An adaptive multilevel thresholding method with chaotically-enhanced Rao algorithm

Yagmur Olmez1 • Abdulkadir Sengur2 • Gonca Ozmen Koca1 • Ravipudi Venkata Rao3

Received: 21 October 2021 / Revised: 7 February 2022 / Accepted: 11 August 2022
Published online: 9 September 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
Multilevel image thresholding is a well-known technique for image segmentation. Recently, various metaheuristic methods have been proposed for the determination of the thresholds for multilevel image segmentation. These methods are mainly based on metaphors and they have high complexity and their convergences are comparably slow. In this paper, a multilevel image thresholding approach is proposed that simplifies the thresholding problem by using a simple optimization technique instead of metaphor-based algorithms. More specifically, in this paper, Chaotic enhanced Rao (CER) algorithms are developed where eight chaotic maps namely Logistic, Sine, Sinusoidal, Gauss, Circle, Chebyshev, Singer, and Tent are used. Besides, in the developed CER algorithm, the number of thresholds is determined automatically, instead of manual determination. The performances of the developed CER algorithms are evaluated based on different statistical analysis metrics namely BDE, PRI, VOI, GCE, SSIM, FSIM, RMSE, PSNR, NK, AD, SC, MD, and NAE. The experimental works and the related evaluations are carried out on the BSDS300 dataset. The obtained experimental results demonstrate that the proposed CER algorithm outperforms the compared methods based on PRI, SSIM, FSIM, PSNR, RMSE, AD, and NAE metrics. In addition, the proposed method provides better convergence regarding speed and accuracy.

Keywords Multilevel thresholding • Image segmentation • Metaheuristic methods • Chaotic search • Rao algorithm

1 Introduction
Image segmentation is a challenging and essential step in image processing and pattern recognition, where an image is segmented into several classes based on its characteristics, such as intensity, texture, and contrast. Thresholding is the simplest and one of the most
popular methodologies of image segmentation methods based on the partition of the pixels. Therefore, it possesses extensive applications in different fields such as infrared imaging [16], medical imaging [17], microscopic imaging [15], remote sensing [32], etc. Based on the number of thresholds, it can be classified into two groups as bi-level and multilevel thresholding. Bi-level thresholding is the simplest method that divides the image into two parts as object and background using one threshold value. Multilevel thresholding divides the image into multiple classes and uses more than two threshold values.

In 2014, multilevel thresholding was employed by Tuba with Otsu’s maximum variance and Kapur entropy [37]. In the study using the Intel i7–4770 K CPU, it is stated that Otsu’s thresholding method required approximately two months to find seven thresholds, 40 years for eight thresholds, and approximately 10,000 years for nine thresholds. The problem of computational complexity in multilevel thresholding can be solved by using optimization algorithms. Agrawal et al. [3] developed a multilevel thresholding method based on the Coral reef optimization algorithm. The authors found the optimal threshold values by minimizing the objective function obtained from the diagonal class entropy (DCE). The proposed method was compared with 2-D Otsu’s method and other multi-level thresholding methods (Entropy of Kapur, Saliency Map [2], KL-MPSO [47], and Entropy-based [5]). The superiority of the method was then demonstrated through the experimental results.

Pare et al. [22] presented a new segmentation approach for color multilevel thresholding based on gray level co-occurrence matrix (GLCM). To handle the shortcomings of the GLCM related to computational complexity and stability, the authors utilized the Differential Evaluation (DE) algorithm to find the most appropriate threshold values. It was stated that the developed method performed better than the other optimization algorithms (Backward Search Optimization Algorithm, Particle Swarm Optimization (PSO), Bacteria Foraging Optimization (BFO), and traditional GLCM algorithm). Besides, a herd-based algorithm for color multilevel image segmentation using the comprehensive energy function was implemented by Pare et al. [24]. Shao et al. [30] proposed a new method for segmenting ultrasound images using multilevel thresholding based on a different search algorithm. Bhandari et al. [6] applied the cuckoo and wind-powered optimization algorithm for segmenting satellite images with multilevel thresholding based on Kapur’s entropy. Bao et al. [4] applied a new hybrid algorithm, combining Harris Hawks (HH) and DE optimization algorithms for multilevel color image segmentation. Jia et al. [12] developed a new algorithm called self-adaptive moth-flame optimization to improve color image segmentation. Kapur’s entropy method and Otsu’s thresholding method were used as objective functions. The results were compared with eight other meta-heuristic algorithms (multi-verse optimizer, whale optimization algorithm, standard moth-flame optimization algorithm, etc.) It was reported that the proposed optimization algorithm was better than the other algorithms. Pare et al. [23] presented a modified fuzzy entropy function to achieve multilevel thresholding of color images at different levels. The search performance and image segmentation performance were improved by using the Levy flight-guided firefly algorithm.

Apart from these, a Differential Evolutionary Adaptive Harris Hawks Optimization Algorithm [42] is proposed for multilevel thresholding based on 2-D practical Masi entropy. Selection operators and differential position concepts in the DE algorithm are used to improve the performance of the HHO algorithm. The performance of the modified optimization algorithm is shown by comparing state of art algorithms based on multilevel thresholding. Xing proposes an improved emperor penguin optimization algorithm [43] based on multilevel color image thresholding using Berkeley images, Satellite images, and plant canopy images.
To develop the searchability emperor penguin optimization algorithm opposition-based learning, the Gaussian mutation, and the Levy flight strategies are employed. Yue et al. propose a modified hybrid bat algorithm for multilevel thresholding based on Otsu’s method and Kapur’s entropy [44]. The algorithm’s local search ability is improved with the crossover operation. The thresholding method is assessed by a set of images using different levels of thresholds. The proposed optimization algorithm is compared with genetic algorithm (GA), gravitational search algorithm (GSA), particle swarm optimization (PSO), whale optimization algorithm (WOA), improved salp swarm algorithm (LSSA), and basic bat algorithm (BA). Houssein et al. suggest an improved equilibrium optimizer for the multi-level thresholding segmentation based on fuzzy entropy [10]. The local search ability of the equilibrium optimizer method is enhanced by combining the standard operators of the optimizer with the dimension learning hunting. The proposed segmentation method is applied to CT images of COVID-19. The performance of the proposed method has been proven by a series of experimental studies. A hybrid whale optimization algorithm is proposed to color image multilevel thresholding [1]. In this study, a novel elimination method is proposed for the optimization algorithm. Based on this elimination method and WOA, a new segmentation model is developed. The segmentation model is applied to ten images from Berkeley University Dataset and it is compared with a set of well-known algorithms in terms of the objective values, peak signal-to-noise ratio index, structural similarity index (SSIM), features similarity index (FSIM), and CPU time. In the experimental studies, it is seen that the proposed approach is superior to the other algorithms in terms of PSNR and FSIM indexes. Swain et al. multilevel thresholding approach is proposed based on differential exponential entropy [36]. In addition, a new Equilibrium-Cuckoo Search Optimizer algorithm is suggested to obtain optimal thresholds. The comparisons are fulfilled with Cuckoo Search Algorithm and Equilibrium Optimizer. The experimental results are evaluated in terms of PSNR, SSIM, and FSIM and it is shown that the proposed approach outperforms the other techniques.

It is observed from the literature review that many optimization algorithms are based on the metaphor of some natural phenomena or behavior of animals, fishes, insects, societies, cultures, planets, musical instruments, etc. However, it would be better if the researchers focus on developing simple optimization techniques that can provide effective solutions to complex problems instead of looking for developing metaphor-based algorithms. Many of the metaphor-based algorithms are complex and are not good enough to solve varieties of engineering problems. Furthermore, many of these algorithms require tedious tuning of their algorithm-specific parameters in addition to the tuning of common control parameters such as population size and the number of iterations to obtain the best results. These algorithms may not contribute to the progress of the optimization field [26, 33]. Rao [26] developed three simple and powerful algorithm-specific parameter-less and metaphor-less optimization algorithms, named Rao algorithms. These algorithms are gaining wide acceptance by the optimization research community [8, 9, 28, 31, 39, 41].

This study focuses on improving the Rao-1 algorithm based on various chaotic mappings. Chaos is a nonlinear phenomenon that refers to an arbitrary situation in a deterministic system. It has the basic properties of randomness, universality, and the law of scaling. Chaos theory is concerned with the initial conditions in which a system or chaotic movement can pass through all stages in the range under the chosen chaotic behavior without repeating itself within certain limits. Therefore, it is more ergonomic and advantageous to use chaotic theory instead of irregular randomness in optimization problems. The use of chaotic mapping techniques as random number generators in heuristic algorithms is an increasingly common topic. These
mapping techniques, which have a wide spectrum, increase the performance by increasing the variety in the random selection of metaheuristic algorithms.

As mentioned earlier, the Rao optimization approach is a metaphor-free algorithm that uses a simple optimization technique. Moreover, when the above-mentioned studies are examined, it is seen that the number of thresholds is manually determined as a user parameter, which yields a fixed segment number in the segmentation process. It is worth mentioning that giving fixed threshold numbers to all input images produces the same number of segments for all images that is not applicable in all areas. These lead us to propose a new multilevel thresholding method based on the chaotically enhanced Rao algorithm with an adaptively adjustable segment number of images. Thus, it allows the proposed method to be applied to images in all areas without the need to give the number of segments. However, in studies conducted in various fields such as infrared images [16], medical images [17], microscope Images [15], ultrasound images [30], satellite images [6], etc. are segmented by giving a different number of threshold values. Segmentation for each level is evaluated separately and the appropriate level is tried to be found. In addition to being a very laborious and time-consuming process, it also requires user experience. The proposed adaptive CER algorithm stands out as it has a high accuracy that can be applied in a short time and can be easily applied to images in all areas without exhaustive manual adjustment and evaluations.

The study has been carried out in two parts. Firstly, virtual images with different segment numbers have been created, and the proposed CER algorithm is applied to create virtual images. The reliability of the developed method has been proved primarily with those images. When the proposed CER algorithm is applied to the images, it has been shown that the level numbers and threshold values are found correctly for each virtual image. Considering the proposed method for the created image having the number of the n-segment, the fitness value (Otsu’s max. variance) increases until it reaches the nth level. After reaching the n-level, this value remains constant. In the second part of the study, to validate the efficiency of the proposed method on reel images, extensive experimental studies have been realized on 300 images from the BSDS300 dataset. The results are analyzed by objective and subjective evaluations. The assessments are based on twelve image quality measurement indices and human-made segmentations (ground-truth images). To compare the proposed CER algorithm and five other thresholding methods, namely artificial bee colony (ABC), gravitational search algorithm (GSA), differential evolution (DE), exponential Kbest gravitational search algorithm (eKGSA), and chaotic Kbest gravitational search algorithm (cKGSA), and chaotic Kbest gravitational search algorithm (cKGSA) algorithms are used.

The main contributions of the study can be summarized as follows:

1. An improved Rao algorithm based on chaos theory is developed and is applied the multilevel image thresholding.
2. A simple, less complex, and metaphor-free algorithm is developed.
3. Automatic determination of the number of thresholds for each input image is carried out.
4. Determination of the best chaotic mapping in multilevel image thresholding is handled.

More details about the study are presented in the following sections. The rest of the article is structured as follows. In Section 2, a brief description of gray level co-occurrence matrix (GLCM), Otsu’s maximum variance method for the multilevel thresholding, Rao algorithms, and chaotic maps is presented. In Section 3, we propose a novel adaptive CER algorithm and its details. The application of the proposed method to image segmentation, experimental results, and analysis are presented in Section 4. The concluding remarks are given in Section 5.
2 Preliminaries

This section introduces the basics of the gray level co-occurrence matrix, and the Otsu’s maximum variance method.

2.1 Gray level co-occurrence matrix

A co-occurrence matrix is a matrix that calculates the frequency of pairs of pixels with identical gray values in an image and embeds the distribution of grayscale gradients using edge information. Quadratic statistical properties are extracted using the GLCM matrix. It is calculated based on the relative distance between the pixel pairs and their directions. The GLCM matrix of an image of $mxn$ size is created as a square matrix of $LxL$ dimensions. $L$ shows the number of gray levels in the given image. The $(i, j)$ element of a gray level co-occurrence matrix $M$ denotes the number of pixel pairs at relative position $d$ and $\phi$ orientation where the first pixel is the $i^{th}$ gray level and the second pixel is the gray level $j$, in a linear spatial relation. $G$ is created for each of the four quantized directions or angles (horizontal: $0^\circ$, diagonal: $45^\circ$, vertical: $90^\circ$, and anti-diagonal: $135^\circ$) and distance $d$.

\[
G = \frac{1}{4} \left( M_{d,0^\circ} + M_{d,45^\circ} + M_{d,90^\circ} + M_{d,135^\circ} \right)
\]  

(1)

Then we normalize the GLCM matrix as:

\[
G = \frac{g(i,j)}{\sum_{i=1}^{L}g(i,j)\sum_{j=1}^{L}g(i,j)}
\]  

(2)

2.2 Otsu’s maximum variance method

Otsu’s method, which is an unsupervised, nonparametric and automated method and also known as the inter-class variance method was proposed in 1979 [21]. In this algorithm, the optimal threshold value is obtained by finding the maximum difference between the average gray level of the object (foreground) and the gray level of the background. The image is segmented with the obtained optimal threshold value. For bi-level, the image is divided into two classes with one-threshold value ($T$) as $C_0$ and $C_1$ by fitness function $f:= \sigma_o^2 + \sigma_1^2$. The variances of two classes belonging to foreground and background are given in (3) and (4), respectively.

\[
\sigma_o^2 = w_0(\mu_0-\mu_T)^2
\]  

(3)

\[
\sigma_1^2 = w_1(\mu_1-\mu_T)^2
\]  

(4)

where,

\[
\mu_0 = \sum_{i=1}^{t} \frac{ip_i}{w_0}
\]  

(5)

\[
\mu_1 = \sum_{i=t+1}^{L} \frac{ip_i}{w_1}
\]  

(6)
$w_0$ and $w_1$ are class probabilities for foreground and background and can be expressed by:

$$w_0(t) = P_r(C_0) = \sum_{i=1}^{t} p_i$$  \hspace{1cm} (7)

$$w_1(t) = P_r(C_1) = \sum_{i=t+1}^{L} p_i = 1 - w_0(t)$$ \hspace{1cm} (8)

For multilevel thresholding, the image is divided into $m$ classes as $\{s_1, s_2, \ldots, s_m\}$ by fitness function $f := \sigma^2_0 + \sigma^2_1 + \ldots + \sigma^2_m$. The variances of the $m$ classes are:

$$\sigma^2_0 = w_0 (\mu_0 - \mu_T)^2,$$

$$\sigma^2_1 = w_1 (\mu_1 - \mu_T)^2,$$

$$\ldots,$$

$$\sigma^2_m = w_m (\mu_m - \mu_T)^2.$$

Average levels of the segmented classes and class probabilities are given as follows:

$$\mu_0 = \sum_{i=0}^{t-1} \frac{ip_i}{w_0}, \quad w_0(t) = \sum_{i=0}^{t-1} p_i$$ \hspace{1cm} (9)

$$\mu_1 = \sum_{i=t_1}^{t_2-1} \frac{ip_i}{w_1}, \quad w_1(t) = \sum_{i=t_1}^{t_2} p_i$$ \hspace{1cm} (10)

$$\mu_2 = \sum_{i=t_2}^{t_3-1} \frac{ip_i}{w_2}, \quad w_2(t) = \sum_{i=t_2}^{t_3} p_i$$ \hspace{1cm} (11)

$$\mu_m = \sum_{i=t_m}^{t_{m-1}} \frac{ip_i}{w_m}, \quad w_m(t) = \sum_{i=t_m}^{t_{m-1}} p_i$$ \hspace{1cm} (12)

### 3 Adaptive chaotically-enhanced Rao algorithm for multilevel thresholding

In the proposed chaotically-enhanced Rao algorithm, the chaotic behavior of function has been utilized instead of irregular randomness. The use of chaotic mapping techniques as random number generators in heuristic algorithms is an increasingly common topic. These mapping techniques, which have a wide spectrum, improve the performance by increasing the variety in the random selection of metaheuristic algorithms. Many chaotic maps have been described in the literature showing various types of chaotic behavior. Some of these are: Logistic Map, Sine Map, Sinusoidal Map, Gauss Map, etc. The chaotic maps and mathematical expressions used are given in Table 1.

The proposed CER algorithm utilizes eight different chaotic maps to improve the performance of the Rao algorithm. Rao algorithms consist of three simple and metaphor-free optimization algorithms recently developed by Rao [26, 31]. The steps of the Rao algorithms are given in Algorithm 1.
Algorithm 1 Rao Algorithm

Step 1: Initialize Parameters
- \( f(x) \): Fitness Function
- \( N \): Size of Population
- \( T_{\text{max}} \): Maximum Number of Iterations
- \( \text{MaxGen} \): Maximum Function Evaluations

Step 2: Start the optimization process

1: Calculate fitness function
2: Determine the best and worst solutions
3: Update the new solutions as \( x' \) according to (13)
4: If \( f(x') \) is better than \( f(x) \), then \( x = x' \)
5: If \( f(x') \) is not better than \( f(x) \), then keep the previous solution.
6: If termination criterion is met, then report the solution
7: If not, then goto 2.

It has been seen in the study that the Rao algorithms have obtained very successful results for solving constrained/unconstrained engineering optimization problems. Rao algorithms are very simple and uncomplicated compared to the highly complex methods introduced recently, since these algorithms use only simple control parameters such as iteration number, and size of population which are common to all optimization algorithms and do not require any specific control parameters. The Rao algorithms are based on the interactions between the randomly created candidate solutions and the best/worst solutions determined during the iterations for the problem. Of the three Rao algorithms, Rao-1 algorithm is the simplest and hence the same is used in the present study (and called as Rao algorithm). According to Rao-1 algorithm, the candidate solutions are updated using (13).

\[
x'_{j,k} = x_{j,k} + r_1 (\text{best}_{j,i} - \text{worst}_{j,i}) \tag{13}
\]

Table 1 The Mathematical Formulas of Various Chaotic Maps

| Chaotic Map   | Formula |
|--------------|---------|
| Logistic Map | \( z_{i+1} = \alpha z_i (1 - z_i) \), \( \alpha = 4 \) |
| Sine Map     | \( z_{i+1} = \alpha \sin(\pi z_i) \), \( \alpha = 2.3 \) |
| Sinusoidal Map | \( z_{i+1} = \alpha z_i^2 \sin(\pi z_i) \) |
| Gauss Map    | \( z_{i+1} = \begin{cases} 1 \\ \text{mod}(z_i, 1) \\ 1 \end{cases}, \quad z_i \neq 0 \\ z_i = 0 \) |
| Circle Map   | \( z_{i+1} = z_i + a - \frac{b}{2} \sin(2\pi z_i) \mod(1) \), \( a \in [0, 1] \), \( b \in [0, 4\pi] \) |
| Chebyshev Map| \( z_{i+1} = \cos(\cos^{-1}(z_i)) \) |
| Singer Map   | \( z_{i+1} = \mu(-13,30875z_i^4 - 28,75z_i^3 - 23,31z_i^2 + 7,86z_i), \quad \mu \in [0.9 - 1.08] \) |
| Tent Map     | \( z_{i+1} = \begin{cases} \frac{z_i}{10} \\ \frac{z_i}{3} (1-z_i) \end{cases}, \quad z_i < 0.7 \\ z_i \geq 0.7 \) |
If we examine the integration of chaotic maps to the Rao algorithm, the formulation of the new candidate solutions can be calculated as:

\[ x'_{j,k} = x_{j,k} + z_t (\text{best}_{j,i} - \text{worst}_{j,i}) \]  

(14)

The proposed \( x'_{j,k} \) chaotically changes its values over the iterations. \( z_t \) the parameter represents the chaotic value obtained by chaotic maps. Let’s examine \( z_t \) for the logistic map, which is one of the used chaotic maps.

\[ z_{t+1} = \mu z_t (1-z_t) \]  

(15)

where \( \mu \) is a positive constant and control parameter. \( z_t \) is the chaotic value at \( t^{th} \) iteration. The value of \( \mu \) is chosen as 4 to make the logistic map completely chaotic. The \( z \) variable has better ergodicity at this value. Other chaotic maps’ formulations can be seen in Table 1.

The pseudo-code of the CER algorithm is given in Algorithm 2. The detailed process of the CER algorithm is briefly described as follows.

**Step 1. Initialization** The population is initialized as \( n \). \([ub, lb]\) is denotes the the ranges of the candidate solutions and taken as [1255], maximum number of the iteration is taken as 1000 and is denoted as max_level. The defined termination criterion to determine the number of adaptive threshold levels is expressed in eps. At this step, it is also determined which chaotic map will be used.

**Step 2. Start the optimization process** Along with the he optimization process, the population is initialized in the range of [1255] and the fitness values are calculated for each candidate solution in the initial population.

**Step 3. Selection** The best and worst solutions are determined according to calculated fitness values (Otsu’s Maximum Variance).

**Step 4. Update the population** In this step, the iteration loop is initialized. The candidate solutions are updated according to Eq. (13). The fitness values of the new population are calculated.

**Step 5. Re-Selection** The best and worst solutions of the new population are selected. If the best fitness of the new population is better than the fitness value (fgbest) of the previous best solution, the newly found best fitness (best_fitness) is assigned instead of fgbest and the newly found candidate solution is assigned as the best candidate solution (gbest).

**Step 6. Testing the number of segment level** The error value is calculated by taking the difference between the best variance value in the previous cycle and the optimal variance values in the current cycle. The optimization process breaks if the error value is less than the specified eps value. If not then the process is continued by increasing the number of level.

**Step 7. Update the previous_gbest and previous_fgbest** If the optimal fitness value of the current level is better than the previous value, the current best fitness value is assigned as the global best fitness (fgbest). The best solution of the current level is assigned to the global best solution (gbest). The process is continued until the termination conditions are met.
Algorithm 2 CER Algorithm for Adaptive Multilevel Thresholding

**Algorithm 2. CER Algorithm for Adaptive Multilevel Thresholding**

**Input:** \( p \)

**Output:** level, gbest, fgbest, error

---

**Step 1: Initialization**

1. Determine the initial parameters, \([ub, lb]\), \(max\_iteration\), \(max\_level\), pop, \(eps\)  
2. Define the Chaotic Map

---

**Step 2: Start the optimization process**

3: while level < \(max\_level\)  
4: Initialize the population  
5: Compute the fitness value  

---

**Step 3: Selection**

6: Determine the fgbest & gbest

---

**Step 4: Update the population**

5: for \(i=1:max\_iter\)  
6: Update the solutions  
7: Compute the fitness values of new solutions

---

**Step 5: Re-Selection**

8: Determine the best and worst fitness values  
9: Determine the best and worst values  
10: if \(best\_fitness > fgbest\)  
11: Assign \(best\_fitness\) to fgbest  
12: Assign \(best\) to gbest  
13: end if  
14: end for

---

**Step 6: Testing the number of segment level**

15: Compute the error  
16: if \(error < eps\)  
17: break;  
18: else  
19: increase the number of levels  

---

**Step 7: Update the previous_gbest and previous_fgbest**

20: Assign fgbest to previous_fgbest  
21: Assign gbest to previous_gbest

---

**Step 8: Return the gbest and fgbest**

---

In the proposed CER algorithm, 2d Otsu’s thresholding method is used as the objective function. The descriptions of the algorithm’s parameters and their notations are represented in Table 2.

An image can be defined as \(f(x, y)\) and let the \(m\)-threshold values \(\{T_1^*, T_2^*, ..., T_m^*\}\) divide the image into \(m + 1\) classes as \(S_i\), where \(i = 1, 2, ..., m + 1\) in the multilevel thresholding. The \(S_1\) and \(S_{m+1}\) are the foregrounds and background classes, respectively, and \(S_i = 2, 3, ..., m\) are the intermediate classes. For the multilevel thresholding, the different classes can be described as:

\[
\begin{align*}
S_1 & \in \{0, T_1-1\} \\
S_2 & \in \{T_1, T_2-1\} \\
S_3 & \in \{T_2, T_3-1\} \\
& \vdots \\
S_{m+1} & \in \{T_m, L-1\}
\end{align*}
\]
According to 2d Otsu’s thresholding, optimal thresholding can be calculated as:

\[
\{ T_1^*, T_2^*, \ldots, T_m^* \} = \operatorname{arg\ max}(f_1, f_2, \cdots, f_n)
\]  

(17)

As can be seen in (16), to solve optimal solution set \( \{ T_1^*, T_2^*, \ldots, T_m^* \} \), the fitness values of every candidate solution as \( \text{fitness} := \{ f_1, f_2, \cdots, f_n \} \) and the fitness is maximized according to (17) and \( f \) can be calculated as:

\[
f = \sigma_0^2 + \sigma_1^2 + \cdots + \sigma_m^2
\]  

(18)

The block diagram of the proposed image segmentation method is displayed in Fig. 1. The detailed process of the proposed image segmentation method is briefly described as follows.

**Step 1. Initialization** The parameters of the segmentation method is initialized. Max_iter and the max_level represent the number of the maximum iteration and the maximum level, respectively. Pop denotes the candidate optimal thresholds. Ub and lb. is upper bound and the lower bound of the threshold values, respectively. RUNS determines how many times the algorithm will run. Eps is the value of the termination condition for the error value.

**Step 2. Choose one of the chaotic maps given in Table 1** The proposed CER algorithm utilizes eight different chaotic maps to improve the performance of the Rao algorithm. In this step, one of the chaotic maps is selected.

**Step 3. Read the original image** The images in the dataset are read sequentially. The median filter is applied to the read image and resized. A few sample images selected from the mentioned dataset are given in the flowchart of the method given in Fig. 1.

**Step 4. Compute the histogram of the given image** The 2D histogram of the read image is constructed with GLCM. The 2D histograms of the selected images are also illustrated in the flowchart.

**Step 5. Go to the CER algorithm** Constructed histogram is sent to the CER algorithm. In the beginning, the candidate solutions are initialized in the CER algorithm. The candidate solutions and the constructed 2D histogram belonging to the read image are sent to Otsu’s method. The variances of the candidate solutions are calculated according to Otsu’s thresholding method. In the population, the solution with the maximum variance is selected as the optimal threshold for the read image. In addition to the optimal thresholds, the number of the optimal level is determined in the CER algorithm.

**Step 6. Segment image according to the optimal thresholding value** The images are segmented using obtained optimal thresholds. The segmented versions of the selected five images can be

| Table 2 Description of the algorithm parameters and their notations |
|---------------------------------------------------------------|
| Description                           | Notation      | Value   |
|---------------------------------------|---------------|---------|
| Maximum Number Of Iterations          | max_iteration | 1000    |
| Lower Band                            | Lb            | 1       |
| Upper Band                            | Ub            | 255     |
| Population Size                       | Pop           | 20      |
| Minimum Value Of Error                | Eps           | 0       |
| Maximum Number Of Thresholding Level  | max_level     | 50      |
| Control parameter (chaotic maps)      | \( \alpha \)  |         |
| Histogram Constructed By GLCM         | P             |         |
| Number Of Thresholding Level          | Level         |         |
| Best Solution In The Whole Population | global_best_sol |       |
| The Fitness Value Of The Best Solution| global_best_fitness |     |
Step 1. Initialization
(max_iter, max_level, pop, ub, lb, RUNS, eps)

Step 2: Choose one of the Chaotic Maps given in Table 1.
(Logistic Map, Sine Map, Sinusoidal Map, Gauss Map, Tent Map,
Chebyshev Map, Circle Map, Singer Map)

Step 3: Read the original image
(A few sample images taken from BSDS300 dataset are given below)

Step 4: Compute the histogram of given image

Step 5: Go to CER Algorithm
(Given in Algorithm 2)

Step 6: Segment image according to optimal thresholding values

Step 7: Evaluate segmented image with measuring metrics
(BDE, PRI, GCE, VOI, SSIM, FSIM, PSNR, RMSE, NAE, CC, AD, MD)

Fig. 1 Flowchart of the proposed segmentation method

seen in the flowchart. The segmentation algorithm was applied to 300 images one by one, and the measurement results were given by averaging the 300 images.
Step 7. Evaluate segmented image with measuring metrics In the dataset, each image is segmented by five different subjects on average (ground-truth images). All measuring metrics are calculated by comparing the segmented images and the ground truth images. Each segmented image is compared with the five different subjects one by one, and the results are obtained by averaging the five different measurement values.

The initial parameters required for the proposed method in the algorithm are decided. These parameter values and their descriptions are given in Table 2. Here, the maximum number of iterations is taken as 1000. The maximum threshold level number selected for the algorithm that finds the number of threshold levels adaptively is determined as 50. The number of candidate solutions is taken as 20 for each algorithm. The lower and upper limit values of the candidate solutions are given as 1 and 255, respectively. For all methods applied, the same number of maximum iterations, maximum threshold levels, the lower and upper limit values of the candidate solutions in the population, and the number of candidate solutions are used.

The Rao algorithm, which uses simple optimization techniques, is used to find the optimal threshold values. Since the chaotic theory is more ergonomic and advantageous in optimization problems than irregular randomness, the Chaotically-Enhanced Rao algorithm is developed to find optimal values in multi-level thresholding. There are many suggested chaotic maps in the literature, the most common of which have been emphasized.

To determine which chaotic map is more successful in the multi-level image thresholding, eight different chaotic maps, whose formulas are given in Table 2, are applied to the proposed method to improve the performance of the Rao algorithm in image segmentation. For this purpose, as stated before, Logistic, Sine, Sinusoidal, Gauss, Circle, Chebyshev, Singer, and Tent chaotic mapping methods were used. The $z_t$ value in the candidate solutions update equation given in (14) is determined according to the equation of the selected chaotic map and the new values of the candidate solutions updated chaotically. After all the initial parameters and selections are made, the images selected from the BSDS300 dataset are read sequentially, and the 2d histograms are obtained by the GLCM method. The threshold number and thresholds values of each image are obtained with the proposed CER algorithm. The images are divided into segments according to the threshold number and threshold values. The ground truth images made by different people in the BSDS300 data set are compared with the segmented images.

Ground truth images of a few sample images selected randomly from the data set are given in Fig. 2. The dataset contains 300 original images and each image is segmented by five different subjects on average. All metrics are calculated by using these ground-truth images.

It can be seen in Fig. 2, segmentation is made by five different subjects on average using different segment numbers and values for each image. Segmentation performance is evaluated separately for each image with 12 indices, which are most widely used to measure image segmentation performance. The advantages of chaotic maps among each other and the performance comparisons of the proposed multi-level image segmentation method with DE, ABC, GSA, cGSA, and eKGSA methods are given in the experimental analysis and results section.

4 Experimental analysis and results

The experiments are conducted using MATLAB R2020a on Intel core-i5 (9th Gen.) CPU with 16 GB RAM under the Windows 10 environment. Berkeley-Benchmark Segmentation Dataset is used to illustrate the effectiveness of the proposed method. The thresholding level is
determined adaptively with the proposed CER algorithm. The sample images from the BSDS300 dataset are resized to $320 \times 240$. Some sample images taken from the BSDS300 dataset and their histograms (1d & 2d histograms) along with the obtained threshold values by the proposed segmentation method are illustrated in Fig. 3.

To demonstrate the accuracy of the proposed method, the method is first applied to virtual created images. In Fig. 4, a virtual-gray image with a segment number of five and its histograms are given. The created virtual image with five-level and its 1d & 2d histograms are given in Fig. 4. Optimal values obtained when the suggested adaptive multilevel thresholding method is applied to the given image are shown in the 1d histogram in Fig. 4b. The proposed multilevel thresholding method for the 1–25 number of levels are applied to the given virtual-gray image and the fitness values are observed in Fig. 5. It is concluded that as the number of levels increases, the fitness value constantly

| Test Images | Ground-Truth Images |
|-------------|---------------------|
| ![Image 1](image1.png) | ![Ground-Truth Image 1](groundtruth1.png) |
| ![Image 2](image2.png) | ![Ground-Truth Image 2](groundtruth2.png) |
| ![Image 3](image3.png) | ![Ground-Truth Image 3](groundtruth3.png) |
| ![Image 4](image4.png) | ![Ground-Truth Image 4](groundtruth4.png) |
| ![Image 5](image5.png) | ![Ground-Truth Image 5](groundtruth5.png) |

**Fig. 2** Ground truth images of the five test images
increases until it reaches the 5-level, and remains constant after this value. The error graph of the fitness values is also given in Fig. 5b.

Besides, the proposed method is applied to the 3-level and 4-level artificial images given in Fig. 6. The threshold values obtained with the CER algorithm are visualized with 1d-

| Test Images | 1D Histograms | 2D Histograms |
|-------------|-------------|-------------|
| ![Image](image1.png) | ![Histogram1](histogram1.png) | ![2D Histogram1](2d_histogram1.png) |
| ![Image](image2.png) | ![Histogram2](histogram2.png) | ![2D Histogram2](2d_histogram2.png) |
| ![Image](image3.png) | ![Histogram3](histogram3.png) | ![2D Histogram3](2d_histogram3.png) |
| ![Image](image4.png) | ![Histogram4](histogram4.png) | ![2D Histogram4](2d_histogram4.png) |

Fig. 3 Original images and their histograms. In a 2d-histogram, the x-axis & y-axis correspond to the pixel values, and the z-axis corresponds to the frequency of each level of intensity. In the 1d-histogram & 2d-histogram, the range of pixel values is from 0 to 255.
histograms. To examine the performance of the proposed method on real images, the method is applied to the images in the BSDS300 dataset.

The fitness values, computational times, and the optimal threshold values obtained when 1–10 thresholding level is applied to 5 sample images selected from the specified data set are given in Table 4. The fitness values obtained are given in Fig. 6 with graphics for a better understanding. When Table 4 and Fig. 6 were examined, it can be seen that the fitness value increased until the actual segment number in real images as well as in virtual images. When the number of segments is increased further, it is seen in Fig. 7 that the fitness value remains constant or decreases.

The proposed multilevel thresholding method was applied to 300 images in the BSDS300 dataset to compare the performances of 8 chaotic maps applied to the Rao algorithm. In each method, the segmentation method, which finds the adaptive threshold level based on Otsu’s thresholding method, is applied. Image histograms are constructed by GLCM. For a more reliable evaluation, the well-known performance metrics of image segmentation such as Boundary Displacement Error (BDE) [29], Variance of Information (VOI) [29], Global Consistency Error (GCE) [29], Probability Rand Index (PRI), peak signal to noise ratio (PSNR) [29], Structural Similarity (SSIM) [40], Feature Similarity (FSIM) [46], Root Mean Squared Error (RMSE) [38], Cross-Correlation (CC) [35, 38], Average difference (AD) [38], Maximum Difference (MD) [38], Normalized Absolute Error (NAE) [18] are calculated.

BDE calculates the average error of displacement of boundary pixels between the segmented images. It is defined as the distance between the pixel on the other image boundary and the closest pixel. A lower BDE value is preferred for better segmentation.

GCE calculates the extent to which one segmentation can be observed as an improvement of the other. The GCE index is in the range of 0–1. The lower the value, the better the segmentation. VOI

![Virtual Image and its histograms](image)

**Fig. 4** Virtual Image and its histograms a Original Image b 1d-histogram c 2d-histogram with threshold values obtained with the proposed method
Fig. 5 Fitness Values (a) and Error (b) Graphics Belong to Created Virtual Image with 5 number of segment level

Fig. 6 Created Virtual Gray Scale Images with 3–4 number of segment level and their 1d-histograms with threshold values
calculates the amount of randomness in a segmented image by defining the distance between two segmentations in terms of average conditional entropy.

PRI is an indicator of similarities between two regions in an image. This index ranges between 0 and 1. It should be higher for better segmentation. The SSIM is an indicator that measures the similarities between the original image and the segmented image. A higher SSIM value means better performance. Feature similarity index (FSIM) is an index that evaluates the similarities between the original image and the segmented image.

PSNR is used to compare images with different dynamic ranges. A higher PSNR value indicates a better thresholding performance. As the threshold number increases, the CC value increases. A higher CC value indicates a better thresholding performance. AD is the average of the difference between the reference image and the segmented image. Low AD value is preferred for better segmentation.

MD takes the maximum error between the two images. A smaller MD value indicates better segmentation. NAE calculates the normalized absolute difference between two images. Low NAE value is preferred for good segmentation.

Estimated values are found for 300 segmented images, and the values in Table 3 are obtained by taking the average values of all images with adaptive threshold levels, and the best results are indicated in bold. The values obtained in Table 3 aim to show both the

![Fig. 7](image-url) a Sample images selected from BSDS300, b their fitness values graphs, and c CPU Times Graphs
|                  | Original Rao [26] | Rao with Logistic | Rao with Sine | Rao with Sinusoidal | Rao with Gauss | Rao with Circle | Rao with Chebyshev | Rao with Singer | Rao with Tent |
|------------------|------------------|-------------------|--------------|--------------------|---------------|----------------|-------------------|----------------|--------------|
| BDE [29]         | 10.6399          | 10.6961           | 10.678       | 10.661             | 10.6366       | 10.5747        | 10.7019           | 10.4845        | 10.6666      |
| PRI [29]         | 0.6571           | 0.6573            | 0.6553       | **0.6582**         | 0.6563        | 0.6525         | 0.6572            | 0.6504         | 0.6573       |
| VOI [29]         | 3.2278           | 3.2325            | 3.2184       | 3.2397             | **3.1709**    | 3.0750         | 3.2499            | **3.0694**     | 3.2176       |
| GCE [29]         | 0.4489           | 0.4491            | 0.4451       | 0.4507             | 0.4438        | 0.4305         | 0.4523            | 0.4262         | 0.4479       |
| SSIM [40]        | 0.5754           | 0.5759            | 0.5730       | 0.5771             | 0.5674        | 0.5520         | **0.5776**        | 0.5468         | 0.5719       |
| FSIM [46]        | 0.7195           | 0.7194            | 0.7175       | 0.7201             | 0.7156        | 0.7073         | **0.7206**        | 0.7076         | 0.7171       |
| RMSE [38]        | 37.2159          | 37.1993           | 37.3643      | 37.1183            | 37.8773       | 38.9668        | **37.0342**       | 39.3624         | 37.5154      |
| PSNR [29]        | 17.1129          | 17.1202           | 17.0719      | 17.1459            | 16.9411       | 16.6611        | **17.1568**       | 16.5686         | 17.0284      |
| AD [38]          | 28.9327          | 28.9031           | 29.0916      | 28.8071            | 29.8157       | 31.2260        | **28.6930**       | 31.6779         | 29.1868      |
| CC [38]          | 1.6665           | 1.6643            | 1.6721       | 1.6635             | 1.6941        | 1.7422         | **1.6563**        | 1.7559         | 1.6758       |
| MD [38]          | 113.330          | **113.170**       | 113.740      | 113.350             | 114.460       | 115.050        | 113.4300          | 115.5900         | 114.0400     |
| NAE [38]         | 0.2774           | 0.2767            | 0.2789       | 0.2757             | 0.2857        | 0.2997         | 0.2748            | **0.3045**     | 0.2798       |
superiority of chaotic maps and the performance of the proposed adaptive method. For each method, the number of maximum iterations is taken as 1000. The population size is taken as 20, with 30 runs per algorithm. Other parameters are given in Table 2. From the statistical analysis considering the average BDE, PRI, VOI, GCE, SSIM, FSIM, RMSE, PSNR, NK, AD, SC, MD, and NAE values of the 300 test images, it is found that chaos-based Rao algorithms are better in obtaining the optimal threshold values. Besides, it is found that Chebyshev Map ranked first among the other seven chaotic maps in obtaining the optimal threshold values.

The level numbers and fitness values, and optimal solutions obtained for 5 sample images with adaptive segmentation methods are given in Table 4. As seen in Table 4, optimal threshold solution was selected according to the fitness values. As the highest fitness value showed the best solution, the threshold values according to the highest fitness value are used in segmentation process. This situation was also indicated in Fig. 7.

The computational times of the simple Rao algorithm and eight different chaotic maps applied to the Rao algorithm are given comparatively in Fig. 8. The computational times of the chaotically enhanced Rao algorithm with Chebyshev map and the other five different maps are compared. The comparative results are given in Fig. 9. It can be seen in Fig. 9, the Chebyshev-Rao algorithm is generally converging to the optimal point in a shorter time than the other algorithm. In Figs. 8 and 9, ten test images are selected randomly from the dataset.

The statistical tests are essential to show the performances of the optimization algorithms [14, 45]. For this purpose, a set of a statistical tests is conducted. The results of the proposed CER algorithm on a few sample images are summarized in Table 5. It includes best, worst, mean, median, and standard deviations.

It is shown in Fig. 8 that the Circle and Singer chaotic maps are worse than the others. The Sinusoidal, and, Sine maps are applied to the Rao algorithm, it is seen that the algorithms generally converge to the optimal point in less time than the simple Rao algorithm. However, the Chebyshev-Rao and the Logistic-Rao are generally converging to the optimal point in the shortest time. Gauss and Tent maps have competitive results with the simple Rao algorithm.

It is essential to compare the proposed method with other optimization methods to show the effectiveness of the proposed CER algorithm. These comparisons can be seen in Table 6. As seen in Table 6, different optimization techniques such as DE, ABC, and GSA, which have different strategies and formulations, are used to solve the multilevel thresholding problem. These techniques are inspired by biology and physics. To evaluate the chaotic behavior of the proposed method, chaos-enhanced Kbest GSA (cKGSA) [19] is also applied to solve the problem of multilevel thresholding.

From the statistical analysis (see in Table 6) considering the average PRI, SSIM, FSIM, RMSE, PSNR, AD, and NAE values of the 300 test images, it is found that the proposed CER algorithm produces better threshold values where higher PRI, SSIM, FSIM, and PSNR scores are obtained than the compared approaches. Similarly, lower RMSE, AD, and NAE scores that show better performance are obtained by the proposed method. The segmented images with 3-level and 5-level segmentations are given in Fig. 11 for comparison. As seen in Fig. 10, we used the JET colormap array for coloring the segments [11] (Fig. 11).

In order to investigate the complexity of the proposed adaptive CER algorithm, the computational times for the algorithm with varying size of the input is given in Table 7.
Table 4. The best values for five sample images obtained by the proposed approach with different chaotic maps

| Algorithm     | Level | Fitness Value \((\times 10^3)\) | Optimal Solution | Algorithm     | Level | Fitness Value \((\times 10^3)\) | Optimal Solution |
|---------------|------|----------------------------------|------------------|---------------|------|----------------------------------|------------------|
| Test          | 5    | 3.7272                           | 61 86 99,124,176 | Test          | 5    | 6.3938                           | 84,120,135,148,171 |
| Image 1       |      |                                  |                  | Image 2       |      |                                  |                  |
| Logistic Map  | 6    | 3.7589                           | 43 76 94,102,123,174 | Logistic Map  | 5    | 6.3938                           | 84,120,135,148,171 |
| Sine Map      | 6    | 3.7589                           | 43 76 94,102,123,174 | Sine Map      | 5    | 6.3938                           | 84,120,135,148,171 |
| Sinusoidal Map| 6    | 3.7589                           | 43 76 92,102,124,174 | Sinusoidal Map| 5    | 6.3938                           | 84,120,135,148,171 |
| Gauss Map     | 5    | 3.7276                           | 41 76 92,122,174  | Gauss Map     | 4    | 6.3897                           | 84,124,147,171   |
| Circle Map    | 4    | 3.7029                           | 60 93,119,172     | Circle Map    | 5    | 6.3858                           | 90,127,147,173,255 |
| Chebyshev Map | 6    | 3.7589                           | 43 61 86,99,124,176 | Chebyshev Map | 7    | 6.3928                           | 83,116,130,144,149,173 |
| Singer Map    | 3    | 3.6546                           | 62 92,132         | Singer Map    | 5    | 6.3692                           | 83,111,149,174,255 |
| Tent Map      | 6    | 3.7589                           | 43 76 94,102,123,174 | Tent Map      | 5    | 6.3938                           | 84,120,135,148,171 |
| Test          | 5    | 1.1989                           | 38 90,114,127,172 | Test          | 6    | 5.8503                           | 64 85,94,103,128,198 |
| Image 3       |      |                                  |                  | Image 4       |      |                                  |                  |
| Logistic Map  | 6    | 1.1990                           | 37 88,112,119,129,172 | Logistic Map  | 5    | 5.8494                           | 66 86,102,128,198 |
| Sine Map      | 6    | 1.1990                           | 37 88,112,119,129,172 | Sine Map      | 6    | 5.8503                           | 64 85,94,103,128,198 |
| Sinusoidal Map| 6    | 1.1990                           | 37 88,112,119,129,172 | Sinusoidal Map| 4    | 5.8385                           | 66 94,128,198    |
| Gauss Map     | 5    | 1.1985                           | 38 92,114,123,169 | Gauss Map     | 5    | 5.8494                           | 66 86,102,128,198 |
| Circle Map    | 4    | 1.1934                           | 26 84,126,173     | Circle Map    | 4    | 5.8184                           | 63 97,121,200    |
| Chebyshev Map | 5    | 1.1989                           | 38 90,114,127,172 | Chebyshev Map | 5    | 5.8494                           | 66 94,128,198    |
| Singer Map    | 4    | 1.1940                           | 59 96,126,167     | Singer Map    | 5    | 5.8249                           | 2 70,101,127,203 |
| Tent Map      | 6    | 1.1990                           | 37 88,112,119,129,172 | Tent Map      | 6    | 5.8503                           | 64 85,94,103,128,198 |
| Test          | 5    | 9.9925                           | 101,130,145,154,166,199 | Test          | Circle Map  | 4    | 9.9454                           | 97,133,142,201   |
| Image 5       |      |                                  |                  | Image 5       |      |                                  |                  |
| Logistic Map  | 6    | 9.9925                           | 101,130,145,154,166,199 | Logistic Map  | 6    | 9.9925                           | 101,130,146,160,167,200 |
| Sine Map      | 6    | 9.9925                           | 101,130,145,154,166,199 | Sine Map      | 4    | 9.9695                           | 101,130,148,204  |
| Sinusoidal Map| 6    | 9.9925                           | 101,130,145,154,166,199 | Sinusoidal Map| 4    | 9.9695                           | 101,130,145,154,166,199 |
| Gauss Map     | 6    | 9.9907                           | 100,129,144,157,168,200 | Gauss Map     | 6    | 9.9925                           | 101,130,145,154,166,199 |
Table 5 The statistical results for CER algorithm based Chebysev map

| Test Image | Best     | Worst    | Mean     | Median    | STD      |
|------------|----------|----------|----------|-----------|----------|
| Test Image 1 | 3.7389e+03 | 2.8086e+03 | 2.9010e+03 | 2.8086e+03 | 0.2923e+03 |
| Test Image 2 | 6.3928e+03 | 5.7183e+03 | 5.7856e+03 | 5.7183e+03 | 0.2127e+03 |
| Test Image 3 | 1.1989e+03 | 1.0823e+03 | 1.0939e+03 | 1.0823e+03 | 0.0365e+03 |
| Test Image 4 | 5.8494e+03 | 4.9979e+03 | 5.0822e+03 | 4.9979e+03 | 0.2666e+03 |
| Test Image 5 | 9.9925e+03 | 8.6565e+03 | 8.7891e+03 | 8.6565e+03 | 0.4195e+03 |
| Test Image 6 | 6.1357e+03 | 5.0642e+03 | 5.1714e+03 | 5.0642e+03 | 0.3388e+03 |
| Test Image 7 | 2.8420e+03 | 1.8297e+03 | 1.9309e+03 | 1.8297e+03 | 0.3201e+03 |
| Test Image 8 | 2.7513e+03 | 1.9558e+03 | 2.0354e+03 | 1.9558e+03 | 0.2515e+03 |
| Test Image 9 | 1.3200e+03 | 1.0066e+03 | 1.0379e+03 | 1.0066e+03 | 0.0991e+03 |
| Test Image 10 | 1.3324e+03 | 0.8192e+03 | 0.8705e+03 | 0.8192e+03 | 0.1623e+03 |
According to Table 7, the suggested adaptive CER algorithm’s time is:

$$t(n) = 2n + 6$$

for input of size $n$. Then, we can say that:

$$t(n) = n + 6/2$$

In light of these assumptions, we can say that as $n$ increases, $t(n)$ grows like $n$ or $t(n) = O(n)$. The graph of the time complexity belonging to the proposed adaptive CER algorithm is given in Fig. 10.

![Graph of time complexity](image_url)
5 Conclusions

In this study, a multilevel image thresholding algorithm that can adaptively adjust the segment number of images is proposed. The multilevel thresholding process is considered an optimization problem, and Otsu’s between-class variance is used as the objective function. The proposed method has been developed based on a recently published Rao algorithm which is a simple and powerful metaphor-less algorithm. The proposed method is called as chaotically-enhanced Rao (CER) algorithm. In the proposed algorithm, chaotic behavior has been utilized instead of irregular randomness. It has fast convergence to the target point and a more precise solution. The main advantage of the CER algorithm is that it requires only the common control parameters such as the maximum number of iterations and population size and there is no need to tune any algorithm-specific parameters.
Unlike many studies, the adaptive CER algorithm determines the number of segments for each image. The number of segments for each image is unique, instead of segmenting all images in the dataset by a specified fixed segment number, the segment number for each image in the dataset is automatically found, and images are segmented with the proposed method.

The proposed method is primarily applied to the created virtual images. In this case, when the results obtained are examined, it is seen that each image has a unique threshold level number, and it is extremely difficult to adjust this manually. With the adaptive CER algorithm, it is observed that the threshold level and fitness value are directly proportional, and the fitness value increases as the number of levels increases, but when it come to the number of segments specific to the image, the fitness value remains constant or decreases. With the developed algorithm in this framework, segmentation performance has improved, and the threshold level for each image has been found effectively and accurately. The proposed method is also applied to the original images in the BSDS300 data set. To evaluate the segmented images, well-known metrics, SSIM, FSIM, PSNR, RMSE, BDE, GCE, NAE, MD, VOI, PRI, CC, AD are utilized. Compared with the well-established and recently published image thresholding techniques, it is shown with 12 different quality measurement indices that the proposed method has obtained more effective results than the other methods.

Future studies aim to develop adaptive methods to provide more precise segmentation with more optimum threshold values for images by expanding studies on different methods such as energy-based thresholding methods [34], fuzzy entropy [20, 25], diagonal class entropy [3]. In addition to these, the proposed method and future methods will be tested for different application areas such as satellite images, medical images, nature images, and real-time images.

**Declarations**

**Conflict of interest** The authors declare that they have no conflicts of interest.

| n   | Time (sec) |
|-----|------------|
| 1   | 26,908     |
| 2   | 47,777     |
| 3   | 69,669     |
| 4   | 92,274     |
| 5   | 11,5945    |
| 6   | 13,8024    |
| 7   | 16,1839    |
| 8   | 18,5731    |
| 9   | 21,0005    |
| 10  | 23,1309    |
| 11  | 25,5399    |
| 12  | 27,8315    |
| 13  | 30,3581    |
| 14  | 32,5964    |
| 15  | 34,9095    |
| 16  | 37,1114    |
| 17  | 39,5955    |
| 18  | 42,1488    |
| 19  | 44,0502    |
| 20  | 47,0547    |

Table 7 The computational times for the algorithm with size of the input
References

1. Abdel-Basset M, Mohamed R, AbdelAziz NM, Abouhawwash M (2022) HWOA: a hybrid whale optimization algorithm with a novel local minima avoidance method for multi-level thresholding color image segmentation. Expert Syst Appl 190:116145. https://doi.org/10.1016/j.eswa.2021.116145
2. Achanta R, Estrada F, Wils P, Süsstrunk S (2008) Salient region detection and segmentation. In: Computer vision systems. Springer Berlin Heidelberg, Berlin, pp 66–75
3. Agrawal S, Panda R, Abraham A (2020) A novel diagonal class entropy-based multilevel image thresholding using coral reef optimization. IEEE Trans Syst Man Cybern Syst 50:4688–4696. https://doi.org/10.1109/TSMC.2018.2859429
4. Bao X, Jia H, Lang C (2019) A novel hybrid Harris hawks optimization for color image multilevel thresholding segmentation. IEEE Access 7:76529–76546
5. Ben IA (2017) A two-dimensional multilevel thresholding method for image segmentation. Appl Soft Comput 52:306–322
6. Bhandari AK, Singh VK, Kumar A, Singh GK (2014) Cuckoo search algorithm and wind driven optimization based study of satellite image segmentation for multilevel thresholding using Kapur’s entropy. Expert Syst Appl 41:3538–3560
7. Feoktistov V (2006) Differential Evolution. In: Differential evolution: in search of solutions. Springer US, Boston, pp 1–24
8. Grzywinski M, Atmaca B, Dede T, Venkata Rao R (2020) The size optimization of steel braced barrel vault structure by using Rao-1 algorithm. Sigma J Eng Nat Sci 38:1415–1425
9. Hosny M, Kamel S, El-Dabah M et al (2021) Optimal reactive power dispatch with time-varying demand and renewable energy uncertainty using Rao-3 algorithm. IEEE Access PP:1–1. https://doi.org/10.1109/ACCESS.2021.3056423
10. Houssein EH, Helmy BE, Oliva D, Jangir P, Premkumar M, Elngar AA, Shaban H (2022) An efficient multi-thresholding based COVID-19 CT images segmentation approach using an improved equilibrium optimizer. Biomed Signal Process Control 73:103401. https://doi.org/10.1016/j.bspc.2021.103401
11. Jet colormap array - MATLAB jet. https://www.mathworks.com/help/matlab/ref/jet.html. Accessed 31 Dec 2021
12. Jia H, Ma J, Song W (2019) Multilevel thresholding segmentation for color image using modified moth-flame optimization. IEEE Access 7:44097–44134
13. Karaboga D, Basturk B (2007) Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems. In: Foundations of fuzzy logic and soft computing. Springer, Berlin, Heidelberg, pp 789–798
14. Li M-W, Wang Y-T, Geng J, Hong W-C (2021) Chaos cloud quantum bat hybrid optimization algorithm. Nonlinear Dyn 103:1167–1193. https://doi.org/10.1007/s11071-020-06111-6
15. Luo TL, Eisenberg MC, Hayashi MAL, Gonzalez-Cabezas C, Foxman B, Marrs CF, Rickard AH (2018) A sensitive thresholding method for confocal laser scanning microscope image stacks of microbial biofilms. Sci Rep 8:13013. https://doi.org/10.1038/s41598-018-31012-5
16. Manda MP, Kim HS (2020) A fast image thresholding algorithm for infrared images based on histogram approximation and circuit theory. Algorithms 13:207. https://doi.org/10.3390/a1300207
17. Maoolood IY, Al-Salhi YEA, Lu S (2018) Thresholding for medical image segmentation for Cancer using fuzzy entropy with level set algorithm. Open Med (Wars) 13:374–383. https://doi.org/10.1515/med-2018-0056
18. Mittal H, Saraswat M (2018) An optimum multi-level image thresholding segmentation using non-local means 2D histogram and exponential Kbest gravitational search algorithm. Eng Appl Artif Intell 71:226–235
19. Mittal H, Pal R, Kulhari A, Saraswat M (2016) Chaotic Kbest gravitational search algorithm (CKGSA). In: 2016 ninth international conference on contemporary computing. Noida, India, pp 1–6
20. Naji Alwerfali HS, Al-qaness AA, Abd Elaziz M et al (2020) Multi-level image thresholding based on modified spherical search optimizer and fuzzy entropy. Entropy 22:328. https://doi.org/10.3390/e22030328
21. Nobuyuki O (1979) A threshold selection method from gray-level histograms. IEEE Trans Syst Man Cybern 9:62–66
22. Pare S, Kumar A, Singh GK (2017) Color multi-level thresholding using grey-level co-occurrence matrix and differential evolution algorithm. In: 2017 international conference on communication and signal processing (ICCPSP), pp 0096–0100
23. Pare S, Bhandari AK, Kumar A, Singh GK (2018) A new technique for multilevel color image thresholding based on modified fuzzy entropy and Lévy flight firefly algorithm. Comput Electr Eng 70:476–495
24. Pare S, Kumar A, Bajaj V, Singh GK (2019) A context sensitive multilevel thresholding using swarm based algorithms. IEEE/CIA A J Autom Sin 6:1471–1486. https://doi.org/10.1109/JAS.2017.7510697
25. Raj A, Gautam G, Sheikh Abdullah S et al (2019) Multi-level thresholding based on differential evolution and Tsallis Fuzzy entropy. Image Vis Comput 91. https://doi.org/10.1016/j.imavis.2019.07.004
26. Rao RV (2020) Rao algorithms: three metaphor-less simple algorithms for solving optimization problems. Int J Ind Eng Comput 11:107–130. https://doi.org/10.5267/j.ijieec.2019.6.002
27. Rashedi E, Nezamabadi-pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179:2232–2248
28. Ravipudi J (2020) Synthesis of linear, planar, and concentric circular antenna arrays using Rao algorithms. Int J Appl Evol Comput 11:31–49. https://doi.org/10.4018/IJAEC.2020070103
29. Sathyar B, Manavalan R (2011) Image segmentation by clustering methods: performance analysis. Int J Comput Appl 29:27–32. https://doi.org/10.5120/3688-5127
30. Shao D, Xu C, Xiang Y, Gui P, Zhu X, Zhang C, Yu Z (2019) Ultrasound image segmentation with multilevel threshold based on differential search algorithm. IET Image Process 13:998–1005
31. Sharma S, Singh B, Aneja M (2021) Classification of Parkinson disease using binary Rao optimization algorithms. Expert Syst. https://doi.org/10.1111/exsy.12674
32. Shen L, Huang X, Fan C (2018) Double-group particle swarm optimization and its application in remote sensing image segmentation. Sensors 18:1393. https://doi.org/10.3390/s18051393
33. Sörensen K (2015) Metaheuristics—the metaphor exposed. Int Trans Oper Res 22:3–18. https://doi.org/10.1111/itor.12001
34. Srikanth R, Bikshalu K (2021) Multilevel thresholding image segmentation based on energy curve with harmony search algorithm. Ain Shams Eng J 12:1–20. https://doi.org/10.1016/j.asej.2020.09.003
35. Srinivasan PN, Norwawi N, Amiripalli SS, Deepalakshmi P (2021) Secured compression for 2D medical images through the manifold and fuzzy trapezoidal correlation function. Gazi Univ J Sci. https://doi.org/10.35378/gujs.884880
36. Swain M, Tripathy TT, Panda R, Agrawal S, Abraham A (2022) Differential exponential entropy-based multilevel threshold selection methodology for colour satellite images using equilibrium-cuckoo search optimizer. Eng Appl Artif Intell 109:105459. https://doi.org/10.1016/j.engappai.2021.105459
37. Tuba M (2014) Multilevel image thresholding by nature-inspired algorithms: a short review. Comput Sci J Mold 22:318–338
38. Varnan SC, Jagan A, Kaur J, Jyoti D, Rao DS (2011) Image quality assessment techniques pn spatial domain. Int J Comput Sci Technol 2:177–184
39. Venkata Rao R, Keesari H (2020) Rao algorithms for multi-objective optimization of selected thermodynamic cycles. Eng Comput 37:3409–3437. https://doi.org/10.1007/s00366-020-01008-9
40. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13:600–612
41. Wang L, Wang Z, Liang H, Huang C (2019) Parameter estimation of photovoltaic cell model with Rao-I algorithm. Optik 210:163846. https://doi.org/10.1016/j.ijleo.2019.163846
42. Wuinava A, Kumar Naik M, Panda R, Jena B, Abraham A (2020) A differential evolutionary adaptive Harris hawks optimization for two dimensional practical Masi entropy-based multilevel image thresholding. J King Saud Univ – Comput Inf Sci 34:3011–3024. https://doi.org/10.1016/j.jksuci.2020.05.001
43. Xing Z (2020) An improved emperor penguin optimization based multilevel thresholding for color image segmentation. Knowl-Based Syst 194:105570. https://doi.org/10.1016/j.knosys.2020.105570
44. Yue X, Zhang H (2020) Modified hybrid bat algorithm with genetic crossover operation and smart inertia weight for multilevel image segmentation. Appl Soft Comput 90:106157. https://doi.org/10.1016/j.asoc.2020.106157
45. Zhang Z, Hong W-C (2021) Application of variational mode decomposition and chaotic grey wolf optimizer with support vector regression for forecasting electric loads. Know-Based Syst 228:107297. https://doi.org/10.1016/j.knosys.2021.107297
46. Zhang L, Zhang L, Mou X, Zhang D (2011) FSIM: A feature similarity index for image quality assessment. IEEE Trans Image Process 20:2378–2386
47. Zhaoa X, Turk M, Li W et al (2016) A multilevel image thresholding segmentation algorithm based on two-dimensional K–L divergence and modified particle swarm optimization. Appl Soft Comput 48:151–159

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

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.
Affiliations

Yagmur Olmez 1 · Abdulkadir Sengur 2 · Gonca Ozmen Koca 1 · Ravipudi Venkata Rao 3

Abdulkadir Sengur
ksengur@gmail.com

Gonca Ozmen Koca
gonca.ozmen@gmail.com

Ravipudi Venkata Rao
ravipudirao@gmail.com

1 Department of Mechatronics Engineering, Faculty of Technology, University of Firat, 23119 Elazig, Turkey
2 Department of Electrical and Electronics Engineering, Faculty of Technology, University of Firat, 23119 Elazig, Turkey
3 Department of Mechanical Engineering, Sardar Vallabhbhai National Institute of Technology, Surat, Gujarat 395007, India