Int. J. Nonlinear Anal. Appl. In Press, 1–12
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2025.33478.5013

An improved grasshopper optimization algorithm for multilevel thresholding image segmentation

Leila Amiri[a], Abdolah Chalechale[a,*], Maryam Taghizadeh[b]

[a] *Department of Computer Engineering and Information Technology, Razi University, Kermanshah, Iran*

[b] *Faculty of Information Technology, Kermanshah University of Technology, Kermanshah, Iran*

(Communicated by Haydar Akca)

Abstract

Multilevel thresholding is one of the most common, straightforward, and effective image segmentation algorithms. The most important issue in this method is choosing an appropriate threshold value. In such a way that by defining worthy thresholds, the image can be more accurately segmented. The Otsu approach is suitable for establishing the thresholds at two levels, but as the number of thresholds increases, the performance of Otsu diminishes in terms of time and segmentation accuracy. On the other hand, optimization techniques can be effective to address these challenges. As a result, it is used with optimization techniques to improve time and segmentation accuracy. In this paper, we propose an improved grasshopper optimization approach to enhance the quality of the segmented image and its accuracy. In the proposed method, multilevel thresholding image segmentation is performed by employing the Otsu method as an objective function. This research aims to enhance the grasshopper algorithm to improve image segmentation outcomes. For this purpose, various modifications were applied to the grasshopper method. The proposed algorithm is evaluated on some known images and compared with several optimization algorithms. The resultant modified grasshopper method outperforms other evolutionary algorithms like Whale, Firefly, and Artificial Bee Colony (ABC) optimization algorithms. The proposed IGOA algorithm outperforms other approaches at PSNR metric for threshold levels of 32 and 64 for 87.5% and 100% of images, respectively. Additionally, at SSIM metric, for both threshold levels of 32 and 64, it overcomes other approaches for 100% of images.

Keywords: Otsu, entropy, meta-heuristic algorithm, evolutionary optimization algorithm
2010 MSC: Primary: 68U10; Secondary: 68W50, 90C59

1 Introduction

Image segmentation is a fundamental core and a pre-processing step in image processing where it is utilized in many computer vision tasks such as region proposal generation [1], object detection [2], image captioning understanding [3], and image medical analysis [4, 5, 6]. This technique divides an image into several non-overlapping regions. Here, a tag is allocated to each pixel in the segmented image, and the pixels with the same tag have similar attributes and features. In terms of color, brightness, or texture, all pixels in each region are identical whereas the adjacent

*Corresponding author
Email addresses: a9659amiri@gmail.com (Leila Amiri), chalechale@razi.ac.ir (Abdolah Chalechale), m.taghizadeh@kut.ac.ir (Maryam Taghizadeh)

Received: March 2024 *Accepted:* November 2025

regions are different from each other. In other words, image segmentation is able to separate object regions from the background. Various approaches can be used for image segmentation directly or indirectly. For example, sparse coding [7], and image thresholding [8] can be directly employed for image segmentation. Other ideas can be indirectly applied to image segmentation such as game theory [9, 10], meta-heuristic algorithms [11], and category learning [12, 13].

Image thresholding is a popular, simplest, and most effective image segmentation technique. The main aim of the threshold-based segmentation algorithms is to find an acceptable threshold value. This technique is categorized into two types, including single-level and multilevel thresholding. The single-level thresholding is a technique for splitting the pixels of an image using only one threshold. The obtained result is a binary image. However, this technique is not suitable for gray and color image segmentation. When the pixels of an image are clustered using multiple thresholds, it's referred to as a multilevel thresholding method. The image is divided into several classes. Multilevel thresholding is more accurate than a single-level one in terms of segmenting the image and representing more information about objects. In sum, choosing a single threshold or multiple thresholds is the most significant challenge in image segmentation. While appropriate thresholds are selected for image segmentation, high accuracy, and more precisely segmented regions are provided. However, in a multilevel thresholding problem, the complexity of threshold determination time and the time required to compute image segmentation are very challenging. Briefly, achieving the appropriate thresholds in multilevel thresholding is a difficult task. Therefore, multilevel thresholding is an attractive topic for many researchers nowadays.

A practical approach in the single-level thresholding techniques is to use the histogram of the image. This approach segments the image based on the distribution of the pixels. Here, the pixels that have values greater than the threshold are assigned to an identical label, and the rest pixels are considered the background. The histogram-based approaches are more efficient and accurate compared to other image segmentation techniques since they examine the pixels only once. The histogram is computed from all image pixels, and the peaks and valleys in the histogram curve are used to find the classes. In contrast, the multilevel thresholding technique based on the histogram is inefficient due to both the complexity of computation and the difficulty of implementation. The Otsu method [14] is one of the most well-known histogram-based methods that divides a binary image into two levels black and white by choosing an optimal threshold by maximizing the class variance. In summary, single-level thresholding is usually an easy process, with the threshold located at the bottom of the histogram between the two peaks. The Otsu method for multilevel thresholding is more complex. In fact, the computational time expands considerably with the number of thresholds, and hence, it is inefficient. Totally, all these methods are based on traditional exhaustive methods which are inefficient in terms of computation. Therefore, a careful implementation of the optimization algorithm is essential to obtain the optimal solution in adequate time. Meta-heuristic methods are widely used to find multilevel thresholds. As a result, these methods have attracted considerable attention from researchers.

It is important to note that Optimization algorithms are powerful tools used to find the best solution to a given problem within a specified set of constraints. In other words, optimization algorithms reduce the time for solving a problem and discovering the optimal solution out of a vast number of options. Each optimization algorithm is an adaptive model out of various models from different domains, each with its cost function (objective function, fitness function, evaluation function). In the context of multilevel thresholding, these algorithms aim to determine the optimal threshold values that maximize a certain objective function. In fact, the multilevel thresholding segmentation can be resolved using a combination of the Otsu method and the optimization algorithms [15, 16, 17, 18]. A significant advantage of this approach is to reduce the computation time. Moreover, they can explore a large search space and predict near-optimal solutions even for complex problems. Other advantages include flexibility and efficiency. In some cases, combining optimization algorithms with Otsu's method can provide advantages. For example, an optimization algorithm could be used to find multiple threshold values, while Otsu's method could be used to refine the initial estimates. This approach can be particularly useful for images with complex intensity distributions or multiple distinct objects. Many studies are now investigating the effectiveness of meta-heuristic techniques to overcome the limitations and hard optimization problems such as Otsu. For instance, genetic algorithm [19], particle swarm optimization [16], cuckoo search [20], and flower pollination algorithm [21] are used to optimize the targeted criteria.

This study investigates the Otsu method as an objective function to determine the optimal thresholds for the optimization algorithms such as the Grasshopper [22] , Artificial Bee Colony [23], Firefly [24] , and Whale [25] optimization algorithms. Furthermore, an improved version of the grasshopper algorithm is presented in the present study. The grasshopper optimization algorithm is a meta-heuristic optimization algorithm that was presented in 2017 [22] inspired by the social behavior of grasshopper insects in nature. This algorithm has attracted the attention of many researchers due to its simplicity, efficiency, scalability, and high power in solving various optimization problems. Additionally, the improved version of GOA can enhance exploration capabilities and effectively address complex problems with multiple local optima. Experimental results have shown that the accuracy of the segmentation method and the quality of the segmented image are enhanced by the proposed method.

The purpose of this paper is mainly to improve the segmentation accuracy by combining the optimization and Otsu algorithms. An improved version of the grasshopper algorithm is given in this study. The proposed strategy enhances the accuracy of the segmentation process and the quality of the segmented image. The main contributions of this paper are (1) to propose an enhanced version of the grasshopper algorithm which can assist us in achieving more precise solutions in the multilevel thresholding method, (2) to evaluate the proposed algorithm based on increasing the number of thresholds, (3) to compare the proposed algorithm with several optimization algorithms, and the proposed algorithm outperforms them all.

The present paper is structured as follows: In Section 2, a summary of some of the studies in the field of image segmentation is given. The proposed method is provided and evaluated in Section 3 and Section 4, respectively. Finally, Section 5 concludes the paper and points to future studies.

2 Research Background

In this section, we give a brief description of the existing works in multilevel thresholding image segmentation. To overcome the high computation time, a multilevel thresholding image segmentation was proposed based on the honey bee mating algorithm and cooperative learning [26]. The population initialization strategy of the method is led to be more efficient.

El Aziz et al. introduced a multilevel thresholding technique based on the Firefly algorithm for minimizing the entropy difference between the original image and the segmented image, which has promising image quality and accuracy [27]. Ishak [28] suggested a two-dimensional multilevel thresholding method for gray image segmentation utilizing quantum genetic optimization techniques, evolutionary differences, and a particular objective function to discover multiple thresholds. Naidu et al. [29] utilized the firefly optimization algorithm for image segmentation. This optimization is performed by maximizing Shannon or fuzzy entropy.

The authors in [30] used a multilevel search-based thresholding algorithm in 2019 for image segmentation. They employ Kapur's entropy and Otsu techniques with increasing class variance for determining and finding multilevel thresholds. The obtained results indicated the supremacy of the method compared to different algorithms such as firefly, particle swarm optimization, and genetic algorithm. A state transition algorithm (STA) is employed to select the optimal parameters of the fitting function based on the normalized histogram of the image [31]. To further enhance the segmentation performance, a denoising process is also accomplished after image segmentation. The best thresholds are found by minimizing the overall probability error. The method has shown a good achievement in assessing natural and medical images.

In 2018, Kotte et al. proposed a new multilevel thresholding approach, namely improved differential search (IDS) [32]. Applying the effective objective function, including Otsu and Kapur's entropy, this program selects the best thresholds. The computations are satisfactory, and the segmented image's quality is acceptable. In the work introduced by [33], a multi-level thresholding method was proposed using the improved slime mold algorithm (ASMA) in which sma is combined with an artificial bee colony (ABC). The authors performed a comparative experiment between ASMA and 11 other algorithms using 30 test functions. The experimental results have shown that ASMA has been able to provide high-quality solutions and almost does not suffer from premature convergence.

A multi-objective threshold image segmentation method based on the boost marine predator algorithm (BMPA) was proposed for infrared image error detection using the Kapoor evaluation function [34]. This method uses adaptive weights and opposition-based learning to strengthen MPA optimization and multi-objective problems. Furthermore, electrical equipment in infrared images is used to evaluate the error detection ability of the relevant method.

A multilevel thresholding method, namely the improved Slime Mould Algorithm (DASMA), was proposed by Zhao et al. [35]. The method is threefold: the diffusion mechanism is utilized to increase the population's diversity and to prevent falling into local optima. After finding the optimal solution, the algorithm is employed for Renyi's entropy multilevel threshold image segmentation. The target data set is CT images. In [36] a multi-level thresholding method for color images was proposed which uses a combination of two algorithms, including the whale optimization algorithm (WOA) and the local minima avoidance method (LMAM). It should be noted that the Otsu method is used as the objective function. In the study [37], a multi-level thresholding method is performed using an improved Chimp optimization algorithm (IChOA) on thermographic images. Opposition-based learning (OBL) is used in IChOA to increase its population diversity in the search space, and Lévy flight is employed to increase IChOA utilization. Chaotic enhanced Rao (CER) algorithm is developed using chaos theory to determine multiple threshold values. The advantages of the algorithm are simplicity with less complexity [38]. Recently, a new method based on combining

Algorithm 1: Pseudo code of the grasshopper optimization algorithm

Input: Image histogram
Output: Optimal thresholds
MaxIter=100, *Iter*=1

Initialize the population of Grasshoppers
Define GOA variables and objective function
Evaluate the fitness of each grasshopper and select the best solution
While *Iter* $\leq$ *MaxIter*
    Update C1 and C2
    For each grasshopper do
        Calculate the objective function
        Update the grasshoppers' position
        Bring current grasshopper back if it goes outside the boundaries
    End For
    Iter=Iter+1
End While
Return the best threshold

arithmetic optimization algorithm and Harris Hawks optimizer is introduced to improve exploitation phase [39]. It was used for multilevel thresholding image segmentation and indicated better quality than meta-heuristic methods.

3 Proposed Method

The proposed algorithm, namely the improved grasshopper optimization algorithm (IGOA) is explained in more detail in this section. As mentioned before, in optimization techniques, the objective function is extremely important. Therefore, it is essential to be carefully and accurately designed. The objective function should be constructed to get the best solution as quickly as possible and avoid getting stuck in the local minimum. The Otsu method is used as the objective function in meta-heuristic algorithms to find the optimal thresholds in images to improve the image segmentation performance in terms of increasing segmentation accuracy and increasing segmentation quality. The Grasshopper optimization algorithm is briefly described in 3.1, while the proposed approach is described in 3.2. Subsection 3.3 and 3.4 describe computational complexity and generalization of the GOA, respectively. Then, we explain applications og IGOA in more detail.

3.1 Grasshopper optimization algorithm

In the GOA, each grasshopper represents a possible solution to an optimization problem. Grasshoppers move in a space and update their position based on the positions of other grasshoppers and the best solution found so far. This movement is similar to the movement of grasshoppers in nature when searching for food or shelter. There are two types of relationships between grasshoppers as follows: social attraction and social repulsion. This social attraction causes grasshoppers to move towards promising areas while This social repulsion causes grasshoppers to explore the search space thoroughly. The Grasshopper algorithm starts the optimization with a random set of solutions to find the most optimal response in the search space and to achieve global optimization. Each Grasshopper is defined as a k-vector up to get the optimal thresholds, where up is the vector of the defined thresholds:

$$\vec{u_p} = (t_1, ...t_k) \quad subject \quad to \quad 0 < t_1 < ... < t_k < H \tag{3.1}$$

In the present study, the thresholds chosen by each Grasshopper are randomly determined in the range $[V_{max}, V_{min}]$. Notice that two values V_{max} and V_{min} are upper bound and lower bound for each search agent, respectively. In Eq. (3.1), H represents the maximum gray region. Each search agent updates its position. The status update is repeated until the termination conditions are satisfied. Finally, the position and the value of the best target, i.e., the optimal threshold, are the outputs of the best approximation to the global optimum. Algorithm. 1 shows the steps of determining the optimal threshold values by GOA in this study.

3.2 Improved grasshopper optimization algorithm (IGOA)

The optimization algorithms determine the value of the optimal thresholds or the position of the search agents in determining multilevel threshold image segmentation. Each meta-heuristic algorithm considers a number of agents that find the ideal thresholds over the number of iterations. Each search factor decides the threshold's location in the search space. In this paper, the search space is the same as the image histogram and the k is the number of the searched thresholds.

In this study, the objective function selects the best thresholds by maximizing within-class variance. According to Eq. (3.2), the initial image M is classified into k + 1 classes using the optimal k thresholds.

$$\begin{aligned} C_1 &= \{p(i,j) \in M \mid I_{min} \leq p(i,j) \leq t_1 - 1\} \\ C_2 &= \{p(i,j) \in M \mid t_1 \leq p(i,j) \leq t_2 - 1\} \\ C_k &= \{p(i,j) \in M \mid t_{k-1} \leq p(i,j) \leq t_k - 1\} \\ C_{k+1} &= \{p(i,j) \in M \mid t_k \leq p(i,j) \leq I_{max} - 1\} \end{aligned} \tag{3.2}$$

Based on Eq. (3.2), $p(i,j)$ is the pixel value of the matrix M in row i and column j. Two values I_{min} and I_{max} are the minimum and maximum pixel brightness in the M matrix, respectively and t_1, t_2, t_k and t_{k+1} are the corresponding thresholds, C_1, C_2, C_k and C_{k+1} are the classes belonging to each image. The threshold t_k is obtained by maximizing the objective function $F(t_1, ..., t_k)$ as given in Eq. (3.3).

$$t_1, ..., t_k = \max_{t_1,...,t_k} F(t_1, ..., t_k) \tag{3.3}$$

The image histogram containing the component N_p is presumed to be the required search space. The image pixels at different locations are classified into their specific groups based on the intended thresholds according to select appropriate thresholds by the optimization process. The number of the variables defined for each search factor (the number of each search factor's dimensions) is equal to the number of the searched thresholds, as indicated in the corresponding algorithms. We perform the following modifications.

- In this study, a constant $O = 0.75$ was added to GOA to improve its efficiency.
- We adjusted the GOA equations in the learning phase to enhance the balance between the exploration and exploitation phases:

$$C_1 = C\frac{ub_d - lb_d}{2} \tag{3.4}$$

$$C1 = log|\sum_{j=1}^{N} C(3\frac{ub_d - lb_d}{2})| \tag{3.5}$$

$$C2 = C1 + O \tag{3.6}$$

where, ub_d and lb_d are the upper and lower bounds of the search space, respectively. We replace Eq. (3.5) with Eq. (3.4) and insert a constant value to *C2*. We utilize Eq. (3.12) to solve the optimization problem based on ([22]). We inserted a constant value f to compute function s according to Eq. (3.10) and Eq. (3.11) where this function defines the strength of social forces. In the following, Eq. (3.7) shows the distance between the i-grasshopper and the *j-th* grasshopper. In addition, the distance function d_{ij} has been modified by dividing the d_{ij} using a logarithmic expression called *c3* that is computed by Eq. (3.8) and (3.9). Lastly, the value of X^d is obtained by Eq. (3.12) where X^d is the location of the grasshopper in the d-dimension. In Eq. (3.12, $\widehat{TD}$ shows the value of the Dth dimension in the target as the best solution found so far. More information has been presented in ([22]).

$$d_{ij} = |x_j - x_i| \tag{3.7}$$

$$c3 = log|2K + f + c1| \tag{3.8}$$

$$d_{ij} = |x_j - x_i|/c3 \tag{3.9}$$

$$s(r) = f \times e^{(-r/l)} - e^{-r} \tag{3.10}$$

$$s(r) = f \times e^{(-r/l)} - e^{-r} + f \tag{3.11}$$

$$X_i^d = c(\sum_{j=1}^{n} c\frac{ub_d - lb_d}{2} s(|X_j^d - X_i^d|)\frac{x_j - x_i}{d_{ij}} + \widehat{TD}) \tag{3.12}$$

The IGOA proposed is utilized the log function. The logarithm is used for the logarithmic derivative process converting products into sums. This function is useful because the functions to optimize are usually products with exponential, and detection of the extrema requires differentiation. Finally, the sum statement is more manageable as the factors are complex.

Notice that the inner c contributes to the decrease of repulsion/attraction forces between grasshoppers related to the number of iterations, as the outer c moderates the search converges around the target as the iteration count increases. Authors [22] suggested a random behavior by applying random coefficients to both terms in Eq.(3.12).

3.3 Computational complexity

To compute computational complexity, important parameters such as the number of population (N) and the problem dimension (D) are considered. The GOA is executed by the steps including: parameters definition, initial position of grasshoppers definition, objective function computation, the grasshopper position updating, and repetition of these steps until the stopping condition. Based on these steps, the complexity can be computed by multiplying the number of iterations (identified by T), dimension, and population. Therefore it can be written as $O(N \times D \times T)$.

3.4 Generalization

The Grasshopper Optimization Algorithm (GOA), while a promising meta-heuristic optimization technique, has certain limitations that it is essential consider. For the generalization of GOA, there are limitations described as follows: The GOA may require carefully tuning parameters to achieve optimal performance for specific problem domains. This can be time-consuming and may limit its generalization to a wide range of problems. Moreover, the GOA can get trapped in local optima. And it can prevent finding the global optimum solution. This is particularly problematic for complex and multimodal problems.

In addition, in some cases, GOA may converge prematurely, leading to suboptimal solutions. This can occur when the search agents become too concentrated in a particular region of the search space. The performance of GOA is sensitive to the values of its parameters, such as the population size, convergence criterion, and shape factor. Improper parameter settings can significantly impact the algorithm's effectiveness.

3.5 Application

The Improved Grasshopper Optimization Algorithm (IGOA) has shown promise in various applications. While its performance can vary depending on specific problem characteristics, there are several cases where IGOA has been generally shown to significantly outperform other methods: IGOA's efficient search strategy and reduced computational overhead can handle large-scale problems more effectively than other algorithms. It can handle constraints efficiently, therefore; it can be beneficial in real applications. Likewise, it is able to adapt to changing environments and it is suitable for dynamic optimization problems like self-driving [40]. Moreover, the IGOA can adjust to changing environments which can be beneficial for real-time applications where conditions may vary. Also, the IGOA can be effectively combined with deep learning techniques for real-time image processing tasks, such as video segmentation or medical imaging. The IGOA can be used to tune the hyper-parameters of a deep learning model for real-time foreground-background segmentation in videos. The IGOA can be used to optimize the parameters of an object tracking algorithm based on a neural network such as skin segmentation for tracking in video [41] while real-time image processing requires low latency, and IGOA can help to optimize the models for efficiency. Briefly, IGOA's enhanced exploration capabilities, efficient search strategy, ability to handle constraints, adaptability to dynamic environments. Moreover, the advantage of robustness to noise can be particularly improves these processes.

4 Analysis of the results

We perform two sets of experiments to assess the proposed IGOA for the multilevel thresholding image segmentation method. In the first experiment, the suggested approach is evaluated for several images using various criteria. For this purpose, the evaluation criteria are presented, then the results are given and discussed. The second set of experiments is to compare our method with other optimization algorithms including, WOA, GOA, ABC, and Firefly. In this section, the proposed method is evaluated and compared in terms of the processing time, the Peak Signal to Noise Ratio (PSNR), and the Structural Similarity Index Measure (SSIM) for each algorithm on each image (on average, over 30 runs). The algorithms are implemented for different thresholds such as 2, 3, 4, and 5. We also evaluated the proposed method for large values equal to 32, and 64. Fig. 1 depicts the corresponding set of the gray images used in this evaluation. The method proposed in this paper was simulated in MATLAB 2018b and implemented on a Windows-64bit environment on a laptop having Core2.

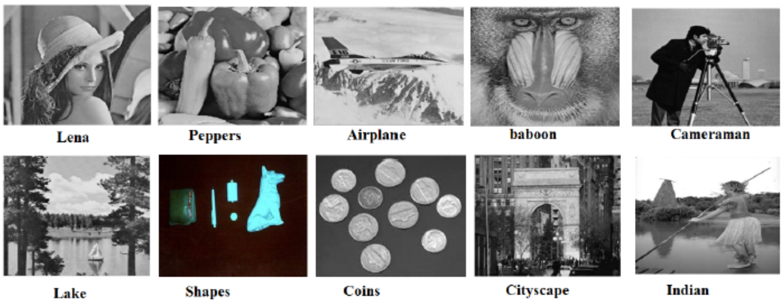

Figure 1: Set of images for evaluation

4.1 Processing time

We compute the processing time for all images in different levels and all optimization algorithms based on 30 runs. Table. 1 indicates the execution time for several selected images for Firefly, WOA, ABC, GOA, and IGOA algorithms. The highest run time is for the IGOA algorithm. As shown in Table 1, the execution time is increased when a high value is considered for the number of threshold levels. On the other hand, the shortest runtime is related to WOA for different thresholds. Furthermore, ABC has a high level of complexity, and it took more time than WOA and GOA.

Table 1: Average execution time for some images in seconds

Image	K	Firefly	WOA	ABC	GOA	IGOA	Image	K	Firefly	WOA	ABC	GOA	IGOA
Lena	2	3.1652	0.2337	3.4495	2.2587	3.5871	Lake	2	3.0544	0.2071	3.0058	2.1361	3.5142
	3	3.3193	0.2994	3.5676	2.4826	5.0314		3	3.3382	0.3367	3.2933	2.8908	5.1778
	4	3.5842	0.3315	3.6504	3.3409	6.0672		4	4.0382	0.9933	3.8713	3.4964	6.4226
	5	3.7367	0.7261	4.2059	4.3324	6.4857		5	4.1204	0.4185	3.2553	4.7529	6.8016
Peppers	2	3.0344	0.3152	3.5041	1.9509	3.9068	Coins	2	5.9684	0.5514	2.6313	1.9531	3.8885
	3	3.3320	0.3251	3.6251	2.8041	5.0718		3	3.5470	0.5754	3.1512	2.4752	4.2662
	4	3.7983	0.3352	3.6422	3.3897	6.4633		4	4.0806	0.7894	3.8425	3.36	5.1475
	5	3.9905	0.5241	3.8355	4.5614	7.1261		5	8.9785	0.7699	4.6845	4.200	8.1093
Cameraman	2	2.9407	0.5142	3.7974	2.32	3.7353	Cityscape	2	3.6524	0.7625	2.6323	0.7698	0.9875
	3	3.3299	0.3845	3.6149	2.2652	4.2348		3	4.6332	0.9685	3.2511	3.1754	3.9748
	4	3.7581	0.6524	4.0702	3.3898	6.0239		4	4.2674	0.3678	3.6426	4.3285	4.7124
	5	4.2065	0.3910	3.7991	4.4063	6.8016		5	5.2387	1.9874	4.6843	5.9125	6.2987

4.2 Structural similarity index measure

SSIM is a metric for measuring the perceptual quality of the image. The closer to 1, the more the representation resembles the original image. The SSIM value for image A and image $\acute{A}$ with size, M $\times$ N is calculated using the following equation ([26]).

$$SSIM(A, \acute{A}) = \frac{(2\mu_1\mu_S + c1)(2\sigma_{1,S} + c2)}{(\mu_1^2 + \mu_S^2 + c1)(\sigma_1^2 + \sigma_S^2 + c2)} \tag{4.1}$$

where μ_1 and μ_S denote the mean density of image A and image $\acute{A}$. Two parameters σ_1 and σ_S, respectively, indicate the standard deviation of image A and image $\acute{A}$, $\sigma_{1,S}$ indicates the covariance between image A and segmented image $\acute{A}$, and c_1 and c_2 as two fixed values equal to 6.50 and 58.52, respectively ([26]).

The average SSIM values of the individual segments for some thresholds are illustrated in Table. 2. As seen in Table. 2, in most images, the IGOA algorithm outperforms other algorithms for various thresholds. Although it does not converge too quickly, it achieves a better and more precise optimal response.

Table 2: Average SSIM values for different images and algorithms during 30 runs

Image	K	Firefly	WOA	ABC	GOA	IGOA	Image	K	Firefly	WOA	ABC	GOA	IGOA
Lena	2	0.7012	0.6969	0.6841	0.7063	**0.7192**	Lake	2	0.6724	0.6711	0.6722	0.6736	**0.6783**
	3	0.7073	0.6473	0.6412	0.7178	**0.7182**		3	0.6658	0.6552	0.6084	0.6681	**0.6692**
	4	**0.7569**	0.6969	0.6821	0.7538	0.7541		4	0.6812	0.6812	0.5719	0.6810	**0.6821**
	5	0.7138	0.7338	0.7132	0.6754	**0.7400**		5	0.6814	0.6852	0.6082	0.6879	**0.7228**
Peppers	2	**0.7156**	0.7156	0.7115	0.7123	0.7154	Shapes	2	0.8259	**0.8269**	0.8125	0.8249	0.8259
	3	0.6832	0.6832	0.6842	0.6975	**0.7016**		3	0.8150	0.8079	0.8058	0.8121	**0.8211**
	4	0.6681	0.6681	0.6885	0.6982	**0.7212**		4	0.8169	0.7417	0.8035	0.8183	**0.8295**
	5	0.6721	0.7180	0.6847	0.6719	**0.7396**		5	0.8275	0.8211	0.8175	0.8249	**0.8296**
Airplane	2	0.8257	0.8267	0.8132	0.8264	**0.8275**	Coins	2	0.8464	**0.8596**	0.7948	0.8480	0.8491
	3	0.8030	0.8022	0.7865	0.8014	**0.8096**		3	**0.8180**	0.8078	0.7912	0.8064	0.8102
	4	0.8087	0.8536	0.7882	0.7935	**0.8441**		4	0.8053	0.7858	0.7925	0.7915	**0.8090**
	5	0.8283	0.8022	0.8065	0.8193	**0.8372**		5	0.7980	0.8116	0.7932	0.7922	**0.8641**
Baboon	2	0.7536	0.7468	0.7425	0.7548	**0.7548**	Cityscape	2	0.8445	0.8445	0.8514	0.8515	**0.8575**
	3	0.7011	0.6957	0.6945	0.7075	**0.7573**		3	0.7548	0.7585	0.7536	0.7568	**0.7586**
	4	0.7168	0.7168	0.7211	0.7214	**0.7388**		4	0.8325	0.8327	0.8345	0.8352	**0.8364**
	5	0.6976	0.6895	0.6859	0.6986	**0.7101**		5	0.8325	0.8375	0.8364	0.8376	**0.8398**
Cameraman	2	0.7812	0.7632	0.7821	0.7874	**0.7991**	Indian	2	0.8346	0.8312	0.8324	0.8419	**0.8429**
	3	0.7710	0.7116	0.4739	0.7774	**0.7791**		3	0.8348	0.8445	0.8465	0.8412	**0.8472**
	4	0.7812	0.7812	0.7886	0.7923	**0.7974**		4	0.8431	0.8362	0.8511	0.8631	**0.8647**
	5	0.7813	0.7059	0.7582	0.7864	**0.7872**		5	0.8511	0.8557	0.8625	0.8627	**0.8782**

4.3 Image quality

PSNR is to evaluate image quality and the PSNR depends on the mean square error (MSE). For image A and image $\acute{A}$ with size m × n, the PSNR value per dB unit is calculated using Eq. (4.2) and Eq. (4.3):

$$MSE = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (A(i,j) - \acute{A}(i,j))^2 \tag{4.2}$$

$$PSNR = 10 \log_{10} (\frac{255^2}{MSE}) \tag{4.3}$$

When the PSNR value is equal to infinity (inf), the image equals the original image. Therefore, the high values of PSNR represent high image quality ([26]). The average PSNR values indicate different ranges for all the thresholds and the corresponding algorithms during 30 runs as summarized in Table. 3. The PSNR value of IGOA is superior to other algorithms for most images, as can be observed in Table. 3.

In addition, Fig. 2 , 3, and 4 show the results of the implementation of five WOA, GOA, IGOA, ABC, and Firefly algorithms for different threshold levels based on the proposed method for several test images.

To further evaluate our method, we select two large values for the number of thresholding levels. All results for SSIM and PSNR are shown in Table. 4 and Table. 5. Obviously, the IGOA algorithm shows promising results compared to other algorithms.

To sum up, the obtained results demonstrated that the IGOA algorithm is practical for multilevel thresholding segmentation for most of the threshold numbers. Although the WOA algorithm has been fast in our experiments, it shows less performance in other metrics for most of the test images.

4.4 Summary and Future Work

Lastly, it is important to point out how to extend the proposed method in new applications and utilize effective deep learning techniques. The Improved Grasshopper Optimization Algorithm (IGOA) can be effectively combined

Table 3: Average PSNR values for different images and algorithms during 30 runs

Image	K	Firefly	WOA	ABC	GOA	IGOA	Image	K	Firefly	WOA	ABC	GOA	IGOA
Lena	2	22.86	22.95	21.35	22.97	**22.99**	Lake	2	21.21	22.16	21.35	22.17	**22.17**
	3	21.37	20.07	21.42	21.34	**21.69**		3	20.63	20.42	20.25	20.79	**21.17**
	4	**23.85**	23.02	22.37	23.14	23.22		4	20.12	**22.25**	19.56	20.18	21.66
	5	20.21	20.43	20.14	20.24	**21.66**		5	20.32	20.43	20.78	21.49	**21.95**
Peppers	2	22.78	22.82	22.82	22.77	**22.82**	Shapes	2	27.76	**27.88**	26.65	27.85	27.87
	3	21.02	20.55	20.55	22.15	**22.25**		3	26.01	25.37	24.39	25.81	**27.28**
	4	21.59	**23.76**	19.60	21.67	23.52		4	27.18	**28.52**	25.43	27.51	28.25
	5	20.90	20.75	**23.58**	20.95	22.97		5	27.96	**28.66**	27.81	28.33	28.54
Airplane	2	24.98	24.98	23.98	24.97	**24.99**	Coins	2	25.78	**26.05**	22.65	25.85	25.91
	3	23.44	23.30	23.30	23.29	**23.83**		3	24.47	24.51	22.65	24.84	**24.95**
	4	24.48	**25.83**	22.54	23.56	25.56		4	24.16	23.79	22.37	24.02	**24.97**
	5	24.97	23.30	23.30	24.16	**25.68**		5	26.37	24.59	23.69	24.15	**26.57**
Baboon	2	24.01	20.57	20.55	24.27	**24.28**	Cityscape	2	21.29	21.64	20.56	21.43	**21.98**
	3	24.29	22.90	21.90	22.80	**25.46**		3	21.63	21.72	20.36	21.46	**21.64**
	4	23.43	**24.61**	20.65	23.09	23.75		4	22.36	**25.75**	22.75	22.85	23.68
	5	22.71	**25.63**	21.42	22.31	22.72		5	22.75	22.77	23.65	22.52	**23.96**
Cameraman	2	24.24	23.39	23.23	24.37	**24.39**	Indian	2	20.36	20.36	20.87	22.67	**22.75**
	3	**21.53**	20.27	20.42	21.08	21.11		3	20.12	21.24	21.69	21.75	**22.65**
	4	21.42	20.45	20.25	21.44	**21.55**		4	19.82	**23.72**	20.78	20.63	21.95
	5	21.51	20.84	20.30	21.26	**21.52**		5	20.74	20.36	20.91	20.85	**21.67**

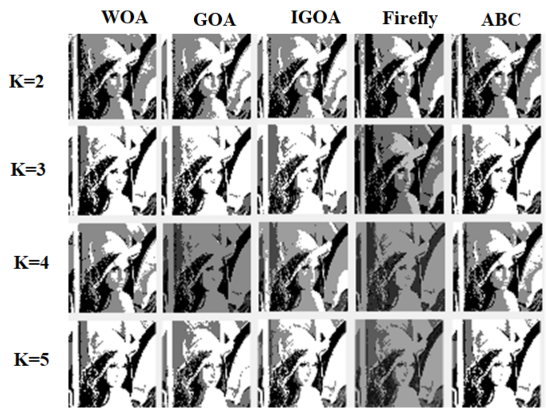

Figure 2: Segmented Lena image based on different algorithms and several values of K

Table 4: Average SSIM values for different images and algorithms during 30 runs for K=32 and K=64

Image	K	Firefly	WOA	ABC	GOA	IGOA	Image	K	Firefly	WOA	ABC	GOA	IGOA
Lena	32	0.8045	0.7230	0.7264	0.8046	**0.9113**	Lake	32	0.8261	0.8436	0.7823	0.8483	**0.8617**
	64	0.9380	0.8157	0.8512	0.9470	**0.9481**		64	0.9169	0.9125	0.8320	0.9193	**0.9707**
Peppers	32	0.8285	0.8378	0.8286	0.8371	**0.8745**	Shapes	32	0.8469	0.8196	0.8397	0.8400	**0.9440**
	64	0.8715	0.7952	0.8658	0.8668	**0.9357**		64	0.8602	0.8226	0.8367	0.8443	**0.8983**
Airplane	32	0.8611	0.8542	0.8542	0.8635	**0.9017**	Coins	32	0.9651	0.8978	0.9537	0.9638	**0.9707**
	64	0.9222	0.8542	0.9158	0.9249	**0.9311**		64	0.9639	0.9407	0.9610	0.9660	**0.9850**
Baboon	32	0.8694	0.8786	0.8796	0.8840	**0.9289**	Cityscape	32	0.9376	0.9387	0.9398	0.9404	**0.9412**
	64	0.9458	0.928	0.9435	0.9463	**0.9808**		64	0.9638	0.9652	0.9675	0.9675	**0.9682**
Cameraman	32	0.8560	0.8112	0.8837	0.8504	**0.8999**	Indian	32	0.9287	0.9225	0.9332	0.9453	**0.9556**
	64	0.8563	0.8514	0.8628	0.8673	**0.9251**		64	0.9222	0.9367	0.9628	0.9742	**0.9784**

with deep learning techniques for various image processing techniques. They can complement each other effectively in hybrid segmentation approaches, leveraging their respective strengths to achieve superior results. Generally, the IGOA is able to explore the search space effectively and therefore, it can be effective to identify optimal parameters for deep learning models. In addition, IGOA can be employed to tune hyper-parameters such as learning rates, batch sizes, and network architectures to improve model performance. It can be utilized to select powerful features from hyperspectral images, reducing computational complexity and improving segmentation accuracy. These suggestions can be implemented and tested in separate studies in the future.

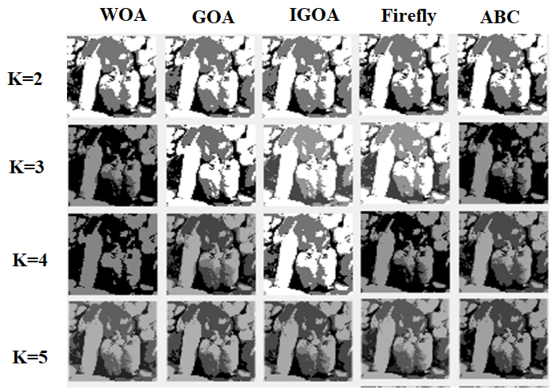

Figure 3: Segmented Peppers image based on different algorithms and several values of K

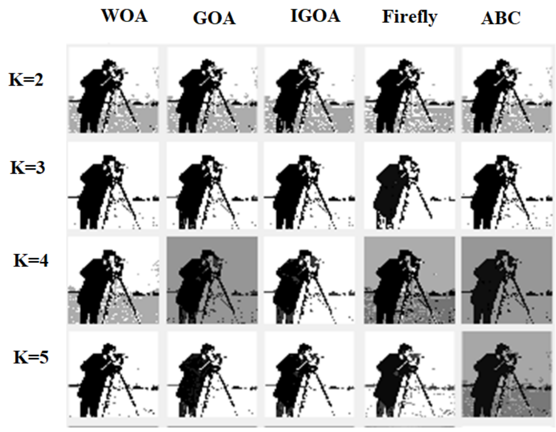

Figure 4: Segmented Cameraman image based on different algorithms and several values of K

Table 5: Average PSNR values for different images and algorithms during 30 runs for K=32 and K=64

Image	K	Firefly	WOA	ABC	GOA	IGOA	Image	K	Firefly	WOA	ABC	GOA	IGOA
Lena	32	22.03	24.42	21.69	22.93	**31.92**	Lake	32	26.38	24.86	25.69	**28.42**	26.66
	64	36.21	24.83	32.63	36.23	**36.31**		64	36.12	25.82	30.63	32.53	**37.19**
Peppers	32	24.79	27.43	26.58	27.93	**28.97**	Shapes	32	32.21	28.64	26.39	27.29	**36.95**
	64	27.10	23.26	26.69	27.66	**34.48**		64	34.72	27.07	28.36	29.80	**34.98**
Airplane	32	25.64	24.36	24.64	25.96	**28.69**	Coins	32	36.93	28.39	28.72	35.34	**37.36**
	64	31.80	25.55	30.65	31.34	**32.52**		64	39.55	31.03	32.36	35.72	**40.74**
Baboon	32	21.88	28.37	20.36	21.73	**31.23**	Cityscape	32	25.11	25.96	25.62	25.29	**29.68**
	64	30.33	31.38	29.68	31.41	**39.02**		64	26.36	32.65	25.36	36.53	**39.86**
Cameraman	32	28.01	26.55	23.83	23.68	**29.57**	Indian	32	13.88	23.85	22.85	23.67	**27.75**
	64	31.28	23.24	26.65	26.41	**32.99**		64	25.93	31.65	24.12	35.63	**38.36**

5 Conclusion

Image segmentation is an important pre-processing technique for many computer vision applications. This paper proposed an improved grasshopper optimization algorithm (IGOA) for multilevel thresholding image segmentation. The proposed method considers the Otsu method as the objective function. The accuracy of the segmentation algorithm was increased by the proposed approach based on SSIM and PSNR metrics. Additionally, we evaluated the proposed method for high values of the number of the level thresholds such as 32 and 64. The experimental results confirm that the proposed algorithm outperforms WOA, ABC, GOA, and Firefly evolutionary algorithms. In the future, deep learning-based segmentation techniques and comparisons can be investigated.

References

[1] M. Taghizadeh and A. Chalechale, *A class-independent flexible algorithm to generate region proposals*, Multimedia Tools Appl. **80** (2021) no. 16, 24697–24717.

[2] Y. Wang, L. Wang, H. Lu, and Y. He, *Segmentation based rotated bounding boxes prediction and image synthesizing for object detection of high resolution aerial images*, Neurocomputing **388** (2020), 202–211.

[3] T. Yao, Y. Pan, Y. Li, and T. Mei, *Hierarchy parsing for image captioning*, Proc. IEEE/CVF Int. Conf. Comput. Vision, 2019, pp. 2621–2629.

[4] R. Islam, S. Imran, Md. Ashikuzzaman, and Md.M. A. Khan, *Detection and classification of brain tumor based on multilevel segmentation with convolutional neural network*, J. Biomed. Sci. Engin. **13** (2020), no. 4, 45–53.

[5] N. Gupta and P. Khanna, *A non-invasive and adaptive CAD system to detect brain tumor from t2-weighted mris using customized otsu's thresholding with prominent features and supervised learning*, Signal Process.: Image Commun. **59** (2017), 18–26.

[6] O. Yousefi, P. Azami, M. Sabahi, R. Dabecco, B. Adada, and H. Borghei Razavi, *Management of optic pathway glioma: A systematic review and meta-analysis*, Cancers **14** (2022), no. 19, 4781.

[7] R. Zandi, *Sparse coding for data augmentation of hyperspectral medical images*, Master's thesis, San Jose State University, 2021.

[8] M. Sezgin and B. Sankur, *Survey over image thresholding techniques and quantitative performance evaluation*, J. Electronic Imag. **13** (2004), no. 1, 146–168.

[9] M. Zarreh, S. Yaghoubi, and H. Bahrami, *Pricing of drinking water under dynamic supply and demand based on government role: a game-theoretic approach*, Water Resources Manag. **38** (2024), no. 6, 2101-2133.

[10] S.A. Rather and S. Das, *Levy flight and chaos theory-based gravitational search algorithm for image segmentation*, Mathematics **11** (2023), no. 18, 3913.

[11] M.J. Ebadi, A. Fahs, H. Fahs, and R. Dehghani, *Competitive secant (BFGS) methods based on modified secant relations for unconstrained optimization*, Optimization **72** (2023), no. 7, 1691–1706.

[12] J.W. Suchow and V. Ashrafimoghari, *The paradox of learning categories from rare examples: A case study of NFTs & the bored ape yacht club*, Proc. Ann. Meet. Cogn. Sci. Soc. **44** (2022).

[13] T. Zhou and W. Wang, *Prototype-based semantic segmentation*, IEEE Trans. Pattern Anal. Machine Intell. **46** (2024), no. 10, 6858–6872.

[14] N. Otsu, *A threshold selection method from gray-level histograms*, IEEE Trans. Systems Man Cyber. **9** (1979), no. 1, 62–66.

[15] M.A. Elaziz, D. Oliva, A.A. Ewees, and S. Xiong, *Multi-level thresholding-based grey scale image segmentation using multi-objective multi-verse optimizer*, Expert Syst. Appl. **125** (2019), 112–129.

[16] B. Akay, *A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding*, Appl. Soft Comput. **13** (2013), no. 6, 3066–3091.

[17] Z.K. Eisham, Md. Haque, Md. Rahman, M.M. Nishat, F. Faisal, M.R. Islam, et al., *Chimp optimization algorithm in multilevel image thresholding and image clustering*, Evolving Syst. **14** (2023), no. 4, 605–648.

[18] H. Gao, Z. Fu, C.M. Pun, H. Hu, and R. Lan, *A multi-level thresholding image segmentation based on an improved artificial bee colony algorithm*, Comput. Electric. Engin. **70** (2018), 931–938.

[19] Z. Peng, L. Wang, L. Tong, H. Zou, D. Liu, and C. Zhang, *Multi-threshold image segmentation of 2d Otsu inland ships based on improved genetic algorithm*, Plos one **18** (2023), no. 8, e0290750.

[20] S. Chakraborty and K. Mali, *A multilevel biomedical image thresholding approach using the chaotic modified cuckoo search*, Soft Comput. **28** (2024), no. 6, 5359–5436.

[21] S. Mahajan, N. Mittal, and A.K. Pandit, *Image segmentation approach based on adaptive flower pollination algorithm and type II fuzzy entropy*, Multimedia Tools Appl. **82** (2023), no. 6, 8537–8559.

[22] S. Saremi, S.A. Mirjalili, and A. Lewis, *Grasshopper optimisation algorithm: theory and application*, Adv. Engin.

Software **105** (2017), 30–47.

[23] M-H. Horng, *Multilevel thresholding selection based on the artificial bee colony algorithm for image segmentation*, Expert Syst. Appl. **38** (2011), no. 11, 13785-–13791.

[24] X-S. Yang, *Firefly algorithms for multimodal optimization*, International symposium on stochastic algorithms, Springer, 2009, pp. 169–178.

[25] S. Mirjalili and A. Lewis, *The whale optimization algorithm*, Adv. Engin. Software **95** (2016), 51–67.

[26] Y. Jiang, W-C. Yeh, Z. Hao, and Z. Yang, *A cooperative honey bee mating algorithm and its application in multi-threshold image segmentation*, Info. Sci. **369** (2016), 171–183.

[27] MA.El Aziz, AA. Ewees, and AE. Hassanien, *Whale optimization algorithm and mothflame optimization for multilevel thresholding image segmentation*, Expert Syst. Appl. **83** (2017), 242–256.

[28] A.B. Ishak, *A two-dimensional multilevel thresholding method for image segmentation*, Appl. Soft Comput. **52** (2017), 306–322.

[29] M. Naidu, P.R. Kumar, and K. Chiranjeevi, *Shannon and fuzzy entropy based evolutionary image thresholding for image segmentation*, Alexandria Engin. J. **57** (2018), no. 3, 1643–1655.

[30] S.J. Mousavirad and H. Ebrahimpour-Komleh, *Human mental search-based multilevel thresholding for image segmentation*, Appl. Soft Comput. **97** (2020), 105427.

[31] J. Han, C. Yang, X. Zhou, and W. Gui, *A new multi-threshold image segmentation approach using state transition algorithm*, Appl. Math. Modell. **44** (2017), 588–601.

[32] S. Kotte, P.R. Kumar, and S.K. Injeti, *An efficient approach for optimal multilevel thresholding selection for gray scale images based on improved differential search algorithm*, Ain Shams Engin. J. **9** (2018), no. 4, 1043–1067.

[33] X. Chen, H. Huang, A.A. Heidari, C. Sun, Y. Lv, W. Gui, G. Liang, Z. Gu, H. Chen, C. Li, et al., *An efficient multilevel thresholding image segmentation method based on the slime mould algorithm with bee foraging mechanism: A real case with lupus nephritis images*, Comput. Bio. Medicine **142** (2022), 105179.

[34] Z. Xing and Y. He, *Many-objective multilevel thresholding image segmentation for infrared images of power equipment with boost marine predators algorithm*, Appl. Soft Comput. **113** (2021), 107905.

[35] S. Zhao, P. Wang, A.A. Heidari, H. Chen, H. Turabieh, M. Mafarja, and C. Li, *Multilevel threshold image segmentation with diffusion association slime mould algorithm and Rényi's entropy for chronic obstructive pulmonary disease*, Comput. Bio. Medicine **134** (2021), 104427.

[36] M. Abdel-Basset, R. Mohamed, N.M. AbdelAziz, and M. Abouhawwash, *Hwoa: A hybrid whale optimization algorithm with a novel local minima avoidance method for multi-level thresholding color image segmentation*, Expert Syst. Appl. **190** (2022), 116145.

[37] E.H. Houssein, M.M. Emam, and A.A. Ali, *An efficient multilevel thresholding segmentation method for thermography breast cancer imaging based on improved chimp optimization algorithm*, Expert Syst. Appl. **185** (2021), 115651.

[38] Y. Olmez, A. Sengur, G.O. Koca, and R.V. Rao, *An adaptive multilevel thresholding method with chaotically-enhanced Rao algorithm*, Multimedia Tools Appl. **82** (2023), no. 8, 12351–12377.

[39] L. Qiao, K. Liu, Y. Xue, W. Tang, and T. Salehnia, *A multi-level thresholding image segmentation method using hybrid arithmetic optimization and Harris hawks optimizer algorithms*, Expert Syst. Appl. **241** (2024), 122316.

[40] Y. Shi, Y. Li, J. Fan, T. Wang, and T. Yin, *A novel network architecture of decision-making for self-driving vehicles based on long short-term memory and grasshopper optimization algorithm*, IEEE Access **8** (2020), 155429—155440.

[41] N. Razmjooy, S. Razmjooy, Z. Vahedi, V.V. Estrela, and G.G.D. Oliveira, *Skin color segmentation based on artificial neural network improved by a modified grasshopper optimization algorithm*, Metaheuristics and Optimization in Computer and Electrical Engineering, Springer, 2020, pp. 169-–185.