ISSN: 0976-9102 (ONLINE)
DOI: 10.21917/ijivp.2021.0344

APPLICATION OF TEACHING LEARNING BASED OPTIMIZATION IN MULTILEVEL IMAGE THRESHOLDING

S. Anbazhagan
Department of Electrical Engineering, Annamalai University, India

Abstract

This paper proposes a Teaching learning-based optimization (TLBO) algorithm for the multilevel image thresholding using Kapur entropy. In image processing, the thresholding arises to help medical imaging, detection, and recognition in making an informed decision about the image. However, they are computationally expensive reaching out to multilevel thresholding since they thoroughly search the optimal thresholds to enhance the fitness functions. In order to validate the chaotic characteristic of multilevel thresholding, a TLBO algorithm is modeled. The proposed model is an algorithm-specific, parameterless algorithm that does not require any algorithm-specific parameters to be controlled by maximizing the Kapur entropy of various classes for image thresholding. The proposed model is compared with recent algorithms to threshold the seven standard benchmark and three test images. The simulation results have higher fitness function values even with the increase of the threshold number with less computation time. The Jaccard measure values are close to 0.99.

Keywords:

Kapur's Entropy, Multilevel Thresholding, Teaching Learning based Optimization

1. INTRODUCTION

Thresholding is a procedure generally utilized in image segmentation. The goal of thresholding is to decide a threshold value to divide the image space into significant regions. Thresholding is a vital advance in many image processing undertakings, for example, automatic recognition of machine-printed or transcribed writings, recognition of object shapes, and image enhancement. The fundamental reason for image thresholding is to decide one (bi-level thresholding) or k (multilevel thresholding) proper threshold values for an image to partition pixels of the image into various regions [1]. In the ongoing years, expanding the multifaceted nature of computerized images, for example, the intensity inhomogeneity, makes multilevel thresholding (MT) approaches drawn considerably more consideration. This is primarily because of its simple usage and low storage memory trademark [2].

The MT changes the image thresholding to an optimization issue where the suitable threshold values are found by maximizing or minimizing a rule. The well-known Otsu's function [3], the threshold is controlled by maximizing the between-class variance. In Kapur's entropy [4], the ideal thresholds are accomplished by maximizing the entropy of various classes. A fuzzy entropy measure is applied for picking the ideal thresholds in [5] while Qiao, Hu, Qian, Luo and Nowinski [6] figured the thresholding rule by investigating the information as far as intensity contrast. Scientists have additionally built up some other best criteria, including fuzzy similarity measure [7], cross-entropy [8], Tsallis entropy [2], [9], Bayesian error [10], Renyi's entropy [11], etc.

Among these methodologies, Kapur entropy picks the best threshold worth by maximizing the entropy of various classes, has pulled in critical consideration from established researchers. In any case, this strategy has an undeniable downside in that the computational complexity nature increments exponentially with an increase in the number of required thresholds. To a limited degree, this confines its application in MT, various methodologies and comparing upgrades have been proposed to dispose of the previously mentioned disadvantages.

Empowered by the effective utilization of the flower pollination (FP) and social spider optimization (SSO) calculations, this paper [12] further analyzes their achievability for taking care of picture thresholding issues by means of an MT approach. As an objective function, Kapur's entropy is utilized to look at the best execution of thresholding pictures utilizing these two optimizations. Acquired outcomes from SSO and FP have been thought about against the swarm optimization (PSO) and the bat algorithm (BA). So as to maximize Kapur's objective function, the spider monkey optimization algorithm is utilized. The standard pictures are pre-tried and contrasted and PSO [13].

The point of whale optimization algorithm (WOA) and moth-flame optimization (MFO) methods is to decide the best thresholds that maximize the Kapur function. The test consequences of the proposed algorithms have been with various swarm methods [14]. Grey wolf optimizer is enlivened from the social and chasing conduct of the grey wolves. This metaheuristic method is applied to MT issues utilizing Kapur entropy function. The exhibitions of the proposed strategy are then contrasted and improved adaptations of PSO and bacterial foraging (BF) optimization based MT strategies [15].

Kotte, Pullakura, and Injeti are concentrated on taking care of the image thresholding issue by consolidating Otsu and Kapur functions with metaheuristic systems, adaptive wind-driven optimization (AWDO) [16]. The Krill Herd optimization [17] embraced to scan for multilevel threshold utilizing Otsu and Kapur objective functions gives better outcomes contrasted with BF, PSO, genetic algorithm (GA), and MFO.

The hybrid optimization algorithm named BMO-DE based on bird mating optimization (BMO) and differential evolutionary (DE) algorithms for multi thresholding using multilevel thresholding based on Kapur and Otsu functions [18]. The proposed method is tried on standard test pictures and contrasted with BF, modified bacterial foraging (MBF), PSO, GA, and hybrid algorithm named PSO-DE. The firefly algorithm (FA) is embraced for taking care of the MT image thresholding issue utilizing Kapur entropy strategy. The proposed method is tried on standard test pictures and Levy flight provides good exploration capability [19].

To maximize Kapur's objective function, another novel metaheuristic algorithm, namely Equilibrium Optimizer (EO), is developed for tackling the multi-thresholding problems. The performance of the algorithm is compared with seven other algorithms like Sine Cosine algorithm (SCA), WOA, Harris Hawks optimization (HHA), Salp Swarm Algorithm (SSA), BA, PSO, and crow search algorithm (CSA) [20].

The proposed Teaching learning-based optimization (TLBO) is an algorithm explicit, the parameterless calculation that doesn't require any method explicit parameters to be tuned for image segmentation dependent on Kapur's entropy. During the refreshing procedure, the nature of every result is assessed utilizing the between-class variance function. As indicated by the fitness function, the result of results is refreshed dependent on the qualities of the TLBO until an end basis is fulfilled. The consequences of the TLBO calculation have been contrasted and other metaheuristic calculations. The exhibition of the distinctive method has been evaluated on seven standard benchmark and three different test pictures utilizing the best fitness values and Jaccard measure.

The remainder of this paper is organized as follows: In Section 2, the issue plan and the meaning of Kapur's entropy and proposed algorithms are presented. The proposed algorithms execution procedure for MT is represented in Section 3. The analyses and results are given in Section 4. At last, the conclusion and future work are recorded in the last section.

2. METHODOLOGY

2.1 MULTILEVEL IMAGE THRESHOLDING CRITERION BASED ON KAPUR ENTROPY

The thresholding procedure performs image thresholding dependent on the data contained in the picture histogram. This is performed by maximizing an objective function that utilizes the chose thresholds as the parameters. Right now, the thresholding strategy to be specific entropy of the segmented classes (Kapur) entropy is utilized. Thresholding utilizing Kapur entropy is a nonparametric thresholding method, which is utilized to partition the whole picture into numerous regions; thus, the entropy and the probability distribution of the image histogram can be maximized. Since Kapur entropy is an entrenched basis, the detailed conversation on Kapur entropy isn't introduced here. Perusers can allude [4], [15] for additional subtleties.

2.2 TLBO

Optimization is the way toward improving something than the past structure. In the course of the most recent decade, the total intelligent conduct of insect or animal groups in the normal world for instance flocks of birds, colonies of ants, schools of fish, swarms of bees, and termites have intrigued the enthusiasm of scientists. The aggregate activity of insects, birds, or animals is distinguished as swarm conduct. Numerous specialists have utilized swarm conduct as a system for taking care of entangled real-world issues. Moreover, all the nature-inspired algorithms require tuning of algorithm parameters for them to work properly. To stay away from this trouble, an optimization algorithm, TLBO, a parameter-free method, is executed right now tackle complex MT problems.

TLBO method is a global optimization algorithm initially created by Rao, Savsani and Vakharia [21]. It is a population-based iterative learning method that shows some basic attributes with other EC methods. Be that as it may, TLBO scans for an ideal through every student attempting to accomplish the experience of the teacher, which is treated as the most learned individual in the general public, accordingly getting the optimum outcomes, instead of through students experiencing genetic activities like selection, crossover, and mutation. Because of its basic idea and high proficiency, TLBO has become an extremely appealing optimization procedure and has been effectively applied to numerous real-world issues [22]–[24].

2.3 OVERVIEW

TLBO is a population-based optimization method in which a gathering of results is executed to arrive at an optimum solution. The TLBO depends because of an educator on the yield of students in the class (yield is considered regarding results or evaluations of learners). The instructor is commonly considered as the most talented individual whose information is imparted to the students. The nature of the educator is considered the result of the students. Consider, two different teachers (T1 and T2) taking a class in two distinct classes (C1 and C2). The consequences of the learners are perceived as far as their evaluations. Assume the mean of the evaluations (M1) got by the learner in class 1 is seen as higher than the mean of the evaluations of the learner in class 2 (M2), it tends to be expressed that T1 is better than T2. In this manner, a great educator creates better mean for the consequences of the learners. Further, by associating among themselves the students can learn. Be that as it may, as a general rule an educator can just move the mean of a class up somewhat relying upon the capacity of the class. This follows an arbitrary procedure relying upon numerous components [21]. The TLBO algorithm is demonstrated dependent on the exchange of information in the study hall condition where, the student first increases information from an instructor and afterward from different students by methods for group discussions, formal interchanges, and so on. The proposed method comprises of two stages to be specific teacher phase (TP) and learner phase (LP). In the TP, an instructor attempts to carry the students to their level as far as information. The result designations are haphazardly conveyed all through the search space. In this manner, among the results, the best solution is chosen [22].

2.4 MATHEMATICAL MODEL OF TLBO

Let, M_i and T_i be the mean and the teacher separately at any iteration i. During the TP, teacher T_i attempts to move mean M_i toward its own level, so new mean is assigned as M_{new} and dependent on the distinction between the current and the new mean, the result is refreshed as [22],

$$Difference\_mean_i = r\left(M_{new} - T_F M_i\right), \tag{1}$$

Where, r is a random number in the range [0, 1] and T_F is a teaching factor which chooses the estimation of mean to be changed. Estimation of T_F can be either 1 or 2 which a heuristic step is and it is chosen arbitrarily as,

$$T_F = round\left[1 + rand\left(0,1\right)\left\{2-1\right\}\right], \tag{2}$$

In view of the distinction mean given by Eq.(1), the current result is changed as,

$$x_{new,i} = x_{old,i} + Difference\_mean_i, \tag{3}$$

In the LP, the learners commonly connect among themselves and the procedure of shared collaboration will in general increment the information on the student. Every student associates arbitrarily with different students and consequently information sharing is encouraged. Consider, two unique students x_i and x_j, to such an extent that $i \neq j$, the modification formula speaking to the learner phase can be communicated as,

$$x_{new,i} = x_{old,i} + r\left(x_i - x_j\right) if\ f\left(x_i\right) < f\left(x_j\right), \quad (4)$$

$$x_{new,i} = x_{old,i} + r\left(x_j - x_i\right) if\ f\left(x_j\right) < f\left(x_i\right), \quad (5)$$

The x_{new} is acknowledged whether it gives better function value. The TLBO algorithm is ended when a foreordained most extreme iteration number (maximum number of generations) is reached [23].

3. EXECUTION OF TLBO FOR TAKING CARE OF MT PROBLEM

The methods connected with the execution of TLBO [23] for dealing with MT issue are as per the following:

Step 1: Read the standard benchmark test image and instate TLBO parameters, for example, population size and most extreme number of generations.

Step 2: *Initialization*: arbitrarily generate population x_i as indicated by population size and number of threshold values as,

$$population = r * U_{L,i} - L_{L,i} + L_{L,i}, \quad (6)$$

where,

$U_{L,i}$ and $L_{L,i}$ corresponds to the maximum and minimum gray levels of the image.

- **a.** Identify the optimal threshold values without violating constraints.
- **b.** Calculate the entropy measures based on the probability distribution of the image among the threshold values.
- **c.** Determine the fitness function value which incorporates the objective function and constraint violations.

Step 3: *Teacher Phase*: The threshold values having maximum fitness value is found and is mimicked as teacher ($x_{teacher}$).

- **a.** Compute the mean of the students ($M_i=[m_1, m_2, ..., m_{nd}]$) in the class column-wise and distinguish the best solution($x_{teacher}$). Move M_i toward $x_{teacher}$ which goes about as new mean (M_{new}).
- **b.** Calculate *Difference_mean* utilizing Eq.(1) and assess x_{new} utilizing Eq.(3). With the new x_{new} values continue from steps 2a to 2c.
- **c.** Compare the fitness values and hold the best.

Step 4: *Learner Phase*: Obtain x_{new} in learners phase utilizing Eq.(4) and Eq.(5). With the new x_{new} values continue from step 2(ii) to 2(iii).

- **a.** Compute and look at the fitness function values and hold the best.

Step 5: *Termination Criterion*: Stop if the most number of generations is reached. In any case goto step 3.

4. EXPERIMENTS AND RESULTS

In this section, the condition of the analyses for the proposed method is presented. The depiction of benchmark pictures is presented right off the bat, at that point, the parameters set for the TLBO method are represented quickly and the quality measurements are utilized to assess the nature of the thresholding procedure. The paper demonstrates the implementation procedure of the TLBO algorithm for taking care of multilevel image thresholding as shown in Fig.1.

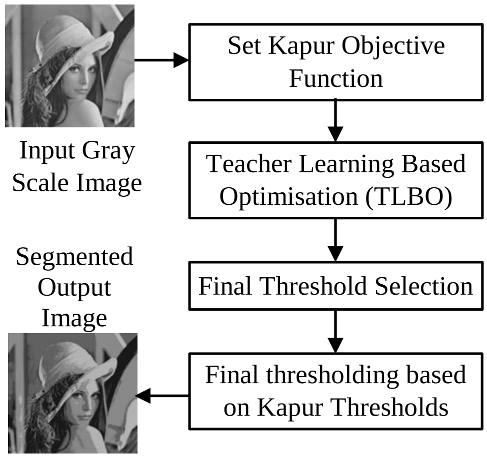


Fig.1. Implementation procedure of the TLBO algorithm

4.1 BENCHMARK IMAGES

The seven standard benchmark test pictures are three generally used images: (a) Cameraman, (b) House, (c) Lena, (d) Lake, (e) Jetplane, (f) Living room and (g) Peppers, as appeared in Fig.2, individually. The size of each tried benchmark picture is 512×512 pixels with 8-bit gray-levels. The three test images are named (a) Obama, (b) Trump, and (c) Modi respectively as shown in Fig.3. Size of the image Obama, Trump, and Modi are 256×256 pixels with 8-bit gray-levels.

4.2 EXPERIMENTAL SETTINGS

In this research, two analyses were directed. In the main set, tests are done on seven benchmark gray-scale pictures, (a) Cameraman, (b) House, (c) Lena, (d) Lake, (e) Jetplane, (f) Living room, and (g) Peppers (refer to Fig.2), with the size of 512×512, Favg (Eq.7), and the Jaccard metric (Eq.8) [24] are utilized to look at image thresholding execution, while in the subsequent set, tests are done on three-test images.

The application and execution of the TLBO method for taking care of MT issues have been uncovered by actualizing on seven standard benchmark test images. To show the adequacy of the proposed algorithm, different three test images have been thought of. The parameters picked to obtain the optimal threshold values are population size = 50 and most number of generations = 100.

The tests were completed on an HCL Laptop with an Intel Core i5 (2.40GHz) processor and 4GB memory. All the algorithms are implemented in Matlab2015 and actualized on Windows 7 - 32 bits.

4.3 SEGMENTED IMAGE QUALITY METRICS

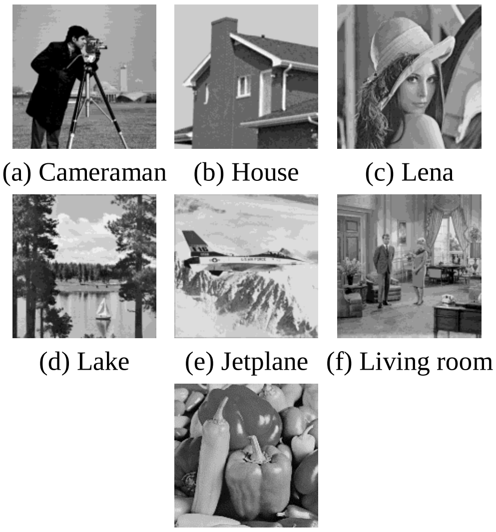
To judge the quality of the algorithm to choose multi-thresholds, the total average of the fitness function and Jaccard measure are utilized. These measurements are utilized to figure the stability of the calculations and are characterized as:

$$F_{avg} = \frac{1}{T}\sum_{i=1}^{T}(F_i), \tag{7}$$

To judge the quality of the algorithm to choose multi-thresholds, the total average of the fitness function and Jaccard measure are utilized. These measurements are utilized to figure the stability of the calculations and are characterized as:

$$J_{ac} = \frac{\left|I_{original} \cap J_{segmentedimage}\right|}{\left|I_{original} \cup J_{segmentedimage}\right|}, \tag{8}$$

It is a measure of similarity for the two sets of images, with a range from 0 to 1. The best method is the one that has a higher estimation of J_{ac}.



(a) Cameraman (b) House (c) Lena

(d) Lake (e) Jetplane (f) Living room

(g) Peppers

Fig.2. Standard benchmark test images

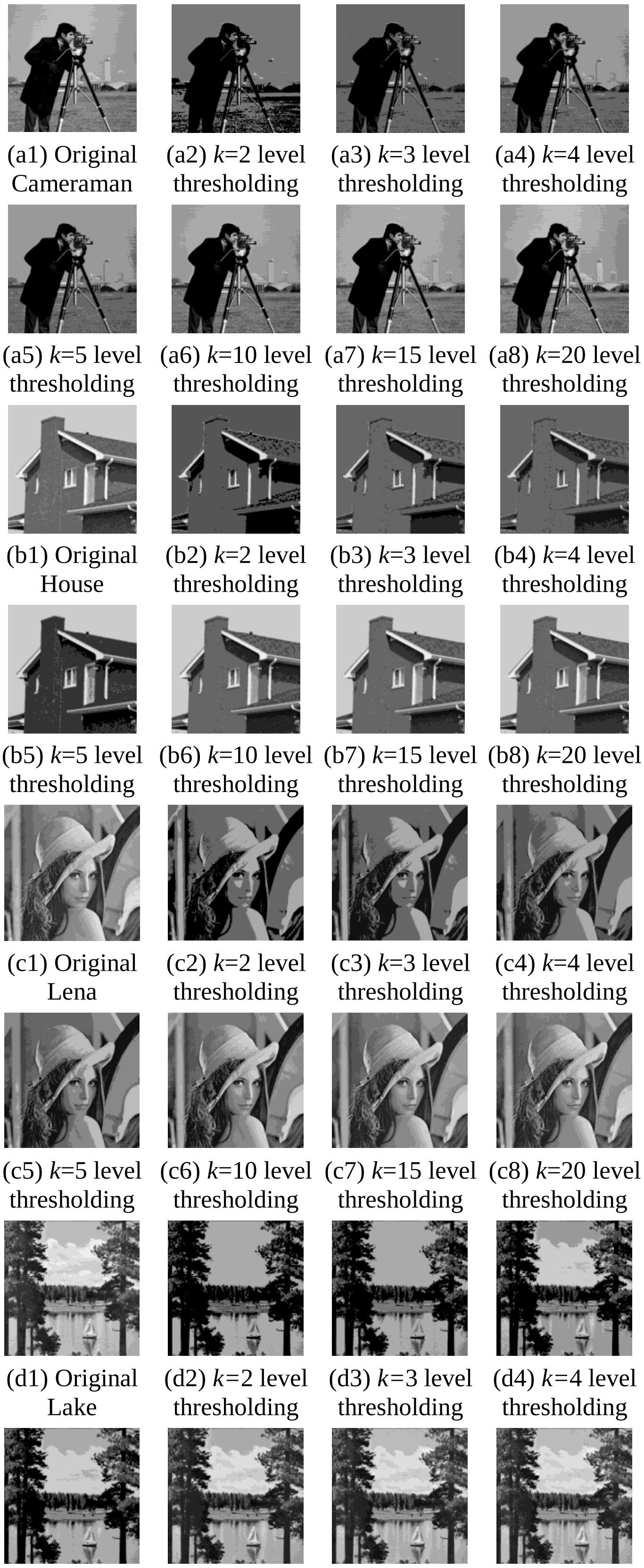
(a) Obama (b) Trump (c) Modi

Fig.3. Three test images

4.4 THE RESULTS AND DISCUSSIONS

Since TLBO is stochastic, it is important to utilize a proper statistical measurement to quantify its efficiency. To keep up similarity with comparable works detailed in the writing [20], the number of thresholds focuses utilized in the test are k = 2, 3, 4, 5, 10, 15, 20, 30, 40, and 50.

The enhanced visualizations of Fig.2 and Fig.3 at (a2) - (a8) for (a1), (b2) - (b8) for (b1), (c2) - (c8) for (c1), (c2) - (c8) for (c1), (d2) - (d8) for (d1), (e2) - (e8) for (e1), (f2) - (f8) for (f1), (g2) - (g8) for (g1), (h2) - (h8) for (h1), (i2) - (i8) for (i1), (j2) - (j8) for (j1) represent different threshold levels k = 2, 3, 4, 5, 10, 15, and 20 respectively are shown in Fig.4 which shows that the nature of the segmented image comes about because of applying the TLBO algorithm.



(a1) Original Cameraman (a2) k=2 level thresholding (a3) k=3 level thresholding (a4) k=4 level thresholding

(a5) k=5 level thresholding (a6) k=10 level thresholding (a7) k=15 level thresholding (a8) k=20 level thresholding

(b1) Original House (b2) k=2 level thresholding (b3) k=3 level thresholding (b4) k=4 level thresholding

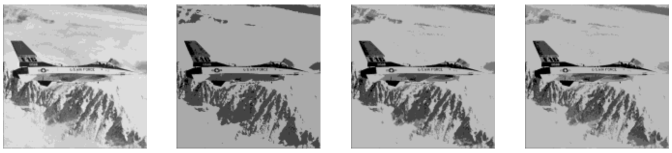
(b5) k=5 level thresholding (b6) k=10 level thresholding (b7) k=15 level thresholding (b8) k=20 level thresholding

(c1) Original Lena (c2) k=2 level thresholding (c3) k=3 level thresholding (c4) k=4 level thresholding

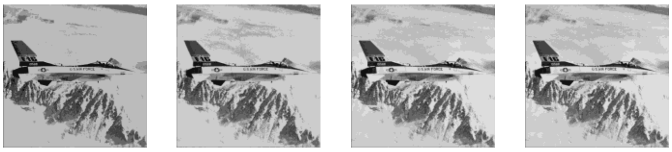
(c5) k=5 level thresholding (c6) k=10 level thresholding (c7) k=15 level thresholding (c8) k=20 level thresholding

(d1) Original Lake (d2) k=2 level thresholding (d3) k=3 level thresholding (d4) k=4 level thresholding

(d5) k=5 level thresholding (d6) k=10 level thresholding (d7) k=15 level thresholding (d8) k=20 level thresholding



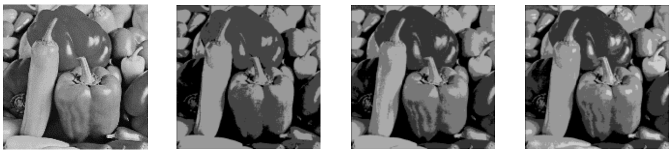
(e1) Original Jetplane (e2) k=2 level thresholding (e3) k=3 level thresholding (e4) k=4 level thresholding



(e5) k=5 level thresholding (e6) k=10 level thresholding (e7) k=15 level thresholding (e8) k=20 level thresholding

(f1) Original Living room (f2) k=2 level thresholding (f3) k=3 level thresholding (f4) k=4 level thresholding

(f5) k=5 level thresholding (f6) k=10 level thresholding (f7) k=15 level thresholding (f8) k=20 level thresholding

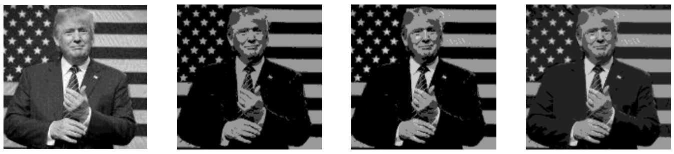


(g1) Original Peppers (g2) k=2 level thresholding (g3) k=3 level thresholding (g4) k=4 level thresholding

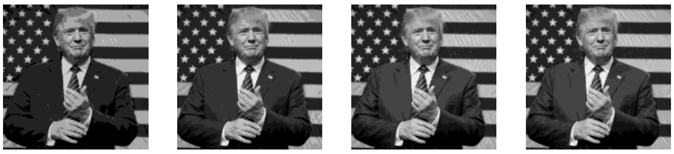
(g5) k=5 level thresholding (g6) k=10 level thresholding (g7) k=15 level thresholding (g8) k=20 level thresholding

(h1) Original Image (h2) k=2 level thresholding (h3) k=3 level thresholding (h4) k=4 level thresholding

(h5) k=5 level thresholding (h6) k=10 level thresholding (h7) k=15 level thresholding (h8) k=20 level thresholding



(i1) Original image (i2) k=2 level thresholding (i3) k=3 level thresholding (i4) k=4 level thresholding



(i5) k=5 level thresholding (i6) k=10 level thresholding (i7) k=15 level thresholding (i8) k=20 level thresholding

(j1) Original image (j2) k=2 level thresholding (j3) k=3 level thresholding (j4) k=4 level thresholding

(j5) k=5 level thresholding (j6) k=10 level thresholding (j7) k=15 level thresholding (j8) k=20 level thresholding

Fig.4. Original and segmented images with various thresholds levels k = 2, 3, 4, 5, 10, 15, and 20 obtained by the TLBO

From Table.1, the optimal thresholds together with the fitness value, and Jaccard measures are computed by the TLBO using Kapur function at various threshold levels k = 2, 3, 4, 5, 10, 15, and 20 to the seven standard benchmark and three test images. Table 2 shows the examination of best average objective function values at various threshold levels k = 2, 3, 4, 5, 10, 15, 20, 30, 40, and 50. Higher is the average objective function value, better is the thresholding execution. It is seen that values got utilizing TLBO are higher when contrasted with different algorithms like SCA, WOA, HHA, SSA, BA, PSO, CSA, and EO. All these algorithms are implemented using the Java programming language for the sake of fair comparison in [20]. In this case, the entropy and the probability distribution of the image histogram can be maximized. Entropy is maximized here, which prompts higher objective function values. The average objective function values increment with increment in the level of thresholds true to form. The number of function assessments increments with more significant levels of thresholding. This is the motivation behind why one watches higher estimations of the average objective function values in Table.2.

The objective function values of the various threshold levels k = 2, 3, 4, 5, 10, 15, and 20 to the three test images are shows the adequacy of the proposed TLBO using Kapur function in Table.1. To judge the quality of the algorithm to choose multi-thresholds, the F_{avg}, and Jaccard measure is utilized.

Table.1. Optimal threshold, fitness value, and Jaccard measures gained by the TLBO

Test images	***k***	**Thresholds**	**Fitness value**	J_{ac}
Cameraman	2	125, 196	12.2717	0.61
	3	44, 102, 196	15.3795	0.78
	4	42, 96, 145, 198	18.5411	0.78
	5	24, 61, 99, 146, 198	21.3137	0.81
	10	20, 45, 70, 96, 120, 145, 169, 191, 210, 232	33.5438	0.82
	15	20, 32, 49, 66, 84, 98, 116, 131, 144, 160, 177, 191, 207, 224, 239	43.1904	0.82
	20	7, 18, 32, 45, 58, 71, 84, 96, 107, 119, 132, 145, 157, 169, 180, 192, 205, 219, 231, 243	51.5686	0.99
House	2	95, 208	10.7304	0.80
	3	47, 97, 208	13.6167	0.97
	4	20, 61, 98, 208	16.2330	0.99
	5	47, 94, 126, 191, 209	18.4336	0.97
	10	20, 47, 71, 96, 121, 148, 176, 203, 210, 231	31.1355	0.99
	15	12, 26, 46, 62, 77, 96, 113, 128, 146, 164, 183, 202, 210, 229, 247	40.8646	0.99
	20	11, 20, 28, 37, 47, 61, 73, 86, 98, 112, 124, 137, 150, 163, 176, 190, 203, 210, 231, 247	49.1109	0.99
Lena	2	97, 164	12.3464	0.71
	3	25, 97, 164	15.5652	1.00
	4	25, 82, 126, 175	18.5372	1.00
	5	25, 64, 97, 137, 179	21.2311	1.00
	10	25, 57, 77, 97, 118, 139, 160, 179, 198, 217	32.6201	1.00
	15	25, 44, 61, 78, 93, 107, 120, 135, 149, 163, 178, 192, 204, 219, 232	41.6122	1.00
	20	25, 35, 43, 52, 61, 71, 81, 92, 103, 114, 124, 136, 148, 160, 172, 183, 193, 205, 217, 229	49.2723	1.00
Lake	2	92, 163	12.5254	0.58
	3	73, 120, 170	15.5654	0.66
	4	71, 113, 157, 196	18.3682	0.67
	5	64, 98, 133, 167, 199	21.0256	0.71
	10	13, 35, 59, 81, 103, 125, 148, 171, 192, 212	32.6228	0.99
	15	13, 28, 44, 62, 76, 91, 105, 119, 133, 148, 163, 177, 193, 211, 228	42.0401	0.99
	20	13, 23, 33, 44, 56, 67, 79, 91, 103, 114, 125, 136, 147, 158, 169, 179, 191, 203, 216, 228	49.9992	0.99
Jetplane	2	71, 173	12.2115	0.97
	3	69, 127, 183	15.5039	0.97
	4	67, 106, 145, 185	18.3121	0.98
	5	16, 61, 104, 137, 187	21.0233	1.00
	10	16, 40, 62, 85, 105, 124, 144, 161, 184, 203	32.6280	1.00
	15	16, 34, 47, 62, 75, 87, 100, 114, 128, 141, 156, 169, 185, 198, 213	41.6851	1.00
	20	16, 27, 39, 51, 63, 74, 85, 95, 106, 117, 128, 138, 149, 160, 170, 181, 191, 201, 209, 217	48.9874	1.00
Living room	2	94, 175	12.4057	0.77
	3	47, 103, 175	15.5523	0.92
	4	47, 98, 149, 197	18.4703	0.93
	5	42, 85, 124, 162, 197	21.1495	0.94
	10	24, 47, 70, 94, 116, 139, 162, 184, 205, 236	32.8997	0.97
	15	16, 31, 46, 61, 77, 92, 107, 122, 138, 153, 169, 184, 201, 223, 236	42.1263	0.98
	20	11, 23, 35, 47, 60, 73, 86, 99, 111, 124, 136, 149, 162, 175, 188, 201, 214, 227, 236, 244	49.8083	0.99
Peppers	2	75, 146	12.6323	0.80
	3	61, 112, 164	15.6861	0.85

	4	45, 80, 125, 171	18.4938	0.90
	5	42, 77, 113, 153, 194	21.2755	0.90
	10	25, 44, 62, 81, 102, 121, 140, 159, 179, 199	32.586	0.94
	15	19, 33, 48, 62, 76, 90, 104, 118, 132, 145, 158, 171, 183, 196, 209	41.5190	0.96
	20	9, 19, 29, 38, 49, 61, 72, 83, 94, 105, 116, 128, 139, 150, 161, 172, 183, 194, 205, 215	48.7589	0.99
Obama	2	86, 162	12.5213	0.73
	3	62, 120, 177	15.6980	0.80
	4	45, 93, 144, 195	18.6618	0.83
	5	42, 86, 129, 172, 215	21.5101	0.84
	10	24, 48, 73, 97, 121, 146, 170, 194, 219, 231	34.0211	0.88
	15	15, 31, 46, 61, 77, 92, 107, 122, 137, 153, 169, 186, 203, 219, 231	43.8116	0.91
	20	13, 23, 33, 44, 55, 65, 76, 88, 100, 111, 122, 134, 146, 159, 173, 187, 203, 219, 231, 244	51.8307	0.92
Trump	2	79, 152	12.6185	0.38
	3	79, 146, 206	16.0828	0.38
	4	40, 81, 147, 206	19.0666	0.88
	5	40, 79, 121, 162, 207	21.9964	0.88
	10	36, 57, 79, 101, 122, 143, 163, 184, 206, 229	33.7248	0.90
	15	21, 38, 53, 66, 79, 96, 112, 128, 144, 160, 175, 190, 206, 222, 238	43.2946	0.95
	20	12, 23, 34, 44, 55, 67, 79, 92, 105, 117, 129, 141, 154, 166, 179, 192, 205, 217, 230, 243	51.3177	0.97
Modi	2	86, 158	12.3887	0.97
	3	86, 151, 205	15.5442	0.97
	4	67, 110, 158, 206	18.5945	0.98
	5	35, 74, 110, 158, 206	21.5131	0.99
	10	23, 42, 64, 86, 110, 135, 158, 180, 205, 228	33.3866	0.99
	15	18, 31, 42, 58, 74, 91, 110, 127, 143, 159, 174, 192, 208, 224, 240	43.1397	0.99
	20	12, 23, 32, 42, 55, 69, 82, 96, 109, 123, 136, 148, 159, 174, 186, 198, 210, 222, 234, 244	51.024	0.99

Table.2. Comparison of average of the fitness function (F_{avg}) values acquired utilizing various optimization algorithms

Methods	**Average of the fitness function (F_{avg}) values**									
	Thresholds levels, *k*									
	2	**3**	**4**	**5**	**10**	**15**	**20**	**30**	**40**	**50**
Cameraman test image										
TLBO	12.2687	15.3786	18.5330	21.3065	33.4784	43.1519	**51.1734**	**63.6507**	**72.7129**	**79.8464**
EO [20]	**12.2844**	**15.4002**	**18.5594**	**21.3264**	**33.5590**	**43.2440**	51.1500	63.3082	71.9235	76.9196
WOA [20]	**12.2844**	**15.4002**	18.5591	21.3258	33.5526	43.2291	51.0529	63.137	70.8257	75.6398
HHA [20]	12.2842	15.393	18.5291	21.2677	33.0439	41.8325	48.9322	62.2145	69.8032	75.3161
SCA [20]	12.2838	15.3947	18.5293	21.2613	33.0859	42.0819	49.3253	59.2768	66.1775	71.2321
PSO [20]	12.2844	15.4002	18.5594	21.3098	33.5461	42.8887	50.5297	59.4927	66.0217	70.6307
BA [20]	12.2837	15.3924	18.5057	21.2804	33.1896	42.5354	50.0055	61.2218	69.3100	74.3661
CSA [20]	12.2840	15.3938	18.5370	21.2750	33.0688	42.0200	49.0690	59.1808	66.0903	70.8393
SSA [20]	**12.2844**	15.3942	18.5565	21.3118	33.4725	42.5775	50.1877	62.0128	69.8982	74.8254
House test image										
TLBO	10.7185	13.6037	16.1567	18.5443	30.9666	**40.7318**	**48.8374**	**61.6891**	**71.7535**	**78.1927**
EO [20]	**10.7627**	**13.6567**	16.0188	18.5264	**31.0564**	40.6614	48.7683	61.1904	70.0402	75.9533
WOA [20]	10.7608	13.6530	16.2098	**18.5586**	30.8506	40.3845	48.3588	60.3357	68.5505	73.0770
HHA [20]	10.7613	13.6129	16.1321	18.4284	30.196	39.2259	46.3133	59.6055	67.485	73.3144
SCA [20]	10.7607	13.5975	16.1486	18.4627	30.1948	39.315	46.4282	56.9888	63.7988	68.7655

PSO [20]	**10.7627**	**13.6567**	**16.2619**	18.5537	30.8606	39.9169	47.0462	56.6042	63.8004	68.7110
BA [20]	10.7531	13.5599	16.1003	18.4104	30.1908	39.4323	47.115	58.8423	67.0638	72.0987
CSA [20]	10.7605	13.5937	16.1053	18.4397	30.3261	39.2596	46.5362	57.1936	63.7188	69.0858
SSA [20]	10.7608	13.6112	16.1485	18.4821	30.5845	39.9713	47.4646	59.1837	67.3441	72.4336
Lena test image										
TLBO	**12.3458**	**15.5624**	**18.5202**	**21.2095**	**32.4513**	**41.4930**	**48.6549**	**60.3792**	**67.1620**	**73.4638**
EO [20]	12.3447	15.3123	18.0000	20.6071	31.4215	40.1678	47.4679	58.6289	65.8907	70.1318
WOA [20]	12.3447	15.3123	18.0104	20.6069	31.4174	40.0730	47.322	57.6599	64.0561	67.9429
HHA [20]	12.3444	15.3097	17.9846	20.5342	30.8171	38.7786	44.9852	56.8433	62.6129	66.7760
SCA [20]	12.3445	15.3108	17.9913	20.5623	31.0681	39.0505	45.2668	54.1149	59.2393	62.8502
PSO [20]	12.3447	15.3123	18.0019	20.6071	31.3751	40.0736	47.0159	56.4881	61.9209	65.2505
BA [20]	12.3443	15.3081	17.9923	20.5186	30.9608	39.4162	45.6805	55.0831	61.3615	66.0501
CSA [20]	12.3444	15.3108	17.9936	20.5664	31.0157	39.0500	45.2688	53.6043	59.3639	62.9774
SSA [20]	12.3447	15.3122	18.0045	20.6009	31.2548	39.8302	46.2318	56.1225	61.9115	66.2675
Lake test image										
TLBO	**12.5250**	**15.5639**	**18.3640**	**21.0141**	**32.6100**	**42.0226**	**49.7892**	**61.6195**	**70.6074**	**76.2493**
EO [20]	12.4920	15.5467	18.3224	20.9908	32.5814	42.0192	49.7770	61.5921	69.5696	74.9453
WOA [20]	12.4920	15.5467	18.3287	20.9939	32.6029	41.8490	49.6482	61.1710	68.3939	72.9805
HHA [20]	12.4919	15.5427	18.3016	20.9214	32.0070	40.6283	47.4934	60.2210	67.3841	72.4774
SCA [20]	12.4919	15.5442	18.3171	20.9626	32.1048	40.912	47.5613	57.4127	63.8521	68.2539
PSO [20]	12.4920	15.5467	18.3288	20.9908	32.4701	41.6167	48.4916	58.0381	64.4272	67.9253
BA [20]	12.4916	15.5307	18.3041	20.9134	32.2548	41.0659	48.3781	59.1389	67.3117	72.4789
CSA [20]	12.4918	15.5433	18.3152	20.9434	32.0538	40.9355	47.5522	57.2208	63.7721	68.2573
SSA [20]	12.4920	15.5466	18.3136	20.9847	32.4047	41.5014	49.1386	59.9185	67.6693	71.9562
Jetplane test image										
TLBO	12.2114	15.5012	18.3084	**21.0597**	**32.3563**	**41.3267**	**48.6178**	**59.9582**	66.1903	**72.9442**
EO [20]	**12.2607**	**15.5534**	**18.3666**	20.9681	31.9308	40.6672	47.8430	58.9933	**66.2906**	70.5503
WOA [20]	**12.2607**	**15.5534**	18.3665	20.967	31.9289	40.5819	47.7750	58.0223	64.3415	68.4693
HHA [20]	12.2605	15.5460	18.3483	20.9043	31.3379	39.2368	45.3242	57.4042	64.0238	67.8117
SCA [20]	12.2604	15.5485	18.3524	20.9181	31.5621	39.5422	45.5483	54.5935	59.5460	64.2197
PSO [20]	**12.2607**	**15.5534**	**18.3666**	20.9677	31.9256	40.6508	47.1357	56.4425	61.9342	65.0632
BA [20]	12.2589	15.5361	18.3394	20.9071	31.52	39.7686	46.3258	56.0881	63.0703	66.3891
CSA [20]	12.2602	15.5480	18.3526	20.9212	31.5612	39.6246	45.5278	54.0246	59.8482	63.4959
SSA [20]	12.2606	15.5531	18.3646	20.9617	31.7915	40.1488	47.0391	56.9536	63.5957	67.8002
Living room test image										
TLBO	12.4055	15.5503	18.4648	21.1396	**32.8712**	42.0081	49.5879	61.2393	69.7757	75.2012
EO [20]	**12.6968**	**15.9376**	**18.9441**	**21.7281**	33.8699	**43.6058**	**51.5558**	**63.6198**	**71.9313**	**77.3192**
WOA [20]	**12.6968**	**15.9376**	18.9436	**21.7281**	33.8571	43.4949	51.3161	63.5894	71.2537	76.2743
HHA [20]	12.6966	15.9334	18.9280	21.6701	33.4419	42.3966	49.4925	62.8503	70.0848	75.0848
SCA [20]	12.6965	15.9348	18.9319	21.7000	33.4898	42.4250	49.4525	59.6960	66.1912	70.846
PSO [20]	**12.6968**	**15.9376**	18.9440	21.7265	33.7946	42.9211	50.0175	60.0761	66.0772	70.3306
BA [20]	12.6964	15.9304	18.9163	21.6723	33.5225	42.8895	50.4389	61.4662	69.5373	74.7284
CSA [20]	12.6966	15.9342	18.9323	21.6996	33.4205	42.4708	49.5924	59.5823	66.3864	71.0637
SSA [20]	**12.6968**	**15.9376**	18.9419	21.7128	33.7467	43.1396	50.7400	62.1823	69.7134	74.9288
Peppers test image										
TLBO	**12.6323**	**15.6852**	**18.5255**	**21.2722**	**32.5463**	**41.3180**	**48.5786**	**60.0273**	**67.9122**	**73.7977**
EO [20]	12.5888	15.6215	18.4494	21.1693	32.4063	41.236	48.5733	59.6414	67.1298	71.5093

WOA [20]	12.5888	15.6215	18.4508	21.169	32.4201	41.1888	48.3582	59.1248	65.7456	69.8383
HHA [20]	12.5887	15.6205	18.4385	21.1163	31.7660	39.8494	45.9843	58.2227	64.4569	68.8863
SCA [20]	12.5887	15.6208	18.4426	21.138	31.9880	40.1784	46.4947	55.1098	61.1123	64.9364
PSO [20]	12.5888	15.6215	18.4478	21.1693	32.4042	41.0936	47.7139	56.9219	62.2085	65.4690
BA [20]	12.5878	15.6187	18.4383	21.1128	32.0420	40.3131	46.8885	57.306	64.0379	68.0441
CSA [20]	12.5886	15.6207	18.4432	21.1421	32.0107	40.2436	46.3308	55.124	60.7316	64.4661
SSA [20]	12.5888	15.6213	18.4449	21.1637	32.3245	40.7725	47.6883	58.076	65.4988	69.1572

5. CONCLUSION

This paper addresses TLBO based solution algorithm for taking care of Kapur entropy issues in MT. The proposed algorithm is executed on seven standard benchmark pictures and different three test pictures are taken for the examination so as to show its efficacy. The acquired solution gives the greatest entropy and the probability distribution of the image histogram that guarantees the best thresholding. The numerical outcomes are contrasted and the current writing algorithms that show the proposed algorithm is increasingly powerful in finding the global optimal solution for image thresholding issues. The proposed algorithm is appropriate for thresholding of any size and gives the greatest average objective function values for standard benchmark test pictures. The empowering simulation results show that the proposed approach is fit for getting progressively efficient, excellent solutions, stable combination attributes, and great computational efficiency. In the future, this algorithm can be applied to other entropy measures.

REFERENCES

[1] S. Dey, S. Bhattacharyya and U. Maulik, “Quantum Behaved Multi-Objective PSO and ACO Optimization for Multi-Level Thresholding”, *Proceedings of International Conference on Computational Intelligence and Communication Networks*, pp. 242-246, 2014.

[2] A.K. Bhandari, A. Kumar and G.K. Singh, “Tsallis Entropy based Multilevel Thresholding for Colored Satellite Image Segmentation using Evolutionary Algorithms”, *Expert Systems with Applications*, Vol. 42, No. 22, pp. 8707-8730, 2015.

[3] N. Otsu, “A Threshold Selection method from Gray-Level Histograms”, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 9, No. 1, pp. 62-66, 1979.

[4] J.N. Kapur, P.K. Sahoo and A.K. 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.

[5] L.K. Huang and M.J.J. Wang, “Image Thresholding by Minimizing the Measures of Fuzziness”, *Pattern Recognition*, Vol. 28, No. 1, pp. 41-51, 1995.

[6] Y. Qiao, Q. Hu, G. Qian, S. Luo and W.L. Nowinski, “Thresholding based on Variance and Intensity Contrast”, *Pattern Recognition*, Vol. 40, No. 2, pp. 596-608, 2007.

[7] X. Li, Z. Zhao and H.D. Cheng, “Fuzzy Entropy Threshold Approach to Breast Cancer Detection”, *Information Sciences Applications*, Vol. 4, No. 1, pp. 49-56, 1995.

[8] C.H. Li and P.K.S. Tam, “An Iterative Algorithm for Minimum Cross Entropy Thresholding”, *Pattern Recognition Letters*, Vol. 19, No. 8, pp. 771-776, 1998.

[9] K. Li and Z. Tan, “An Improved Flower Pollination Optimizer Algorithm for Multilevel Image Thresholding”, *IEEE Access*, Vol. 7, pp. 165571-165582, 2019.

[10] J. Kittler and J. Illingworth, “Minimum Error Thresholding”, *Pattern Recognition*, Vol. 19, No. 1, pp. 41-47, 1986.

[11] S. Zarezadeh and M. Asadi, “Results on Residual Renyi Entropy of Order Statistics and Record Values”, *Information Sciences*, Vol. 180, No. 21, pp. 4195-4206, 2010.

[12] S. Ouadfel and A. Taleb-Ahmed, “Social Spiders Optimization and Flower Pollination Algorithm for Multilevel Image Thresholding: A Performance Study”, *Expert Systems with Applications*, Vol. 55, pp. 566-584, 2016.

[13] S.S. Pal, S. Kumar, K. Kashyap, Y. Choudhary and M. Bhattacharya, “Multi-Level Thresholding Segmentation Approach based on Spider Monkey Optimization Algorithm”, *Proceedings of International Conference on Computer and Communication Technologies*, pp. 273-287, 2016.

[14] M.A. El Aziz, A.A. Ewees and A.E. Hassanien, “Whale Optimization Algorithm and Moth-Flame Optimization for Multilevel Thresholding Image Segmentation”, *Expert Systems with Applications*, Vol. 83, pp. 242-256, 2017.

[15] A. K. M. Khairuzzaman, and S. Chaudhury, “Multilevel Thresholding using Grey Wolf Optimizer for Image Segmentation”, *Expert Systems with Applications*, Vol. 86, pp. 64-76, 2017.

[16] S. Kotte, R.K. Pullakura and S.K. Injeti, “Optimal Multilevel Thresholding Selection for Brain MRI Image Segmentation based on Adaptive Wind Driven Optimization”, *Measurement*, Vol. 130, pp. 340-361, 2018.

[17] K.B. Resma and M.S. Nair, “Multilevel Thresholding for Image Segmentation using Krill Herd Optimization Algorithm”, *Journal of King Saud University-Computer and Information Sciences*, Vol. 12, No, 3, pp. 1-13, 2018.

[18] M. Ahmadi, K. Kazemi, A. Aarabi, T. Niknam and M.S. Helfroush, “Image Segmentation using Multilevel Thresholding based on Modified Bird Mating Optimization”, *Multimedia Tools and Applications*, Vol. 78, No. 16, pp. 23003-23027, 2019.

[19] A. Sharma, R. Chaturvedi, S. Kumar and U.K. Dwivedi, “Multi-Level Image Thresholding based on Kapur and Tsallis Entropy using Firefly Algorithm”, *Journal of Interdisciplinary Mathematics*, Vol. 23, No. 2, pp. 563-571, 2020.

[20] M. Abdel-Basset, V. Chang and R. Mohamed, “A Novel Equilibrium Optimization Algorithm for Multi-

Thresholding Image Segmentation Problems", *Neural Computing and Applications*, Vol. 34, No. 1, pp.1-34, 2020.

[21] R.V. Rao, V.J. Savsani and D.P. Vakharia, "Teaching-Learning-Based Optimization: An Optimization Method for Continuous Non-Linear Large-Scale Problems", *Information Sciences*, Vol. 183, No. 1, pp. 1-15, 2012.

[22] R.V. Rao, V.J. Savsani and J. Balic, "Teaching-Learning-Based Optimization Algorithm for Unconstrained and Constrained Real-Parameter Optimization Problems", *Engineering Optimization*, Vol. 44, No. 12, pp. 1447-1462, 2012.

[23] R.V. Rao and V.D. Kalyankar, "Parameter Optimization of Machining Processes using a New Optimization Algorithm", *Materials and Manufacturing Processes*, Vol. 27, No. 9, pp. 978-985, 2012.

[24] M.A. Elaziz, A.A. Ewees and A.E. Hassanien, "Hybrid Swarms Optimization based Image Segmentation", *Proceedings International Conference on Hybrid Soft Computing for Image Segmentation*, pp. 1-21, 2016.