

# Ensemble Searching: A New Concept of Heuristic Search Algorithms and Its Application in Multilevel Thresholding Optimization

Ehsan Ehsaeyan\*

Electrical Engineering Department, Sirjan University of Technology, Sirjan, Iran; ehsaeyan@sirjantech.ac.ir

## ABSTRACT

Multilevel thresholding is recognized as a fast and effective technique for image segmentation. Although exhaustive search provides a comprehensive solution, its computational complexity increases with the number of threshold levels. This paper introduces a novel meta-heuristic search algorithm called Ensemble Searching (ES), designed to tackle complex nonlinear optimization problems. The focus is on applying ES to image multilevel thresholding. Initially, the population is divided into predefined groups, each guided by an evolutionary algorithm that independently searches for better positions within the search space. If an algorithm encounters a local optimum, a diversity-maintaining mechanism is activated to relocate the group. Throughout the iterative process, all algorithms share the best global solution (Gbest). The proposed structure's effectiveness is evaluated using ten test images and the energy curve method. Kapur's entropy, a well-established measure, is used to assess the algorithm's performance. A comparative analysis with eight different search algorithms demonstrates the proposed framework's rapid convergence, confirming its efficiency and effectiveness.

**Keywords**— Image segmentation, multilevel thresholding, ensemble searching, energy curve, Kapur entropy, swarm intelligence.

## 1. Introduction

Image segmentation is crucial for analyzing and interpreting images. Multi-level thresholding is a foundational technique for isolating significant objects across various domains, including image analysis, character recognition, target recognition, MPEG-4 object-based coding, map processing, and computer vision. Image segmentation methodologies are categorized into parametric and non-parametric approaches. Parametric methods estimate statistical parameters for two or more classes, which can be time-intensive and heavily reliant on initial conditions. Conversely, non-parametric methods determine threshold values by optimizing evaluation criteria, such as Otsu and Kapur measurements. The primary challenge in multi-level thresholding is identifying threshold values that maximize these criteria. Meta-heuristic algorithms have shown great promise in multi-level image thresholding, but there

are still several research gaps that need to be addressed:

1. Automatic Threshold Determination: Determining the optimal number of thresholds automatically remains a difficult task. Most current methods require manual input or predefined parameters, which can limit their applicability in real-world scenarios.
2. Diversity Preservation: Ensuring diversity in the population of solutions is crucial for the success of meta-heuristic algorithms. Techniques to maintain or enhance diversity during the optimization process are still being explored.
3. Hybrid Approaches: Combining meta-heuristic algorithms with other optimization techniques or machine learning methods could potentially improve performance. However, finding the right combination and integration strategy is a complex task.

 <http://dx.doi.org/10.22133/ijwr.2024.460411.1225>

**Citation** E. Ehsaeyan, "Ensemble Searching: A New Concept of Heuristic Search Algorithms and Its Application in Multilevel Thresholding Optimization", *International Journal of Web Research*, vol.7, no.3, pp.15-27, 2024, doi: <http://dx.doi.org/10.22133/ijwr.2024.460411.1225>.

\*Corresponding Author

Article History: Received: 31 January 2024 ; Revised: 15 May 2024; Accepted: 28 May 2024.

Copyright © 2024 University of Science and Culture. Published by University of Science and Culture. This work is licensed under a Creative Commons Attribution-Noncommercial 4.0 International license (<https://creativecommons.org/licenses/by-nc/4.0/>). Noncommercial uses of the work are permitted, provided the original work is properly cited. (<https://creativecommons.org/licenses/by-nc/4.0/>). Noncommercial uses of the work are permitted, provided the original work is properly cited.

To tackle these gaps, a new framework of meta-heuristic algorithms is proposed in this paper. The main contributions of this work are listed as:

1. A novel structure of meta-heuristic algorithm is proposed in this paper, which leverages the benefits of all combined algorithms.

2. The introduced methodology is equipped with a Darwinian rule, which improves the diversity of the population.

3. After conducting numerical evaluations and nonparametric statistical analysis, it was determined that the proposed method consistently delivers results that are highly comparable to, and often superior to, standalone algorithms.

## 2. Related Works

Nature-inspired algorithms frequently face sub-optimal regions, especially in high-dimensional spaces, and need finely-tuned parameters to perform effectively. The complexity introduced by numerous control factors represents a significant limitation of heuristic search algorithms. In response to these challenges, a variety of algorithms have emerged over recent decades [1-2]. In the realm of multilevel thresholding, methods leveraging metaheuristic algorithms have been introduced, including krill herd [3], fruit fly [4], thermal exchange [5], pigeon [6], differential evolution [7-10], human-mental [11], elephant herd [12], symbiotic organisms [13], moth swarm [14], grasshopper [15], and bird mating [16]. Cuckoo Search (CS) algorithm was applied in multilevel thresholding to reduce its complexity [17]. A chaotic version of Darwinian Particle Swarm Optimization was suggested to fly away from the local optima [18]. Chakraborty et al. modified Particle Swarm Optimization (PSO) algorithm by decomposing high dimensional population into some one-dimensional populations to escape from premature converge [19]. In multilevel thresholding literature, Gao et al. improved Artificial Bee Colony (ABC) algorithm by defining adaptive parameters and accelerating convergence rate [20]. Water Cycle algorithm is another bio-inspired search optimizer which was considered as an efficient method and was applied to color images [21]. A multilevel image segmentation method by bat algorithm with Kapur, Otsu, Renyi and Shannon entropies was published for color images [22]. Another research, a non-local mean 2D histogram was introduced for multilevel thresholding and used by gravitational search algorithm [23]. Oliva et al. applied antlion optimizer and sine cosine algorithms to thresholding, using energy curve, to consider the spatial information of neighbour pixels [24]. In another research, a novel thresholding method using animal migration optimization was reported for image segmentation [25]. Kotte et al. implemented adaptive Wind Driven algorithm to find the threshold values of MRI images and the comparative results were presented in view of between-class and Kapur entropy [26].

This study presents a new structure for heuristic search algorithms, improving the efficiency of optimization methods. It employs multilevel thresholding and introduces a parallel framework that reduces iterations and prevents individuals from getting trapped in local sub-regions. The algorithm is applied to the energy curve of gray images. Typically, eight renowned methods—differential evolution (DE) [27], particle swarm optimization (PSO) [28], bat algorithm search (BAT) [29], flower pollination algorithm (FPA) [30], artificial bee colony (ABC) [31], harmony search (HS) [32], grey wolf optimizer (GWO) [33], and whale optimization algorithm (WOA) [34]—are implemented and evaluated against benchmark images using Kapur entropy. The subsequent sections of this work are organized as follows: The next section reviews the related works on this issue. Section 3 outlines the fundamental theory of multilevel thresholding, histogram, energy curve, and Kapur criterion. Section 4 details the proposed ensemble searching algorithm. Section 5 presents empirical results, performance evaluation, and search capability. The final section discusses conclusions and directions for future research.

## 3. Image Thresholding

Multilevel thresholding is a technique used to segment an image  $I$  into multiple regions by applying  $t$  threshold values. In the context of a grayscale image with  $L$  intensity levels, multilevel thresholding can be mathematically represented as Equ(1):

$$\begin{aligned} R_0 &= \{g(x, y) \in I \mid 0 \leq g(x, y) \leq t_1 - 1 \\ &\vdots \\ R_K &= \{g(x, y) \in I \mid t_K \leq g(x, y) \leq L - 1 \end{aligned} \quad (1)$$

In the given context,  $g(x, y)$  represents a pixel in the image,  $t_i$  where ( $i=1, \dots, k$ ) denotes a threshold value, and  $K$  is the total number of thresholds. The process of image thresholding constitutes a  $K$ -dimensional problem. To address this optimization challenge, the optimal set of thresholds ( $t_1, t_2, \dots, t_K$ ) is determined to optimize a cost function, such as those proposed by Otsu and Kapur. Utilizing an exhaustive search algorithm to identify these thresholds becomes impractical and time-intensive as the problem scales to two or more levels. The complexity of this task is equivalent to a  $K$ -combination of  $L$  elements as Equ(2):

$$\binom{L}{K} = \frac{L!}{(L-K)!K!} \quad (2)$$

Figure 1.a illustrates the computational load for  $L=256$ . Despite threshold repetitions, multilevel thresholding remains a  $K$ -dimensional challenge, requiring exploration within a discrete  $K$ -dimensional hypercube ranging from  $0$  to  $L-1$  along each axis. Consequently, the problem's complexity is

on the order of  $L^K$ . Figure 1.b depicts viable solutions when  $K=3$ . These visual representations underscore that an exhaustive search is not only inefficient but also demands significant computational resources to identify optimal thresholds. This inefficiency necessitates the adoption of bio-inspired search algorithms for resolving multilevel thresholding issues.

### 3.1. Image Thresholding Fundamentals

Given an image with  $L$  gray levels ranging from  $\{0, 1, \dots, L-1\}$ , the normalized histogram can be derived as Equ(3):

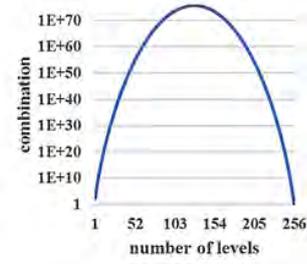
$$p_i = \frac{h(i)}{N} \quad 0 \leq i \leq L-1 \quad (3)$$

In this context,  $p_i$  represents the probability of intensity level  $i$ , while  $N$  signifies the total count of pixels within the image, and  $h(i)$  indicates the number of pixels that share the identical gray intensity  $i$ . The equation  $N = \sum_{i=0}^{L-1} p_i$  encapsulates this relationship. However, this approach overlooks the neighborhood factor, a crucial element often missed in histograms. In multilevel thresholding, another key criterion is the energy function. To accurately define an image's energy function, it is essential to first introduce a pixel's neighborhood system. The neighborhood mask  $N$  of order  $d$  for a pixel at position  $(i, j)$  is defined as  $N_{pq}^d = \{(i+u, j+v), (u, v) \in N^d\}$  [35]. Our focus is solely on the second-order neighborhood, denoted as  $d=2$ . The second-order neighborhood mask ( $N_{ij}^2$ ) is depicted in Figure 2.

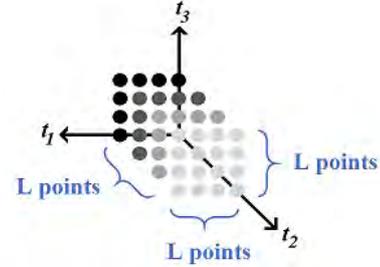
The energy function is computed for each distinct gray level. For a specific gray level  $l$ , we construct a binary matrix  $B_l$  of dimensions  $m \times n$  (matching the original image's size) defined by  $B_l = \{b_{ij}, 1 \leq i \leq m, 1 \leq j \leq n\}$ , where  $b_{ij} = 1$  if  $(g(x, y) > l)$ ; otherwise,  $b_{ij} = -1$ . Essentially,  $B_l$  designates whether a pixel in the original image possesses an intensity that is either below or above the threshold level  $l$ . In a similar vein, we introduce another matrix  $C$  as  $C = \{c_{ij}, 1 \leq i \leq m, 1 \leq j \leq n\}$ , populated entirely with ones, that is,  $c_{ij} = 1$  for all  $(i, j)$ . The energy function at gray level  $l$  is then formulated as Equ(4):

$$E_l = \sum_{i=1}^m \sum_{j=1}^n \sum_{pq \in N_{ij}^2} (c_{ij}c_{pq} - b_{ij}b_{pq}) \quad (4)$$

The matrix  $C$  is constructed to ensure the positive energy condition, where  $E_l \geq 0$ . Unlike the histogram diagram, the energy curve incorporates spatial information from adjacent pixels. This results in a more seamless and accurate differentiation between various objects within the image.



a) complexity of thresholding problem



b) Searching space  $K = 3$

Figure 1. image thresholding problem

$(i-1, j-1)$	$(i-1, j)$	$(i-1, j+1)$
$(i, j-1)$	$(i, j)$	$(i, j+1)$
$(i+1, j-1)$	$(i+1, j)$	$(i+1, j+1)$

Figure 2. Neighbourhood of pixel  $(i, j)$

### 3.2. Kapur entropy

The Kapur method, an entropy-driven criterion, seeks to centralize the probability density function (PDF) distribution for each segment within the histogram [36]. Initially introduced for bi-level thresholding, it focuses on finding the optimal threshold to segregate the object from its background. This methodology has been subsequently extended to multilevel thresholding and has been incorporated into numerous research investigations [37]. The issue of thresholding is described as Equ(5).

$$\text{Maximize } f(t_0, \dots, t_K) = \sum_{i=0}^K H_i$$

Where

$$H_0 = - \sum_{i=0}^{t_1-1} \frac{p_i}{\omega_0} \ln \frac{p_i}{\omega_0}, \quad \omega_0 = \sum_{i=0}^{t_1-1} p_i$$

⋮

$$H_K = - \sum_{i=t_K}^{L-1} \frac{p_i}{\omega_K} \ln \frac{p_i}{\omega_K}, \quad \omega_K = \sum_{i=t_K}^{L-1} p_i \quad (5)$$

The Kapur fitness function is established as the objective function within heuristic search algorithms. The goal is to ascertain a set of multiple thresholds  $t_1, \dots, t_K$  that will optimize this function.

4. The Proposed Structure

There are two major problems in heuristic search algorithms:

1-If the global optimum is not presented to individual routines, it will not be detected. Therefore, the method of searching the space is crucial. Each algorithm has a unique technique to cover the space. For example, combining a circular search like the Whale Optimization Algorithm (WOA) with a linear search like the Crow Search Algorithm (CSA) can be more effective.

2-All heuristic search algorithms struggle with getting stuck in suboptimal regions. Avoiding premature convergence is a significant weakness, and many strategies have been proposed in the literature to address this issue. In this study, the introduced algorithm incorporates a strategy to bypass local optima.

With these assumptions, the ensemble searching structure can be proposed. Figure 3 shows the framework of the ensemble searching. Ensemble searching structures outperform other methods because meta-heuristic algorithms have different procedures to reach the global optimum. Each procedure can lead the algorithm to obtain the best answer; however, they are not complete, and the success rate of each algorithm is limited. For example, PSO searches the space linearly following *Gbest*, while WOA moves in a spiral. By combining these two algorithms, a method is created that can search the space both linearly and spirally. Additionally, this combination helps the algorithm escape local optima.

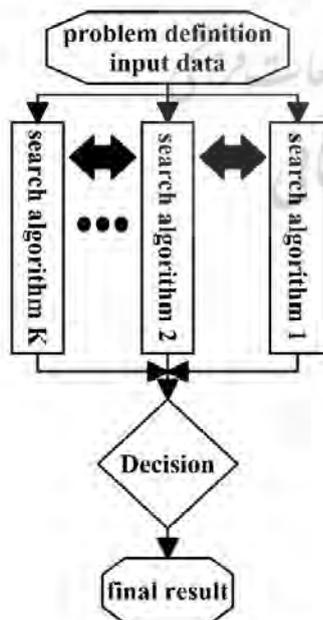


Figure 3. Structure of ensemble searching

To improve the efficiency of searching, Darwinian theory is also applied. Darwinian theory, also referred to as natural selection or survival of the fittest, posits that the robust members of a population thrive and reproduce, while the weaker ones perish and are weeded out. This concept is operationalized in the algorithm through a parameter called MOTIONLESS, which tracks the iterations a group remains stationary. The methodology of Darwinian ensemble searching is depicted in Figure. 4.

The implementation process unfolds as outlined: An initial population is established by randomly generating *N* particles, each with *K* dimensions as Equ(6):

$$\vec{x}_i = L_{min} + (L_{max} - L_{min}) \times rand \quad (6)$$

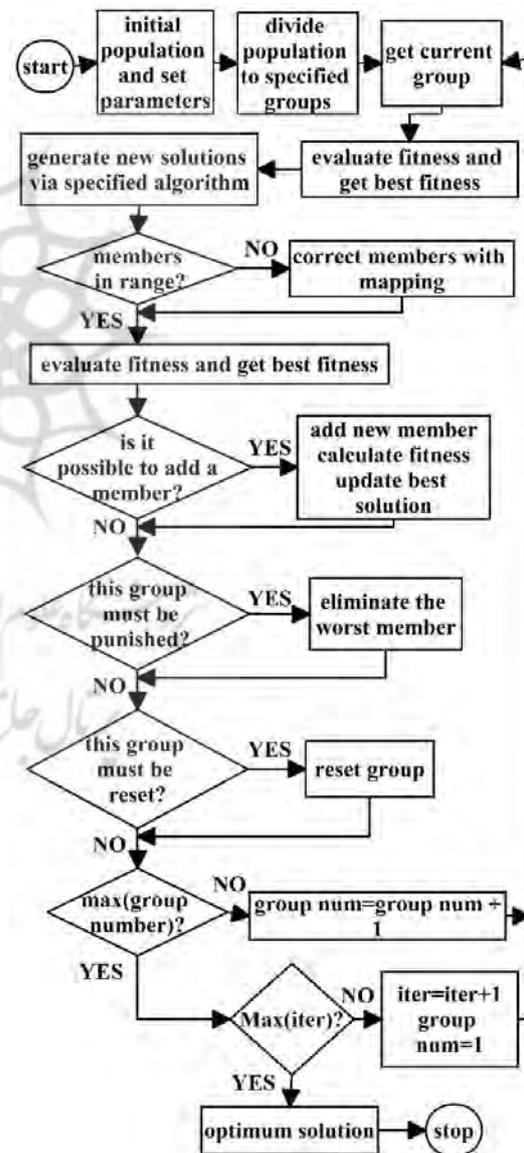


Figure 4. Flowchart of Darwinian ensemble searching

Within this framework, rand represents a random number selected from the interval  $([0,1])$ , while  $L_{min}$  and  $L_{max}$  denote the minimum and maximum grayscale levels of the image, respectively, for  $i=1, \dots, N$ . To avoid expending time on infeasible solutions,  $L_{min}$  and  $L_{max}$  are not fixed at 0 and 255. Next, the initial solutions are determined by calculating the fitness values of the particles. The algorithm then sets various control parameters, such as weighting values, maximum iterations, the range of group numbers, a cap on group members, and MOTIONLESS. It organizes members into  $G$  groups, each steered by a nature-inspired algorithm aiming for the Global optimum. Concurrent operation of groups within the same search domain bolsters search capabilities and hastens convergence.

To circumvent entrapment in sub-optimal zones, a safeguarding mechanism is in place. The fitness of each group member is assessed, with the premier point discovered by a group designated as GroupBest, and the supreme position identified by all participants termed GBest. The ensemble search structure is managed by incorporating GroupBest into the algorithms' update equations. The new position for each individual  $i$  is thus established as Equ(7):

$$X_i^{t+1} = \text{previous terms} + \alpha_1 \times r_1 \times (\text{GroupBest}_i^t - X_i^t) + \alpha_2 \times r_2 \times (\text{GBest} - X_i^t) \quad (7)$$

Here  $\alpha_1$  and  $\alpha_2$  represent the step sizes for exploration and exploitation, respectively,  $r_1$  and  $r_2$  come from  $[0,1]$  distribution and  $t$  indicates current iteration. *Previous terms* depends on updating the relation of each algorithm. For example, in Cuckoo Search algorithm (CS), the updating rule is given by Equ(8):

$$X_i^{t+1} = X_i^t + \alpha(X_i^t - \text{GBest})\text{Levy}(\lambda) \quad (8)$$

The location of nest  $i$  ( $i = 1, 2, \dots, N$ ) in the next generation  $t + 1$  is denoted by  $x_i^{t+1}$ . This position is determined by three factors: a step-size control parameter ( $\alpha$ ), the current best solution found in the iteration (GBest), and a random walk term ( $\text{Levy}(\lambda)$ ) drawn from a Levy distribution. Equation (9) expresses this relationship

$$X_i^{t+1} = X_i^t + \alpha_1 \times \text{Levy}(\lambda) \times (\text{PBest}_i^t - X_i^t) + \alpha_2 \times \text{Levy}(\lambda) \times (\text{GBest} - X_i^t) \quad (9)$$

In the optimization process, GBest guides the particles towards significant regions, while Pbest assists them in discovering new optimal solutions. Consider  $X_i^t = [x_{i1}^t \dots x_{iK}^t]$  as the newly generated position in a K-dimensional space during iteration  $t$ . The subsequent step involves verifying the viability of these solutions. Should a position fall outside the

predefined boundaries, it will be adjusted through a specific mapping technique to ensure its correctness as Equ(10):

$$x_{ip}^t = \begin{cases} L - 1 & x_{ip}^t > L - 1 \\ x_{ip}^t & 0 \leq x_{ip}^t \leq L - 1 \\ 0 & x_{ip}^t < 0 \end{cases}, p = 1, \dots, K \quad (10)$$

The fitness scores of all candidates are reassessed, and the PBest, GroupBest, and GBest metrics are modified as needed. When a group discovers an improved solution, its stagnancy count remains at zero, signifying that its GroupBest is refreshed each cycle. A group maintaining zero stagnancy with fewer members than the maximum allowed may introduce a new member, created randomly as in the initial population setup. Conversely, a group's inability to enhance the GroupBest results in a penalty through the expulsion of a member. The member with the poorest fitness level is located and expelled. Rather than resetting the group's stagnancy count to zero after this removal, it is instead set to (11) [38].

$$\text{motionless} = SC_C^{\max} \left[ 1 - \frac{1}{N_{kill} + 1} \right] \quad (11)$$

Where  $SC_C^{\max}$  represents the maximum acceptable stagnancy of groups, and  $N_{kill}$  indicates the number of individuals removed from a weak group. If a group loses too many members, it suggests that it is converging to a local optimum and cannot update its position, necessitating a reset to save time.

## 5. Experimental Results

To assess the performance of the suggested algorithms, eight search algorithms (DE, PSO, BAT, WOA, GWO, FPA, ABC, and HS) were applied to the test images depicted in Figure 5 to highlight the advantages of our approach. These test images have been selected from the University of Waterloo's repository [39]. This repository includes several classical image datasets that are widely used in machine learning and computer vision research. The collection includes both photographic and synthetic images, providing a diverse range of graphical material to assess the performance of image processing algorithms. The image repository is categorized into three sets:

- Greyscale Set 1: 12 small greyscale images
- Greyscale Set 2: 12 medium greyscale images
- Color Set: 8 large full-color images

Figure 6 illustrates the energy curves associated with the images.

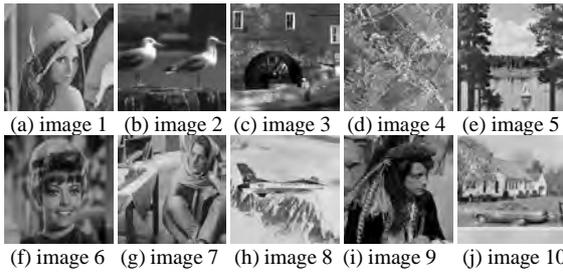


Figure. 5. Benchmark Images for Assessing the Performance of the Newly Developed Algorithm

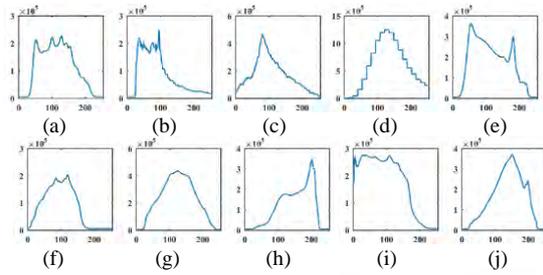


Figure. 6. Energy curve of test images  
 (a) Test 1, (b) Test 2  
 (c) Test 3, (d) Test 4  
 (e) Test 5, (f) Test 6  
 (g) Test 7, (h) Test 8  
 (i) Test 9, (j) Test 10

Kapur's entropy method was utilized to evaluate the effectiveness of various approaches and the precision of the resulting solutions. The Peak Signal-to-Noise Ratio (PSNR) is a metric that quantifies the quality of a reconstructed image compared to the original, defined by Equ(12):

$$PSNR = 20 \log (255 / RMSE)$$

$$RMSE = \sqrt{\sum_{i=1}^M \sum_{j=1}^N (I(i, j) - I'(i, j))^2 / MN} \quad (12)$$

In this context,  $M$  and  $N$  represent the dimensions of the test image. The terms  $I(i, j)$  and  $I'(i, j)$  correspond to the original and segmented images, respectively. The Root Mean Squared Error (RMSE) is used to quantify the error between the original and segmented images. The Structural Similarity Index (SSIM) is a metric that measures the similarity between the original and processed image and is calculated as Equ(13):

$$SSIM = \frac{(2\mu_I\mu_{I'}+c_1)(2\sigma_{II'}+c_2)}{(\mu_I^2+\mu_{I'}^2+c_1)(\sigma_I^2+\sigma_{I'}^2+c_2)} \quad (13)$$

In this equation,  $\mu$  and  $\sigma$  denote the mean and variance of the images  $I$  and  $I'$ , respectively. The term  $\sigma_{II'}$  represents the covariance between  $I$  and  $I'$ . The constants  $c_1$  and  $c_2$  are parameters associated with the pixel values of the images. The Feature Similarity Index (FSIM) is defined as Equ(14) [40]:

$$FSIM = \frac{\sum_{x \in \Omega} S_L(x) PC_m(x)}{\sum_{x \in \Omega} PC_m(x)} \quad (14)$$

Let  $\Omega$  denote the entire image domain. The term  $S_L(x)$  represents the similarity parameter, while  $PC_m(x)$  signifies the phase consistency. These parameters are defined as (15):

$$PC_m(x) = \max(PC_1(x), PC_2(x))$$

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

$$S_{PC}(x) = \frac{2PC_1(x) \times PC_2(x) + T_1}{PC_1^2(x) \times PC_2^2(x) + T_1}$$

$$S_G(x) = \frac{2G_1(x) \times G_2(x) + T_2}{G_1^2(x) \times G_2^2(x) + T_2} \quad (15)$$

$PC_1(x)$  and  $PC_2(x)$  indicate the phase consistency and  $\alpha$ ,  $\beta$ ,  $T_1$  and  $T_2$  are constants parameters. It is observed that an increase in PSNR, SSIM, or FSIM values typically signifies enhanced image segmentation quality. These metrics tend to rise in conjunction with elevated threshold levels, which correlates with improved segmentation precision. The control parameters for the algorithms presented are detailed in Table 1. These parameters have been selected based on their optimal performance as reported in the original publications and have been fine-tuned through a trial-and-error process.

For a balanced evaluation of the search algorithms, each is run for 50 iterations with a population size of 40. The Kapur method serves as the fitness function, optimized by all algorithms during the testing phase to determine the ideal threshold levels.

Tables 2 to 6 display the outcomes for varying threshold values  $K=3,4,5$ . The fitness values shown represent the average of 30 runs per algorithm, with additional data pertaining to the optimal solution identified in these trials. Four algorithms—PSO, DE, BAT, FPA—comprise the first ensemble, ENSEMBLE 1 (ENS1). Similarly, ENSEMBLE 2 (ENS2) includes ABC, HS, GWO, WOA. The Darwinian outcomes for both ENS1 and ENS2 are also documented in these tables (DENS1 and DENS2). Figures 7 and 8 illustrate the segmented images using the Kapur method for level 5, while Figure 9 depicts the convergence curves for test images based on the Kapur method over 30 iterations.

The data from these tables indicate that both ENSEMBLE 1 and 2 outperform individual algorithms. Moreover, the Darwinian variants of ENSEMBLE 1 and 2 exhibit superior performance compared to other algorithms in metrics such as PSNR, SSIM, FSIM, or the mean objective function. As the thresholding size increases, the efficacy of other search methods diminishes, often converging

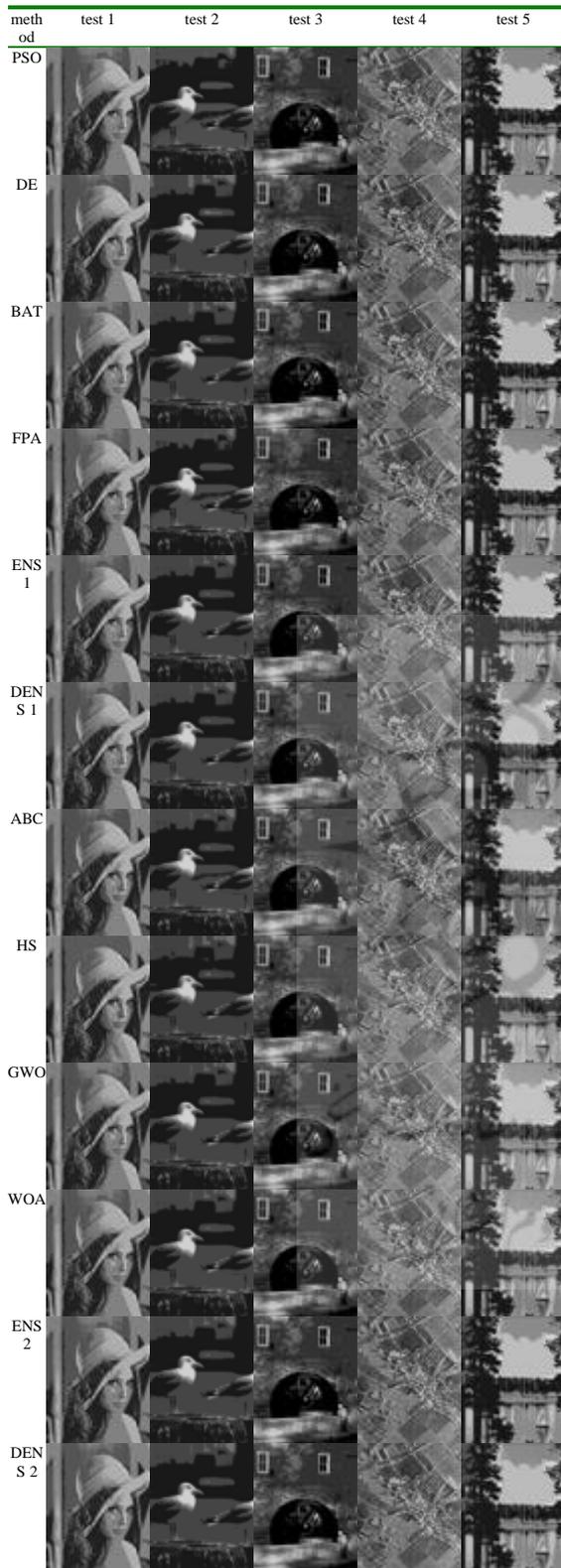


Figure 7. Segmented images by Kapur entropy (level = 5)

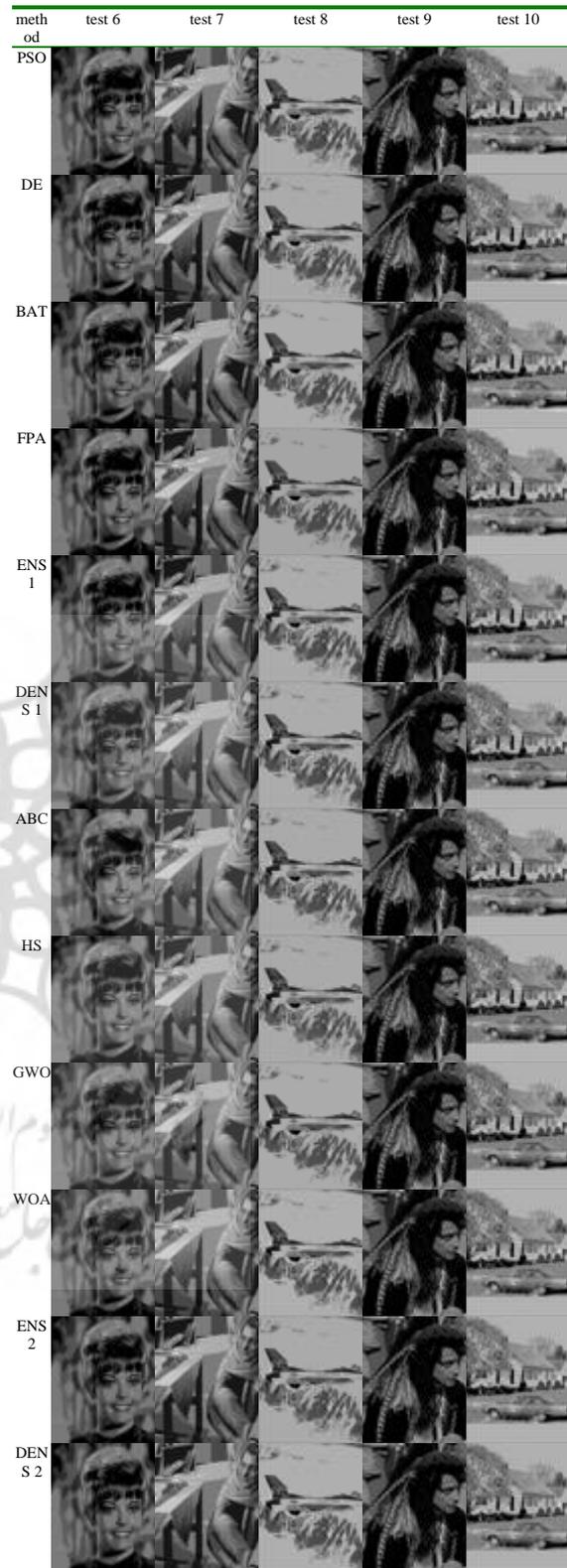


Figure 8. Segmented images by Kapur entropy (level = 5)

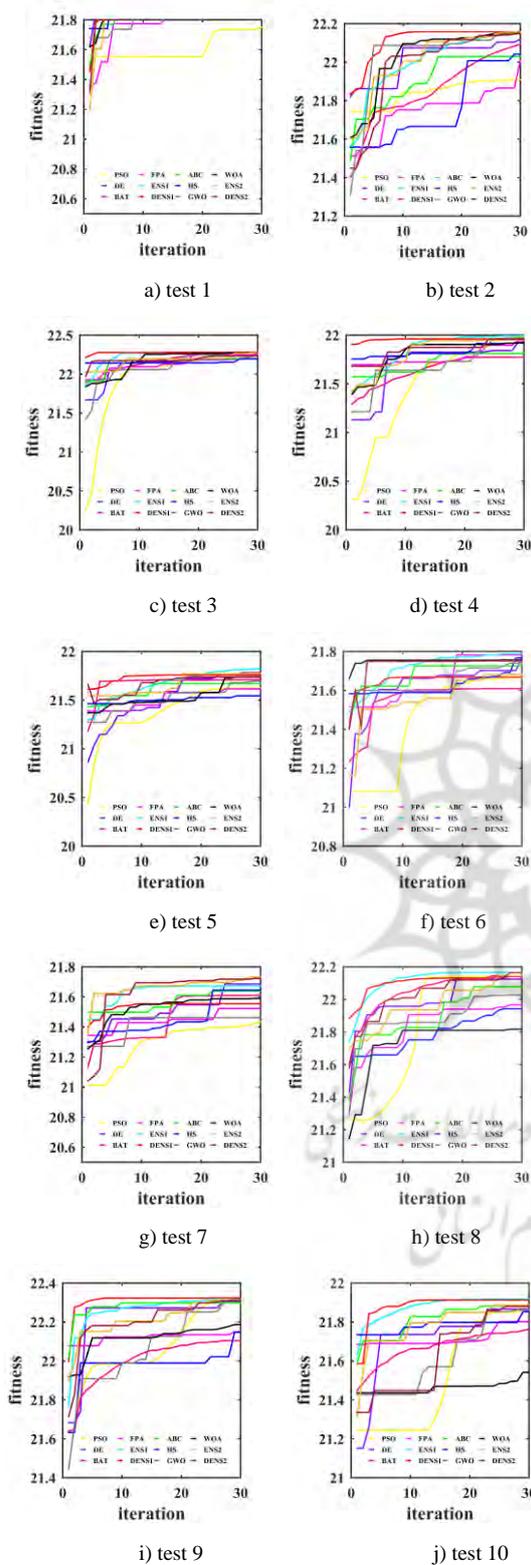


Figure 9. Convergence Trajectories Utilizing the Kapur Method (Level 5)

Table 1. Optimization Parameters for Metaheuristic Search Algorithms

Algorithm	Parameters
ABC	$limit = 0.1 \times population$
BAT	$loudness = 0.5$
	$pulse\ rate = 0.5$
	$frequency\ range: [0,1]$
DE	$crossover\ probability = 0.1$
FPA	$scaling\ factor = 0.5$
GWO	$probability\ switch = 0.7$
	$\alpha = 2 - \frac{1}{max\ generation}$
HS	$Consideration\ Rate: 0.75$
	$Pitch\ adjusting\ rate: 0.5$
	$new\ harmonies = 0.1 \times population$
PSO	$velocity\ range: [-2,2]$
	$Cognitive\ constant = 1.5$
	$Social\ constant = 1.5$
WOA	$logarithmic\ spiral = 1$
Darwinian Ensemble	$group\ population\ range = [8,12]$
	$maximum\ stagnancy = 2$

Table 2. Average Fitness Scores Across Algorithms Using the Kapur Method

method	K	test 1	test 2	test 3	test 4	test 5	test 6	test 7	test 8	test 9	test 10
PSO	3	15.85	16.13	<b>16.38</b>	15.92	15.71	16.20	15.76	15.89	16.28	15.81
	4	19.00	19.12	19.43	18.98	18.73	19.05	18.72	18.94	19.35	18.82
	5	21.98	21.87	22.23	21.89	21.64	21.68	21.58	21.91	22.26	21.79
	3	<b>15.90</b>	<b>16.14</b>	<b>16.38</b>	15.94	<b>15.74</b>	<b>16.21</b>	<b>15.78</b>	16.06	<b>16.29</b>	<b>15.86</b>
	4	<b>19.19</b>	<b>19.17</b>	19.44	19.06	18.75	19.05	18.74	19.15	19.40	19.02
DE	3	15.86	<b>16.14</b>	16.37	15.91	15.71	16.19	15.76	15.94	16.28	15.81
	4	19.09	19.11	19.40	18.99	18.73	19.02	18.70	19.07	19.34	18.93
	5	22.03	21.94	22.22	21.89	21.60	21.64	21.57	22.02	22.21	21.77
	3	15.89	<b>16.14</b>	16.37	15.93	15.73	16.20	15.77	16.02	16.28	15.85
	4	19.10	19.15	19.41	19.03	18.74	19.03	18.73	19.11	19.36	18.96
BAT	3	15.90	<b>16.14</b>	<b>16.38</b>	<b>15.95</b>	<b>15.74</b>	<b>16.21</b>	<b>15.78</b>	<b>16.07</b>	<b>16.29</b>	<b>15.86</b>
	4	<b>19.19</b>	<b>19.17</b>	<b>19.45</b>	19.09	<b>18.78</b>	<b>19.06</b>	18.75	<b>19.16</b>	<b>19.41</b>	<b>19.03</b>
	5	22.16	22.03	22.26	21.97	21.71	21.74	21.66	22.15	22.31	21.88
	3	15.89	<b>16.14</b>	<b>16.38</b>	<b>15.95</b>	<b>15.74</b>	<b>16.21</b>	<b>15.78</b>	<b>16.07</b>	<b>16.29</b>	15.84
	4	19.16	<b>19.17</b>	<b>19.45</b>	<b>19.11</b>	<b>18.78</b>	<b>19.06</b>	<b>18.76</b>	<b>19.16</b>	<b>19.41</b>	<b>19.03</b>
FPA	3	15.90	<b>16.14</b>	<b>16.38</b>	15.94	<b>15.74</b>	16.20	15.77	16.04	<b>16.29</b>	15.85
	4	19.18	19.17	19.43	19.06	18.75	19.04	18.74	19.13	19.38	18.99
	5	22.10	22.09	22.21	21.87	21.70	21.69	21.66	22.07	22.24	21.84
	3	15.89	16.13	16.37	15.93	15.73	16.20	15.77	<b>16.04</b>	16.28	15.84
	4	19.09	19.15	19.41	19.02	18.73	19.03	18.72	19.10	19.37	18.95
ENS 1	3	15.91	<b>16.14</b>	<b>16.38</b>	15.92	15.73	<b>16.21</b>	<b>15.78</b>	16.06	<b>16.29</b>	15.82
	4	19.17	19.15	19.44	19.08	18.74	19.05	18.76	19.14	19.40	18.98
	5	22.13	22.10	22.23	21.95	21.74	21.73	21.67	22.10	22.24	21.86
	3	15.89	<b>16.14</b>	<b>16.38</b>	15.92	15.73	16.20	<b>15.78</b>	16.05	<b>16.29</b>	15.85
	4	19.14	19.16	19.44	19.05	18.75	19.04	18.74	19.11	19.40	18.98
ENS 2	3	15.91	<b>16.14</b>	<b>16.38</b>	15.93	<b>15.74</b>	<b>16.21</b>	<b>15.78</b>	<b>16.06</b>	<b>16.29</b>	15.85
	4	19.19	<b>19.18</b>	<b>19.45</b>	<b>19.09</b>	18.77	<b>19.06</b>	18.75	19.14	19.40	19.01
	5	22.14	22.14	22.25	21.96	21.76	21.75	21.70	22.13	22.30	21.87
	3	<b>15.91</b>	<b>16.14</b>	<b>16.38</b>	<b>15.95</b>	<b>15.74</b>	<b>16.21</b>	<b>15.78</b>	<b>16.06</b>	<b>16.29</b>	<b>15.86</b>
	4	<b>19.20</b>	<b>19.18</b>	<b>19.45</b>	<b>19.09</b>	<b>18.78</b>	<b>19.06</b>	<b>18.77</b>	<b>19.15</b>	<b>19.41</b>	<b>19.02</b>
DENS 2	3	<b>15.91</b>	<b>16.14</b>	<b>16.38</b>	<b>15.95</b>	<b>15.74</b>	<b>16.21</b>	<b>15.78</b>	<b>16.06</b>	<b>16.29</b>	<b>15.86</b>
	4	<b>19.20</b>	<b>19.18</b>	<b>19.45</b>	<b>19.09</b>	<b>18.78</b>	<b>19.06</b>	<b>18.77</b>	<b>19.15</b>	<b>19.41</b>	<b>19.02</b>
	5	<b>22.15</b>	<b>22.15</b>	<b>22.26</b>	<b>21.97</b>	<b>21.79</b>	<b>21.76</b>	<b>21.71</b>	<b>22.14</b>	<b>22.31</b>	<b>21.88</b>

Table 3. Threshold Values Determined by Various Algorithms Using the Kapur Method

method	K	test 1	test 2	test 3	test 4	test 5
PSO	3	33,107,178	74,130,191	61,128,196	40,114,188	82,136,192
	4	33,98,162,219	67,113,161,208	54,105,154,206	40,92,145,203	25,85,139,193
	5	33,83,129,177,221	25,78,123,170,214	50,94,135,176,216	40,80,128,167,214	25,82,133,188,227
DE	3	33,105,178	75,130,191	61,128,196	40,115,188	82,136,192
	4	33,99,163,219	25,79,136,195	56,106,155,206	40,95,148,203	74,130,188,228
	5	33,82,132,176,221	25,69,112,158,207	48,91,132,174,215	26,68,115,162,208	26,77,136,188,226
BAT	3	33,107,178	74,130,191	61,128,196	40,114,188	82,136,192
	4	33,98,161,219	25,79,136,195	54,105,154,206	40,94,148,202	80,133,188,227
	5	33,83,129,175,220	25,68,113,160,206	50,94,134,174,215	40,80,126,172,212	24,85,134,186,226
FPA	3	33,108,181	75,129,191	59,127,196	40,108,186	82,136,193
	4	34,93,155,219	25,80,135,196	56,110,152,208	40,93,144,199	82,129,188,227
	5	33,89,129,174,222	25,76,122,168,212	45,88,131,178,217	40,75,118,159,205	27,78,133,187,229
ENS 1	3	33,107,178	74,130,191	61,128,196	40,114,188	82,136,192
	4	33,98,162,219	25,79,136,195	54,105,154,206	40,94,148,202	80,133,188,227
	5	33,83,130,176,221	25,69,114,162,208	50,94,135,176,215	40,80,125,170,213	25,83,135,188,227
DENS 1	3	33,107,178	74,130,191	61,128,196	39,113,187	82,136,192
	4	33,98,162,219	25,79,136,195	54,105,154,206	40,94,148,202	80,133,188,227
	5	33,83,130,176,221	25,69,114,162,208	50,94,135,176,215	40,80,125,170,214	25,83,134,188,227
ABC	3	33,107,178	75,130,191	61,128,196	40,112,186	82,136,192
	4	33,101,163,220	25,78,136,195	55,107,160,212	40,96,146,198	81,131,185,226
	5	33,81,132,175,220	25,71,116,163,210	48,96,135,174,218	26,69,124,167,212	24,77,122,186,228
HS	3	33,107,179	75,129,191	61,129,196	40,111,187	80,135,191
	4	32,93,154,218	25,78,133,193	57,107,154,206	40,93,149,199	81,134,186,226
	5	33,90,140,175,220	25,65,113,156,206	48,92,139,183,217	40,82,118,169,208	26,83,128,176,227
GWO	3	33,107,178	74,130,191	61,128,196	40,112,188	82,136,192
	4	33,96,162,220	25,79,136,195	53,104,153,206	40,94,150,202	83,135,188,227
	5	33,81,127,172,220	25,69,114,162,209	50,95,136,175,215	26,67,113,160,209	26,81,132,186,227
WOA	3	33,107,178	74,130,191	61,128,196	40,112,188	82,136,192
	4	33,95,163,219	25,79,136,195	54,104,154,206	40,95,146,203	80,134,186,227
	5	33,85,129,171,220	25,70,114,162,209	49,94,137,176,216	26,67,115,161,209	25,80,129,186,227
ENS 2	3	33,107,178	74,130,191	61,128,196	40,114,188	82,136,192
	4	33,97,161,219	25,79,136,195	53,105,154,206	40,94,149,202	80,132,189,227
	5	33,83,129,174,221	25,69,114,162,208	50,94,136,176,215	26,67,115,161,206	24,87,138,189,227
DENS 2	3	33,107,178	74,130,191	61,128,196	40,115,188	82,136,192
	4	33,98,161,219	25,79,136,195	53,105,155,206	40,93,148,202	81,131,187,227
	5	33,85,131,175,220	25,69,114,162,208	49,94,135,177,216	26,67,114,161,208	25,82,134,187,227
PSO	3	58,112,171	74,130,186	71,147,223	63,126,194	75,148,224
	4	47,88,129,173	23,77,130,187	65,116,168,223	58,115,172,213	61,115,170,224
	5	44,84,124,162,187	23,76,124,172,219	33,75,124,172,223	45,89,131,174,213	51,93,135,179,225
DE	3	58,112,171	74,129,185	71,148,223	64,127,195	76,147,224
	4	46,88,130,174	23,78,131,185	65,117,170,223	57,112,173,213	60,115,169,224
	5	47,86,124,163,187	23,72,122,174,219	35,77,125,171,223	46,87,133,173,215	52,90,133,175,224
BAT	3	58,112,171	74,130,186	71,147,223	63,126,194	75,148,224
	4	46,87,129,173	23,77,132,187	62,114,167,223	57,115,172,213	60,113,169,224
	5	44,81,119,161,187	23,74,123,173,220	32,74,122,173,223	42,86,130,174,213	51,93,135,178,224
FPA	3	57,111,171	74,129,186	70,149,223	64,126,194	74,144,224
	4	49,88,131,175	24,81,131,184	62,121,167,223	59,115,173,212	59,117,175,224
	5	48,86,124,163,187	22,73,126,167,216	38,74,122,165,223	43,88,133,178,212	50,90,137,182,226
ENS 1	3	58,112,171	74,130,186	71,147,223	63,126,194	75,148,224
	4	47,88,129,173	23,78,133,187	64,116,168,223	58,115,172,213	60,113,169,224
	5	44,83,123,162,187	23,74,123,173,220	34,76,124,172,223	44,88,132,175,214	51,93,135,178,225
DENS 1	3	58,112,171	74,130,186	71,147,223	63,126,194	75,148,224
	4	46,87,128,172	23,78,133,187	64,116,168,223	58,115,172,213	60,113,169,224
	5	44,83,123,162,187	23,74,123,173,220	34,76,124,172,223	44,88,132,175,214	50,92,134,178,225
ABC	3	58,111,171	73,129,185	72,146,223	64,126,195	73,147,224
	4	47,87,128,172	23,75,133,186	60,114,164,223	61,117,171,214	62,117,172,224
	5	52,87,126,163,187	23,78,128,176,220	32,77,125,170,222	43,85,127,176,214	50,88,136,180,225
HS	3	58,111,170	75,130,187	72,146,223	64,127,194	76,150,225
	4	46,89,129,171	24,86,139,189	59,112,170,223	53,115,170,212	57,110,168,224
	5	47,90,123,162,187	22,76,119,171,221	37,80,118,170,223	39,88,131,181,214	36,83,130,180,225
GWO	3	58,112,171	74,130,186	71,147,223	63,126,194	75,148,224
	4	46,87,128,173	23,78,132,187	63,113,166,223	56,114,172,213	58,112,170,224
	5	44,81,122,162,187	23,75,125,173,220	33,78,125,171,223	43,86,131,175,215	48,88,131,177,225
WOA	3	58,112,171	74,130,186	70,147,223	63,126,194	76,149,224
	4	47,87,129,173	23,74,127,185	66,118,169,223	59,115,172,213	59,112,168,224
	5	43,84,122,162,187	23,72,124,173,221	35,77,126,175,223	40,86,129,175,214	52,94,133,178,225
ENS 2	3	58,112,171	74,130,186	72,148,223	63,126,194	75,148,224
	4	46,87,129,173	23,77,132,187	63,113,168,223	59,114,172,213	60,112,167,224
	5	42,83,121,162,187	23,73,123,173,220	33,76,121,170,223	43,86,130,174,214	53,95,136,177,225
DENS 2	3	58,112,171	74,130,186	71,147,223	63,126,194	74,147,224
	4	46,87,128,172	23,78,133,188	64,115,168,223	56,115,172,213	58,113,169,224
	5	45,83,122,162,187	23,74,122,172,219	34,76,126,175,223	44,88,131,175,214	48,88,132,176,225

Table 4. PSNR Outcomes for Various Algorithms Derived from the Kapur Method

method	K	test 1	test 2	test 3	test 4	test 5	test 6	test 7	test 8	test 9	test 10
PSO	3	16.2	16.4	18.0	15.7	15.4	18.1	17.0	13.5	18.1	14.84
	4	16.9	17.3	19.1	18.6	17.9	20.3	18.8	16.9	18.6	17.97
	5	19.8	20.7	20.0	20.2	17.9	20.7	19.3	17.9	21.2	20.19
DE	3	16.2	16.4	18.0	15.7	15.4	18.1	17.1	13.6	18.1	14.85
	4	16.9	20.1	19.2	17.7	15.7	20.2	18.9	17.2	18.6	17.92
	5	19.7	21.4	20.1	18.7	18.3	20.6	19.2	17.7	21.1	19.92
BAT	3	16.2	16.4	18.0	15.7	15.4	18.1	17.0	13.5	18.1	14.84
	4	17.0	20.1	19.1	19.4	15.5	20.3	18.8	16.8	18.6	17.83
	5	19.9	21.2	20.0	20.1	17.9	20.9	19.2	18.0	21.1	20.24
FPA	3	16.1	16.4	17.8	15.2	15.4	18.2	17.0	13.7	18.1	14.70
	4	17.3	20.2	19.0	18.9	15.3	20.2	19.0	16.6	18.6	18.17
	5	20.0	20.9	19.8	20.7	18.2	20.6	19.3	16.7	21.0	20.25
ENS 1	3	16.2	16.4	18.0	15.7	15.4	18.1	17.0	13.5	18.1	14.84
	4	16.9	20.1	19.1	19.4	15.5	20.3	18.8	16.9	18.6	17.83
	5	19.8	21.2	20.0	20.1	18.0	20.7	19.2	17.9	21.1	20.13
DENS 1	3	16.2	16.4	18.0	15.5	15.4	18.1	17.0	13.5	18.1	14.84
	4	16.9	20.1	19.1	19.4	15.5	20.4	18.8	16.9	18.6	17.83
	5	19.8	21.2	20.0	20.1	17.9	20.7	19.2	17.9	21.1	20.08
ABC	3	16.2	16.4	18.0	15.5	15.4	18.2	17.1	13.4	18.1	14.77
	4	16.8	20.3	19.0	17.9	15.6	20.4	18.7	16.3	18.7	18.13
	5	19.7	21.1	19.7	18.1	17.3	20.4	19.2	17.7	21.0	20.09
HS	3	16.1	16.4	17.7	15.5	15.5	18.2	17.0	13.4	18.1	14.79
	4	17.4	20.2	19.2	17.9	15.6	20.3	18.9	17.2	18.7	17.63
	5	19.6	20.9	19.8	19.8	18.8	20.6	19.1	17.6	21.0	19.70
GWO	3	16.2	16.4	18.0	15.6	15.4	18.1	17.0	13.5	18.1	14.84
	4	16.8	20.1	19.0	18.2	15.5	20.4	18.8	16.6	18.6	17.80
	5	20.0	21.2	19.8	19.9	18.1	20.7	19.3	17.7	21.1	19.83
WOA	3	16.2	16.4	18.0	15.6	15.4	18.1	17.0	13.5	18.1	14.88
	4	16.7	20.1	19.2	17.6	15.7	20.3	18.8	17.1	18.6	17.74
	5	20.0	21.3	19.8	20.1	17.8	20.7	19.1	18.4	21.1	20.10
ENS 2	3	16.2	16.4	18.0	15.7	15.4	18.1	17.0	13.6	18.1	14.84
	4	16.9	20.1	19.0	18.2	15.5	20.3	18.8	16.9	18.6	17.70
	5	19.9	21.2	19.9	20.0	17.9	20.8	19.2	17.6	21.1	20.12
DENS 2	3	16.2	16.4	18.0	15.7	15.4	18.1	17.0	13.5	18.1	14.80
	4	17.0	20.1	19.0	19.2	15.5	20.4	18.8	16.9	18.6	17.82
	5	19.9	21.2	19.9	20.0	18.0	20.7	19.3	18.4	21.1	19.79

Table 5. SSIM Outcomes for Various Algorithms Derived from the Kapur Method

method	K	test 1	test 2	test 3	test 4	test 5	test 6	test 7	test 8	test 9	test 10
PSO	3	0.63	0.36	0.53	0.79	0.51	0.51	0.58	0.73	0.47	0.63
	4	0.64	0.41	0.59	0.88	0.69	0.57	0.72	0.80	0.49	0.73
	5	0.71	0.73	0.62	0.91	0.67	0.58	0.74	0.82	0.59	0.79
DE	3	0.63	0.35	0.53	0.79	0.51	0.51	0.58	0.74	0.46	0.64
	4	0.64	0.72	0.59	0.86	0.52	0.57	0.72	0.80	0.50	0.74
	5	0.71	0.77	0.63	0.90	0.68	0.58	0.74	0.82	0.58	0.78
BAT	3	0.63	0.36	0.53	0.79	0.51	0.51	0.58	0.73	0.47	0.63
	4	0.64	0.72	0.59	0.89	0.50	0.57	0.72	0.80	0.49	0.73
	5	0.71	0.77	0.62	0.91	0.65	0.59	0.74	0.82	0.60	0.78
FPA	3	0.63	0.35	0.53	0.79	0.51	0.51	0.58	0.74	0.47	0.64
	4	0.65	0.72	0.58	0.89	0.49	0.56	0.73	0.79	0.49	0.73
	5	0.71	0.74	0.62	0.92	0.68	0.57	0.73	0.81	0.59	0.79
ENS 1	3	0.63	0.36	0.53	0.79	0.51	0.51	0.58	0.73	0.47	0.63
	4	0.64	0.72	0.59	0.89	0.50	0.57	0.72	0.80	0.49	0.73
	5	0.71	0.77	0.62	0.91	0.67	0.58	0.74	0.82	0.59	0.79
DENS 1	3	0.63	0.36	0.53	0.78	0.51	0.51	0.58	0.73	0.47	0.63
	4	0.64	0.72	0.59	0.89	0.50	0.57	0.72	0.80	0.49	0.73
	5	0.71	0.77	0.62	0.91	0.67	0.58	0.74	0.82	0.59	0.79
ABC	3	0.63	0.35	0.53	0.79	0.51	0.51	0.58	0.73	0.47	0.63
	4	0.63	0.72	0.58	0.87	0.48	0.57	0.72	0.79	0.48	0.73
	5	0.71	0.75	0.62	0.89	0.65	0.56	0.74	0.80	0.60	0.79
HS	3	0.63	0.35	0.52	0.79	0.52	0.51	0.58	0.73	0.46	0.64
	4	0.65	0.72	0.58	0.86	0.49	0.57	0.72	0.80	0.50	0.73
	5	0.70	0.77	0.62	0.91	0.69	0.58	0.74	0.82	0.60	0.78
GWO	3	0.63	0.36	0.53	0.79	0.51	0.51	0.58	0.73	0.47	0.63
	4	0.64	0.72	0.59	0.87	0.50	0.57	0.72	0.80	0.50	0.73
	5	0.72	0.77	0.62	0.91	0.67	0.59	0.74	0.82	0.59	0.79
WOA	3	0.63	0.36	0.53	0.79	0.51	0.51	0.58	0.73	0.47	0.63
	4	0.65	0.72	0.59	0.86	0.49	0.57	0.72	0.80	0.49	0.73
	5	0.71	0.76	0.62	0.91	0.66	0.59	0.74	0.82	0.60	0.79
ENS 2	3	0.63	0.36	0.53	0.79	0.51	0.51	0.58	0.74	0.47	0.63
	4	0.64	0.72	0.58	0.87	0.51	0.57	0.72	0.80	0.49	0.73
	5	0.71	0.77	0.62	0.91	0.67	0.59	0.74	0.82	0.59	0.80
DENS 2	3	0.63	0.36	0.53	0.79	0.51	0.51	0.58	0.73	0.47	0.64
	4	0.64	0.72	0.58	0.89	0.49	0.57	0.72	0.80	0.49	0.73
	5	0.71	0.77	0.62	0.91	0.67	0.58	0.74	0.82	0.59	0.79

Table 6. FSIM Outcomes for Various Algorithms Derived from the Kapur Method

method	K	test 1	test 2	test 3	test 4	test 5	test 6	test 7	test 8	test 9	test 10
PSO	3	0.74	0.76	0.81	0.94	0.85	0.72	0.84	0.78	0.80	0.76
	4	0.76	0.80	0.87	0.97	0.87	0.77	0.85	0.85	0.83	0.85
	5	0.82	0.79	0.89	0.98	0.87	0.78	0.86	0.86	0.89	0.89
DE	3	0.73	0.76	0.81	0.94	0.85	0.72	0.84	0.78	0.80	0.77
	4	0.76	0.78	0.86	0.96	0.85	0.77	0.85	0.85	0.83	0.85
	5	0.82	0.82	0.89	0.98	0.88	0.78	0.86	0.86	0.89	0.88
BAT	3	0.74	0.76	0.81	0.94	0.85	0.72	0.84	0.78	0.80	0.76
	4	0.76	0.78	0.87	0.97	0.85	0.77	0.85	0.85	0.83	0.85
	5	0.82	0.83	0.89	0.98	0.86	0.79	0.86	0.86	0.89	0.89
FPA	3	0.74	0.76	0.81	0.94	0.85	0.72	0.84	0.78	0.80	0.76
	4	0.77	0.78	0.85	0.97	0.84	0.77	0.85	0.84	0.83	0.85
	5	0.82	0.80	0.89	0.97	0.88	0.78	0.87	0.85	0.89	0.89
ENS 1	3	0.74	0.76	0.81	0.94	0.85	0.72	0.84	0.78	0.80	0.76
	4	0.76	0.78	0.87	0.97	0.85	0.77	0.85	0.85	0.83	0.85
	5	0.82	0.83	0.89	0.98	0.87	0.78	0.86	0.86	0.89	0.89
DENS 1	3	0.74	0.76	0.81	0.94	0.85	0.72	0.84	0.78	0.80	0.76
	4	0.76	0.78	0.87	0.97	0.85	0.77	0.85	0.85	0.83	0.85
	5	0.82	0.83	0.89	0.98	0.87	0.78	0.86	0.86	0.89	0.89
ABC	3	0.74	0.76	0.81	0.94	0.85	0.72	0.84	0.77	0.80	0.76
	4	0.76	0.78	0.86	0.96	0.85	0.77	0.85	0.84	0.82	0.85
	5	0.82	0.82	0.89	0.98	0.86	0.78	0.86	0.85	0.89	0.89
HS	3	0.74	0.76	0.80	0.94	0.85	0.72	0.84	0.77	0.80	0.77
	4	0.77	0.78	0.86	0.96	0.85	0.77	0.84	0.85	0.83	0.84
	5	0.82	0.83	0.89	0.97	0.88	0.79	0.86	0.86	0.89	0.89
GWO	3	0.74	0.76	0.81	0.94	0.85	0.72	0.84	0.78	0.80	0.76
	4	0.76	0.78	0.87	0.97	0.85	0.77	0.85	0.85	0.83	0.84
	5	0.83	0.83	0.89	0						

within local sub-regions and failing to yield satisfactory threshold values. In contrast, the Darwinian ENSEMBLE 1 and ENSEMBLE 2 consistently achieve higher objective function values after more than 30 runs, surpassing other algorithms in both exploration and exploitation of the search space. Consequently, the Darwinian ENSEMBLE approach, grounded in the Kapur method, proves to be an effective strategy for multi-level image thresholding. It avoids futile searches in non-essential areas and adeptly maintains the diversity of the search agents.

To assess the convergence velocity, a comparative analysis of processing times for diverse methods was conducted using the aforementioned benchmarks. Figure 10 illustrates the mean execution duration for various algorithms applied to test images. The data indicates that ENSEMBLE algorithms exhibit satisfactory computational efficiency in problem resolution.

## 6. Conclusion

This paper introduces a novel search framework for heuristic algorithms, specifically designed for object-based MPEG-4 coding. The framework incorporates Darwinian natural selection to enhance population diversity. The Kapur method is used as a segmentation criterion, with results demonstrated for three threshold values ( $K=3, 4, 5$ ) over 30 trials. Darwinian theory guides the algorithm's search strategy, improving the movement of search agents towards the optimal target. The combination of Darwinian theory and shared spatial information among algorithms has significantly improved segmentation for various benchmark images, showing notable enhancements in the objective function, PSNR, SSIM, and FSIM metrics compared to individual heuristic search methods. The only drawback of the proposed method is that it is time-consuming. Although the ensemble benefits from combined algorithms and better convergence, the execution time increases, and parallel hardware is needed for implementation. Future studies will explore the application of Ensemble Searching to other areas such as feature selection and assess the method's effectiveness in identifying optimal solutions.

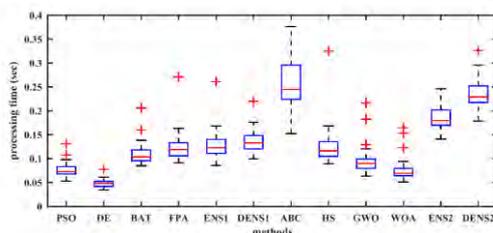


Figure 10. Mean Execution Duration for Diverse Algorithmic Procedures

## Declarations

### Funding

This research did not receive any grant from funding agencies in the public, commercial, or non-profit sectors.

### Authors' contributions

Ehsan Ehsaeyan: Study design, acquisition of data, interpretation of the results, statistical analysis, drafting the manuscript;

### Conflict of interest

The authors declare that no conflicts of interest exist.

## References

- [1] Chouhan SS, Kaul A, Singh UP (2019) Image Segmentation Using Computational Intelligence Techniques: Review. *Arch Computat Methods Eng* 26:533–596. <https://doi.org/10.1007/s11831-018-9257-4>
- [2] 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. *Computers & Electrical Engineering* 70:476–495. <https://doi.org/10.1016/j.compeleceng.2017.08.008>
- [3] He L, Huang S (2020) An efficient krill herd algorithm for color image multilevel thresholding segmentation problem. *Applied Soft Computing* 89:106063. <https://doi.org/10.1016/j.asoc.2020.106063>
- [4] Ding G, Dong F, Zou H (2019) Fruit fly optimization algorithm based on a hybrid adaptive-cooperative learning and its application in multilevel image thresholding. *Applied Soft Computing* 84:105704. <https://doi.org/10.1016/j.asoc.2019.105704>
- [5] Xing Z, Jia H (2020) Modified thermal exchange optimization based multilevel thresholding for color image segmentation. *Multimed Tools Appl* 79:1137–1168. <https://doi.org/10.1007/s11042-019-08229-1>
- [6] Wang Y, Zhang G, Zhang X (2019) Multilevel Image Thresholding Using Tsallis Entropy and Cooperative Pigeon-inspired Optimization Bionic Algorithm. *J Bionic Eng* 16:954–964. <https://doi.org/10.1007/s42235-019-0109-1>
- [7] J. De Jesús Rubio, E. Orozco, D. A. Cordova, M. A. Hernandez, F. J. Rosas, and J. Pacheco, "Observer-based differential evolution constrained control for safe reference tracking in robots," *Neural Networks*, vol. 175, p. 106273, Jul. 2024, doi: 10.1016/j.neunet.2024.106273.
- [8] L. Wang, Y. He, X. Wang, Z. Zhou, H. Ouyang, and S. Mirjalili, "A novel discrete differential evolution algorithm combining transfer function with modulo operation for solving the multiple knapsack problem," *Information Sciences*, vol. 680, p. 121170, Oct. 2024, doi: 10.1016/j.ins.2024.121170.
- [9] Y. Huang, L. Xing, B. Li, and B. Zhou, "PC-SSRDE: A paradigm crossover-based differential evolution algorithm with search space reduction," *Information Sciences*, vol. 681, p. 121188, Oct. 2024, doi: 10.1016/j.ins.2024.121188.
- [10] Y. Zhan, Z. Hu, J. Kou, N. Hong, and Q. Su, "A constrained differential evolution algorithm for calculation of two-phase equilibria at given volume, temperature and moles," *Journal of Computational Physics*, vol. 518, p. 113320, Dec. 2024, doi: 10.1016/j.jcp.2024.113320.
- [11] Mousavirad SJ, Ebrahimpour-Komleh H (2019) Human mental search-based multilevel thresholding for image

- segmentation. *Applied Soft Computing* 105427. <https://doi.org/10.1016/j.asoc.2019.04.002>
- [12] Chakraborty F, Roy PK, Nandi D (2019) Oppositional elephant herding optimization with dynamic Cauchy mutation for multilevel image thresholding. *Evol Intel* 12:445–467. <https://doi.org/10.1007/s12065-019-00238-1>
- [13] Chakraborty F, Nandi D, Roy PK (2019) Oppositional symbiotic organisms search optimization for multilevel thresholding of color image. *Applied Soft Computing* 82:105577. <https://doi.org/10.1016/j.asoc.2019.105577>
- [14] Bhandari AK, Rahul K (2019) A context sensitive Masi entropy for multilevel image segmentation using moth swarm algorithm. *Infrared Physics & Technology* 98:132–154. <https://doi.org/10.1016/j.infrared.2019.03.010>
- [15] Bhandari AK, Rahul K (2019) A novel local contrast fusion-based fuzzy model for color image multilevel thresholding using grasshopper optimization. *Applied Soft Computing* 81:105515. <https://doi.org/10.1016/j.asoc.2019.105515>
- [16] Ahmadi M, Kazemi K, Aarabi A, et al (2019) Image segmentation using multilevel thresholding based on modified bird mating optimization. *Multimed Tools Appl* 78:23003–23027. <https://doi.org/10.1007/s11042-019-7515-6>
- [17] Pare S, Kumar A, Bajaj V, Singh GK (2017) An efficient method for multilevel color image thresholding using cuckoo search algorithm based on minimum cross entropy. *Applied Soft Computing* 61:570–592. <https://doi.org/10.1016/j.asoc.2017.08.039>
- [18] Suresh S, Lal S (2017) Multilevel thresholding based on Chaotic Darwinian Particle Swarm Optimization for segmentation of satellite images. *Applied Soft Computing* 55:503–522. <https://doi.org/10.1016/j.asoc.2017.02.005>
- [19] Chakraborty R, Sushil R, Garg ML (2019) An Improved PSO-Based Multilevel Image Segmentation Technique Using Minimum Cross-Entropy Thresholding. *Arab J Sci Eng* 44:3005–3020. <https://doi.org/10.1007/s13369-018-3400-2>
- [20] Gao H, Fu Z, Pun C-M, et al (2018) A multi-level thresholding image segmentation based on an improved artificial bee colony algorithm. *Computers & Electrical Engineering* 70:931–938. <https://doi.org/10.1016/j.compeleceng.2017.12.037>
- [21] Kandhway P, Bhandari AK (2019) A Water Cycle Algorithm-Based Multilevel Thresholding System for Color Image Segmentation Using Masi Entropy. *Circuits Syst Signal Process* 38:3058–3106. <https://doi.org/10.1007/s00034-018-0993-3>
- [22] Mishra S, Panda M (2018) Bat Algorithm for Multilevel Colour Image Segmentation Using Entropy-Based Thresholding. *Arab J Sci Eng* 43:7285–7314. <https://doi.org/10.1007/s13369-017-3017-x>
- [23] Mittal H, Saraswat M (2018) An optimum multi-level image thresholding segmentation using non-local means 2D histogram and exponential Kbest gravitational search algorithm. *Engineering Applications of Artificial Intelligence* 71:226–235. <https://doi.org/10.1016/j.engappai.2018.03.001>
- [24] Oliva D, Hinojosa S, Elaziz MA, Ortega-Sánchez N (2018) Context based image segmentation using antlion optimization and sine cosine algorithm. *Multimed Tools Appl* 77:25761–25797. <https://doi.org/10.1007/s11042-018-5815-x>
- [25] Rahkar Farshi T (2019) A multilevel image thresholding using the animal migration optimization algorithm. *Iran J Comput Sci* 2:9–22. <https://doi.org/10.1007/s42044-018-0022-5>
- [26] Kotte S, Pullakura RK, Injeti SK (2018) Optimal multilevel thresholding selection for brain MRI image segmentation based on adaptive wind driven optimization. *Measurement* 130:340–361. <https://doi.org/10.1016/j.measurement.2018.08.007>
- [27] Bilal, Pant M, Zaheer H, et al (2020) Differential Evolution: A review of more than two decades of research. *Engineering Applications of Artificial Intelligence* 90:103479. <https://doi.org/10.1016/j.engappai.2020.103479>
- [28] Elbes M, Alzubi S, Kanan T, et al (2019) A survey on particle swarm optimization with emphasis on engineering and network applications. *Evol Intel* 12:113–129. <https://doi.org/10.1007/s12065-019-00210-z>
- [29] Chawla M, Duhan M (2015) Bat Algorithm: A Survey of the State-of-the-Art. *Applied Artificial Intelligence* 29:617–634. <https://doi.org/10.1080/08839514.2015.1038434>
- [30] Abdel-Basset M, Shawky LA (2019) Flower pollination algorithm: a comprehensive review. *Artif Intell Rev* 52:2533–2557. <https://doi.org/10.1007/s10462-018-9624-4>
- [31] Karaboga D, Gorkemli B, Ozturk C, Karaboga N (2014) A comprehensive survey: artificial bee colony (ABC) algorithm and applications. *Artif Intell Rev* 42:21–57. <https://doi.org/10.1007/s10462-012-9328-0>
- [32] Manjarres D, Landa-Torres I, Gil-Lopez S, et al (2013) A survey on applications of the harmony search algorithm. *Engineering Applications of Artificial Intelligence* 26:1818–1831. <https://doi.org/10.1016/j.engappai.2013.05.008>
- [33] Hatta NM, Zain AM, Sallehuddin R, et al (2019) Recent studies on optimisation method of Grey Wolf Optimiser (GWO): a review (2014–2017). *Artif Intell Rev* 52:2651–2683. <https://doi.org/10.1007/s10462-018-9634-2>
- [34] Gharehchopogh FS, Gholizadeh H (2019) A comprehensive survey: Whale Optimization Algorithm and its applications. *Swarm and Evolutionary Computation* 48:1–24. <https://doi.org/10.1016/j.swevo.2019.03.004>
- [35] Ghosh S, Bruzzone L, Patra S, et al (2007) A Context-Sensitive Technique for Unsupervised Change Detection Based on Hopfield-Type Neural Networks. *IEEE Trans Geosci Remote Sensing* 45:778–789. <https://doi.org/10.1109/TGRS.2006.888861>
- [36] Kapur JN, Sahoo PK, Wong AKC (1985) A new method for gray-level picture thresholding using the entropy of the histogram. *Computer Vision, Graphics, and Image Processing* 29:273–285. [https://doi.org/10.1016/0734-189X\(85\)90125-2](https://doi.org/10.1016/0734-189X(85)90125-2)
- [37] Symbiotic Organisms Search Algorithm for multilevel thresholding of images, Bū, s ranur Küçüku ʒurlu a, \*, Eyüp Gedikli b, *Expert Systems With Applications* 147 (2020) 113210, DOI:10.1016/j.eswa.2020.113210
- [38] Ghamisi P, Couceiro MS, Benediktsson JA, Ferreira NMF (2012) An efficient method for segmentation of images based on fractional calculus and natural selection. *Expert Systems with Applications* 39:12407–12417. <https://doi.org/10.1016/j.eswa.2012.04.078>
- [39] “Repository.” Accessed: Sep. 15, 2024. [Online]. Available: <https://links.uwaterloo.ca/Repository.html>
- [40] Lin Zhang, Lei Zhang, Xuanqin Mou and D. Zhang, “FSIM: A Feature Similarity Index for Image Quality Assessment,” *IEEE Trans. on Image Process.* 20, 2378–2386 (2011).



**Ehsan Ehsaeyan** was born in Sirjan, Iran, in 1982. He received the B.S. degree in electrical engineering from Shahed University, Tehran, in 2005, the M.S. degree from the Shahid Bahonar University, Kerman, Iran, in 2008, and the Ph.D. degree in communication engineering from Shiraz

University, Shiraz, in 2021. Since 2012, he has been an Assistant Professor with the Electrical Engineering Department, Sirjan University of Technology, Sirjan, Iran. his research interests include image processing and natural-inspired algorithms.

