Journal of Artificial Intelligence and Data Mining (JAIDM), x(x): xxx-xxx, xxxx

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research paper

Enhancing Image Segmentation with Darwinian Grey Wolf Optimizer: A Novel Multilevel Thresholding Approach

Ehsan Ehsaeyan*

Electrical Engineering Department, Sirjan University of Technology, Sirjan, Iran.

Article Info

Article History:
Received 30 April 2024
Revised 07 August 2024
Accepted 22 August 2024

DOI:10.22044/jadm.2024.14469.2550

Keywords:
Image thresholding; Multilevel Segmentation; Grey Wolf Optimizer; Darwinian rule.

**Corresponding author: Ehsaeyan@sirjantech.ac.ir (E. Ehsaeyan).*

Abstract

This paper presents a novel approach to image segmentation through multilevel thresholding, leveraging the speed and precision of the technique. The proposed algorithm, based on the Grey Wolf Optimizer (GWO), integrates Darwinian principles to address the common stagnation issue in metaheuristic algorithms, which often results in local optima and premature convergence. The search agents are efficiently steered across the search space by a dual mechanism of encouragement and punishment employed by our strategy, thereby curtailing computational time. This is implemented by segmenting the population into distinct groups, each tasked with discovering superior solutions. To validate the algorithm's efficacy, 9 test images from the Pascal VOC dataset were selected, and the renowned energy curve method was employed for verification. Additionally, Kapur entropy was utilized to gauge the algorithm's performance. The method was benchmarked against four disparate search algorithms, and its dominance was underscored by achieving the best outcomes in 20 out of 27 cases for image segmentation. The experimental findings collectively affirm that the Darwinian Grey Wolf Optimizer (DGWO) stands as a formidable instrument for multilevel thresholding.

1. Introduction

Multilevel thresholding has gained popularity as an effective segmentation method for extracting significant objects from images. Multilevel thresholding approaches are generally divided into these categories: parametric and non-parametric methods. The parametric approach utilizes statistical parameters from multiple classes, which can be time-intensive and heavily reliant on initial conditions. Non-parametric methods directly optimize criteria for threshold estimation, avoiding assumptions about data distribution. Figure 1 provides a summary of these techniques as documented in the literature. Fuzzy clustering is a special way of grouping image pixels together based on their shades of gray. The Expectation Maximization (EM) algorithm, a density-based unsupervised method, seeks the maximum likelihood estimates of parameters from a dataset [1]. Edge detection addresses image segmentation by identifying and linking edges or pixels between regions with abrupt intensity changes, forming closed object boundaries [2].

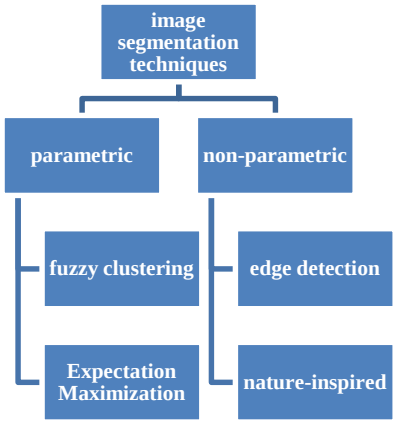


Figure 1. Classification of segmentation algorithms.

A variety of thresholding techniques inspired by natural phenomena have been explored, including those based on krill herds [3], thermal exchanges [4] and human cognition [5]. However, these nature-inspired algorithms often converge on suboptimal solutions, particularly in high-dimensional spaces, and require carefully calibrated parameters to function effectively. The Grey Wolf Optimizer (GWO) is a notable natural-inspired method characterized by its minimal control parameters and rapid convergence, making it well-suited for complex problems. A primary limitation of the GWO is its lack of adaptability to specific problem conditions. To address this shortcoming, some researchers have refined the classical GWO [6] or integrated it with other metaheuristic algorithms [7]. Furthering the review of literature, Erwin and Yuningsih applied GWO for segmenting retinal blood vessels [8], and Khairuzzaman and Chaudhury developed a GWO-based multilevel image thresholding technique [9]. This paper first establishes the foundation by covering the multilevel thresholding problem, explaining energy curves and the Kapur criterion (Section 2). Section 3 then details the Grey Wolf Optimizer. Building on this groundwork, a novel Darwinian Grey Wolf Optimizer (DGWO) is introduced for optimal threshold determination (Section 3). The efficacy of DGWO is evaluated in Section 4. For comparison, the performance of four established methods – Modified Snake Optimizer (MSO) [10], Snow Ablation Optimizer (SAO) [11], improved Salp Swarm Algorithm (SDSSA) [12], enhanced Whale Optimization Algorithm (CVWOA) [13]– is assessed on benchmark images using the energy curve.

2. Fundamental Concept

2.1. Image Thresholding

In image segmentation, a technique called multilevel thresholding can be used to divide a grayscale image I into distinct regions $t+1$. This is achieved by identifying thresholds t that correspond to the image's intensity levels L.

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

Each pixel in the image, denoted by $g(x,y)$, has an intensity value. We define $t_i \ (i = 1,...,K)$ as the threshold values, where K is the total number of thresholds used.

2.2. Basic Theory Of Energy Curve

For a given intensity level, l, a binary image is created with the same dimensions $m \times n$ as the original image. The elements of B_l are defined as: $B_l = \{ b_{ij}, 1 \le i \le m, 1 \le j \le n \}$ where if $g(x,y) > 1$, then $b_{ij} = 1$; Otherwise, $b_{ij} = -1$. Another matrix, called C, is created with all ones. This means every element in C (represented as c_{ij}) has a value of 1, regardless of its position (i, j) within the matrix. energy function helps us understand how important it is to consider the brightness level l when processing the image:

$$E_L = \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{p,q \in N_{ij}^2} \left(c_{ij} c_{pq} - b_{ij} b_{pq} \right) \quad (2)$$

The matrix C is defined with all elements set to one ($c_{ij} = 1$ for all i and j) to ensure a non-negative energy condition $E_L \ge 0$.

2.3. Kapur Method

In image segmentation, the Kapur method utilizes entropy to create histograms with well-defined peaks for each class [14]. Let's delve into the formulation of the multilevel thresholding problem:

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

Where

$$\begin{aligned} H_0 &= -\sum_{i=0}^{t_1-1} \frac{p_i}{\omega_0} \ln \frac{p_i}{\omega_0}, \omega_0 = \sum_{i=0}^{t_1-1} p_i \\ &\vdots \\ H_K &= -\sum_{i=t_K}^{L-1} \frac{p_i}{\omega_K} \ln \frac{p_i}{\omega_K}, \omega_K = \sum_{i=t_K}^{L-1} p_i \end{aligned} \quad (3)$$

Where p_i denotes the probability of intensity i.

3. The Proposed Algorithm

Grey Wolf Optimizer (GWO) is a simple natural-inspired method introduced in 2014 [15]. The implementation of GWO can be summarized as follows: Assume that N individuals have randomly initialized positions in a K-dimensional space.

$$X = \begin{bmatrix} \vec{x}_1 \\ \vdots \\ \vec{x}_N \end{bmatrix} = \begin{bmatrix} x_{11} & \cdots & x_{1K} \\ \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{NK} \end{bmatrix} \quad (4)$$

Within the group, the three most successful wolves (alpha, beta, and delta) are identified through competition. The remaining wolves follow the lead of these top performers to improve their own performance as described below:

$$\vec{D} = \left| \vec{C}.\vec{x}_p(t) - \vec{x}(t) \right| \quad (5)$$

$$\vec{x}(t-1) = \vec{x}_p(t) - \vec{A}.\vec{D} \quad (6)$$

Where t is the iteration, $\vec{x}_p(t)$ is the location of the prey, $\vec{A}$ and $\vec{C}$ are identified as:

$$\vec{A} = 2\vec{a}.\vec{r}_1 - \vec{a} \quad (7\text{-a})$$

$$\vec{C} = 2\vec{r}_2 \quad (7\text{-b})$$

In these equations, $\vec{r}_1$ and $\vec{r}_2$ are random numbers drawn from $U(0,1)$ and $\vec{a}$ is decreased from 2 to 0. In every iteration, α, β and δ are determined by the fitness calculation and omega wolves are update their positions as

$$\vec{D}_\alpha = \left| \vec{C}_1.\vec{x}_\alpha - \vec{x} \right| \quad (8\text{-a})$$

$$\vec{D}_\beta = \left| \vec{C}_2.\vec{x}_\beta - \vec{x} \right| \quad (8\text{-b})$$

$$\vec{D}_\delta = \left| \vec{C}_3.\vec{x}_\delta - \vec{x} \right| \quad (8\text{-c})$$

$$\vec{x}_1 = \vec{x}_\alpha - \vec{A}_1.\vec{D}_\alpha \quad (9\text{-a})$$

$$\vec{x}_2 = \vec{x}_\beta - \vec{A}_2.\vec{D}_\beta \quad (9\text{-b})$$

$$\vec{x}_3 = \vec{x}_\delta - \vec{A}_3.\vec{D}_\delta \quad (9\text{-c})$$

$$\vec{x}(t+1) = \frac{1}{3}(\vec{x}_1 + \vec{x}_2 + \vec{x}_3) \quad (10)$$

This research proposes a Darwinian Grey Wolf Optimizer (DGWO) for multilevel thresholding, addressing the challenge of balancing exploration (finding new possibilities) and exploitation (refining promising areas) to avoid getting stuck in sub-optimal solutions. Inspired by Darwinian theory's "survival of the fittest," DGWO employs a parameter called STAGNANCY. STAGNANCY tracks how long a group of wolves (algorithm elements) fails to improve. If stagnation exceeds a limit, the group is penalized by losing a member, similar to a wolf pack struggling in a bad hunting ground. By eliminating the least effective member, the group is encouraged to explore new areas. Notably, the stagnation counter isn't reset to zero after this penalty but is assigned a specific value (to be determined) to prevent immediate relapse.

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

This process considers two factors: a maximum allowed time period where groups aren't making progress $SC_C^{\max}$, and the number of individuals removed from a low-performing group N_{kill}.If this group is confined to local areas, it may become stagnant there. Therefore, stagnancy is not reset to zero but is set according to Equation (11). A lower stagnancy value encourages reinforcement of the random search, while a higher value may result in time wasted on searching within local areas. Equation (11) represents a moderate degree of success as proposed in the referenced study [16].

Evolutionary principles inspired by Darwinian theory improve computational efficiency and reduce runtime execution. This is achieved by allowing the algorithm to escape sub-optimal regions. If a group gets stuck on a decent solution, the search there stops. Instead, a new group starts searching in a different area. This helps the algorithm explore the whole problem space, making it more diverse. Figure 2 illustrates the main steps of the DGWO algorithm in a flowchart. To begin, a population of *N* randomly generated solution vectors is created by Eq. (12). Each solution vector has *K* dimensions.

$$x_l = L_{\min} + \left(L_{\max} - L_{\min}\right) \times rand \quad (12)$$

The algorithm initializes solutions by assigning a random value between 0 and 1 to each individual. $L_{\min}$ and $L_{\max}$ define the valid range for image gray levels (excluding 0 and 255 to avoid unnecessary exploration). Each initial solution's fitness is then evaluated. The algorithm defines control parameters like maximum iterations, allowable group size range, and a stagnation threshold. Finally, the algorithm divides the population into groups and runs them concurrently on the same search space, improving search efficiency and convergence speed.

Several groups, acting like miniature GWOs, work together to find the absolute best solution (Global optimum). To prevent them from getting stuck in areas that aren't as good (sub-optimal regions), a system is in place. Each group member's performance (fitness) is assessed, and the best position discovered by a group is designated as $GroupBest$. A new position in K-dimensional space at generation t will be represented as $X_i^t = \left[x_{i1}^t ... x_{iK}^t \right]$, generated by Eq. (12). Next, it is checked if the solutions are valid (feasible). If a position falls outside the designated range, a specific mapping procedure will be applied to bring it back within bounds by Eq. (13). Fitness of all individuals is continuously measured.

$$x_{ip}^t = \begin{cases} L-1 & x_{ip}^t > L \\ x_{ip}^t & 0 \le x_{ip}^t \le L \\ 0 & x_{ip}^t < 0 \end{cases} \quad (23)$$

$$p = 1, ..., K$$

If a group finds a better position, it remains active (stagnancy stays at zero) and its best solution *GroupBest* is updated. Active groups under a population limit can add a new randomly generated member. Likewise, active groups within a group limit are rewarded with the ability to form a new group. This new group inherits half its members from the successful group and the rest are random, promoting exploration around promising areas [16]. Conversely, failing groups that can't improve their best solution are penalized by removing the least fit member.

The DGWO algorithm iteratively refines a population of groups, deleting those that get stuck in suboptimal solutions ("local optima") and replacing them with new groups based on successful ones. All groups are evaluated in each iteration, adding or removing members as needed, until a stopping criterion like a maximum number of iterations or a good enough solution (acceptable fitness) is met. The details of this process are further explained in pseudocode format in Figure 3.

4. EXPERIMENTAL RESULTS

To assess the performance of the proposed algorithms, four well-established search algorithms were implemented on the test images selected from the Pascal VOC dataset [17], as presented in Figure 4. These algorithms, namely SAO, MSO, CVWOA and SDSSA, were chosen due to their documented effectiveness in image segmentation tasks. Figure 5 illustrates the corresponding energy curves of the test images.

For comparison of efficiency and solution accuracy across the different methods, Kapur entropy was chosen as the metric. The algorithms were implemented in MATLAB on a system with an i7 2.5 GHz CPU, 8 GB RAM, and Windows 10 (64-bit). The performance of the search algorithms was evaluated using the energy curve of the benchmark images, the obtained threshold values, mean fitness function, and PSNR, SSIM, and FSIM metrics. To clarify, Peak Signal-to-Noise Ratio (PSNR) is defined as

$$PSNR = 20\log\left(255/RMSE\right) \qquad (34\text{-a})$$

$$RMSE = \sqrt{\sum_{i,j}^{M,N}\left(I(i,j)-I'(i,j)\right)^2 \Big/ MN} \qquad (14\text{-b})$$

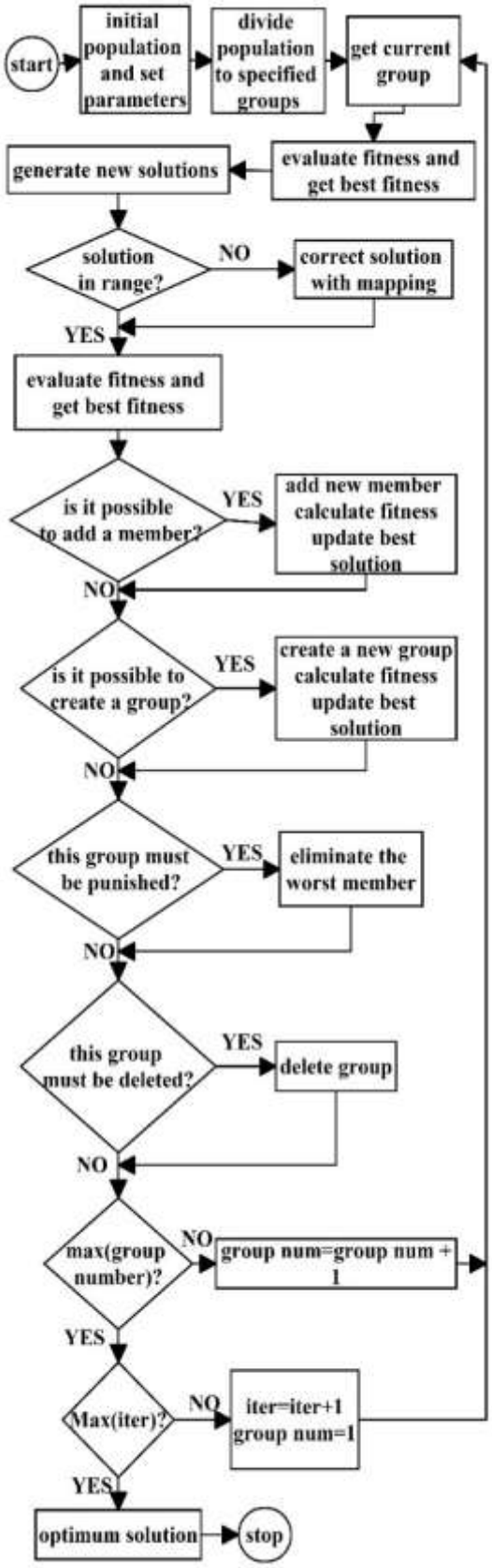


Figure 2. Flowchart of the proposed DGWO.

Start
Set parameters:
Iteration, initial group population, minimum group population, maximum group population, initial group number, minimum group number, maximum group number, STAGNANCY,
first population:
Generate Random Initial Population according Eq. (11)
Divide population to *initial group number*
Find the best solution within a group (GroupBest)
While iter<iter_maximum
Get current group
For j=1:*group_number*
Update individuals location according Eq. (10)
Correct infeasible solution according Eq. (13)
Calculate fitness and update *GroupBest*
Create a new particle if possible
Create a new group if possible
Eliminate worst particle
Eliminate inefficient group
Calculate fitness and update *GroupBest* and STAGNANCY
end
end
identify Best *GroupBest*
Stop

Figure 3. Pseudo code of the proposed DGWO.

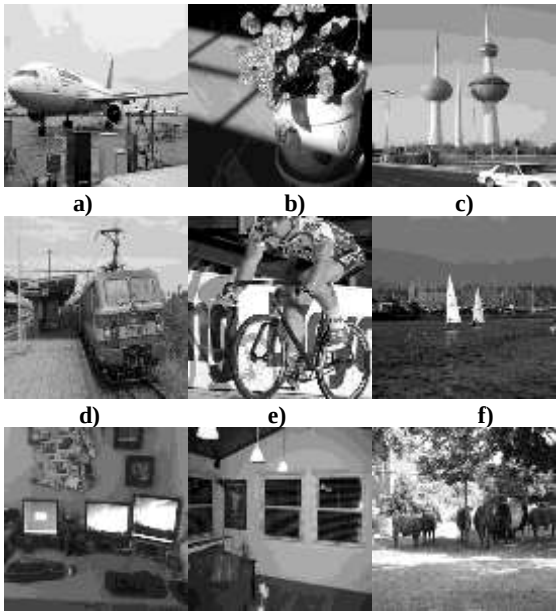


Figure 4. Benchmark Images from Pascal Voc dataset (a) Image 1, (b) Image 2, (c) Image 3, (d) Image 4 (e) Image 5, (f) Image 6, (g) Image 7, (h) Image 8 (i) Image 9.

M and N represent the dimensions (width and height) of the test image. The original image is denoted as $I(i,j)$ and the segmented image as $I'(i,j)$. Root Mean Squared Error (RMSE) measures the difference between the original and segmented images. The Structural Similarity Index Measure (SSIM) describes how similar the original and segmented images are, defined as

$$SSIM = \frac{(2\mu_I\mu_{I'} + c_1)(2\sigma_{II'} + c_2)}{(\mu_I^2 + \mu_{I'}^2 + c_1)(\sigma_I^2 + \sigma_{I'}^2 + c_2)} \quad (45)$$

The Feature Similarity Index (FSIM) is defined in [18] and depends on several statistical properties of the images I and I'. These properties include the means μ of each image, their respective variances σ, and the covariance between them $\sigma_{II'}$. Additionally, constants c_1 and c_2 are incorporated to account for variations in pixel values.

$$FSIM = \frac{\sum_{x\in\Omega} S_L(x)PC_m(x)}{\sum_{x\in\Omega} PC_m(x)} \quad (56)$$

In this equation, Ω represents the entire image. $S_L(x)$ stands for the similarity parameter, and $PC_m(x)$ refers to the phase consistency. These terms are calculated as

$$PC_m(x) = \max(PC_1(x), PC_2(x)) \quad (67\text{-a})$$

$$S_L(x) = [S_{PC}(x)]^{\alpha} [S_G(x)]^{\beta} \quad (17\text{-b})$$

$$S_{PC}(x) = \frac{2PC_1(x)\times PC_2(x) + T_1}{PC_1^2(x)\times PC_2^2(x) + T_1} \quad (17\text{-c})$$

$$S_G(x) = \frac{2G_1(x)\times G_2(x) + T_2}{G_1^2(x)\times G_2^2(x) + T_2} \quad (77\text{-d})$$

$PC_1(x)$ and $PC_2(x)$ are the phase consistency of two image blocks. Meanwhile, α, β, T_1, and T_2 are constant parameters used in the segmentation process. Generally, higher values of PSNR, SSIM, or FSIM indicate better image segmentation quality. These metrics tend to increase with stricter thresholds (higher threshold values) as the segmentation becomes more precise, focusing on capturing only the most relevant details.

Table 1 details the control parameters used by the introduced methods. These parameters were chosen based on the original research papers and fine-tuned through a trial-and-error process. To ensure consistency across algorithms, the initial population for all search methods is selected from a uniform distribution between 0 and 255. The objective is to identify the optimal threshold values for segmenting the image into four, eight, and twelve levels. To facilitate a fair comparison, all algorithms are run with 100 iterations and a population size of 100. Kapur entropy serves as the objective function in this experiment. It allows us to evaluate the algorithms' ability to escape sub-optimal solutions, a key factor for effective image segmentation.

The following tables (Tables 2 to 5) present the segmentation results obtained using Kapur's

method with different threshold values (K = 4, 8, and 12). The tables show the average fitness score (averaged over 30 runs of each algorithm) and other data associated with the best solution found in these runs.

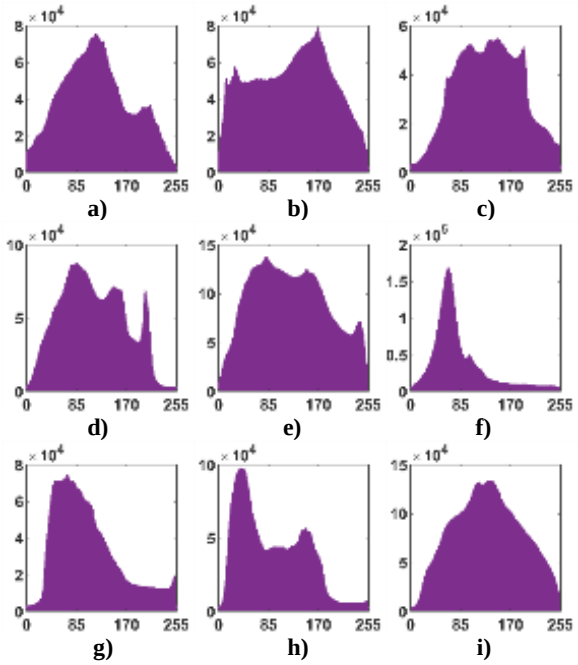


Figure 5. Analysis of energy curves for selected images. (a Image 1, (b) Image 2, (c) Image 3, (d) Image 4 (e) Image 5, (f) Image 6, (g) Image 7, (h) Image 8 (i) Image 9.

The analysis of the tables shows that DGWO outperforms other algorithms in all metrics, including PSNR, SSIM, FSIM, and the mean objective function. DGWO performed very well on test images, achieving the best rank on 20 out of the 27 in terms of mean fitness. In terms of PSNR, DGWO achieved the highest rank with 12 values. Notably, the effectiveness of alternative search methods diminishes as the thresholding size increases. These algorithms become trapped in local optima, hindering their ability to identify optimal threshold values. Conversely, DGWO consistently achieves superior objective function values across 30 runs. This success stems from its efficient exploration and exploitation of the search space, surpassing other algorithms in this capability. Therefore, DGWO, when combined with the Kapur method, establishes itself as a highly effective approach for image multi-level thresholding. Its strength lies in its ability to avoid exploration of irrelevant areas and effectively manage the diversity of its search agents. Figure 6 illustrates the segmentation results for Test Image 1, providing a better visual comparison between algorithms. Figure 7 depicts the convergence curves for test images processed using the Kapur method with a single run and 100 iterations. These curves demonstrate that DGWO generally achieves superior performance and converges more rapidly compared to other algorithms. This behavior suggests that DGWO is adept at locating optimal points within the search space.

Table 1. Search algorithm controls.

algorithm	parameters
MSO	$initial\ c_1 = 0.5$
	$initial\ c_2 = 0.05$
	$initial\ c_3 = 2$
	$Runtime = 0.2\,s$
	$MaxRuntime = 30\,s$
SAO	$M = 0.35\ to\ 0.6$
	$\beta_1 = 0.4$
	$\beta_2 = 0.9$
SDSSA	$k = 1$
	$v_0 = 0$
CVWOA	$a = 2\ to\ 0$
	$Seed = 4$
DGWO	$number\ of\ groups\ range = [1, 4]$
	$group\ population\ range = [7, 13]$
	$\max\ stagnancy = 2$

Table 2. Mean fitness comparison of algorithms with Kapur's entropy.

img	K	CVWOA	MSO	SAO	SDSSA	DGWO
1	4	19.401	19.412	19.385	19.419	**19.421**
	8	28.881	29.643	29.641	29.777	**29.781**
	12	38.118	37.772	**38.407**	37.571	38.386
2	4	19.542	19.524	19.430	19.541	**19.543**
	8	29.692	29.832	29.774	29.805	**29.921**
	12	**38.170**	37.853	38.053	37.772	38.143
3	4	19.428	19.397	19.295	19.431	**19.432**
	8	29.592	29.506	29.753	29.812	**29.937**
	12	38.145	37.678	38.354	37.910	**38.400**
4	4	19.261	19.270	19.182	**19.280**	19.280
	8	29.472	29.456	29.127	29.486	**29.577**
	12	37.790	37.270	37.323	37.480	**37.844**
5	4	19.452	19.458	19.412	19.469	**19.472**
	8	29.872	29.716	29.434	29.761	**29.930**
	12	37.269	37.836	38.143	37.711	**38.143**
6	4	19.292	19.268	19.025	19.293	**19.294**
	8	**29.764**	29.535	29.638	29.592	29.716
	12	37.376	37.338	37.634	37.995	**38.141**
7	4	19.307	19.282	19.172	19.306	**19.308**
	8	**29.872**	29.610	29.579	29.795	29.750
	12	37.546	37.913	37.360	37.773	**38.383**
8	4	19.266	19.281	19.251	19.288	**19.288**
	8	29.471	29.460	29.275	29.378	**29.560**
	12	37.836	37.224	**38.024**	37.795	37.686
9	4	19.225	19.338	19.299	19.344	**19.346**
	8	**29.684**	29.505	29.607	29.568	29.680
	12	38.215	37.408	38.184	37.622	**38.251**

Table 3. PSNR evaluation of various algorithms for multi-level thresholding with Kapur's method.

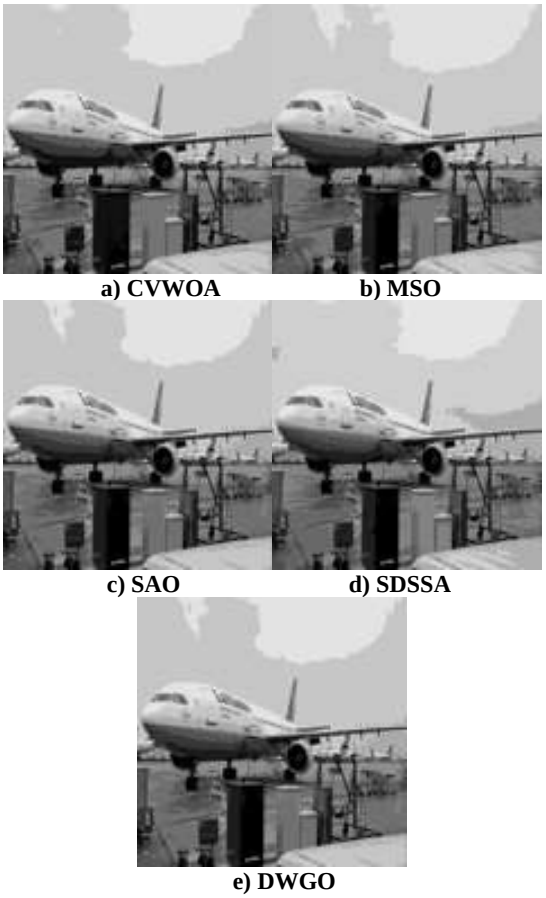
img	K	CVWOA	MSO	SAO	SDSSA	DGWO
1	4	19.73	**19.77**	19.13	19.73	19.74
	8	19.78	23.49	22.58	**24.64**	23.17
	12	26.36	26.53	27.12	26.55	**27.12**
2	4	19.70	19.68	19.67	19.72	**19.73**
	8	23.82	24.49	22.83	24.40	**24.75**
	12	27.73	24.15	26.46	26.40	**27.90**
3	4	18.11	18.16	**18.33**	18.19	18.19
	8	**23.67**	22.39	22.54	21.85	22.89
	12	26.95	26.46	**27.27**	26.59	26.83
4	4	17.14	**17.38**	16.44	17.28	17.29
	8	**24.55**	24.10	20.39	24.04	23.83
	12	26.69	26.98	25.18	25.40	**27.70**
5	4	18.34	**18.62**	17.31	18.47	18.34
	8	23.53	23.84	23.32	23.32	**23.87**
	12	24.56	25.99	26.27	23.71	**26.48**
6	4	19.32	18.92	14.98	**19.42**	19.35
	8	23.12	**24.84**	22.72	24.67	23.37
	12	23.49	27.20	26.77	26.91	**28.19**
7	4	18.22	18.21	15.91	18.26	**18.38**
	8	**24.11**	23.32	23.18	24.09	23.99
	12	24.54	25.64	26.71	24.91	**26.75**
8	4	**19.51**	18.70	18.85	18.96	18.85
	8	22.83	23.73	23.99	22.83	**24.42**
	12	27.68	24.31	**27.78**	27.24	26.25
9	4	18.18	18.49	18.26	**18.69**	18.67
	8	23.49	22.90	**23.55**	22.83	23.45
	12	26.43	25.69	26.77	25.35	**26.78**

Table 4. SSIM evaluation of various algorithms for multi-level thresholding with Kapur's method.

img	K	CVWOA	MSO	SAO	SDSSA	DGWO
1	4	0.812	0.804	0.773	**0.814**	0.813
	8	0.862	0.874	0.880	0.876	**0.885**
	12	**0.919**	0.908	0.918	0.912	0.918
2	4	0.524	**0.529**	0.525	0.525	0.522
	8	**0.755**	0.729	0.601	0.722	0.728
	12	0.842	0.622	0.789	0.803	**0.906**
3	4	0.870	**0.872**	0.821	0.869	0.869
	8	0.921	0.903	0.920	0.923	**0.924**
	12	**0.940**	0.935	0.938	0.937	0.938
4	4	0.682	0.683	**0.702**	0.676	0.677
	8	0.840	0.830	0.844	0.853	**0.867**
	12	0.879	**0.891**	0.879	0.887	0.886
5	4	**0.772**	0.770	0.701	0.763	0.754
	8	0.872	0.884	**0.887**	0.878	0.883
	12	0.893	0.906	0.908	0.873	**0.929**
6	4	0.722	0.716	0.486	0.727	**0.730**
	8	0.820	0.853	0.807	**0.854**	0.825
	12	0.797	0.881	0.894	0.889	**0.911**
7	4	0.705	**0.739**	0.581	0.700	0.711
	8	0.851	0.842	0.851	**0.854**	0.852
	12	0.862	0.875	**0.897**	0.862	0.887
8	4	**0.658**	0.606	0.628	0.622	0.612
	8	0.763	0.807	**0.860**	0.839	0.852
	12	0.896	0.865	**0.897**	0.884	0.892
9	4	0.700	0.701	0.695	**0.708**	0.708
	8	0.864	0.839	0.851	0.847	**0.865**
	12	0.916	0.909	0.925	0.900	**0.926**

Table 5. FSIM evaluation of various algorithms for multi-level thresholding with Kapur's method.

img	K	CVWOA	MSO	SAO	SDSSA	DGWO
1	4	0.814	0.817	0.797	0.817	**0.818**
	8	0.862	0.880	0.878	**0.882**	0.880
	12	0.917	0.909	**0.920**	0.909	0.919
2	4	0.800	**0.804**	0.796	0.800	0.800
	8	0.884	**0.900**	0.874	0.889	0.897
	12	0.930	0.907	0.925	0.917	**0.945**
3	4	**0.836**	0.834	0.800	0.834	0.834
	8	0.902	0.874	0.898	0.902	**0.904**
	12	0.923	0.923	0.923	0.922	**0.926**
4	4	0.771	**0.776**	0.776	0.773	0.773
	8	0.887	0.880	0.879	0.897	**0.902**
	12	0.919	0.922	0.911	0.919	**0.934**
5	4	0.844	**0.846**	0.818	0.843	0.839
	8	0.916	**0.922**	0.917	0.915	0.920
	12	0.924	0.944	**0.950**	0.916	0.949
6	4	**0.766**	0.761	0.616	0.764	0.764
	8	0.847	0.871	0.837	**0.872**	0.854
	12	0.827	0.895	0.908	0.913	**0.927**
7	4	0.738	**0.767**	0.671	0.733	0.742
	8	0.868	0.861	0.864	0.868	**0.872**
	12	0.876	0.889	0.904	0.894	**0.905**
8	4	**0.793**	0.771	0.772	0.777	0.777
	8	0.854	0.876	0.891	0.882	**0.894**
	12	0.926	0.892	**0.932**	0.926	0.912
9	4	0.805	0.825	**0.833**	0.829	0.829
	8	0.921	0.910	0.921	0.917	**0.922**
	12	0.957	0.938	0.958	0.947	**0.960**



a) CVWOA **b) MSO**

c) SAO **d) SDSSA**

e) DWGO

Figure 6. The comparison of segmentation results: Test 1, based on Kapur method (level = 8).

Execution time is considered in this section, and the studied algorithms are compared based on speed performance. Figure 8 demonstrates the time results calculated for each algorithm. Here's the ranking of algorithms by processing time (fastest to slowest): SDSSA, MSO, DWGO, CVWOA and SAO. DGWO obtained the third rank among five algorithms, indicating that it has a relatively acceptable computation time for solving problems.

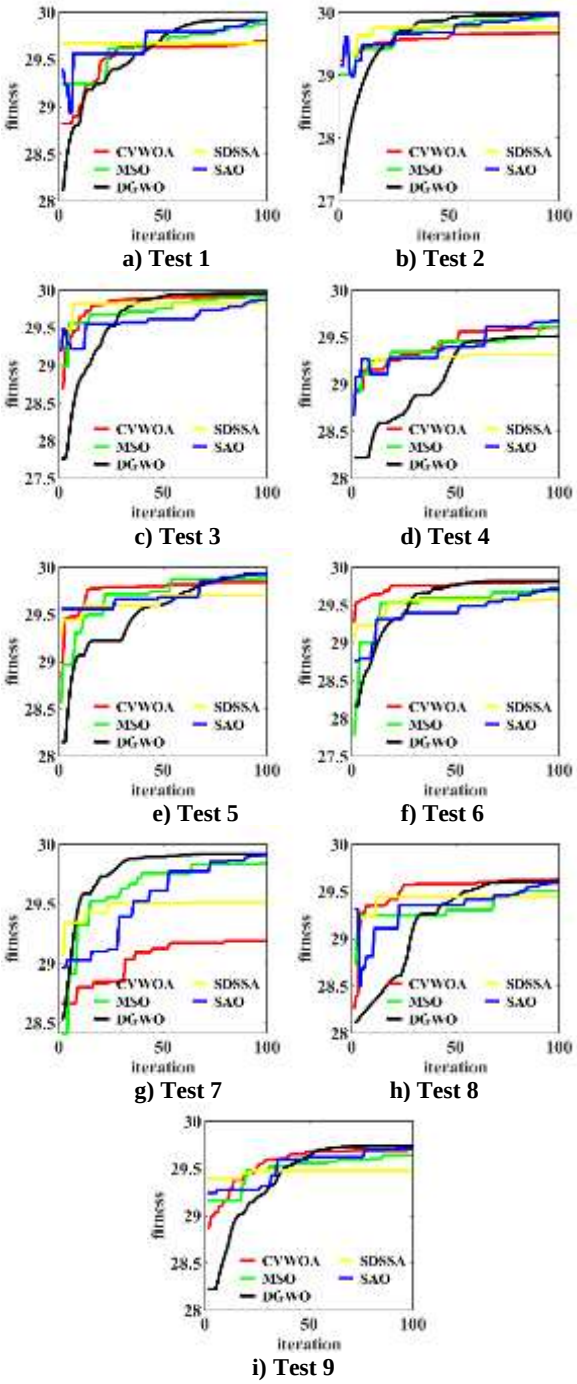


Figure 7. Convergence of Kapur's entropy-based thresholding (level 8).

The analysis proceeds by examining natural-inspired algorithms on fifteen functions of CEC 2017 dataset [19]. Various optimization benchmarks are tested to assess the algorithms' robustness. To review the definition of $F1$ to $F15$ functions, refer to [19]. With a population size of 50 and 30 independent runs, the results are displayed in Table 6. Table 6 reveals DGWO's excellent performance on 8 of 15 values and its ability to deliver good solutions in most cases. This suggests DGWO searches areas with superior sensitivity compared to other classical algorithms, leading to faster convergence to the global optimum. This advantage likely stems from a well-organized exploration mechanism that prevents individuals from getting trapped in local optima.

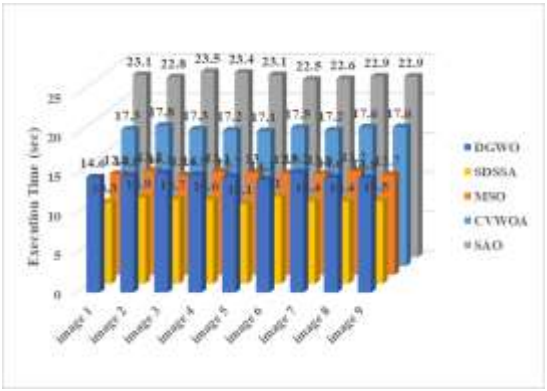


Figure 8. Comparison of execution times for common algorithms.

Table 6. mean fitness results of running various algorithms for 30 times on CEC 2017 benchmark functions $(D=15)$

func	CVWOA	MSO	SAO	SDSSA	DGWO
F1	3.78E+05	3.54E+05	**1.37E+05**	2.99E+05	1.91E+05
F2	1.80E+03	1.85E+03	**1.63E+03**	2.34E+03	1.72E+03
F3	2.44E+04	3.49E+04	2.27E+04	3.82E+04	**2.03E+04**
F4	1.15E+03	7.13E+02	9.72E+02	7.92E+02	**5.92E+02**
F5	**6.00E+02**	1.22E+03	7.12E+02	9.06E+02	6.55E+02
F6	1.25E+03	1.60E+03	1.11E+03	1.64E+03	**8.63E+02**
F7	1.24E+03	1.45E+03	1.69E+03	1.40E+03	**8.79E+02**
F8	2.72E+03	3.77E+03	3.22E+03	2.62E+03	**1.98E+03**
F9	**2.82E+03**	6.52E+03	4.97E+03	5.21E+03	3.66E+03
F10	3.97E+03	2.95E+03	2.84E+03	2.57E+03	**2.02E+03**
F11	1.40E+06	1.12E+06	1.31E+06	1.43E+06	**7.54E+05**
F12	1.25E+06	9.85E+05	1.34E+06	1.18E+06	**8.69E+05**
F13	2.85E+03	3.37E+03	**1.73E+03**	3.41E+03	1.99E+03
F14	1.76E+04	2.65E+04	3.27E+04	**1.56E+04**	1.66E+04
F15	**2.70E+05**	5.45E+05	3.29E+05	3.10E+05	2.92E+05

5. CONCLUSION

This study introduces a novel image segmentation algorithm for object-based MPEG-4 coding, combining the Grey Wolf Optimizer (GWO) with Darwinian theory concepts. By incorporating natural selection, the algorithm enhances population diversity. Utilizing the Kapur method, results are presented for three threshold values over 30 runs. Darwinian theory guides the search strategy within the search space, optimizing the movement of search agents towards the optimal solution. This combined approach (DGWO) is

applied to benchmark images, demonstrating significant improvements in objective function, PSNR, SSIM, and FSIM criteria compared to recent heuristic search methods. Future research will explore applying DGWO to other applications, such as feature selection, to evaluate its effectiveness in finding optimal solutions.

References

[1] F. Fakouri, M. Nikpour, and A. Soleymani Amiri, "Automatic Brain Tumor Detection in Brain MRI Images Using Deep Learning Methods," *Journal of Ai and Data Mining*, vol. 12, no. 1, pp. 27–35, Jan. 2024.

[2] L. F. Lyu and W. D. Zhu, "Operational Modal Analysis of a Rotating Structure Subject to Random Excitation Using a Tracking Continuously Scanning Laser Doppler Vibrometer via an Improved Demodulation Method," *Journal of Vibration and Acoustics*, vol. 144, no. 1, Jun. 2021.

[3] L. He and S. Huang, "An efficient krill herd algorithm for color image multilevel thresholding segmentation problem," *Applied Soft Computing*, vol. 89, p. 106063, Apr. 2020.

[4] Z. Xing and H. Jia, "Modified thermal exchange optimization based multilevel thresholding for color image segmentation," *Multimedia Tools and Applications*, vol. 79, no. 1–2, pp. 1137–1168, Oct. 2019.

[5] S. J. Mousavirad and H. Ebrahimpour-Komleh, "Human mental search-based multilevel thresholding for image segmentation," *Applied Soft Computing*, p. 105427, Apr. 2019.

[6] M. H. Nadimi-Shahraki, S. Taghian, and S. Mirjalili, "An improved grey wolf optimizer for solving engineering problems," *Expert Systems with Applications*, vol. 166, p. 113917, Mar. 2021.

[7] X. Zhang, Q. Lin, W. Mao, S. Liu, Z. Dou, and G. Liu, "Hybrid Particle Swarm and Grey Wolf Optimizer and its application to clustering optimization," *Applied Soft Computing*, vol. 101, pp. 107061–107061, Mar. 2021.

[8] Erwin and T. Yuningsih, "Detection of Blood Vessels in Optic Disc with Maximum Principal Curvature and Wolf Thresholding Algorithms for Vessel Segmentation and Prewitt Edge Detection and Circular Hough Transform for Optic Disc Detection," *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, vol. 45, no. 2, pp. 435–446, Aug. 2020.

[9] A. K. M. Khairuzzaman and S. Chaudhury, "Multilevel thresholding using grey wolf optimizer for image segmentation," *Expert Systems with Applications*, vol. 86, pp. 64–76, Nov. 2017.

[10] H. Song, J. Wang, J. Bei, and M. Wang, "Modified snake optimizer based multi-level thresholding for color image segmentation of agricultural diseases," *Expert Systems with Applications*, pp. 124624–124624, Jun. 2024.

[11] Mohamed Abd Elaziz, Mohammed A.A. Al-qaness, Rehab Ali Ibrahim, A. A. Ewees, and Mansour Shrahili, "Multilevel thresholding Aerial image segmentation using comprehensive learning-based Snow ablation optimizer with double attractors," *Egyptian Informatics Journal*, vol. 27, pp. 100500–100500, Sep. 2024.

[12] H. Guo *et al.*, "Multi-threshold Image Segmentation based on an improved Salp Swarm Algorithm: Case study of breast cancer pathology images," *Computers in Biology and Medicine*, vol. 168, pp. 107769–107769, Jan. 2024.

[13] J. Shi, Y. Chen, Z. Cai, Ali Asghar Heidari, H. Chen, and X. Chen, "Multi-threshold image segmentation based on an improved whale optimization algorithm: A case study of Lupus Nephritis," *Biomedical Signal Processing and Control*, vol. 96, pp. 106492–106492, Oct. 2024.

[14] J. N. Kapur, P. K. Sahoo, and A. K. C. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," *Computer Vision, Graphics, and Image Processing*, vol. 29, no. 3, pp. 273–285, Mar. 1985.

[15] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, Mar. 2014.

[16] P. Ghamisi, M. S. Couceiro, J. A. Benediktsson, and N. M. F. Ferreira, "An efficient method for segmentation of images based on fractional calculus and natural selection," *Expert Systems with Applications*, vol. 39, no. 16, pp. 12407–12417, Nov. 2012.

[17] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes Challenge: A Retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, Jun. 2014.

[18] Lin Zhang, Lei Zhang, Xuanqin Mou, and D. Zhang, "FSIM: A Feature Similarity Index for Image Quality Assessment," *IEEE Transactions on Image Processing*, vol. 20, no. 8, pp. 2378–2386, Aug. 2011.

[19] kooaslansefat, "CEC 2017 Benchmark," *Kaggle.com*, Mar. 07, 2023. https://www.kaggle.com/code/kooaslansefat/cec-2017-benchmark (accessed Aug. 27, 2024).

ارتقاء بخش بندی تصویر با استفاده از الگوریتم گرگ خاکستری داروینی: یک رویکرد آستانه‌ای چندسطحی جدید

احسان احسائیان

گروه آموزشی مهندسی برق، دانشگاه صنعتی سیرجان، ایران.

ارسال ۲۰۲۴/۰۴/۳۰؛ بازنگری ۲۰۲۴/۰۸/۰۷؛ پذیرش ۲۰۲۴/۰۸/۲۲

چکیده:

این مقاله یک رویکرد جدید برای بخش بندی تصویر با استفاده از آستانه‌گذاری چندسطحی سریع و دقیق ارائه می‌کند. در الگوریتم پیشنهادی، برای حل مشکل رایج در الگوریتم های فراابتکاری، که منجر به گیر افتادن در نقاط بهینه محلی و همگرایی زودرس می شود، الگوریتم گرگ خاکستری (GWO) و اصول داروینی ادغام شده است. عوامل جستجو توسط یک مکانیسم دوگانه تشویق و تنبیه به طور موثر در فضای جستجو به کار گرفته می شوند و در نتیجه زمان محاسباتی کاهش پیدا می کند. این امر با تقسیم‌بندی جمعیت به گروه‌های مجزا انجام می‌شود که هر کدام وظیفه کشف راه‌حل‌های برتر را برعهده دارند. برای اعتبارسنجی کارایی الگوریتم، ۹ تصویر آزمایشی از مجموعه دیتاست Pascal Voc انتخاب شد و نتایج براساس منحنی انرژی ارائه شد. همچنین از آنتروپی Kapur برای سنجش عملکرد الگوریتم پیشنهادی استفاده شد. این روش با چهار الگوریتم جستجوی متفاوت محک زده شد و برتری آن با دستیابی به بهترین نتایج در ۲۰ مورد از ۲۷ مورد برای بخش‌بندی تصویر اثبات شد. یافته‌های تجربی در مجموع تأیید می‌کنند که الگوریتم گرگ خاکستری داروینی (DGWO) به عنوان ابزاری قدرتمند برای آستانه‌گذاری چندسطحی می‌باشد.

کلمات کلیدی: آستانه گذاری چندسطحی، بخش بندی تصویر، الگوریتم گرگ خاکستری، قانون داروینی.