A Novel Improved Bat Algorithm Based Image Multi-Thresholding

Safia Yassad Mokhtari and Mohamed Tahar Kimour

Embedded Systems Laboratory (LASE),
University of Badji Mokhtar, Annaba, 23000, Algeria
safiay03@yahoo.fr, kimour@yahoo.com

Abstract: Image segmentation is a very important activity in computer vision, where critical applications are highly dependent on the efficacy of such activity. To enhance the efficiency of such automated activity, meta-heuristic algorithms to optimally elucidate multi-level image segmentation problems have been proposed in the literature. Because of the advantages in terms of efficiency and convergence speed of the bat algorithm, this paper presents a novel improvement of such algorithm for solving the image multi-thresholding problem. The algorithm leads to speed up the convergence and increase diversity through the utilization of an appropriate crossover operator and chaotic sequences, with the use of Kapur's entropy as the optimized objective function. The proposed method produces segmented images with optimal values for the threshold in few iterations. Through the comparative analysis based on standard deviation, peak signal to noise ratio (PSNR) and segmented image quality, it is observed that the effectiveness of the proposed method, validated using different standard test images, outperforms well-known metaheuristic-based optimization techniques.

Keywords: Multi-thresholding, Image segmentation, Bat optimization algorithm, Entropy.

1. Introduction

Image segmentation is a fundamental process in many image, video, and computer vision applications. It has been widely applied in various fields, such as medical image analysis, image classification, object recognition, object tracking and motion estimation, and so on [1-2-3]. As a commonly used image segmentation algorithm, threshold segmentation selects proper threshold to divide image into different areas or classes [6-8]. Typically, thresholding is one of the most commonly used techniques in image segmentation due to its efficiency and simplicity, accuracy and robustness [2-5]. It is based on histogram, where in an image pixels in a region can share their intensities. it distinguish with light and dark regions [7-8].

However, since proper segmentation depends on adequately computed thresholds, determination of the multilevel thresholds is crucial in image segmentation. The automatic selection of a robust optimum n-level threshold remains a challenge in segmentation of images. As it is time-consuming, reliable and accurate image segmentation process is challenging to achieve. Hence, researches were interested with employing heuristic and meta-heuristic algorithms to elucidate multi-level image segmentation problems optimally.

By introducing the optimization methods, one reduces the time consumption and computation and produces better robustness and accuracy by optimizing objective functions. To achieve an objective functions optimization, the usually used means of the determination of optimal threshold values, is the analysis of the histogram characteristics [4].

Kurban et al. [5] had conducted comparative studies of the applications of evolutionary and swarm-based methods in image multi-thresholding. According to the statistical analyses, population-based methods are more precise in solving image multi-thresholding problems.
Various metaheuristic algorithms such as Particle Swarm Optimization (PSO) [23][39-40], Bat algorithm [12][22], and its variant IBA[25], have been applied to multilevel thresholding.

PSO is a population-based optimization algorithm. It searches for a solution by altering the directions of individuals, called particles. The algorithm starts with a random initialization of the each particle's position and velocity within the parameter space. It searches for the global optimum in a multi-dimensional parameter space, using both the best position found by all

Received: October 31th, 2017. Accepted: June 30th, 2019
DOI: 10.15676/ijeei.2019.11.2.13

particles and the best one found by itself in the search process. In order to find the minimum or (maximum) value of a user-defined objective function, velocity and position of each particle are updated in successive iterations.

As one of the most promising meta-heuristic methods, Bat algorithm (BA) [12] is a bio-inspired algorithm that has been proved its efficiency in various applications. In order to improve standard BA, [25] presents a modified BA algorithm to improve its exploration capabilities. The new step of each bat is controlled by the position vector, the global best position, and frequency. However, one problem found in the standard PSO, BA, and IBA is that they could easily fall into local optima in many optimization problems.

One reason for them to converge to local optima is that particles in PSO or bats in BA can quickly converge to the best position once the best position has no change in a local optimum. at times the standard bat algorithm as well as its modification versions may get trapped into local optima when they are applied to some difficult problems.

In this paper, we propose a Novel Improved Bat Algorithm (NIBA) to search for multilevel thresholds using Kapur's entropy for image segmentation. The proposed modification lies in the evolution rules and the technique to build the initial population using chaotic sequences concept for solving the image multi-thresholding problem. The algorithm leads to speed up the convergence and increase diversity through the utilization of a genetic algorithm crossover and chaotic sequences, with the use of Kapur’s entropy as the optimized objective function.

NIBA uses Kapur’s optimal entropy threshold method [16-17], without requiring prior knowledge, while producing good segmentation results. The experimental results show that multilevel image segmentation with NIBA has exciting advantages in terms convergence speed and efficient segmentation quality, giving better results than the particle swarm algorithm and previous modified Bat algorithms.

The remainder of the paper is organized as follows. Section 2 describes the multilevel thresholding problem and presents Kapur’s objective functions. Section 3 describes the original BA and NIBA algorithms adopted to search for the optimal multilevel thresholds, respectively. Section 4 shows the experimental results of applying NIBA and compares them with those of PSO [14] and the recently improved BA [25] using some standard benchmark images. Finally, our conclusions and future works are discussed in Section 5.

2. Multilevel Image thresholding

Due to its simplicity, compact storage space, fast execution, cost, and real-time applicability, image thresholding has been widely adopted for image segmentation [26, 27]. It is based on the information contained in the global gray value of the image histogram.

The process of image thresholding has to be formulated by taking image elements or image features as parameters to get the optimized objective function values with the purpose of getting close to the optimal thresholds [29].

Multilevel Image thresholding needs a set of thresholds. Based on that, the image can be segmented into different regions. Thresholding attempts to identify and extract an object from its background on the basis of the distribution of gray levels or texture in the image object [28].

A. Pixel grouping

Assume that an image can be represented by L gray levels. The gray level for each pixel can be represented by I(x,y), where x,y represents the coordination point of a pixel. Then the output image can be formulated by Equation (a):

$$g(x,y)=\begin{cases} v1 \qquad \square 0 \le I(\mathrm{x,y}) \le s1 \\ v2si \le I(\mathrm{x,y}) \le si+1 \\ vmsm \le I(\mathrm{x,y}) \le L \end{cases} \qquad \text{(a)}$$

Where s_i (i = 1, 2 . . . , m) stands for i^{th} threshold and m is the number of thresholds. As the threshold value is optimized, the image can be segmented into m + 1 regions.

The key point is to determine s_i and its optimization. To realize the optimization of the thresholds, the objective function has to be initialized. The maximization or minimization of the objective function represents the optimal value and also ensures the optimization of image multi-thresholding results.

S={s_1, s_2, . . . , s_m} indicates different thresholds. Therefore, image multi-thresholding can be described as the problem of searching optimal values of the elements of S that achieves the optimal objective function. By maximizing the objective function to determine the optimal thresholds, we use the Kapur's entropy [17], which is a well-known technique to solve the thresholding problem

B. Concept of Kapur's Entropy for Image Thresholding

The intelligent optimization algorithm is linked with the image multi-thresholding through objective functions to get better segmentation results. In doing so, the population based segmentation method using Kapur's entropy [17] could produce better image thesholding. Kapur's method can be easily extended from bi-level thresholding to multi-level thresholding, and, with the entropy reaching maximization, the optimal thresholds are naturally deployed in the image's histogram.

Entropy of the discrete information can be obtained by the probability distribution p = p_i, where p_i, is the probability of the system in possible state i [30].

The probability for each gray level i is represented by its relative occurrence frequency, equalized by the total number of gray levels as shown in Equation (b):

$$pi = hi / \sum_{i=1}^{n} h_i \; i = 0, \ldots\ldots, L-1 \text{(b)}$$

Kapur's entropy is used to measure the compactness and separability of classes. For MT, Kapur's entropy can be described as in the following equations (c):

$$H0 = -\sum_{i=0}^{t1-1} \frac{pi}{\omega 0} ln \frac{pi}{\omega 0}, \omega 0 = \sum_{i=0}^{t1-1} pi$$

$$H1 = -\sum_{i=t1}^{t2-1} \frac{pi}{\omega 1} ln \frac{pi}{\omega 1}, \omega 1 = \sum_{i=t1}^{t2-1} pi$$

$$Hj = -\sum_{i=tj}^{tj+1-1} \frac{pi}{\omega 1} ln \frac{pi}{\omega j}, \omega j = \sum_{i=tj}^{tj+1-1} pi$$

$$Hm = -\sum_{i=tm}^{tL-1} \frac{pi}{\omega 1} ln \frac{pi}{\omega m}, \omega m = \sum_{i=tm}^{tL-1} pi$$

Thus, the function f (S) can be obtained by Equation (d).

$f(S) = \sum_{i=1}^{m} -H_i \, , S = s1, s2, \ldots, sm(d)$ Where, S represents a vector quantity of thresholds.

The optimization process searches the values of S that maximize the objective function defined by the equation (d). To do this, we propose a novel improved bat algorithm (NIBA) that introduces new mechanisms to diversify the initial population and to increase its convergence.

3. Standard Bat Algorithm (BA)

BA is an optimization algorithm introduced by Yang [12]. It is inspired by the echolocation behavior of natural bats in locating their foods. It is used for solving various optimization problems. The main advantage of BA is that it can provide very quick convergence at a very initial stage by switching from exploration to exploitation. It is potentially more powerful than PSO and GA [25].

BA is found very efficient in solving difficult problems. This algorithm has been advanced hurriedly and has been practical in different optimization jobs. BA achieves the searches via updating the status of each bat. At the beginning, it randomly initiates a group of bats (random solutions) with a specific position and velocity for each [22-24]. It updates the status (its velocity and position) of each bat using the recorded best position experienced for this bat and the best position of the whole swarm until now. Each bat flies randomly with velocity v_i at position x_i with a fixed frequency, varying wavelength and loudness to seek for prey [22][31]. The frequency factor f controls the step size of a solution in BA [12]. This factor is assigned to a random value for each bat (solution) between upper and lower boundaries [$f_{\min}$, $f_{\max}$]. The position x_i and velocity v_i of each bat should be defined and updated during the optimization task [22].

The flow of BA can briefly be described as follows [12].

A. Initialization of Bat Population

Initial population is randomly generated from real valued vectors with dimension d and a given number of bats and, by taking into account lower and upper boundaries.

$$x_{ij} = x_{min\ i} + rand\,(0,1)(x_{max\ j} - x_{min\ j}) \tag{1}$$

Where i=1, 2,… n, j=1, 2,…. d, x_{minj} and x_{maxj} are lower and upper boundaries for dimension j respectively.

B. Evolution process

The new solutions x_t^t and velocities v_t^t at time step t are computed by the following equations [12]:

$$f_i = f_{\min} + (f_{\max} - f_{\min})\,\beta \tag{2}$$

$$v_i^t = v_i^{t-1} + (x_i^t \text{-xbest})\ f_i \tag{3}$$

$$x_i^t = x_i^{t-1} + v_i^t \tag{4}$$

Where β ϵ [0, 1] indicates randomly generated number, xbest represents current global best solutions (exploration) that is located when all solutions are compared with each other among all the n bats.

In every iteration, a random number is generated and compared with a pulse emission rate r_i, and $r_i \in [0, 1]$.

If the random number is greater than r_i, a local search part of the algorithm (exploitation) is performed by generating a new position around the current best solutions, as follows:

$$X_{old} = X_{old} + \varepsilon \overline{A^t} \tag{5}$$

Where $\overline{A^t}$, is the average loudness of all bats at this time step. ε∈[-1, 1] is a random number and represents the direction and intensity of random-walk.

C. Update Process of Loudness and Pulse Emission Rate

As the loudness usually reduces once a bat has found its prey, while the rate of pulse emission increases, the loudness can be selected as any value of convenience.

Loudness (A_i) and pulse emission rate (r_i) should be updated only when the global near best solution is updated and the randomly generated number is smaller than A_i. Loudness (A_i) and pulse emission rate (r_i) are updated by the following equations [25]:

$$A_i^{t+1} = \alpha * A_i^t \tag{6}$$

$$r_i^{t+1} = r_i^0[1 - e^{-\gamma t}] \tag{7}$$

Where α and γ are constants included in [−1, 1]. The algorithm iterates until the termination criteria is met.

The loudness A_i and the rate r_i of pulse emission have to be updated as the iterations proceed. As the loudness decrease once a bat has found its prey, while the rate of pulse emission increases, the loudness can be chosen as any value of convenience. When the loudness reaches the minimum A_{min}, it means that the bat found the prey and stop emitting any sound.

The choice of parameters requires some experimenting [25]. Each bat should have different values of loudness and pulse emission rate. The loudness and emission rates are updated only if the new solutions are improved, meaning that these bats walk towards the optimal solution.

The standard bat algorithm has many advantages; one of them is that it can get quick convergence at initial stages by switching from exploration to exploitation. This makes it an efficient algorithm when a quick solution is needed [31-32] [37-39]. In order to improve the performance, many modifications have been added to increase the diversity of the solution and to enhance the performance of the standard Bat algorithm as mentioned previously.

4. Proposed Improved Bat Algorithm

Bat algorithm is simple to implement and produces good results. However, based on some experiments, it is powerful in intensification, but at times it may get trapped into local optima when it is applied to some difficult problems such as the multi-thresholding segmentation image. Therefore, we propose an improved version of bat algorithm adopted to search for multilevel thresholds using Kapur's entropy as a objective function to optimize.

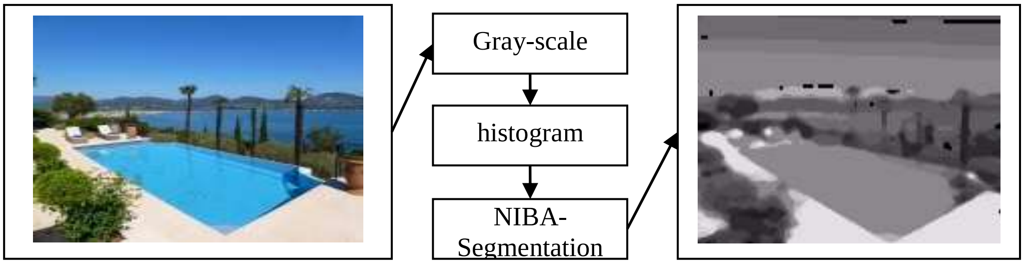

Figure 1. Flow chart of our NIBA-based multi-thresholding segmentation image.

The flow chart of the proposed method to multilevel image thresholding segmentation is illustrated in Figure. 1. Indeed, the input image is first converted into a grayscale image after removing the noise, and is used to generate a histogram. The latter is further given to our New Improved Bat Algorithm (NIBA) to obtain multi-level thresholds using Kapur's entropy as an objective function. The obtained thresholds are used to segment the image into a number of regions. Our NIBA algorithm aims to speed up convergence and increase diversity in the initial population. This was done by a modification of the standard bat algorithm with integrating the chaotic sequences and an appropriate crossover mechanism.

A. Generating the initial population by using chaos method

The initialization mode of population greatly influences the convergence performance of the evolutionary algorithms for all populations. Thus, it is needed to make the initial populations distribute as evenly as possible in the whole search space [39]. To do this, we propose the use of the Chaotic sequence instead of the random generation of the initial individuals of the population. Recently, chaos combined with metaheuristic algorithms and produced good results [35-36]. Evolutionary optimization algorithms can enhance its capability of searching global best solution using chaotic sequences [35].

The complex behavior of non-linear deterministic system is defined by chaos [33-34]. Chaos has non-repetition property and for this it searches best solution faster than any searching strategy that depends upon the probability distribution. It also has ergodicity property [33]. Chaotic sequence shows ergodicity property which helps in better searching. We employ logistic equation to produce initial population. Below is the Logistic mapping equation (8):

$$x_{m+1,n} = \mu x_{m,n}\left(1 - x_{m,n}\right) \quad (n=1.2\ldots, N{:}m=1.2\ldots m) \qquad (8)$$

where, when $X_{m,n} \in [0,1]$, and $0 < \mu \leq 4$, the system is in a chaos state and its track is with favorable ergodicity [38].

$X_{m,n}$ was transformed according to Formula 9 to obtain the initial population with size of M and dimension of N.

$$x_{m,n} = L_n + x_{m,n}(U_n - L_n) \qquad (9)$$

where U_n and L_n are respectively the upper bound and lower bound of variables or bats in the n^{th} dimension. Producing uniform population by using Logistic chaos mapping can improve the searching efficiency of the algorithm.

B. Escaping from the Local optimum

On the condition, once the position of the optimal bat is locally optimal, the algorithm is highly likely to be premature. Indeed, in the standard BA algorithm, the flying direction of each bat is almost determined based on the global best solution. We denote this direction the fast one. Moreover, each bat always flies to the best bat of the whole swarm, and also hangs around its own best position experienced. In doing so, we argue that it is dangerous when the best bat of the whole swarm is trapped into a local optimum [23]. In order to reduce the possibility of being trapped into the local optimum, we propose creating another direction for each bat, instead of just one direction.

We consider the velocity evolution using the current best solution as a fast direction bat evolution (formula (3)) and when using the worst solution in the velocity evolution as a slow direction bat evolution (formula (10)).

$$v_i^t = v_i^{t-1} + (x_i^t - x_{worst})\, f_i \tag{10}$$

Since the slow direction might not always be worse than the fast direction, the new algorithm will choose the best one from the two possible flying directions to update each bat.

Thus, NIBA algorithm chooses the best one from the two potential flying directions to update each bat. This ensures the enlarging the global searching space of bats, and enables them to avoid being trapped into a local optimum too early and in the same time.

Both the worst information mechanism and the best information mechanism compete with each other. The competition not only further decreases the possibility of being trapped into a local optimum, but also makes the search converge to optimal solutions, although its complexity is only two times of the standard BA. In doing so, NIBA improves the possibility of finding the global optimum in the search space. Since it has to calculate two potential velocities and positions for each bat in the swarm, its time complexity is twice of the standard BA.

The optimization process of NIBA searches the values of (x_{i1}, x_{i2}, …, x_{im}): that maximize the objective function. The latter represents the fitness function.

$$f(x_i) = \sum_{i=1}^{m} -H_i \ , \quad x_i = x_{i1}, x_{i2}, \ldots, x_{im}$$

C. NIBA-based multilevel thresholding

In order to enhance accuracy and speed of multilevel image thresholding, we propose a Novel Improved Bat algorithm (NIBA). NIBA searches the optimal threshold values, on which the image segmentation is performed. To do this, we use Kapur's entropy as objective function to be optimized. Thus, we start the segmentation process by the initialization of the bat population n number of solutions, each of which is a D-dimension vector. For every solution representing candidate threshold vector. Xi denotes the i^{-th} bat position in the population, which indicates a candidate thresholds and its fitness will be measured by Kapur's entropy function. The Pseudo-code of NIBA algorithm is described below by Algorithm 1.

Algorithm 1. Image segmentation with NIBA.
Input: RGB Image
Output: The final segmented image with selected thresholds based on the optimal parameters.

1. Image input, noise remove, convert to gray scale;
2. Fix the NIBA parameters (Iter: maximum number of iterations, n: number of individuals in the population, m: dimension, (f_{min}, f_{max}), r_i, α and $\gamma \in [-1, 1]$;

3. Initialize bat population Xi (i = 1, 2, . . . , n) with velocity V_i , with size of n and dimension of m, according to the chaotic model depicted by the equations 8 and 9;
x_i ($x_{i1}, x_{i2}, \ldots, x_{im}$): represents a vector of the threshold values in the image. $x_i \in [0, L-1]$. m is the number of desired classes (segments) in the image. L is the maximum pixel gray scale //intensity in the image.
4. Define frequency f_i at x_i;
5. Initialize pulse emission rate r_i and loudness A_i;
6. Calculate for each x_i the fitness function;
7. Sort all x_i according to the fitness value in ascending order;
8. While (t<Iter)
9. Generate new solutions using Equations (2, 3, 4) for the fast direction solution and Equations (2, 10, 4) for slow direction solution;
10. Select the solution among the two new solutions;
11. If (rand > ri)
12. Select a solution among the best solutions;
13. Generate a local solution around the selected best solution;
14. End If
15. If (rand< A_i and $f(x_i) < f(xbest)$)
16. Accept new solutions;
17. Increase r_i reduce Ai;
18. End If
19. Ranks the bats and find current best solution;
20. End While
21. Display results.

5. Experiments and Discussion

The multilevel image thresholding problem deals with finding optimal thresholds within the range [0, L−1] that maximize the function defined by:

$$f(S) = \sum_{i=1}^{m} -H_i \quad S = s1, s2, \ldots, sm$$

The dimension of the optimization problem is the number of thresholds m, and the search space is delimited by the interval [0, L −1]. In this study our proposed NIBA-based image multithresholding method was compared against three other methods that use standard populations based meta-heuristic techniques such as PSO [14] and IBA [25]. The proposed algorithm has been tested under a set of benchmark images. These images are widely used in the multilevel image segmentation literature to test different methods (Cameramen, Lena, Baboon, etc.).

The experiments were carried on Intel core I5 platform with a 2.5 GHz processor and 4 GB memory, under the Windows 7 operating system. The quality of the solution of the method that utilized the NIBA and the others that employed PSO and IBA are compared on the basis of the value of the best fitness, which is calculated using the entropy of partition. Parameters for NIBA are fixed as follows:

Parameter	Signification	value
Nb	Number of bats	40
MaxIter	Maximum of iterations	50
L	Gray-scale of test image	256
ri	Rate of pulse emission	[0,1]
Ai	Loudness	[1,2]

Furthermore, to show the visual quality and stability of the algorithm respectively, results are also compared using standard deviation and peak signal to noise ratio (PSNR). PSNR is used as quality measurement between original image and resulted segmented image. PSNR provides the information of similarity of an image against original image using the rooted mean square error (RMSE) of each pixel as defined by Eq.20.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{X}\sum_{j=1}^{Y}(I(i,j)-g(i,j))^2}{X*Y}} \tag{20}$$

Where I and g are respectively original image and segmented image of size X × Y. Eq.21 represents the PSNR, measured in decibel.

$$PSNR = 20\log\frac{255}{RMSE} \tag{21}$$

To analyze the stability of the algorithm, standard deviation of the latter is calculated using the Eq.22.

$$\text{standarddeviation} = \sqrt{\frac{\sum_{i+1}^{m}\sigma_i-\mu}{m}} \tag{22}$$

m is the total number of executions, σ_i is the best objective value of i^{th} execution and μ is the mean value of σ.

Both the optimization techniques PSO and IBA were operated in their standard version and its published one respectively. The size of population in PSO, IBA and in NIBA algorithms were set to 40. For fair comparison, both algorithms were executed for the same number of iterations: 60. The number of thresholds calculated in this paper was 3-10.

Table 1: Objective function and optimal threshold based on Kapur's entropy criteria.

Image Data Set	clstrs	Entropy			Optimal Threshold		
		PSO	IBA	NIBA	PSO	IBA	NIBA
Lena	2 3 4	0.68612387 1.04407915 1.34352337	0.68312703 1.07461005 1.33324010	0.69312714 1.08461015 1.39324111	105 81.130 67.103.155	104 70.176 64.100.158	102 80.136 68.103.156
Baboon	2 3 4	0.68412634 1.08717311 1.31451189	0.68822211 1.05831141 1.34618113	0.69312711 1.09831155 1.38628943	92 69.135 44.121.153	90 67.137 44.101.157	96 62.130 44.100.168
Camera men	2 3 4	0.65212339 1.09837147 1.32205431	0.65312112 1.09140632 1.31817261	0.68312714 1.09940670 1.35917481	145 132.165 121.144.29	144 138.168 120.149.18	140 145.178 120.189.17

Table 2. Comparison of computation time in sec.

Image Data Set	clstrs		Entropy	
		PSO	IBA	NIBA
Lena	2 3 4	0.65345479 0.75930551 0.81496440	0.68345400 0.75240057 0.85196461	0.5/643300 0.65028122 0.74160358
Baboon	2 3 4	0.68820231 0.77100527 0.82905440	0.68601234 0.74111520 0.82105444	0.59577639 0.67716752 0.73666621
Cameramen	2 3 4	0.62471106 0.69171003 0.76007316	0.61445107 0.69111250 0.76087380	0.53576717 0.59570314 0.69496825

For PSO, other control parameters w_max =0.4, w_min=0.1, c1=1, c_2=2 were used as defined in [38]. Table 1 provides the maximum entropy evaluated for the entire test images and the value of the optimal thresholds corresponding to best entropy by both the algorithms. From the experiments, it has been examined that the NIBA produces higher entropy than the results obtained by the PSO for various levels thresholding.

To investigate the stability of both of the optimization technique and visual quality of segmented image, standard deviation and PSNR respectively are used and presented in Table 3 for test images. A higher value of PSNR indicates a better quality of thresholding. For all the test images, the proposed algorithm NIBA proves to be better than PSO.

Results of standard deviation in Table 3 shows that NIBA has better stability in comparison to PSO. The standard deviation of execution time for an image repeated for 30 executions is given in Table 3. Table 2 shows the computation time of proposed method and it has been observed that the computation time of NIBA is less than the PSO algorithm.

Table 3. PSNR and standard deviation of entropy.

Image Data Set	clstrs	PSNR			Standard Deviation		
		PSO	IBA	NIBA	PSO	IBA	NIBA
Lena	2	11.00505776	11.00005752	10.51505754	0.00000054	0.00000054	0.00000011
	3	13.20908840	13.30024730	13.98324250	0.00046983	0.00046213	0.00000170
	4	16.00343358	16.11601090	16.99990714	0.00398558	0.00398558	0.00000464
Baboon	2	10.48588785	10.48588785	10.48588785	0.00000068	0.00000067	0.00000005
	3	14.01702554	14.36831907	13.56831907	0.00058510	0.00050515	0.00000256
	4	15.19555078	15.20680883	15.00000883	0.00154086	0.00154084	0.00000446
Cameraman	2	9.67709504	9.69095080	9.60000508	0.00000115	0.00000105	0.00000041
	3	12.65414213	12.74397724	12.70097929	0.00058510	0.00058012	0.00000286
	4	15.20164430	15.32594882	15.32004882	0.00256040	0.00241022	0.00001098

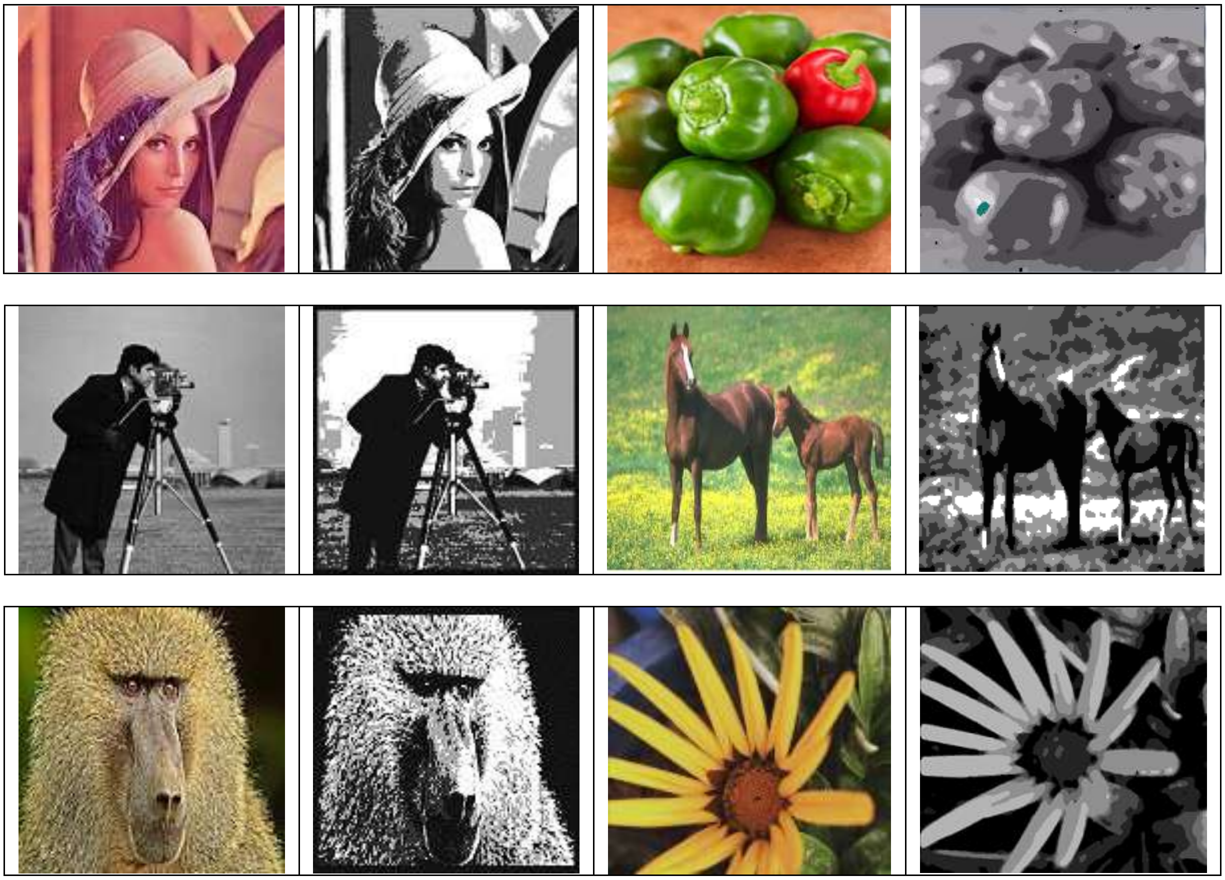

Figure 2. the input test images and their segmentation results using NIBA.

6. Conclusion and Future works

In this paper, a multi-thresholding method based on Kapur's entropy using a novel enhanced bat search algorithm has been proposed. It has better characteristics in searching optimal thresholds. Multilevel image segmentation is done based on histogram for gray scale image. For segmenting image, multiple threshold values are obtained the modified bat algorithm where we have introduced the chaotic concept and a new crossover operator. Through experiments, the performance of our method has been evaluated through the use of appropriate test images and compared with PSO and IBA on various parameters such as standard deviation of entropy and PSNR. The results obtained from test images demonstrated that NIBA performs better than PSO and IBA in terms of entropy, PSNR, stability and computation time. For future scope, fuzzy entropy may also be used in combination with multilevel thresholding for image segmentation.

7. Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

8. References

[1]. S. Masood, M. Sharif, A. Masood, M. Yasmin, and M. Raza, “A survey on medical image segmentation,” Current Medical Imaging Reviews, vol. 11, no. 1, pp. 3–14, 2015.

[2]. P. Ghamisi, M. S. Couceiro, F.M. L. Martins, and J. A. Benediktsson, “Multilevel image segmentation based on fractional-order Darwinian particle swarm optimization,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 5, pp. 2382–2394, 2014.

[3]. M. Waseem Khan, “A survey: image segmentation techniques*,” International Journal of Future Computer and Communication*, vol. 3, no. 2, pp. 89–93, 2014.

[4]. V. Osuna-Enciso, E. Cuevas, and H. Sossa, “A comparison of nature inspired algorithms for multi-threshold image segmentation,” *Expert Systems with Applications*, vol. 40,no. 4, pp. 1213–1219, 2013.

[5]. T. Kurban, P. Civicioglu, R. Kurban, and E. Besdok, “Comparison of evolutionary and swarm based computational techniques for multilevel color image thresholding,” *Applied Soft Computing Journal*, vol. 23, pp. 128–143, 2014.

[6]. N. Otsu, “A threshold selection method from gray-level histograms,” *Automatica*, vol. 11, no. 285–296, pp. 23–27, 1975.

[7]. M.-H. Horng, “Multilevel thresholding selection based on the artificial bee colony algorithm for image segmentation,” *Expert Systems with Applications*, vol. 38, no. 11, pp. 13785–13791, 2011.

[8]. S. Sarkar, G. R. Patra, and S. Das, “A differential evolution based approach for multilevel image segmentation using minimum cross entropy thresholding,” *in Swarm, Evolutionary, and Memetic Computing*, pp. 51–58, Springer, Berlin, Germany, 2011.

[9]. Y. Xue, Y. Zhuang, T. Ni, S. Ni, and X. Wen, “Self-adaptive learning based discrete differential evolution algorithm for solving CJWTA problem,” *Journal of Systems Engineering and Electronics*, vol. 25, no. 1, pp. 59–68, 2014.

[10]. S. Agrawal, R. Panda, S. Bhuyan, and B. K. Panigrahi, “Tsallis entropy based optimal multilevel thresholding using cuckoo search algorithm,” *Swarm and Evolutionary Computation*, vol.11, pp. 16–30, 2013.

[11]. P.D. Sathya and R.Kayalvizhi, “Optimal multilevel thresholding using bacterial foraging algorithm,” *Expert Systems with Applications*, vol. 38, no. 12, pp. 15549–15564, 2011

[12]. X.-S. Yang, "A new meta-heuristic bat-inspired algorithm," in Nature inspired cooperative strategies for optimization (NICSO 2010), ed: Springer, 2010, pp. 65-74.

[13]. Akay, B. A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding. *Applied Soft Computing*, 13(6):3066–3091, June 2013.

[14]. Kennedy, J. and Eberhart, R. Particle swarm optimization. *In IEEE International Conference on Neural Networks*, 1995.

[15]. Sarkar, S. and Das, S. Multilevel image thresholding based on 2D histogram and maximum Tsallis entropy - a differential evolution approach. *IEEE Transactions on Image Processing*, 22(12):4788–4797, December 2013.

[16]. Tao, W., Jin, H., and Li, L. Object segmentation using ant colony optimization algorithm and fuzzy entropy. *Pattern Recognition Letters*, 28(7):788–796, May 2007.

[17]. 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, 1985

[18]. S. and E. U. Kucuksille, “Improved Bat Algorithm (IBA) on Continuous Optimization Problems,” *Lect. Notes Softw. Eng*., vol. 1, no. 3, pp. 279–283, 2013.

[19]. C. Zhen, Z. Yongquan, and L. U. Mindi, “A simplified Adaptive Bat Algorithm Based on Frequency,” *J. Computer. Inf. Syst*., vol. 9: 16 (201, pp. 6451–6458, 2013.

[20]. A. H. and X. S. Yang, “Chaotic bat algorithm,” *J. Comput. Sci*., vol. 5, no. 2, pp. 224–232, 2014.

[21]. M. Wasi, U. Kabir, and M. S. Alam, “Bat Algorithm with Self-adaptive Mutation: A Comparative Study on Numerical Optimization Problems,” *Int. J. Comput. Appl*., vol. 100, no. 10, pp. 975–8887, 2014.

[22]. A. Fouad Ali, “Accelerated Bat Algorithm for Solving Integer Programming Problems.,” *Egypt. Comput. Sci. J*., vol. 39, no. 1, p. 25, 2015.

[23]. Yusra Sabri, Hermaga Santos Zein, and Erwin Yusuf, " Optimal Cost Valuation for Renewable Power Plants Using PSO in Rural Area”, *International Journal on Electrical Engineering and Informatics* - Volume 7, Number 4, December 2015.

[24]. Adis Alihodzic and Milan Tuba, Improved Bat Algorithm Applied to Multilevel Image Thresholding, Scientific World Journal Volume 2014, Article ID 176718, 16 pages.

[25]. Hegazy Zaher, Mahmoud El-Sherbieny, Naglaa Ragaa Saeid, Eman Oun, An Improved Approach for Bat Algorithm, *International Journal of Advanced Research in Computer Science and Software Engineering*, Volume 7, Issue 2, February 2017 ISSN: 2277 128X.

[26]. V. Osuna-Enciso, E. Cuevas, and H. Sossa, “A comparison of nature inspired algorithms for multi-threshold image segmentation,” *Expert Systems with Applications*, vol. 40, no. 4, pp. 1213–1219, 2013.

[27]. L. Dong, G. Yu, P. Ogunbona, and W. Li, “An efficient iterative algorithm for image thresholding,” *Pattern Recognition Letters*, vol. 29, no. 9, pp. 1311–1316, 2008.

[28]. P.-S. Liao, T.-S. Chen, and P.-C. Chung, “A fast algorithm for multilevel thresholding,” *Journal of Information Science and Engineering*, vol. 17, no. 5, pp. 713–727, 2001.

[29]. Linguo Li, Lijuan Sun, Jian Guo, Chong Han, Jian Zhou and Shujing Li, *A Quick Artificial Bee Colony Algorithm for Image Thresholding, Information* 2017, 8, 16;

[30]. S. Sakthi Priya, P. Rishika Menon, M. Vasanthi, I. Thivya Roopini, Improved PSO algorithm approach in Gray scale image multi-level thresholding, *International Journal of Engineering And Computer Science* ISSN: 2319-7242 Volume 5 Issue 5 May 2016, Page No. 16453-16457

[31]. B. Vijay Kumar and N. V. Srikanth, “Bat Algorithm and Firefly Algorithm for Improving Dynamic Stability of Power Systems Using UPFC”, *International Journal on Electrical Engineering and Informatics* - Volume 8, Number 1, March 2016

[32]. X-S. Yang, ‘‘Bat algorithm: Literature review and applications,’’ *International Journal of Bio-Inspired Computation*, 2013, 5(3):141–149.

[33]. Sheikholeslami, R., Kaveh, A.: A Survey of Chaos Embedded Meta-Heuristic Algorithms. Int. *J. Optim. Civil. Eng*. 3(4), 617-633 (2013).

[34]. Boccaletti, S., Grebogi, C., Lai, Y., C., Mancini, H., Maza, D.: The control of chaos: Theory and applications. *Physics Reports*: 329,103-197 (2000).

[35]. Caponetto, R., Fortuna, L., Fazzino, S., Xibilia, M., G.: Chaotic Sequences to Improve the Performance of Evolutionary Algorithms. *IEEE Transaction on Evolutionary Computation*. 7, 289-304 (2003).

[36]. Coelho, L. d. S., Mariani, V. C.: Use of chaotic sequences in a biologically inspired algorithm for engineering design optimization. *Expert Systems with Applications*. 34, 1905-1913 (2008).

[37]. Yang, X.S.: Nature-Inspired Metaheuristic Algorithms, 2nd Edition, Luniver Press, (2010).
[38]. Xuesong Yan, Qinghua Wu, Yuanyuan Fan, Qingzhong Liang and Chao Liu, An Improved Particle Swarm Optimization Algorithm for Traveling Salesman Problems, *International Journal of Control and Automation* Vol. 10, No. 2 (2017), pp.187-200.
[39]. Shaligram Agrawal, Tuli Bakshi, and Dipika Majumdar," Optimal Power Pooling for a Multiple Area Power System through PSO", *International Journal on Electrical Engineering and Informatics* - Volume 6, Number 2, June 2014
[40]. James Kennedy and Russell Eberhart. Particle swarm optimization. In Neural Networks, 1995. Proceedings., *IEEE International Conference* on, volume 4, vol.4, Nov 1995.

Yassad Mokhtari Safia is a doctoral student and same time is a lecturer in the department of computer science, Annaba University, Algeria. She received her M.Sc. degree in embedded systems and mobility in 2014 from Annaba University, Algeria. Her research interests include Embedded vision, Optimization algorithms, and Data science.

Mohamed Tahar Kimour is currently a professor in the department of computer science, Annaba University, Algeria. He received his Ph.D. degree in computer science in 2005 from Annaba University, Algeria. His research interests include Embedded systems, Optimization algorithms, Data science and machine learning techniques.