|| Volume 2 ||Issue 1 ||August 2017||ISSN (Online) 2456-0774
INTERNATIONAL JOURNAL OF ADVANCE SCIENTIFIC RESEARCH AND ENGINEERING TRENDS

RENYI'S ENTROPY BASED MULTILEVEL THRESHOLD SELECTION BASED ON BACTERIAL FORAGING ALGORITHM

P.D. Sathya

Assistant Professor, Department of Electronics and Communication Engineering, Annamalai University,India.
pd.sathya@yahoo.in

Abstract— An original stochastic streamlining way to deal with take care of multilevel thresholding issue in picture division utilizing bacterial foraging (BF) method is introduced. The BF calculation depends on the rummaging conduct of E. Coli microscopic organisms which is available in the human digestive tract. The proposed BF calculation is utilized to expand Renyi's entropy work. The utility of the proposed method is appropriately exhibited by considering a few benchmark test pictures and the outcomes are contrasted and those acquired from particle swarm optimization (PSO) and genetic algorithm (GA) based techniques. Exploratory outcomes show that the proposed calculation could exhibit upgraded execution in correlation with PSO and GA as far as arrangement quality and soundness. Also, applying the proposed technique to deal with picture, the calculation speed is sped up and the quality is gotten to the next level.

KEY WORDS: *multilevel thresholding, Renyi's Entropy, Bacterial foraging*

I INTRODUCTION

Image thresholding which extracts object from the background in an input image is one of the most common applications in image analysis. Foe example, in automatic recognition of machine printed or a hand-written text, in shape recognition of objects, and in image enhancement, thresholding is a necessary step for image preprocessing.

Among the image thresholding methods, bi-level thresholding separates the pixels of an image into two regions i.e. the object and the background. One region contains pixels with gray values smaller than the threshold value and other contains pixels with gray values larger than the threshold value. Further, if the pixels of an image are divided into more that two regions, this is called multilevel thresholding. In general, the threshold is located at the obvious and deep valley of the histogram. However, when the valley is not so obvious, it is very difficult to determine the threshold.

During the past decade, many research studies have been devoted to the problem of selecting the appropriate threshold value. Sahoo et al. [1] have presented a thorough survey of a variety of thresholding techniques. Among those techniques, global, histogram based algorithms [2] are widely used to determine the threshold, and they can be classified as parametric and non-parametric approaches.

In the parametric approaches [3], the gray level distribution of each class is assumed to have a probability density function. It is usually assumed to be a Gaussian distribution. One attempts to find an estimate of the parameters of the distribution that will best fit the given histogram data in the least squares sense. The result is typically a nonlinear optimization problem that is computationally expensive and time-consuming to find the solution.

In the non-parametric approaches, one is to find the thresholds that separate the gray-level regions of an image in an optimum manner according to some discriminate criteria such as the between-class variance [4], entropy [5], and cross entropy [6]. The non-parametric approaches are computationally efficient and simple to implement, compared to the parametric approaches.

In bi-level thresholding the existing non-parametric methods are robust and computationally fast for time-critical applications. However, the computational complexity of those methods is exponentially increased and the selected thresholds generally become less credible as the number of classes to be separated increases. Moreover, to segment complex images, multilevel thresholding method is required. In multilevel image thresholding, pixels can be classified into many classes, not just foreground and background. To mitigate this problem, many methods have been proposed for multilevel thresholding [7-10].

In [8], the Otsu's function is modified by a fast recursive algorithm along with a look-up-table for multilevel thresholding. In [9], Lin has proposed a fast thresholding computation using Otsu's function. Another fast multilevel thresholding technique has been proposed by Yin [10].

Various deterministic methods have been applied to solve multilevel thresholding problem in image segmentation. Several techniques using genetic algorithms (GAs) have also been proposed to solve the multilevel thresholding problem [11], [12]. The particle swarm optimization (PSO) has been applied to the multilevel thresholding for image segmentation [13].

In this paper, the BF algorithm is employed to solve the multilevel thresholding problem in image segmentation. The algorithm is based on the foraging (methods for locating, handling and ingesting food) behavior of E. Coli bacteria present in the human intestine. It was successfully used to solve various kinds of engineering problems [14-16]. The proposed BF algorithm has been compared with the PSO and

GA methods over six benchmark images with respect to the following performance measures: Solution quality, convergence speed and PSNR (Peak to Signal Noise Ration) measure. It has been shown that the BF algorithm offers superior performance than the PSO and GA.

II. PROBLEM FORMULATION WITH RENYI'S ENTROPY

Global threshold selection methods usually use the gray-level histogram of the image. The optimal thresholds are determined by optimizing some criterion function obtained from the gray-level distribution of the image. Similar to the Kapur entropy, Sahoo et al [ ] proposed a new thresholding technique using Renyi's entropy. The method has utilized two probability distributions (object and background) which are derived from the original gray-level distribution of an image. Renyi's entropy method was originally developed for bi-level thresholding.

Let there be L gray levels in a given image and these gray levels are in the range {0, 1, 2… (L-1)}. Then one can define $P_i = h(i) / N$, $(0 \le i \le (L-1))$ where h (i) denotes number of pixels for the corresponding gray-level L and N denotes total number of pixels in the image which is equal to $\sum_{i=0}^{L-1} h(i)$.

Renyi's bi-level thresholding can be described as follows:

$$f(t) = \arg\max[H_0^{\alpha}(t) + H_1^{\alpha}(t)] \quad (1)$$

and

$$H_0^{\alpha}(t) = \frac{1}{1-\alpha}\ln\sum_{i=0}^{t-1}\left(\frac{P_i}{P^A}\right)^{\alpha}, \quad P^A = \sum_{i=0}^{t-1} P_i$$

$$H_1^{\alpha}(t) = \frac{1}{1-\alpha}n\sum_{i=t}^{L-1}\left(\frac{P_i}{P^B}\right)^{\alpha}, \quad P^B = \sum_{i=t}^{L-1} P_i .$$

where α is a positive parameter.

This Renyi's entropy criterion method can also be extended to multilevel thresholding and it is described as follows:

$$f(t) = \arg\max[H_0^{\alpha}(t) + H_1^{\alpha}(t) + H_2^{\alpha}(t) + \ldots\ldots + H_m^{\alpha}(t)] \quad (2)$$

where

$$H_0^{\alpha}(t) = \frac{1}{1-\alpha}\ln\sum_{i=0}^{t_1-1}\left(\frac{P_i}{P^A}\right)^{\alpha}, \quad P^A = \sum_{i=0}^{t_1-1} P_i$$

$$H_1^{\alpha}(t) = \frac{1}{1-\alpha}\ln\sum_{i=t_1}^{t_2-1}\left(\frac{P_i}{P^B}\right)^{\alpha}, \quad P^B = \sum_{i=t_1}^{t_2-1} P_i$$

$$H_2^{\alpha}(t) = \frac{1}{1-\alpha}\ln\sum_{i=t_2}^{t_3-1}\left(\frac{P_i}{P^C}\right)^{\alpha}, \quad P^C = \sum_{i=t_2}^{t_3-1} P_i \quad \ldots.$$

$$H_m^{\alpha}(t) = \frac{1}{1-\alpha}\ln\sum_{i=t_m}^{L-1}\left(\frac{P_i}{P^m}\right)^{\alpha}, \quad P^m = \sum_{i=t_m}^{L-1} P_i .$$

III. BACTERIAL FORAGING ALGORITHM

BF algorithm is a newly introduced evolutionary optimization algorithm that mimics the foraging behavior of Escherichia coli (commonly referred to as E. coli) bacteria. BF algorithm was first introduced by Passino [25]. There are successful applications of BF algorithm in image processing, such as Image Watermarking [26, 27], Image Enhancement [28], Image Circle Detection [29] and Filtering [30].

BF models the movement of E. coli bacterium moves using a pattern of two types of movements: tumbling and swimming. Tumbling refers to a random change in the direction of movement, and swimming refers to moving in a straight line in a given direction. A bacterium in a neutral medium alternates between tumbling and swimming movements.

Suppose it is desired to search for the position X in an N-dimensional space. Let Xi be the initial position of bacterium i in the search space, i = 1, 2, . . . S, where S is the number of bacteria. In biological bacteria populations, S can be as high as 109 and N is three. Let F (Xi) represent an objective function. Let F (Xi) < 0, F (Xi) = 0, and F (Xi) > 0 represent the bacterium at location Xi in nutrient rich, neutral, and noxious environments, respectively. Chemotaxis is a foraging behavior that captures the process of optimization, where bacteria to climb up the nutrient concentration gradient (i.e., bacteria try to achieve positions having lower values of F (Xi) and avoid being at positions Xi, where F (Xi) ≥ 0).

The bacterium i at position Xi takes a chemotactic step j with the step size C(i) and evaluates itself for objective function F (Xi) at each step. If at position Xi (j + 1), the objective value F is better than at position Xi (j), then another step of same size C(i) in the same direction will be taken again, if that step resulted in a position with a better value than at the previous step. This is referred to as a swimming step. Swimming is continued until for a maximum number of steps Ns. After Nc chemotactic steps, a reproduction steps is taken in which the population is sorted in ascending order of the objective function value F and least healthy bacteria are replaced by the copies of the healthier bacteria. After Nre reproduction steps, an elimination-dispersal step is taken. Here, a bacterium is eliminated and a new bacterium is created at a random location in the search space with probability ρed. The optimization stops after Ned elimination-dispersal steps.

Bacteria create swarms by means of cell-to-cell signaling via an attractant and a repellant. Cell-to-cell attraction for bacterium i is represented with Fcc (Xg, Xi), i = 1, 2, . . . S. this is defined as follows:

The cell-to-cell signaling Fcc ( ) helps cells to move toward other cells, but not very close to them. In BF algorithm, the maximum number of objective function evaluations is S. Nc . Ns. Nre . Ned. A general biologically inspired thumb-of-rule for choosing the parameters of

BF is: Nc > Nre > Ned. The detailed pseudo-code for BF algorithm is given in Algorithm 1.

Algorithm 3 Pseudo-code for BF algorithm

```
1: Initialize S, N, Nc, Ns, Nre, Ned, ρed, dattract, hrepellant, ωattract, ωrepellant, Xmin and Xmax
2: Initialize Xi randomly for i = 1, 2, . . . , S
3: Initialize C (i) for i = 1, 2, . . . , S
4: Set the loops counters j, k and l to 0
5: //Elimination-Dispersal loop:
6: while l ≤ Ned do
7:    l = l + 1
8:    //Reproduction loop:
9:    while k ≤ Nre do
10:      k = k +1
11:      //Chemotaxis loop:
12:      while j ≤ Nc do
13:        j = j + 1
14:        for each bacterium i = 1, 2, . . . S do
15:          Compute F(i, j, k, l)
16:           Let F (i, j, k, l) = F(i, j, k, l) + Fcc (Xg, Xi)
17:           Let Flast = F (i, j, k, l)
18:          //Tumble:
19:           Generate a N-dimensional random vector ∆m (i), i = 1, 2, . . ., N on [-1, 1]
20:          //Move:
21:         Let
22:         Compute F (i, j + 1, k, l) with X i (j +1, k, l)
23:         //Swim:
24:         Let m = 0
25:         while m < Ns do
26:           Let m = m + 1
27:           if F (i, j + 1, k, l) < Flast then
28:             Let Flast = F (i, j + 1, k, l)
29:             Let
30:             Use this X i (j +1, k, l) to compute new F (i, j + 1, k, l)
31:           else
32:             m = Ns
33:           end if
34:         end while
35:        end for
36:      end while
37:      Compute for each bacterium i, for given k and l
38:      Eliminate Sr fraction of bacteria with highest Fhealth and split the other bacteria into two at their locations.
39:    end while
40:    For each bacterium, with probability ρed eliminate the bacterium and create a new one at a random position.
41: end while
```

IV. EXPERIMENTAL RESULTS AND DISCUSSION

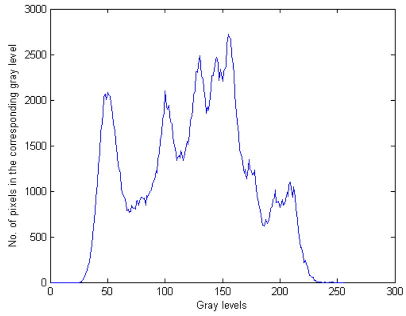
For evaluating the performance of the proposed algorithm, it has been implemented on several test images, the results obtained are compared with the results of well known heuristic algorithms such as PSO and GA. The All tested algorithms belong to the population-based thresholding algorithm. The proposed algorithm is implemented with a core2duo 2 GHz personal computer in MATLAB language. Test images with their corresponding histograms are shown in Fig. 1. Parameters used for the proposed algorithm are summarized in Table I.

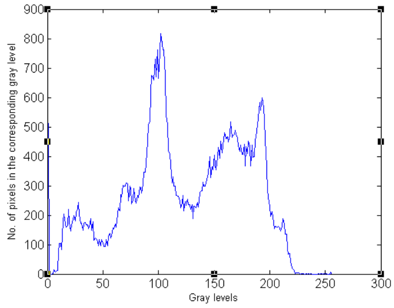
Table 2 shows the optimal thresholds and the corresponding objective values obtained by all the algorithms using Renyi's objective function. Higher value of objective function indicates the better results in the segmentation. As shown in Table 2, the proposed approach offers an improved objective function value over the PSO and GA methods, clearly showing the proposed approach's ability to locate better solutions than other methods.

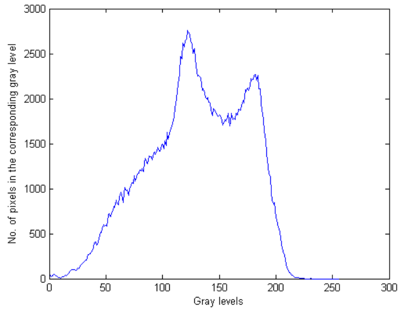
(a) (b) (c)

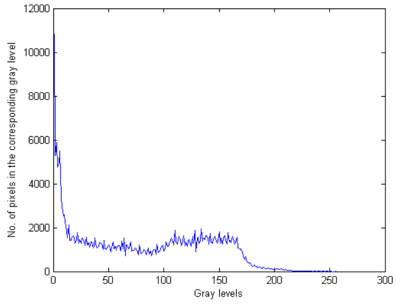
(d) (e) (f)

Fig. 1. Test Images [(a) Lena, (b) Pepper, (c) Baboon, (d) Hunter, (e) Airplane, (f) Butterfly]

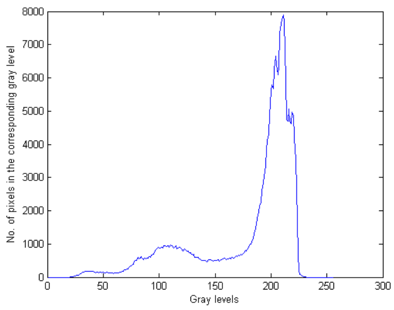


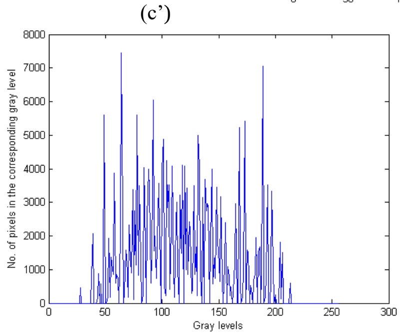






(a’) (b’) (c’)





(d’) (e’) (f’)

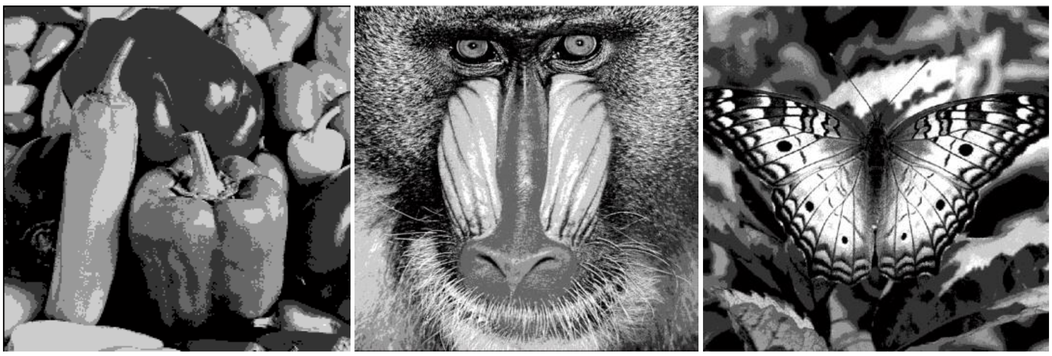
Fig. 2. Histogram of test images [(a’) Lena, (b’) Pepper, (c’) Baboon, (d’) Hunter, (e’) Airplane, (f’) butterfly]

TABLE I. PARAMETERS USED FOR BF METHOD

Parameter	lue
Number of bacterium (s)	0
Number of chemotatic steps (N_c)	0
Swimming length (N_s)	0
Number of reproduction steps (N_{re})	4
Number of elimination of dispersal events (N_{ed})	2
Depth of attractant ($d_{attract}$)	.1
Width of attract ($\omega_{attract}$)	.2
Height of repellent ($h_{repellent}$)	.1
Width of repellent ($\omega_{repellent}$)	0
Probability of elimination and dispersal (P_{ed})	02

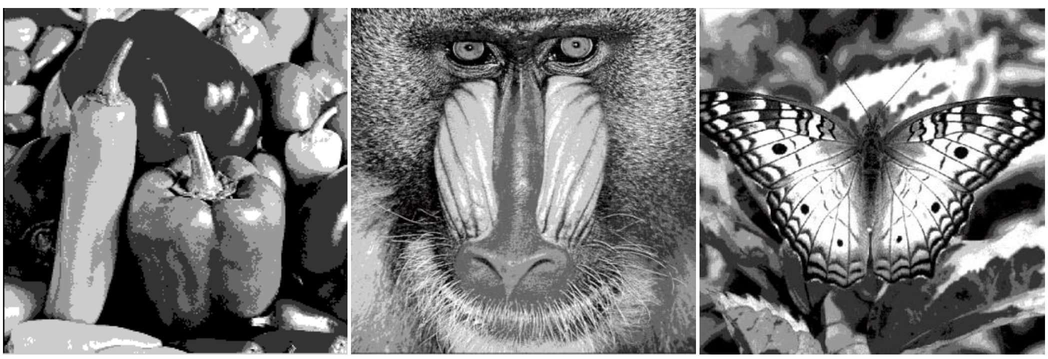
TABLE II. COMPARISON OF OPTIMAL THRESHOLD VALUES AND THEIR OBJECTIVE VALUES OBTAINED BY MCE BASED EVOLUTIONARY ALGORITHMS

Test Images	m	Optimal threshold values			Objective values		
		BF	PSO	GA	BF	PSO	GA
LENA	2	99,166	99,166	99,166	12.3897	12.3897	12.3897
	3	67,122,178	93,137,193	88,168,201	15.3020	15.2254	14.9657
	4	63,94,126,174	75,114,144,178	85,129,149,179	17.6047	17.4717	17.2935
	5	73,104,132,164,194	82,114,147,174,201	87,116,156,177,204	20.6058	20.4798	20.3066
PEPPER	2	80,150	80,150	80,150	12.5809	12.5809	12.5809
	3	72,120,161	74,128,196	88,149,204	15.5931	15.4238	15.2091
	4	82,115,155,197	70,110,145,179	76,106,147,185	17.6938	17.5800	17.3779
	5	53,91,132,167,197	75,117,145,172,200	79,116,150,181,220	21.0780	20.8113	20.6959
BABOON	2	77,143	77,143	77,143	12.2914	12.2914	12.2914
	3	60,118,158	59,128,155	63,87,171	15.2906	15.1011	14.9583
	4	70,107,142,171	73,110,155,191	84,120,160,189	17.3988	17.2551	17.0455
	5	42,88,118,146,175	54,94,128,163,193	69,102,131,161,191	20.7277	20.6136	20.4514
HUNTER	2	91,178	91,178	91,178	12.4967	12.4967	12.4967
	3	51,131,184	85,128,171	61,107,201	15.6301	15.4790	15.3496
	4	45,96,144,184	58,100,132,187	62,104,139,193	17.3064	17.1923	17.0108
	5	55,103,150,182,220	57,93,128,185,208	82,123,150,182,220	21.3092	21.1116	20.9974
AIRPLANE	2	77,170	77,170	77,170	12.2407	12.2407	12.2407
	3	75,114,185	87,121,176	77,118,198	15.3752	15.2436	15.0324
	4	73,114,146,184	73,112,153,193	61,121,153,195	18.2087	18.0972	17.8989
	5	66,99,135,166,191	77,111,144,170,194	67,104,140,183,212	20.7815	20.5993	20.3574
BUTTERFLY	2	96,144	96,144	96,144	10.6317	10.6317	10.6317
	3	69,111,151	99,120,171	66,127,170	12.8903	12.7196	12.5904
	4	84,108,138,170	85,107,138,170	68,114,142,177	15.1106	15.0799	14.8957
	5	71,96,121,144,174	63,99,122,143,165	80,113,137,153,179	17.1092	17.0044	16.8461



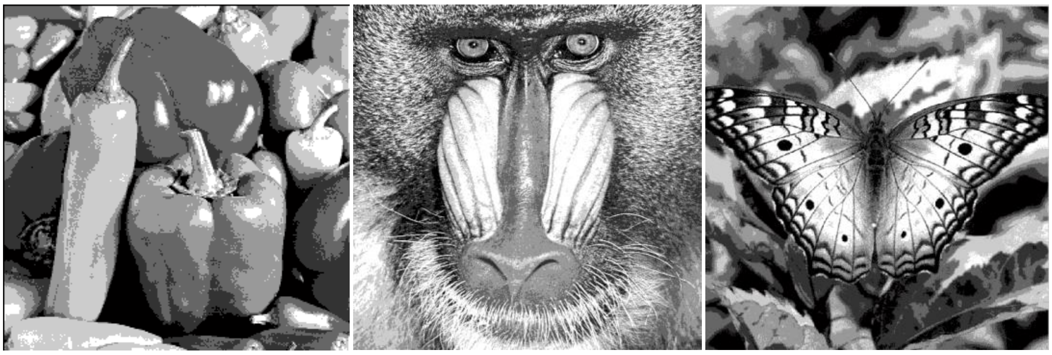
(a) (b) (c)

Fig. 2. Segmented results of GA for pepper, baboon and butterfly images respectively when m = 5 is chosen



(a) (b) (c)

Fig. 3. Segmented results of PSO algorithm for pepper, baboon and butterfly images respectively when m = 5 is chosen



(a) (b) (c)

Fig. 4. Segmented results of the proposed BF algorithm for pepper, baboon and butterfly images respectively when m = 5 is chosen

TABLE III. THE STANDARD DEVIATION VALUE OF FOUR MULTILEVEL THRESHOLDING METHODS

Test Images	m	Standard Deviation		
		BF	PSO	GA
LENNA	2	0.0000	0.0000	0.0000
	3	0.0263	0.0674	0.0864
	4	0.0481	0.0966	0.1415
	5	0.0612	0.1145	0.1750
PEPPER	2	0.0000	0.0000	0.0000
	3	0.0311	0.0948	0.1493
	4	0.0312	0.1425	0.2071
	5	0.0427	0.1980	0.2672
BABOON	2	0.0000	0.0000	0.0000
	3	0.0220	0.0631	0.0984
	4	0.0226	0.0956	0.1255
	5	0.0557	0.1067	0.3191
HUNTER	2	0.0000	0.0000	0.0000
	3	0.0131	0.0444	0.0964
	4	0.0360	0.1056	0.1390
	5	0.0595	0.1200	0.1511
AIRPLANE	2	0.0000	0.0000	0.0000
	3	0.0313	0.0604	0.0775
	4	0.0471	0.0622	0.1079
	5	0.6111	0.0888	0.1854
BUTTERFLY	2	0.0000	0.0000	0.0000
	3	0.0145	0.0519	0.0604
	4	0.0353	0.0972	0.1016
	5	0.0773	0.1086	0.1247

TABLE IV. THE PSNR MEASURE BY FOUR MULTILEVEL THRESHOLDING METHODS

Test Images	m	PSNR (db)		
		BF	PSO	GA
LENNA	2	15.2436	15.2436	15.2436
	3	17.5192	17.3691	16.4330
	4	18.0722	17.6204	17.2404
	5	21.0964	20.5180	20.0175
PEPPER	2	12.7412	12.7412	12.7412
	3	16.4483	15.8646	15.7222
	4	18.9852	17.8329	17.4007
	5	20.3446	20.2480	19.9707
BABOON	2	12.6285	12.6285	12.6285
	3	15.7325	14.9961	14.2945
	4	19.8436	18.6163	18.1994
	5	21.8513	21.4614	20.5676
HUNTER	2	12.7530	12.7530	12.7530
	3	16.2716	15.3867	15.2920
	4	18.2297	17.2325	16.8773
	5	19.5559	18.5961	18.0875
AIRPLANE	2	13.7735	13.7735	13.7735
	3	15.0386	14.5306	14.0115
	4	16.1041	15.4407	15.4115
	5	18.7440	17.6627	17.0447
BUTTERFLY	2	14.2756	14.2756	14.2756
	3	15.6140	15.2239	14.7754
	4	18.7420	17.5003	17.0536
	5	19.8837	18.9507	18.4744

TABLE IV. THE CPU TIME TAKEN BY FOUR MULTILEVEL THRESHOLDING METHODS

Test Images	m	CPU time (Seconds)		
		BF	PSO	GA
LENNA	2	4.0938	4.4313	4.9219
	3	4.2500	4.6878	4.9531
	4	4.5313	4.9844	5.2031
	5	4.6875	5.3594	5.8438
PEPPER	2	4.1250	4.4688	4.8906
	3	4.2031	4.9063	5.1563
	4	4.3281	5.0001	5.5313
	5	4.7500	5.3124	5.5594
BABOON	2	3.7656	3.9844	4.0313
	3	3.9327	4.2969	4.6875
	4	4.1250	4.5313	4.9375
	5	4.3594	4.7500	5.0781
HUNTER	2	4.2813	4.4063	4.9531
	3	4.4750	4.8594	5.2513
	4	4.5016	5.0156	5.4688
	5	4.8343	5.6094	6.2500
AIRPLANE	2	4.1719	4.5781	4.9692
	3	4.2031	4.4500	5.2188
	4	4.3275	4.8594	5.3594
	5	4.5938	5.0938	5.6875
BUTTERFLY	2	4.4825	4.7344	4.9991
	3	4.4375	4.9219	5.1563
	4	4.6250	5.2656	5.5313
	5	4.8125	5.4063	5.8906

The quality of the thresholded images can be evaluated by PSNR measure. The higher value of PSNR means the quality of the thresholded image is better. Obviously, PSNR can be used as a criterion for optimal thresholding. It can be observed from Table III that the results of the proposed method have higher PSNR than the other two methods.

Table IV summarizes the standard deviation values of three different approaches. Through 50 trials, the proposed approach yielded smaller standard deviations of objective function values.

Table V illustrate the CPU time taken to find the optimal threshold values of all the three algorithms. It is seen from the table that the time requirement of the proposed BF method is less and either comparable or better than the other mentioned methods. And also, the CPU time increases with the number of thresholds. So as a whole, it can be said that the BF method is efficient than previously mentioned methods.

To provide the visual comparison, the results of 5-level thresholding in Table 1 are illustrated in Figs. 2, 3 and 4. Fig. 4 shows that the quality of the thresholded images is better by Renyi based BF algorithm than the other two algorithms.

All over, the proposed Renyi's entropy based BF algorithm provides better efficiency, PSNR value and stability. And also, the proposed method converges faster than the other two algorithms.

VI. CONCLUSION

The new evolutionary technique, Bacterial Foraging (BF) is used for solving multilevel thresholding problem, with an endeavor to maximize the Renyi's entropy function. The utility of the proposed algorithm is demonstrated by considering several benchmark test images and it has been compared with other evolutionary algorithms such as PSO and GA methods. Experimental results confirm the potential of the BF method in solving multilevel thresholding problem and show its effectiveness and superiority over PSO and GA. Furthermore, the proposed method is also suitable for other types of images, and can be applied to a wide class of computer vision applications, such as character recognition, watermarking technique and segmentation of wide variety of medical images.

REFERENCES

[1] P. K. Sahoo, S. Soltani, and A. K. C. Wong, "A survey of thresholding techniques," *Computer Vision, Graphics and Image Processing*, vol. 41(2), pp. 233-260, 1988.

[2] C.A. Glasbey, "An analysis of histogram based thresholding algorithms," *CVGIP: Graphical Models and Image Processing*, Vol. 55, pp. 532-537, 1993.

[3] J.S. Weszka, "A survey of threshold selection techniques," *Computer Vision Graphics Image Processing*, Vol. 7, 259-265, 1979.

[4] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transaction on Systems, Man and Cybernetics*, SMC-9(1), pp. 62-66, 1979.

[5] 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, pp. 273-285, 1985.

[6] C.H. Li and C.K. Lee, "Minimum cross entropy thresholding," *Pattern Recognition*, Vol. 26(4), pp. 617-625, 1993.

[7] P. Y. Yin and L. H. Chen, "A fast iterative scheme for multilevel thresholding methods", *Signal Processing*, vol.60, pp. 305-313, 1997.

[8] P. S. T. Liao, S. Chen, and P. C. Chung, "A fast algorithm for multilevel thresholding", *Journal of Information science and Engineering*, vol. 17, pp. 713-727, 2001.

[9] K. C. Lin, "Fast image thresholding by finding zero(s) of the first derivative of between class variance", *Machine Vision and Applications*, vol. 13, pp. 254-262, 2003.

[10] Peng-Yeng Yin and Ling-Hwei Chen, "A fast iterative scheme for multilevel thresholding methods," *Signal Processing*, vol. 60(3), pp. 305-313, 1997.

[11] P. Y. Yin, "A fast scheme for optimal thresholding using genetic algorithms", *Signal processing*, vol. 72, pp 85−95, 1999.

[12] C. C. Lai, D. C. Tseng, "A hybrid approach using Gaussian smoothing and genetic algorithm for multilevel thresholding", *International Journal of Hybrid Intelligent Systems*, vol. 1(3), pp. 143-152, 2004.

[13] M. Maitra, and A. Chatterjee, "A hybrid cooperative-comprehensive learning based PSO algorithm for image segmentation using multilevel thresholding," *Expert Systems with Applications*, Vol. 34, 1341-1350, 2008.

[14] K.M. Passino, Biomimicry of bacterial foraging for distributed optimization and control, IEEE Transactions on Control Systems Magazine 22(3) (2002) 52–67.

[15] Hsiang-Cheh Huang, Yueh-Hong Chen, Guan-Yu Lin, Fuzzy-Based Bacterial Foraging for Watermarking Applications', International Conference on Hybrid Intelligent Systems, Shenyang (2009) pp. 214-217.

[16] Hsiang-Cheh Huang, Yueh-Hong Chen, and Ajith Abraham, Optimized watermarking using swarm-based bacterial foraging, Journal of Information Hiding and Multimedia Signal Processing, 1(1) (2009) 51-58.

[17] M. Hanmandlu, O.P. Verma, N.K. Kumar, M. Kulkarni, A Novel Optimal Fuzzy System for Color Image Enhancement Using Bacterial Foraging, IEEE Transactions on Instrumentation and Measurement 58(2) (2009) 2867-2879.

[18] Sambarta Dasgupta, Arijit Biswas, Swagatam Das and Ajith Abraham 'Automatic circle detection on images with an adaptive bacterial foraging algorithm,' International Conference on Genetic and Evolutionary Computation, Atlanta, USA, (2008) 1695-1696.

[19] K.M. Bakwad, S.S. Pattnaik, B.S. Sohi, S. Devi, P.K. Panigrahi, V.R.S. Sastry Gollapudi, Bacterial foraging optimization technique cascaded with adaptive filter to enhance peak signal to noise ratio from single image, IETE Journal of Research 55(4) (2009) 173-179.

[20] T.K. Das, G.K. Venayagamoorthy, U.O. Aliyu, Bio-inspired algorithms for the design of multiple optimal power system stabilizers: SPPSO and BFA, IEEE Transactions on Industrial Applications 44(5) (2008) pp. 1445-57.