholds

Table 11: Statistical analysis for results in Table 2 using Wilcoxon rank-sum test

Pair Comparison	p-value in increasing order	α_{Holm} as Bonferroni-Holm Correction at 95% confidence level	α_{Holm} as Bonferroni-Holm Correction at 90% confidence level
EMCQ vs GA	0.002	$\alpha_{Holm}=0.0125$, p-value < α_{Holm} , Reject H_0	$\alpha_{Holm}=0.025$, p-value < α_{Holm} , Reject H_0
EMCQ vs TLBO	0.004	$\alpha_{Holm}=0.017$, p-value < α_{Holm} , Reject H_0	$\alpha_{Holm}=0.03$, p-value < α_{Holm} , Reject H_0
EMCQ vs Jaya	0.017	$\alpha_{Holm}=0.025$, p-value < α_{Holm} , Reject H_0	$\alpha_{Holm}=0.05$, p-value < α_{Holm} , Reject H_0
EMCQ vs FP	0.020	$\alpha_{Holm}=0.05$, p-value < α_{Holm} , Reject H_0	$\alpha_{Holm}=0.1$, p-value < α_{Holm} , Reject H_0

Table 12: Statistical analysis for results in Table 5 using Wilcoxon rank-sum test

Pair Comparison	p-value in increasing order	α_{Holm} as Bonferroni-Holm Correction at 95% confidence level	α_{Holm} as Bonferroni-Holm Correction at 90% confidence level
EMCQ vs TLBO	0.002	$\alpha_{Holm}=0.0125$, p-value < α_{Holm} , Reject H_0	$\alpha_{Holm}=0.025$, p-value < α_{Holm} , Reject H_0
EMCQ vs FP	0.002	$\alpha_{Holm}=0.017$, p-value < α_{Holm} , Reject H_0	$\alpha_{Holm}=0.03$, p-value < α_{Holm} , Reject H_0
EMCQ vs GA	0.004	$\alpha_{Holm}=0.025$, p-value < α_{Holm} , Reject H_0	$\alpha_{Holm}=0.05$, p-value < α_{Holm} , Reject H_0
EMCQ vs Jaya	0.006	$\alpha_{Holm}=0.05$, p-value < α_{Holm} , Reject H_0	$\alpha_{Holm}=0.1$, p-value < α_{Holm} , Reject H_0

Table 13: Statistical analysis for results in Table 9 using Wilcoxon rank-sum test

Pair Comparison	p-value in increasing order	α_{Holm} as Bonferroni-Holm Correction at 95% confidence level	α_{Holm} as Bonferroni-Holm Correction at 90% confidence level
EMCQ vs HHUB	0.000	$\alpha_{Holm}=0.007$, p-value < α_{Holm} , Reject H_0	$\alpha_{Holm}=0.014$, p-value < α_{Holm} , Reject H_0
EMCQ vs ABC	0.000	$\alpha_{Holm}=0.008$, p-value < α_{Holm} , Reject H_0	$\alpha_{Holm}=0.017$, p-value < α_{Holm} , Reject H_0
EMCQ vs ABCSCA	0.000	$\alpha_{Holm}=0.01$, p-value < α_{Holm} , Reject H_0	$\alpha_{Holm}=0.02$, p-value < α_{Holm} , Reject H_0
EMCQ vs SCA	0.000	$\alpha_{Holm}=0.0125$, p-value < α_{Holm} , Reject H_0	$\alpha_{Holm}=0.025$, p-value < α_{Holm} , Reject H_0
EMCQ vs FAABC	0.000	$\alpha_{Holm}=0.017$, p-value < α_{Holm} , Reject H_0	$\alpha_{Holm}=0.03$, p-value < α_{Holm} , Reject H_0
EMCQ vs SSO	0.000	$\alpha_{Holm}=0.025$, p-value < α_{Holm} , Reject H_0	$\alpha_{Holm}=0.05$, p-value < α_{Holm} , Reject H_0
EMCQ vs HHB	0.188	$\alpha_{Holm}=0.05$, p-value < α_{Holm} , Cannot Reject H_0	$\alpha_{Holm}=0.1$, p-value < α_{Holm} , Cannot Reject H_0

7. Conclusion and Future Direction

In this research study, a hyper-heuristic based method is proposed for the segmentation of covid-19 image data set using multilevel thresholding. The proposed method consists of two levels i.e. high-level and low-level. At the high-level, Exponential Monte Carlo with Counter (EMCQ) heuristic method is utilized which works like a manager and controller of a set of four low-level heuristics. At the low-level, four heuristic search operators are utilized, namely GA crossover operator, TLBO peer learning operator, FPA global search operator and Jaya algorithm. EMCQ is a simple, robust, fast and parameter free hyper-heuristic and the low-level heuristics are simple operators not complete meta-heuristic algorithms. These features make the proposed methodology lightweight and more efficient than existing methods in the literature. The proposed method is utilized to find optimum threshold values for the problem of multi-level threshold image segmentation and offered best segmentation results in terms of PSNR, MSE, FSIM and SSIM. The proposed method based on the EMCQ hyper-heuristic is tested on 25 images from Covid-19 image data set taken from Kaggle repository. Experimental results from three randomly selected images are then presented which show that EMCQ is capable of providing very good results as compared to other state-of-art meta-heuristic algorithms such as Genetic Algorithm (GA), Teaching Learning-based Optimization (TLBO) algorithm, Flower Pollination Algorithm (FPA) and Jaya algorithm. One 2-D image is also used

for further evaluation of the proposed method against the competing methods. Moreover, the method showed comparable segmentation results against the only hyper-heuristic strategy named Hyper-heuristic Best (HHB) on classical image dataset. Based on the experimental results and above discussed features, it can be concluded that the proposed hyper-heuristic method is a robust and powerful optimization tool for the multi-level thresholding image segmentation problem. In future, the proposed method can be utilized for other problems in image processing such as image enhancement and image registration.

Acknowledgment

This research is supported by the Fundamental Research Grant (Ref: FRGS/1/2023/ICT02/UMP/01/1) titled: “A Multi-Factorial Agent Heroes and Cowards Algorithm based Strategy for Test Redundancy Reduction Problem”. We thank the Ministry of Higher Education, Malaysia and UMPSA for their contribution and support.

References

- Abd Elaziz M., Ewees A.A. & Oliva D. 2020. Hyper-heuristic method for multilevel thresholding image segmentation. *Expert Systems with Applications* **146**: 113201.
- Abdel-Basset M., Abdel-Fatah L. & Sangaiah A.K. 2018. Metaheuristic algorithms: A comprehensive review. In Sangaiah A.K., Zhang Z. & Sheng M. (eds.). *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications*: 185–231. London, UK: Elsevier.
- Abualigah L., Almotairi K.H. & Abd Elaziz M. 2023. Multilevel thresholding image segmentation using meta-heuristic optimization algorithms: Comparative analysis, open challenges and new trends. *Applied Intelligence* **53**(10): 11654-11704.
- Almufti S.M., Shaban A.A., Ali Z.A., Ali R.I. & Dela Fuente J.A. 2023. Overview of metaheuristic algorithms. *Polaris Global Journal of Scholarly Research and Trends* **2**(2): 10-32.
- Burke E.K. & Kendall G. 2014. *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. 2nd Ed. New York, NY: Springer.
- Burke E.K. & Bykov Y. 2017. The late acceptance hill-climbing heuristic. *European Journal of Operational Research* **258**(1): 70-78.
- Burke E., Kendall G., Newall J., Hart E., Ross P. & Schulenburg S. 2003. Hyper-heuristics: An emerging direction in modern search technology. In Glover F. & Kochenberger G.A. (eds.). *Handbook of Metaheuristics*: 457-474. Boston, MA: Springer.
- Chen J., Cai Z., Heidari A.A., Chen H., He Q., Escorcia-Gutierrez J. & Mansour R.F. 2023. Multi-threshold image segmentation based on an improved differential evolution: Case study of thyroid papillary carcinoma. *Biomedical Signal Processing and Control* **85**: 104893.
- Desale S., Rasool A., Andhale S. & Rane P. 2015. Heuristic and meta-heuristic algorithms and their relevance to the real world: A survey. *International Journal of Computer Engineering in Research Trends* **2**(5): 296-304.
- Dhal K.G., Das A., Ray S., Gálvez J. & Das S. 2020. Nature-inspired optimization algorithms and their application in multi-thresholding image segmentation. *Archives of Computational Methods in Engineering* **27**(3): 855-888.
- Din F. & Zamli K.Z. 2018. Hyper-heuristic based strategy for pairwise test case generation. *Advanced Science Letters* **24**(10): 7333-7338.
- Drake J.H., Kheiri A., Özcan E. & Burke E.K. 2020. Recent advances in selection hyper-heuristics. *European Journal of Operational Research* **285**(2): 405-428.
- Dubey S.K., Vijay S. & Pratibha A. 2018. A review of image segmentation using clustering methods. *International Journal of Applied Engineering Research* **13**(5): 2484-2489.
- El Aziz M.A., Ewees A.A. & Hassanien A.E. 2016. Hybrid swarms optimization based image segmentation. In Bhattacharyya S., Dutta P., De S. & Klepac G. (eds.). *Hybrid Soft Computing for Image Segmentation*: 1-21. Cham: Springer.
- Geng X., Chen Z., Yang W., Shi D. & Zhao K. 2011. Solving the traveling salesman problem based on an adaptive simulated annealing algorithm with greedy search. *Applied Soft Computing* **11**(4): 3680-3689.
- Gharehchopogh F.S. & Ibrici T. 2024. An improved African vultures optimization algorithm using different fitness functions for multi-level thresholding image segmentation. *Multimedia Tools and Applications* **83**(6): 16929-16975.
- Glover F. & Greenberg H.J. 1989. New approaches for heuristic search: A bilateral linkage with artificial intelligence. *European Journal of Operational Research* **39**(2): 119-130.
- Hertwig R. & Todd P.M. 2002. Heuristics. In Ramachandran V.S. (ed.). *Encyclopedia of the Human Brain*: 449-

460. San Diego, CA: Academic Press.
- Horé A. & Ziou D. 2010 Image quality metrics: PSNR vs. SSIM. *2010 20th International Conference on Pattern Recognition*, pp. 2366-2369.
- Hosny K.M., Khalid A.M., Hamza H.M. & Mirjalili S. 2022. Multilevel segmentation of 2D and volumetric medical images using hybrid Coronavirus Optimization Algorithm. *Computers in Biology and Medicine* **150**: 106003.
- Houssein E.H., Abdalkarim N., Hussain K. & Mohamed E. 2024. Accurate multilevel thresholding image segmentation via oppositional Snake Optimization algorithm: Real cases with liver disease. *Computers in Biology and Medicine* **169**: 107922.
- Jayaraman K., Srilakshmi K. & Jayaraman S. 2022. Modified flower pollination-based segmentation of medical images. *International Journal of Image and Graphics* **22**(2): 2250026.
- Jia H., Lu C. & Xing Z. 2024. Memory backtracking strategy: An evolutionary updating mechanism for meta-heuristic algorithms. *Swarm and Evolutionary Computation* **84**: 101456.
- Kapur J.N., Sahoo P.K. & Wong A.K.C. 1985. A new method for gray-level picture thresholding using the entropy of the histogram. *Computer Vision, Graphics, and Image Processing* **29**(3): 273-285.
- Karaboga D. & Ozturk C. 2011. A novel clustering approach: Artificial Bee Colony (ABC) algorithm. *Applied Soft Computing* **11**(1): 652-657.
- Khalifa A.F. & Badr E. 2023. Deep learning for image segmentation: A focus on medical imaging. *Computers, Materials & Continua* **75**(1): 1995-2024.
- Khosrowshahli R. Rahnamayan S., Ibrahim A., Asilian Bidgoli A. & Makrehchi M. 2025. Population-level center-based sampling for meta-heuristic algorithms. *Swarm and Evolutionary Computation* **92**: 101827.
- Lee K.Y. & El-Sharkawi M.A. 2002. Modern heuristic optimization techniques with applications to power systems. *Proceedings of the 2002 IEEE Power Engineering Society Winter Meeting*, pp. 114-118.
- Minae S., Boykov Y., Porikli F., Plaza A., Kehtarnavaz N. & Terzopoulos D. 2022. Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **44**(7): 3523-3542.
- Mirjalili S. 2016. SCA: A sine cosine algorithm for solving optimization problems. *Knowledge-Based Systems* **96**: 120-133.
- Mostafa R.R., Houssein E.H., Hussien A.G., Singh B. & Emam M.M. 2024. An enhanced chameleon swarm algorithm for global optimization and multi-level thresholding medical image segmentation. *Neural Computing and Applications* **36**(15): 8775-8823.
- Salavati A.A.H. & Mozafari S. 2015. Provide a hybrid method to improve the performance of multilevel thresholding for image segmentation using GA and SA algorithms. *2015 7th Conference on Information and Knowledge Technology (IKT)*, pp. 1-6.
- Nasser A.B., Kader M.A., Alsewari A.A., Fati S.M., Abdullah N., Diaba S.Y. & Jabbar W.A. 2024. CS-FuzGA-PTS: Maximizing fault detection through optimizing T-way test suite prioritization based on boundary value analysis. *IEEE Access* **12**: 172992-173009.
- Nath S., Agarwal S. & Kazmi Q.A. 2011. Image histogram segmentation by multi-level thresholding using hill climbing algorithm. *International Journal of Computer Applications* **35**(1): 63-72.
- Ohashi T., Aghbari Z. & Makinouchi A. 2003. Hill-climbing algorithm for efficient color-based image segmentation. *Proceedings of the IASTED International Conference on Signal Processing, Pattern Recognition, and Applications*, pp. 17-22.
- Osaba E., Villar-Rodríguez E., Del Ser J., Nebro A.J., Molina D., LaTorre A., Suganthan P.N., Coello Coello C.A. & Herrera F. 2021. A tutorial on the design, experimentation and application of metaheuristic algorithms to real-world optimization problems. *Swarm and Evolutionary Computation* **64**: 100888.
- Otsu N. 1979. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics* **9**(1): 62-66.
- Ouadfel S. & Taleb-Ahmed A. 2016. Social spiders optimization and flower pollination algorithm for multilevel image thresholding: A performance study. *Expert Systems with Applications* **55**: 566-584.
- Qiao L., Liu K., Xue Y., Tang W. & Salehnia T. 2024. A multi-level thresholding image segmentation method using hybrid Arithmetic Optimization and Harris Hawks Optimizer algorithms. *Expert Systems with Applications* **241**: 122316.
- Ryalat M.H., Dorgham O., Tedmori S., Al-Rahamneh Z., Al-Najdawi N. & Mirjalili S. 2023. Harris hawks optimization for COVID-19 diagnosis based on multi-threshold image segmentation. *Neural Computing and Applications* **35**: 6855-6873.
- Salih S.Q., Alsewari A.A., Wahab H.A., Mohammed M.K.A., Rashid T.A., Das D. & Basurra S.S. 2023. Multi-population Black Hole Algorithm for the problem of data clustering. *PLoS ONE* **18**(7): e0288044.
- Satapathy S.C., Raja N.S.M., Rajinikanth V., Ashour A.S. & Dey N. 2018. Multi-level image thresholding using Otsu and chaotic bat algorithm. *Neural Computing and Applications* **29**(12): 1285-1307.
- Sathya P.D., Kalyani R. & Sakthivel V.P. 2021. Color image segmentation using Kapur, Otsu and minimum cross entropy functions based on exchange market algorithm. *Expert Systems with Applications* **172**: 114636.
- Senthilkumaran N. & Vaithegi S. 2016. Image segmentation by using thresholding techniques for medical images. *Computer Science & Engineering: An International Journal* **6**(1): 1-13.

- Shih F.Y. 2017. *Image Processing and Mathematical Morphology: Fundamentals and Applications*. Boca Raton, FL: CRC Press.
- Suresh K. & Sakthi U. 2019. Analysis of heuristic-based multilevel thresholding methods for image segmentation using R programming. *International Journal of Reasoning-based Intelligent Systems* **11**(2): 151-160.
- Szénási S. 2014. Distributed region growing algorithm for medical image segmentation. *International Journal of Circuits, Systems and Signal Processing* **8**: 173-181.
- Thapliyal S. & Kumar N. 2024. A novel algorithm for image segmentation (IP-MH-MLT): Employing an image partitioning technique with metaheuristic parameters to enhance multilevel thresholding. *International Journal of System Assurance Engineering and Management* **15**(9): 4291-4347.
- Vasant P.M. 2012. *Meta-heuristics Optimization Algorithms in Engineering, Business, Economics, and Finance*. Hershey, PA: IGI Global.
- Vijh S., Saraswat M. & Kumar S. 2023. Automatic multilevel image thresholding segmentation using hybrid bio-inspired algorithm and artificial neural network for histopathology images. *Multimedia Tools and Applications* **82**(4): 4979-5010.
- Youssef H., Sait S.M. & Adiche H. 2001. Evolutionary algorithms, simulated annealing and tabu search: A comparative study. *Engineering Applications of Artificial Intelligence* **14**(2): 167-181.
- Zhang L., Zhang L., Mou X. & Zhang D. 2011. FSIM: A feature similarity index for image quality assessment. *IEEE Transactions on Image Processing* **20**(8): 2378-2386.

*Department of Computer Science and IT
Faculty of Computing and Engineering
University of Malakand
Khyber Pakhtunkhwa 18800,
PAKISTAN*

*Email: lmnuom15@gmail.com, fakhruddin@uom.edu.pk, khaliddir@gmail.com,
salam@uom.edu.pk*

*Faculty of Computing
Universiti Malaysia Pahang Al-Sultan Abdullah (UMPSA)
26600 Pekan, Kuantan
Pahang, MALAYSIA
Email: kamalz@umpsa.edu.my**

Received: 25 December 2024

Accepted: 12 February 2025

*Corresponding author