Mathematical Biosciences and Engineering

MBE, 16(6): 6467–6511.
DOI: 10.3934/mbe.2019324
Received: 01 April 2019
Accepted: 30 May 2019
Published: 15 July 2019

http://www.aimspress.com/journal/MBE

Research article

Modified dragonfly algorithm based multilevel thresholding method for color images segmentation

Xiaoxu Peng, Heming Jia* and Chunbo Lang

College of Mechanical and Electrical Engineering, Northeast Forestry University, Harbin 150040, China

*** Correspondence:** Email: jiaheming@nefu.edu.cn.

Abstract: Accurate image segmentation is the preprocessing step of image processing. Multi-level threshold segmentation has important research value in image segmentation, which can effectively solve the problem of region analysis of complex images, but the computational complexity increases accordingly. In order to overcome this problem, an modified Dragonfly algorithm (MDA) is proposed to determine the optimal combination of different levels of thresholds for color images. Chaotic mapping and elite opposition-based learning strategies (EOBL) are used to improve the randomness of the initial population. The hybrid algorithm of Dragonfly Algorithms (DA) and Differential Evolution (DE) is used to balance the two basic stages of optimization: exploration and development. Kapur entropy, minimum cross-entropy and Otsu method are used as fitness functions of image segmentation. The performance of 10 test color images is evaluated and compared with 9 different meta-heuristic algorithms. The results show that the color image segmentation method based on MDA is more effective and accurate than other competitors in average fitness value (AF), standard deviation (STD), peak signal-to-noise ratio (PSNR), structural similarity index (SSIM) and feature similarity index (FSIM). Friedman test and Wilcoxon’s rank sum test are also performed to assess the significant difference between the algorithms.

Keywords: Dragonfly algorithm; multilevel thresholding; Kapur's entropy; minimum cross entropy; Otsu method; elite opposition-based learning; differential evolution

1. Introduction

Image segmentation is a fundamental preprocessing stage in computer vision, and the main goal of which is to partition a given image into several meaningful regions with respect to position, texture, gray level value, etc [1,2]. In the recent decades, many scholars and researchers have done a great deal of work in this field and a number of segmentation techniques have been proposed. Basically, the currently used segmentation techniques can be summarized as three main categories: edge - based method, region-based method, and threshold-based method [3–6]. Among the available segmentation methods, thresholding technique holds the prime position and becomes popular due to the strong robustness, fast computation speed and high accuracy.

The thresholding method can be broadly classified into bi-level and multilevel depending on the number of thresholds required to be determined. Bi-level thresholding method splits the image into two classes: foreground and background, while the multilevel thresholding method can partition the image into several similar parts based on intensity. Over the years, the multilevel thresholding technique has been used extensively and much work has been done on it. Among the various multilevel thresholding techniques, Kapur's entropy, Minimum cross entropy (MCE), and Otsu method (between-class variance) are the most popular ones, which perform much better than other existing methods [7–9]. Kapur's entropy method maximizes the histogram entropy of segmented classes to obtain the optimal threshold values. MCE method minimizes the cross entropy between the original classes and the segmented classes. Otsu method maximizes the between class variance of segmented classes and it has been widely used to solve image segmentation problems. When the number of thresholds is small, the performance of these segmentation techniques is satisfied. However, when the number of thresholds increases, the computational complexity will result in exponential growth since they search the solution space exhaustively to obtain the optimal thresholds.

Therefore, to avoid above weakness, a number of swarm intelligence (SI) techniques have been used extensively and combined with various thresholding methods, such as particle swarm optimization (PSO), genetic algorithm (GA), harmony search optimization (HSO), bat algorithm (BA), ant colony optimization (ACO), etc [10]. For example, Abdul et al. proposed a technique based on grey wolf optimization (GWO) and its combination with Kapur's entropy and Otsu method for image segmentation [11]. Also, Cuevas et al developed and evaluated a multilevel image thresholding method based on DE in 2010 [12]. Genyun Sun et al. utilized a novel hybrid algorithm of gravitational search algorithm (GSA) with GA for multi-level thresholding [13]. The experimental results indicate that GSA-GA has superior or comparative segmentation performance. In favor of satellite image segmentation, Bhandari et al. presented two multi-level thresholding techniques based on cuckoo search (CS) algorithm and wind driven optimization (WDO) using Kapur's entropy [14]. Moreover, Madhubanti and Amitava proposed a bacterial foraging optimization (BFO) based multilevel thresholding technique for magnetic resonance brain image segmentation [15]. They employed BFO to maximize the Kapur's function values. Furthermore, there are many other swarm intelligence algorithms and their modified algorithms, that were utilized for multilevel thresholding including: social spiders optimization (SSO) [16], hybrid differential evolution algorithm [17], modified bacterial foraging (MBF) [18], Patch-Levy-based bees algorithm (PLBA) [19], teaching-learning-based optimization (TLBO) [20], krill herd optimization (KHO) [21] and Lévy flight guided firefly algorithm (LFA) [22]. However, most of these techniques may get stuck in local optimal and the stability of them becomes poor when the number of thresholds is high due to several factors [23]. For instance, generating an inappropriate initial population will affect the convergence performance of proposed algorithm. Besides, the accuracy of solution may be influenced by the transformation of exploration and exploration phases.

In order to overcome the drawbacks above and obtain an efficient method for color image segmentation, MDA based color image segmentation using Kapur's entropy, MCE method and Otsu method is proposed in this paper. The initial and the updating phases of standard DA have been improved through the techniques as follows [24]. In the initial phase of standard DA, the chaotic maps and the EOBL strategy are adopted to enhance the randomness of initial population. Moreover, a hybrid algorithm of dragonfly algorithm and differential evolution is utilized to balance the exploration and exploitation of algorithm for updating phase. In this paper, the performance of the proposed method is validated on ten test color images and the experimental results indicate the appropriate performance of MDA based method over other multilevel thresholding methods. Additionally, several performance evaluation measures such as arithmetic mean (AM), standard deviation (STD), peak signal to noise ratio (PSNR), structural similarity index (SSIM) and feature similarity index (FSIM) are included for quantitative analysis. Besides, a non-parametric Wilcoxon's rank sum test has also been conducted in the paper for statistical analysis [25].

Detailed discussion in regard to the theory and realization of the MDA based method is given in the following sections. In Section 2, the general description of multilevel thresholding methods is introduced. The standard dragonfly algorithm is reviewed in Section 3. In Section 4, the proposed MDA based multilevel thresholding method is introduced in details. The environmental experiment of the proposed method is described in Section 5 and a comprehensive set of experimental results and comparison of other existing methods are provided in Section 6. Finally, the relevant conclusions and the future research directions are drawn in Section 7.

2. Multilevel thresholding

The image threshold method can be summarized as two categories: bi-level thresholding method and multilevel thresholding method. Bi-level thresholding method involves one threshold value which partitions the image into two classes: foreground and background, however if the image is quite complex and contains various objects, the bi-level thresholding method is not very effective [11]. Therefore, multilevel thresholding method is used extensively for image segmentation [26–31]. A brief formulation of three techniques is given in the following subsections. In addition, the RGB images has three basic color components of red, green and blue, so these thresholding techniques are executed three times to determine the optimal threshold values of each color component [32].

2.1. Kapur's entropy

Kapur's method is also an unsupervised automatic thresholding technique, which selects the optimum thresholds based on the entropy of segmented classes [8]. Assuming that $[th_1, th_2, ..., th_n]$ represents the threshold values which divided the image into various classes. Then the object function of Kapur's method can be defined as:

$$H(th_1, th_2, ..., th_n) = H_0 + H_1 + ... + H_n \tag{1}$$

Where

$$H_0 = -\sum_{j=0}^{th_1-1} \frac{p_j}{\omega_0} \ln \frac{p_j}{\omega_0}, \omega_0 = \sum_{j=0}^{th_1-1} p_j \tag{2}$$

$$H_1 = -\sum_{j=th_1}^{th_2\text{-}1} \frac{p_j}{\omega_1} \ln \frac{p_j}{\omega_1}, \omega_1 = \sum_{j=th_1}^{th_2-1} p_j \tag{3}$$

$$H_n = -\sum_{j=th_n}^{L-1} \frac{p_j}{\omega_n} \ln \frac{p_j}{\omega_n}, \omega_n = \sum_{j=th_n}^{L-1} p_j \tag{4}$$

$H_0, H_1, \ldots, H_n$ denote the entropies of distinct classes. $\omega_0, \omega_1, \ldots, \omega_n$ are the probability of each class. In order to obtain the optimal threshold values, the fitness function in Eq. (1) is maximized.

$$f_{Kapur}\left(th_1, th_2, ..., th_n\right) = \arg\max\left\{H\left(th_1, th_2, ..., th_n\right)\right\} \tag{5}$$

2.2. Minimum cross entropy

Minimum cross entropy (MCE) method finds the optimal threshold values based on minimizing the cross entropy between the original image and the segmented image [9]. MCE method can be defined as minimizing the following objective function:

$$M\left(th_1, th_2, ..., th_n\right) = M_G + M_0 + M_1 + ... + M_n \tag{6}$$

where

$$M_G = \sum_{j=0}^{L-1} jp_j \log\left(j\right) \tag{7}$$

$$M_0 = -\sum_{j=0}^{th_1-1} jp_j \log\left(\mu_0\right), \mu_0 = \sum_{j=0}^{th_1-1} \frac{jp_j}{\omega_0}, \omega_0 = \sum_{j=0}^{th_1-1} p_j \tag{8}$$

$$M_1 = -\sum_{j=th_1}^{th_2\text{-}1} jp_j \log\left(\mu_1\right), \mu_1 = \sum_{j=th_1}^{th_2\text{-}1} \frac{jp_j}{\omega_1}, \omega_1 = \sum_{j=th_1}^{th_2-1} p_j \tag{9}$$

$$M_n = -\sum_{j=th_n}^{L-1} jp_j \log\left(\mu_n\right), \mu_n = \sum_{j=th_n}^{L-1} \frac{jp_j}{\omega_n}, \omega_n = \sum_{j=th_n}^{L-1} p_j \tag{10}$$

$M_0, M_1, \ldots, M_n$ represent the cross entropies of distinct classes. $\omega_0, \omega_1, \ldots, \omega_n$ are the probability of each class. M_G is a constant. Thus, the objective function in Eq. (6) can be rewritten as:

$$\eta\left(th_1, th_2, ..., th_n\right) = M_0 + M_1 + ... + M_n \tag{11}$$

$$\begin{aligned}\eta\left(th_1, th_2, ..., th_n\right) = &-\sum_{j=0}^{th_1-1} jp_j \log\left(\frac{\sum_{j=0}^{th_1-1} jp_j}{\sum_{j=0}^{th_1-1} p_j}\right) - \sum_{j=th_1}^{th_2-1} jp_j \log\left(\frac{\sum_{j=th_1}^{th_2-1} jp_j}{\sum_{j=th_1}^{th_2-1} p_j}\right) \\ &- ... - \sum_{j=th_n}^{L-1} jp_j \log\left(\frac{\sum_{j=th_n}^{L-1} jp_j}{\sum_{j=th_n}^{L-1} p_j}\right)\end{aligned} \tag{12}$$

Let $m^0 = \sum_{j=a}^{b-1} p_j$ and $m^1 = \sum_{j=a}^{b-1} jp_j$, then

$$\eta\left(th_1,th_2,...,th_n\right)=-m^1\left(0,th_1-1\right)\log\left(\frac{m^1\left(0,th_1-1\right)}{m^0\left(0,th_1-1\right)}\right)-m^1\left(th_1,th_2-1\right)\log\left(\frac{m^1\left(th_1,th_2-1\right)}{m^0\left(th_1,th_2-1\right)}\right)$$
$$-...-m^1\left(th_n,L-1\right)\log\left(\frac{m^1\left(th_n,L-1\right)}{m^0\left(th_n,L-1\right)}\right) \quad (13)$$

The MCE method search the optimal threshold values by minimizing the objective function in Eq. (13).

$$f_{MCE}\left(th_1,th_2,...,th_n\right)=\arg\min\left\{\eta\left(th_1,th_2,...,th_n\right)\right\} \quad (14)$$

2.3. Otsu method

Otsu method (between-class variance) is a non-parametric thresholding technique, which selects the optimum thresholds based on the between-class variance of segmented classes [7]. Let L represents the number of gray levels in a given image and $\left[th_1,th_2,...,th_n\right]$ denotes the threshold values which are selected to partition the image into various classes. Then the objective function of Otsu method can be defined as:

$$\sigma_B^2\left(th_1,th_2,...,th_n\right)=\sigma_0^2+\sigma_1^2+...+\sigma_n^2 \quad (15)$$

Where

$$\sigma_0^2=\omega_0\left(\mu_0-\mu_T\right)^2,\omega_0=\sum_{j=0}^{th_1-1}p_j\,,\mu_0=\sum_{j=0}^{th_1-1}\frac{jp_j}{\omega_0} \quad (16)$$

$$\sigma_1^2=\omega_1\left(\mu_1-\mu_T\right)^2,\omega_1=\sum_{j=th_1}^{th_2-1}p_j\,,\mu_1=\sum_{j=th_1}^{th_2-1}\frac{jp_j}{\omega_1} \quad (17)$$

$$\sigma_n^2=\omega_n\left(\mu_n-\mu_T\right)^2,\omega_n=\sum_{j=th_n}^{L-1}p_j\,,\mu_n=\sum_{j=th_n}^{L-1}\frac{jp_j}{\omega_n} \quad (18)$$

μ_T denotes the mean intensity for whole image, $\mu_T=\sum_{j=0}^{L-1}jp_j=\omega_0\mu_0+\omega_1\mu_1+...+\omega_n\mu_n$

And Eq. (15) is maximized to obtain the optimal threshold values.

$$f_{otsu}\left(th_1,th_2,...,th_n\right)=\arg\max\left\{\sigma_B^2\left(th_1,th_2,...,th_n\right)\right\} \quad (19)$$

As we know, the computational complexity of these three thresholding techniques above will result in exponential growth as the number of thresholds increase. Under such circumstance, Kapur's entropy, MCE and Otsu method are not very effective for multilevel thresholding. Therefore, MDA is proposed to improve the accuracy and computation speed of thresholding techniques. The ultimate goal of proposed method is to determine the optimal threshold values by optimizing (either maximizing or minimizing) the objective function given in Eq. (1), Eq. (13), and Eq. (15).

3. Dragonfly algorithm

Dragonfly algorithm is inspired by the static and dynamic behaviors of dragonflies in nature. It is firstly proposed by Mirjalili to solve the problem of submarine propeller optimization [24]. In the static swarm, dragonflies create sub-swarms and fly over small areas to hunt the preys. This static behavior represents the exploitation phase of optimization. In the dynamic swarm, however, a mass of dragonflies make the group for migrating in one direction, which represents the exploration phase of optimization. Moreover, five factors are designed to simulate the behaviors of dragonflies, namely separation (S), alignment (A), cohesion (C), attraction towards food (F), and distraction outwards enemy (E). The mathematical model of these factors is given as follows:

The separation of ith dragonfly individual denoted by S_i is given by:

$$S_i = -\sum_{j=1}^{nd} x_i - x_j \tag{20}$$

where nd denotes the number of neighbors. x_i and x_j represent the position of current dragonfly and its neighbor respectively. It's worth noting that if the distance between x_i and x_j is less than the preset value, then x_j is a neighbor of x_i. And the preset value will increase with the number of iterations.

The alignment of ith dragonfly individual denoted by S_i is determined as follows:

$$A_i = \frac{\sum_{j=1}^{nd} v_j}{nd} \tag{21}$$

where v_j is the jth neighboring dragonfly's velocity. A_i shows the velocity consistency of the swarm.

The cohesion of ith dragonfly individual denoted by C_i is computed by:

$$C_i = \frac{\sum_{j=1}^{nd} x_j}{nd} - x_i \tag{22}$$

where x_i is the position of ith dragonfly individual. x_j is the jth neighboring dragonfly individual's position. nd denotes the number of neighbors.

The food source provides an attraction for ith dragonfly individual denoted by F_i and it can be defined as follows:

$$F_i = x_{food} - x_i \tag{23}$$

where x_i shows the position of ith dragonfly individual. x_{food} represents the position of food source.

The distraction outwards an enemy of ith dragonfly individual denoted by E_i is evaluated by:

$$E_i = x_{enemy} + x_i \tag{24}$$

where x_i is the position of ith dragonfly individual. x_{enemy} shows the position of the enemy. It is worth mentioning that the x_{food} and x_{enemy} represent the best and worst position respectively which the swarm of dragonflies has searched for so far.

The position vector of ith dragonfly individual during the interval $[t, t+1]$ can be calculated as follows:

$$x_i^{t+1} = x_i^t + \Delta x_i^{t+1} \tag{25}$$

where

$$\Delta x_i^{t+1} = \left(sS_i + aA_i + cC_i + fF_i + eE_i\right) + \omega \Delta x_i^t \tag{26}$$

Δx indicates the direction of the movement of dragonfly individual. s, a, c, f, and e denote the weight for five factors namely separation, alignment, cohesion, food, and enemy respectively. ω represents the inertia weight. t shows the iteration counter.

It is worthy of noting that if there is no neighbor individual, the current dragonfly will fly around the search space using a random walk (Levy's flight) to improve the performance of algorithm. Under such circumstance, the position of ith dragonfly individual is updated by the equation as follows:

$$x_i^{t+1} = x_i^t + \mathrm{Levy} \times x_i^t \tag{27}$$

Where

$$\mathrm{Levy} = 0.01 \times \frac{r_1 \times \sigma}{|r_2|^{1/\beta}} \tag{28}$$

$$\sigma = \left(\frac{\Gamma(1+\beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right)^{1/\beta} \tag{29}$$

$$\Gamma(x) = (x-1)! \tag{30}$$

r_1 and r_2 are two random numbers in the range of $[0,1]$. β is a constant. Pseudo code of dragonfly algorithm based multilevel thresholding has been given in Figure 1.

Initialize the position of the dragonfly x_i and the step vector $\Delta x\ (i = 1, 2, ..., n)$.
Initialize the food source x_{food} and enemy x_{enemy}.
WHILE $(t < \textit{Maximum number of iterations})$
Calculate the objective value of each dragonfly by using the Eq. (1) for Kapur , Eq. (13) for MCE or Eq. (15) for Otsu.
Update the food source and enemy.
Update s, a, c, f, e, and ω
Evaluate S, A, C, F, and E using Eq. (20) to Eq. (24).
Calculate the number of neighboring individual, nd, of the current dragonfly within the radius.
IF *nd = 0*
Update the position of the current dragonfly using Eq. (27).
ELSE
Update the position of the current dragonfly using Eq. (25).
END IF
Correct the position of the current dragonfly if it is beyond the border.
END WHILE
Return x_{food}, which represents the optimal threshold values of segmentation.

Figure 1. Pseudo codes of dragonfly algorithm based multilevel thresholding.

4. Proposed method

In this section, we give a detailed introduction of the MDA based method that will be used to obtain the optimal threshold values for image segmentation. The chaotic maps and the elite opposition-based learning (EOBL) strategy are utilized to improve the randomness of initial population. Then, a hybrid algorithm of DA and DE is used to balance the two essential phases of optimization, namely exploration and exploitation [33–37]. Besides, the flowchart of MDA for finding the optimal threshold values is shown in Figure 2.

4.1. Initial phase

In this phase, the initial population is generated by an efficient method that combines the chaotic maps and the EOBL strategy [23]. A brief explanation of these two techniques is given in the subsections below.

4.1.1. Chaotic maps

In the standard DA, the initialization of dragonflies is generated randomly. The chaotic maps can explore the search space more efficient than the existing random generators based on probabilities, on account of their special properties including ergodicity, unpredictability, and non-repetition. Therefore, the logistic map is adopted to generate the initial population in this paper with the purpose of improving the diversity of dragonflies. Mathematical definition of the logistic map is as follows:

$$x_{n+1} = \mu \times x_n \left(1 - x_n\right) \tag{31}$$

where μ is the growth rate parameter, $\mu \in \left[0,4\right]$. x_n represents the value obtained at nth step, $x_n \in \left[0,1\right]$.

4.1.2. Elite Opposition-based learning strategy

For a feasible solution, calculate the current solution and evaluate its opposite solution at the same time, then select the better solution as the next generation individual [34]. It is quite clear that this greedy selection method can guarantee the prominent acceleration in convergence as well as reducing the possibility of trapping into local optimal. The EOBL strategy can be summarized as follows:

$$x_j^* = k \cdot (da_j + db_j) - x_j,\ j = 1,2,...,Dim \tag{32}$$

$$da_j = min(x_j),\ db_j = max(x_j) \tag{33}$$

where Dim is the number of dimensions. $x_j \in (a_j, b_j)$, a_j and b_j denote the lower bound and the upper bound of the given search space respectively. k is a generalized coefficient uniformly distributed in the interval of [0,1]. da_j and db_j are the dynamic lower bound and the dynamic upper bound of the *jth* dimension search space, and are calculated according to Eq. (33). By replacing the fixed bound with the dynamic bound, the opposite solution x_j^* can be located in the gradually reduced search space, which promotes faster convergence of the algorithm.

For a problem to be maximized, the opposite solution will be selected if $f\left(x_j^*\right) > f\left(x_j\right)$; otherwise, continue with x_j for further generation [38,39].

4.2. Updating phase

In this phase, the solution is updated through a novel technique that combines the dragonfly algorithm and differential evolution algorithm. More details of the hybrid algorithm will be discussed in the subsection below.

4.2.1. Differential evolution

Differential evolution (DE) algorithm is a population-based stochastic optimization algorithm for solving optimization problems, which is introduced by Price and Storn [33,40]. Basically, the DE algorithm contains two significant parameters, namely mutation scaling factor denoted by SF and crossover probability denoted by CR. Same as the other meta-heuristic algorithms, several operators have been included in the DE algorithm such as mutation, crossover, and selection operators [40–42].

4.2.1.1. Mutation operation

The mutation operation of DE algorithm is defined as follows:

$$m_i^{g+1} = x_{r1}^g + SF * \left(x_{r2}^g - x_{r3}^g\right) \tag{34}$$

where m_i^{g+1} represents the mutant individual in the $(g+1)$th generation. x_{r1}^g, x_{r2}^g, and x_{r3}^g are different individuals from the population. In other words, r_1, r_2, and r_3 cannot be equal. SF is a constant that indicates the mutation scaling factor.

4.2.1.2. Crossover operation

In the process of crossover, the trial individual c_i^{g+1} is selected from the current individual x_i^g or the mutant individual m_i^{g+1} on account of enhancing the diversity of population[43,44]. The crossover operation of DE algorithm is described as:

$$c_i^{g+1} = \begin{cases} m_i^{g+1} & if\ \ rand \le CR \\ x_i^g & if\ \ rand > CR \end{cases} \tag{35}$$

where $rand$ represents a random value which is in the range $[0,1]$. CR is a constant that shows the crossover probability.

4.2.1.3. Selection operation

After the process of selection, the individual of next generation x_i^{g+1} is selected according to the comparison of fitness value between the trail individual c_i^{g+1} and the target individual x_i^g. For a problem to be minimized, the selection operation of DE algorithm can be summarized as follows:

$$x_i^{g+1} = \begin{cases} c_i^{g+1} & if \;\; f\left(c_i^{g+1}\right) < f\left(x_i^g\right) \\ x_i^g & otherwise \end{cases} \tag{36}$$

where f denotes the fitness function value of a given problem.

4.2.2. Hybrid algorithm of DA and DE

In order to improve the exploration and exploitation performance of the proposed algorithm, the hybrid algorithm between the DA and the DE is utilized. On the one hand, the DA algorithm has a satisfied capability of avoiding convergence to the local optimum, thus it is served as global search technique. On the other hand, the DE algorithm is adopted as local search technique, which can increase the precision of solutions.

As we know, the fitness value of current solution indicates its quality. Therefore, we calculate the average fitness value of population in the iterative process to evaluate each particle. All fitness values are presented as absolute values to accommodate Kapur's entropy, MCE and Otsu, threshold techniques. If $|f_i| > |\bar{f}|$, the DE algorithm will be used to update the solution x_i^g using Eq. (34) to Eq. (36). However, if $|f_i| \le |\bar{f}|$, then the current solution will be updated using Eq. (25) or Eq. (27).

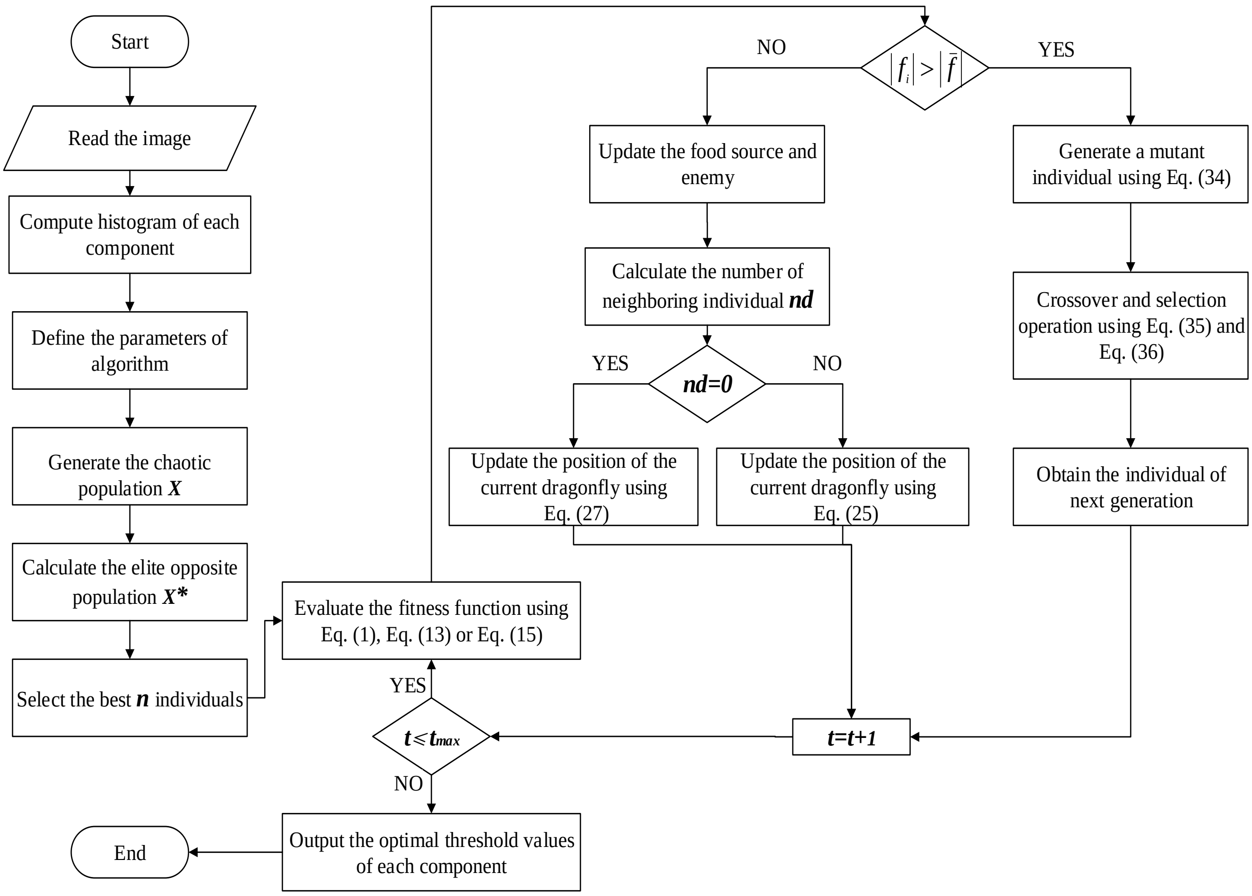

Figure 2. Flowchart of the MDA based method.

5. Experimental environment

5.1. Experimental setup

In order to verify the performance of proposed algorithm, ten color images from Berkley segmentation data set are used. The test images namely Bridge, Building, Cactus, Cow, Deer, Diver, Elephant, Horse, Kangaroo, Lake and their corresponding histograms for each of the color channels (Red, Green, and Blue) are shown in Figure 3. All test images aresize. Moreover, all the experimental series are carried out through simulations in MATLAB R2017a on a computer with the following configuration: Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz and 8GB RAM with Microsoft Windows 10 system (64bit).

5.2. Comparative algorithms

Performance of the proposed algorithm is compared with nine other swarm algorithms that are used extensively. These algorithms are:

1. Dragonfly Algorithm (DA): simulates the static and dynamic behaviors of dragonflies to find an optimal solution [24].
2. Salp Swarm Algorithm (SSA): mimics the original behavior of salp chain for finding food. And this algorithm is easy to implement because of the only parameter [45].
3. Sine Cosine Algorithm (SCA): the algorithm converges to the optimal solution depending on the oscillatory properties of sine and cosine functions [46,47].
4. Ant Lion Optimization (ALO): emulates the hunting mechanism of antlions which can be divided into five main steps such as the random walk of ants, building traps, entrapment of ants in traps, catching preys, and re - building traps [48,49].
5. Harmony Search Optimization (HSO): simulates the behavior of musicians for adjusting the tune of each instrument with their own memory and reaches a state of harmony eventually [50].
6. Bat Algorithm (BA): Idealizes the characteristics of echolocation process of bats. Each bat adjusts the distance of movement by changing the pulse rate and the volume of sound [51,52].
7. Particle Swarm Optimization (PSO): an intelligent algorithm inspired by the migration of birds. Each bird represents a potential solution to the given problem [53,54].
8. Beta Differential Evolution (BDE): A new beta probability distribution has been used for altering of F and CR parameters that are represented as beta differential evolution [55].
9. Elite Opposition-Flower Pollination Algorithm (EOFPA): Using a greedy strategy, individual with the best fitness value in the population is viewed as the elite individual Basic Flower Pollination Algorithm use Lévy flight in global search process. The probability of obtaining better solution in global search is improved and the search space is expanded [56].

The parametric settings of each algorithm are presented in Table 1.

Table 1. Parameters of the algorithms.

Algorithm	Parameters	Values
MDA	Number of dragonflies	30
	No. of iterations	500

Continued on next page

Algorithm	Parameters	Values
DA	Mutation scaling factor SF	0.5
	Crossover probability CR	0.9
	Maximum velocity	25.5
	Number of dragonflies	30
	No. of iterations	500
	Maximum velocitv	25.5
	controlling parameter c_1	$[0,2]$
SSA	Number of salps	30
	No. of iterations	500
	controlling parameter r_1	$[0,2]$
SCA	Population size	30
	No. of iterations	500
ALO	Number of antlions	30
	No. of iterations	500
HSO	Pitch Adjustment Rate	0.3
	Harmony Memory Considering Rate	0.95
	Tuning bandwidth BW	25.5
	Harmony memory size	30
	No. of iterations	500
BA	Loudness	0.25
	Pulse emission rate	0.5
	Maximum frequency	2
	Minimum frequency	0
	Factor updating loudness α	0.95
	Factor updating pulse emission rate γ	0.05
	Scaling factor	4
	Number of bats	30
	No. of iterations	500
PSO	Maximum particle velocity	25.5
	Maximum inertia weight	0.9
	Minimum inertia weight	0.4
	Learning factors c_1 and c_2	2
	Number of particles	30
	No. of iterations	500
BED	The array of scaling factor F	$[0,1]$
	The crossover rate C_R	$[0,1]$
	No. of iterations	500
	Population size	30
EOFPA	No. of iterations	500
	The flowers/pollen gametes	30
	Switch probability	$[0,1]$


Original test images
Histogram of red frame
Histogram of green frame
Histogram of blue frame
(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)
(j)

Figure 3. Original test images named 'Bridge', 'Building', 'Cactus', 'Cow', 'Deer, 'Diver, 'Elephant, 'Horse', 'Kangaroo' and 'Lake' respectively and the corresponding histograms for each of the color channels (Red, Green and Blue).

5.3. Measures of performance

The performance of each algorithm is assessed by the following measures, some of them are used to evaluate the fitness values, and the others are evaluated the quality of segmented images. For the former part, two indices namely arithmetic mean and standard deviation are used and their definitions are as follows:

Arithmetic mean (AM): indicates the center value of sample data and it is defined as:

$$AM = \frac{1}{nr}\sum_{i=1}^{nr} f_i \tag{37}$$

Standard deviation (STD): a value indicates the dispersion of sample data and it is mathematically represented as:

$$STD = \sqrt{\frac{1}{nr-1}\sum_{i=1}^{nr}\left(f_i - AM\right)^2} \tag{38}$$

where nr represents the total number of runs f_i represents the fitness value at the ith run

For the latter part, three indices namely PSNR, SSIM, and FSIM are used and their definitions are as follows:

Peak signal to noise ratio (PSNR): an index which is used to evaluate the similarity of the processed image against the original image and it is defined as:

$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right) \tag{39}$$

MSE represents the mean squared error and is calculated as:

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left[I(i,j) - K(i,j)\right]^2 \tag{40}$$

where $I(i,j)$ and $K(i,j)$ denote the gray level of the original image and the segmented image in the ith row and jth column respectively. M and N denote the number of rows and columns in the image matrix respectively

Structural similarity index (SSIM): a measure of the similarity between the original image and the segmented image, which takes various factors such as brightness, contrast, and structural similarity into account and it is defined as [57]:

$$SSIM(x,y) = \frac{\left(2\mu_x\mu_y + c_1\right)\left(2\sigma_{xy} + c_2\right)}{\left(\mu_x^2 + \mu_y^2 + c_1\right)\left(\sigma_x^2 + \sigma_y^2 + c_2\right)} \tag{41}$$

where μ_x and μ_y denote the mean intensities of the original image and the segmented image respectively. σ_x^2 and σ_y^2 are the standard deviation of the original image and the segmented image respectively. σ_{xy} denotes the covariance between the original image and the segmented image c_1 and

c_2 are constants.

Feature similarity index (FSIM): another measure of the image quality through evaluating the feature similarity between the original image and the segmented image and it is defined as [58,59]:

$$FSIM = \frac{\sum_{x\in\Omega} S_L(x) \times PC_m(x)}{\sum_{x\in\Omega} PC_m(x)} \tag{42}$$

Where, Ω represents the whole image pixel domain, $S_L(x)$ is a similarity score, $PC_m(x)$ denotes the phase consistency measure which is defined as:

$$PC_m(x) = \max\left(PC_1(x), PC_2(x)\right) \tag{43}$$

where $PC_1(x)$ and $PC_2(x)$ represent the phase consistency of two blocks respectively.

$$S_L(x) = \left[S_{PC}(x)\right]^{\alpha} \cdot \left[S_G(x)\right]^{\beta} \tag{44}$$

Where

$$S_{PC}(x) = \frac{2PC_1(x) \times PC_2(x) + T_1}{PC_1^2(x) \times PC_2^2(x) + T_1} \tag{45}$$

$$S_G(x) = \frac{2G_1(x) \times G_2(x) + T_2}{G_1^2(x) \times G_2^2(x) + T_2} \tag{46}$$

$S_{PC}(x)$ denotes the similarity measure of phase consistency, $S_G(x)$ denotes the gradient magnitude of two regions $G_1(x)$ and $G_2(x)$, α , β , T_1 an T_2 are all constants.

Moreover, a higher value of the three indices proposed above indicates a better quality of the segmented image. The values of SSIM and FSIM index are in the range.

6. Results and discussion

This section presents the experimental results of MDA based method and the analysis in terms of accuracy and stability. In order to verity the superiority of proposed MDA method over the other bio-inspired methods, a statistical analysis is also conducted.

6.1. Measures of performance

6.1.1. Hybrid algorithm of DA and DE

The segmented images obtained by all the algorithms using Otsu method, Kapur's entropy, and MCE are presented in Figure 4, Figure 5, and Figure 6 respectively. From the segmented results it is found that visual quality improves as the number of threshold levels is increased and the MDA based method gives higher segmentation performance for most of the considered test images.

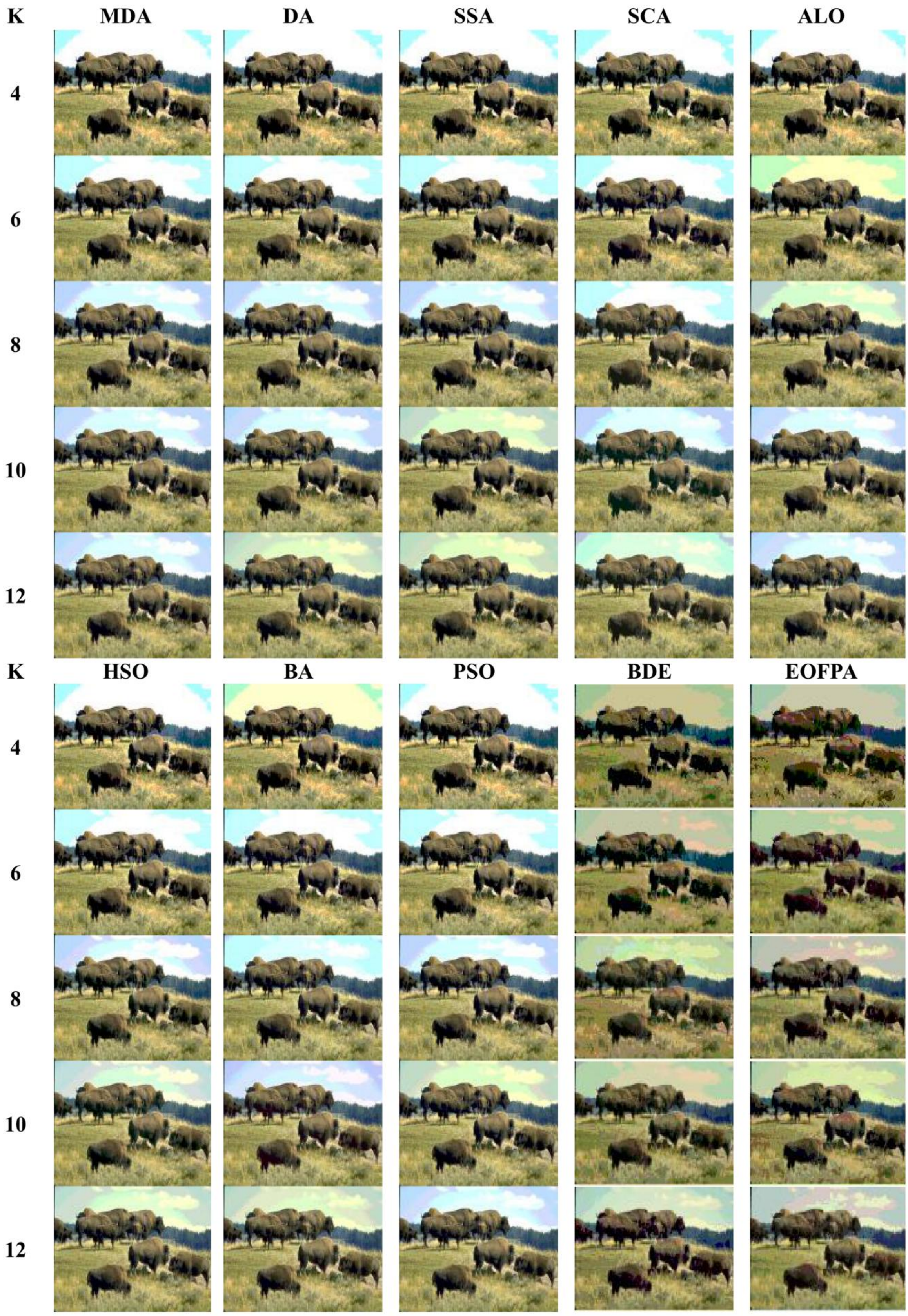

Figure 4. The segmented images for different algorithms using Otsu's method at five levels (Cow).

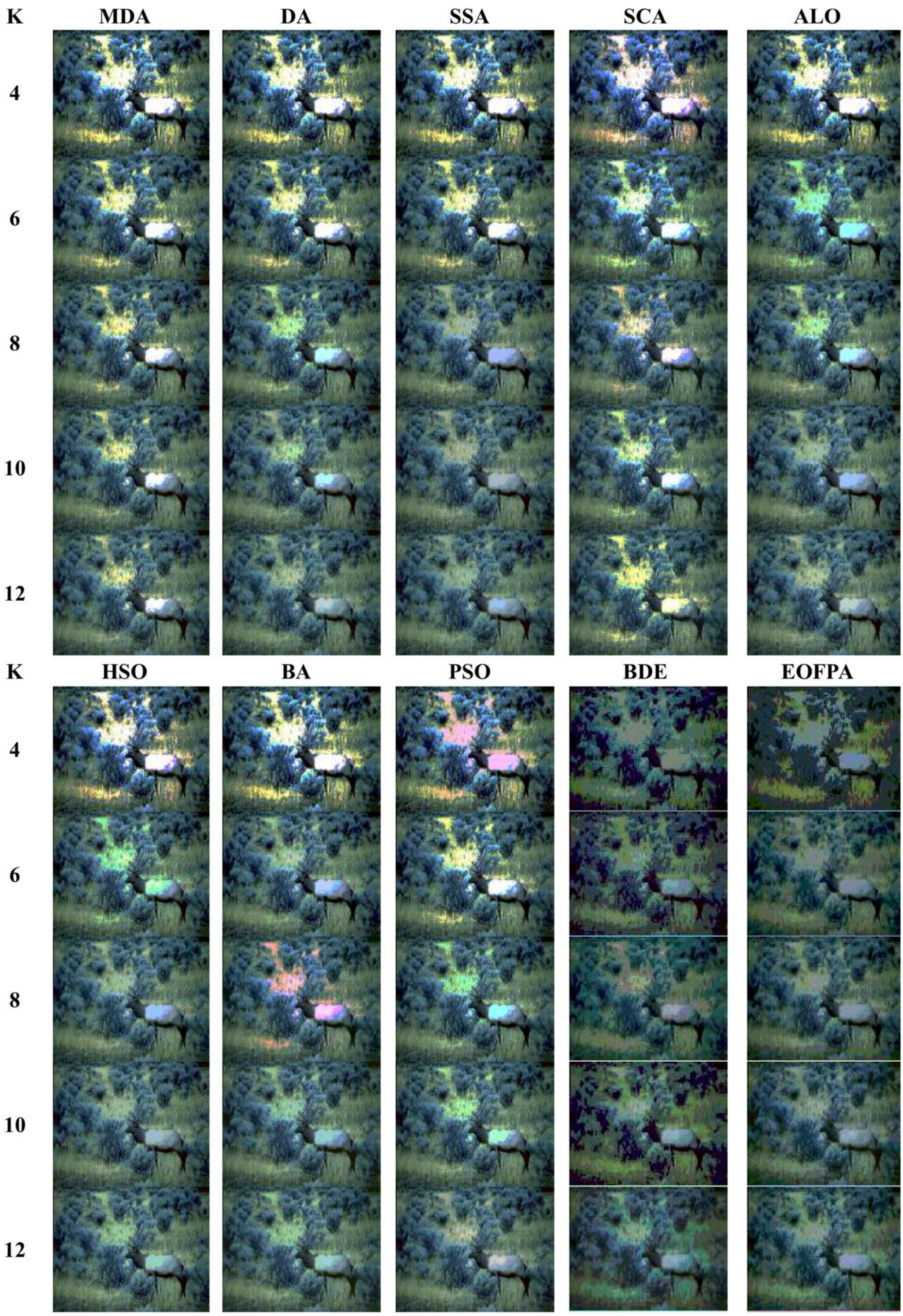

Figure 5. The segmented images for different algorithms using Kapur's entropy at five levels (Deer).

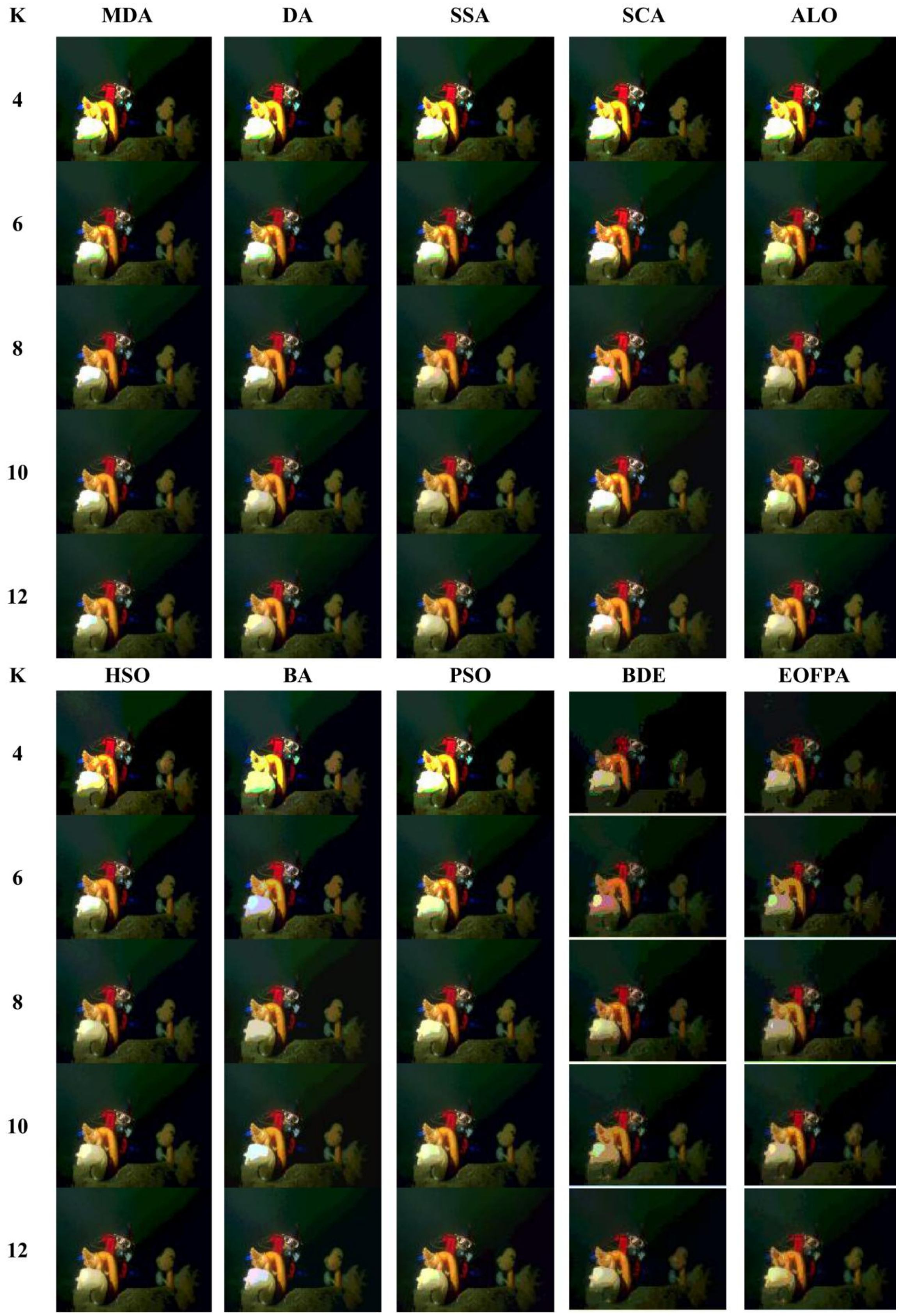

Figure 6. The segmented images for different algorithms using MCE method at five levels (Driver).

There is no guarantee that all methods converge to the same solution each time due to the stochastic nature of meta-heuristic algorithms. Therefore, two indices namely AM and STD are evaluated to analyze the accuracy and stability of all methods. A lower value of STD shows higher stability of the method. Whereas, higher values of AM indicate higher accuracy of the thresholding results. Each algorithm (using Otsu method, Kapur's entropy, and MCE) is run over 100 times to study the effectiveness of MDA. The AM values of fitness functions are given in Tables 2–4. It can be easily seen from these tables that MDA based method obtains higher AM value of fitness function against other methods. Therefore, the proposed technique performs better in terms of segmentation accuracy. The STD values of fitness functions are given in Tables 5–7. It can be observed from these tables that MDA based method obtains lower STD value of the fitness function as compared to other methods in general. This proves that the proposed MDA based method offers better stability and consistency, which is suitable for multilevel segmentation of images. For visual analysis, the convergence curves for fitness function using Otsu method, Kapur's entropy, and MCE are shown in Figures 7–9 (for Levels = 12) for ten test images. It can be seen that MDA based method performs better convergence property than other methods.

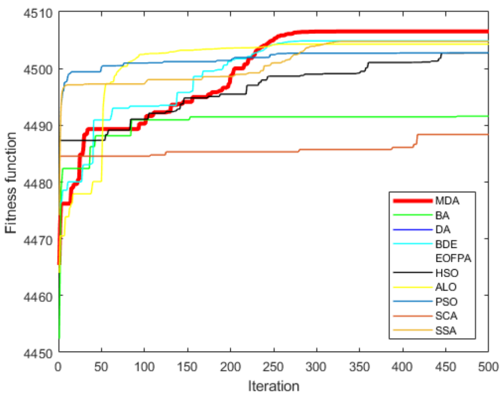

Figure 7. The convergence curves for fitness function using Otsu’s method at 12 levels (Bridge).

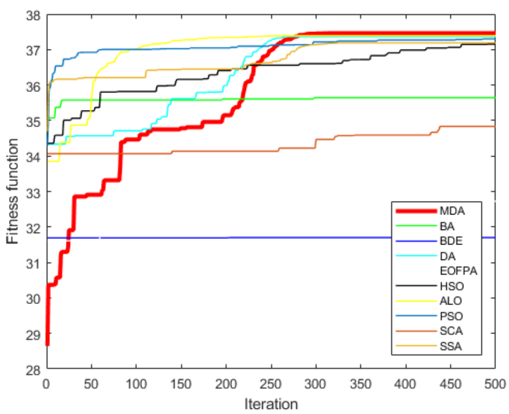

Figure 8. The convergence curves for fitness function using Kapur's entropyat 12 levels (Building).

Table 2. The AM values of fitness functions using Otsu's method.

Images	Level	MDA	DA	SSA	SCA	ALO	HSO	BA	PSO	BDE	EOFPA
	4	**4386.0155**	**4386.0155**	**4386.0155**	4383.2562	**4386.0155**	4385.6741	4385.6859	**4386.0155**	4275.6231	4305.6849
	6	**4456.6941**	4456.6745	4456.6812	4444.5333	4456.6812	4454.8023	4455.0995	4456.6896	4389.0323	4433.0295
Bridge	8	**4484.1180**	4483.9262	4481.7449	4459.9789	4484.1093	4481.4416	4466.3154	4484.1162	4382.9916	4433.2254
	10	**4498.2080**	4497.9168	4497.7713	4482.3727	4498.2009	4494.7658	4471.7015	4497.7081	4459.3658	4433.7415
	12	**4506.5451**	4504.8726	4504.8661	4488.6786	4506.5326	4501.1585	4475.5967	4502.7801	4341.1585	4332.5967
	4	**3666.7889**	**3666.7889**	**3666.7889**	3661.2005	**3666.7889**	3666.4479	3666.5619	**3666.7889**	3535.4479	3598.5619
	6	**3736.1408**	3736.1359	**3736.1408**	3701.9165	3736.1289	3734.3517	3715.7878	3736.1157	3689.3517	3687.7878
Building	8	3765.4064	3763.7146	3761.8726	3732.2340	3765.3964	**3776.4662**	3754.6370	3765.2956	3722.6370	3733.2956
	10	**3780.2478**	3780.0474	3775.2198	3761.4603	3780.2365	3780.1030	3764.3500	3779.2472	3759.7500	3733.9872
	12	**3788.7772**	3785.0051	3784.1982	3764.2379	3785.6140	3784.0790	3767.8448	3787.9557	3698.8448	3780.9557
	4	**2126.2412**	**2126.2412**	**2126.2412**	2122.4910	**2126.2412**	2124.6012	2125.8760	**2126.2412**	2120.6012	2122.8650
	6	**2188.8128**	2188.7833	2188.8074	2170.4086	2188.7851	2187.7032	2173.0367	2188.7902	2174.7042	2169.0577
Cactus	8	**2213.2398**	2211.6703	2209.8554	2185.0069	2213.1713	2210.4804	2200.8495	2212.9784	2199.4804	2198.8485
	10	**2225.0856**	2223.9729	2224.1614	2206.6749	2224.7509	2222.4256	2212.5880	2224.8807	2216.4286	2216.5640
	12	**2231.9191**	2227.8378	2230.5400	2214.0593	2230.9317	2227.4470	2213.0484	2230.2080	2217.4470	2218.0484
	4	**3953.7954**	**3953.7954**	**3953.7954**	3947.6284	3953.7937	3953.6631	3953.3749	**3953.7954**	3947.6651	3949.3589
	6	**4018.8130**	4018.5006	4018.8067	3997.9203	4018.7855	4016.5598	4006.7333	4018.8091	4004.5578	4010.7833
Cow	8	**4048.9214**	4047.7684	4048.9027	4025.8279	4048.9098	4046.5501	4030.1545	4048.7097	4030.5471	4047.1755
	10	**4063.2203**	4062.2329	4062.3868	4046.0778	4060.1107	4058.5699	4051.8095	4062.5712	4048.5669	4049.8045
	12	**4071.2438**	4070.4191	4069.9258	4050.7113	4071.1365	4063.9022	4053.6927	4070.1930	4069.9062	4058.6927
	4	**1122.5798**	**1122.5798**	**1122.5798**	1121.1187	**1122.5798**	1122.2487	1122.5083	1110.4477	1117.2487	1120.9083
	6	**1161.5839**	1161.5775	1161.5775	1142.9409	1161.5705	1154.9956	1147.5510	1152.1515	1097.9556	1148.5540
Deer	8	**1178.1038**	1178.0725	1178.1023	1157.6053	1175.7847	1169.9586	1173.4041	1175.1126	1158.966	1168.4061
	10	**1186.5170**	1186.2849	1180.0212	1166.4972	1185.4142	1178.7451	1172.1178	1183.8460	1168.7851	1169.1168
	12	**1191.1450**	1189.9832	1187.3761	1177.2843	1187.0185	1183.5727	1172.5974	1187.5187	1178.547	1177.5944
	4	**1522.2967**	**1522.2967**	**1522.2967**	1520.5664	**1522.2967**	1519.8700	1521.7928	1522.2640	1507.8700	1519.7928
	6	**1551.5394**	1551.4815	**1551.5394**	1548.3600	1551.5337	1549.0623	1549.0902	1551.4279	1539.0863	1538.0702
Diver	8	**1562.9129**	1562.7854	1561.9790	1556.4619	1561.9507	1559.9760	1558.8788	1562.4085	1559.9760	1558.8788
	10	**1569.5513**	1568.2960	1568.0981	1560.6939	1569.4202	1565.9504	1557.2689	1564.9109	1548.9904	1553.2009
	12	**1572.8927**	1572.8457	1570.6416	1565.8103	1571.7146	1570.6842	1561.5625	1570.6104	1568.8042	1551.5355
	4	**1922.2912**	**1922.2912**	**1922.2912**	1918.9804	**1922.2912**	1921.8603	1922.0942	1922.2682	1919.8643	1920.0542
	6	**1965.3704**	**1965.3704**	1965.2581	1938.3278	1965.3693	1964.1737	1962.4793	1965.1725	1963.1757	1952.4453
Elephant	8	**1984.7096**	1982.4376	1981.5554	1964.4215	1984.6837	1979.0474	1977.1272	1979.8723	1975.0444	1976.1242
	10	**1994.7387**	1993.2369	1989.9457	1978.0828	1992.1249	1987.8605	1984.2041	1987.1609	1985.8365	1983.2043
	12	**2000.4189**	1998.6301	1996.5621	1986.4930	1997.8161	1994.4913	1978.0329	1994.6057	1992.4333	1979.0459
	4	**2310.6909**	**2310.6909**	**2310.6909**	2305.7078	**2310.6909**	2309.9326	2310.3382	2310.6598	2299.9456	2307.3332
	6	**2378.2916**	2378.2825	**2378.2916**	2370.1264	2378.2680	2375.9490	2369.4067	2370.9811	2369.9230	2370.4567
Horse	8	**2406.3611**	2406.3126	2406.3320	2383.0984	2406.2989	2401.2306	2382.1623	2405.4886	2399.2546	2375.1633
	10	**2420.4694**	2418.4885	2418.1694	2399.4467	2416.1505	2411.9233	2399.8047	2418.6889	2401.9343	2397.8467
	12	**2428.4816**	2427.6154	2427.5850	2408.8655	2426.3828	2422.8833	2403.5197	2423.5853	2400.8223	2403.5332
	4	**1114.7964**	**1114.7964**	**1114.7964**	1110.2121	1114.7903	1114.3295	1114.6277	1114.7889	1107.3295	1105.4477
	6	**1164.7521**	1164.7327	1164.7463	1146.3046	1164.7463	1163.3312	1154.3339	1158.7069	1154.3212	1149.3459
Kangaroo	8	**1187.4052**	1187.0806	1186.6851	1165.7543	1187.3525	1183.0318	1167.0394	1186.8491	1179.0338	1170.0344
	10	**1199.0953**	1196.3070	1196.2935	1183.8805	1198.8494	1194.1221	1180.9305	1195.4164	1189.1561	1179.9455
	12	**1205.6955**	1200.5395	1200.9629	1189.7715	1204.7195	1201.8035	1186.1872	1201.2637	1200.8465	1176.1342
	4	**3602.5126**	**3602.5126**	**3602.5126**	3595.0459	**3602.5126**	3602.0153	3602.1261	3602.4966	3599.0442	3600.1445
	6	**3676.1680**	3675.8735	3676.1650	3654.2173	3668.5190	3675.3194	3671.9734	3676.0909	3659.3174	3669.9564
Lake	8	**3705.3747**	3699.8939	3705.3197	3667.8620	3705.3361	3701.3738	3688.8008	3697.8809	3686.3458	3677.8358
	10	**3719.6677**	3719.4646	3719.6472	3691.5485	3719.5805	3712.8384	3706.5417	3716.1525	3710.8335	3705.5865
	12	**3727.6944**	3726.4808	3725.0242	3698.3776	3724.0969	3721.2794	3710.6391	3724.5939	3690.7961	3684.5679

Table 3. The AM values of fitness functions using Kapur's entropy.

Images	Levels	MDA	DA	SSA	SCA	ALO	HSO	BA	PSO	BDE	EOFPA
Bridge	4	**18.1907**	**18.1907**	18.1906	18.1526	18.1445	18.1873	18.1827	**18.1907**	18.1663	18.1597
	6	**23.5682**	23.5675	23.5679	23.3085	23.5678	23.5383	23.5023	23.5680	23.5597	23.5483
	8	**28.3015**	28.2786	28.2814	27.6408	28.3002	28.2265	27.6685	28.2965	28.2197	27.6589
	10	**32.6202**	32.6118	32.6137	30.9182	32.6175	32.5149	30.9871	32.5589	32.5078	30.9861
	12	**36.5456**	36.3356	36.4038	34.5257	35.8899	36.1310	34.6846	36.3876	36.1290	34.6754
Building	4	**18.8295**	**18.8295**	**18.8295**	18.8158	**18.8295**	18.8226	18.8278	**18.8295**	18.8117	18.8256
	6	**24.1205**	24.1102	24.1201	23.8745	24.0527	24.0527	23.9953	24.0966	24.0156	23.9643
	8	**28.9969**	28.9867	28.9965	27.8900	28.9943	28.9057	28.3981	28.9857	28.8521	28.2457
	10	**33.4091**	33.3553	33.3952	31.4147	31.4153	33.0362	31.9753	33.3670	32.9653	31.8533
	12	**37.4610**	37.3651	37.1874	34.8359	37.3891	37.1524	35.6430	37.3001	37.0854	35.3576
Cactus	4	**18.5843**	**18.5843**	**18.5843**	18.5632	**18.5843**	18.5761	18.5818	**18.5843**	18.4797	18.5748
	6	**23.8420**	23.8418	**23.8420**	23.4790	23.8417	23.7550	23.8085	23.8412	23.6854	23.7422
	8	**28.5198**	28.5051	28.4490	27.8225	28.5094	28.3850	27.7631	28.4991	28.3742	27.6854
	10	**32.8542**	32.8325	32.8457	31.3685	32.8479	32.6682	32.0858	32.7519	32.5853	32.0753
	12	**36.8707**	36.7269	36.7470	34.5881	36.7313	36.6221	34.7641	36.6960	36.5586	34.4763
Cow	4	**18.5002**	18.4994	**18.5002**	18.4624	**18.5002**	18.4902	18.4983	**18.5002**	18.4774	18.4333
	6	**23.9812**	23.9795	23.9811	23.7240	23.8746	23.9496	23.8805	23.9796	23.5974	23.8365
	8	**28.8794**	28.8320	28.8785	28.1316	28.7981	28.7163	28.4547	28.7927	28.6873	28.3766
	10	**33.3769**	33.2290	33.2647	32.0085	33.3423	33.1886	31.9696	33.2887	33.0674	31.6773
	12	**37.5478**	37.3152	37.4710	35.6447	37.4301	37.1716	34.9981	37.2740	37.1643	34.4351
Deer	4	**17.7349**	**17.7349**	**17.7349**	17.6786	**17.7349**	17.7135	17.7281	**17.7349**	17.4555	17.2441
	6	**22.8322**	22.8321	22.8319	22.5782	22.8321	22.7880	22.7745	22.8309	22.6470	22.6845
	8	**27.2958**	27.2929	27.2812	26.2300	27.2806	27.2006	26.2772	27.2733	27.1996	26.2692
	10	**31.3387**	31.3160	31.3360	29.2343	30.6775	30.9565	29.5594	31.2597	30.85645	29.5474
	12	**35.0702**	34.4227	34.3552	31.7673	34.4560	34.5034	32.6665	34.4692	34.4694	32.5765
Diver	4	18.3361	18.3358	18.3350	18.2989	**18.3364**	18.3263	18.3232	**18.3364**	18.2883	18.1972
	6	**23.6781**	23.6780	23.6775	23.4056	23.6385	23.6085	23.6370	23.6772	23.4785	23.5850
	8	**28.3676**	28.3511	28.3668	27.4894	28.3655	28.3069	27.8034	28.3593	28.2788	27.6444
	10	**32.6130**	32.5412	32.5953	31.5176	31.9590	32.4207	30.5849	32.5548	32.2977	30.6059
	12	**36.5313**	35.7728	35.9128	33.6363	35.8612	36.0536	33.4780	35.7932	36.0156	33.4620
Elephant	4	**18.1761**	18.1759	**18.1761**	18.1295	**18.1761**	18.1649	18.1713	**18.1761**	18.1086	18.1478
	6	**23.3175**	23.3173	23.3173	23.0015	23.3151	23.2798	23.2584	23.3080	23.1976	23.1658
	8	**27.9489**	27.8447	27.9453	26.9483	27.8597	27.8217	27.7337	27.9263	27.7937	26.9867
	10	**32.1636**	32.0784	32.1632	30.4741	32.1142	31.9377	30.3712	32.0997	31.7812	30.3647
	12	**36.0009**	35.7483	35.7545	32.9324	35.8113	35.5656	33.4534	35.7970	34.8626	32.8634
Horse	4	**18.6122**	**18.6122**	**18.6122**	18.5974	18.6121	18.6081	18.6089	18.6121	18.0771	18.2979
	6	**23.7909**	23.7897	23.7908	23.5302	23.7903	23.7650	23.7601	23.7906	23.6997	22.9791
	8	**28.4407**	28.4312	28.4404	27.8054	28.4385	28.3584	28.1669	28.4277	27.3234	26.1239
	10	**32.6907**	32.5229	32.6719	31.2299	32.6865	32.2931	31.9769	32.6336	32.1891	31.3566
	12	**36.5861**	36.4546	36.4579	34.1298	36.5790	35.9428	33.9804	36.4842	35.6435	33.6689
Kangaroo	4	**18.9363**	18.9362	18.9360	18.8904	18.9215	18.9352	18.9227	**18.9363**	18.7633	18.8644
	6	**24.4756**	24.4707	24.4650	24.2231	24.4742	24.4465	24.3311	24.4713	24.0535	24.1431
	8	**29.4235**	29.4222	29.4202	28.8466	29.4191	29.3775	29.2686	29.4158	29.2777	29.1576
	10	**33.8985**	33.8650	33.8894	32.4266	33.8981	33.6472	31.7512	33.8709	33.5347	31.2442
	12	**38.0564**	37.9305	37.9644	36.4710	37.9993	37.7302	36.7833	37.9278	37.6432	36.6453
Lake	4	**17.7485**	**17.7485**	**17.7485**	17.7140	17.7347	17.7431	17.7477	**17.7485**	17.34767	17.5897
	6	**22.7151**	22.7148	22.7150	22.5244	22.7149	22.6387	22.6811	22.7146	22.5467	22.3551
	8	**27.2343**	27.2271	27.2291	26.4737	27.2342	27.0464	26.8541	27.2195	26.5671	27.2305
	10	**31.3868**	31.3760	31.3488	29.4879	31.3622	31.1878	30.4835	31.3469	30.3657	31.2569
	12	**35.2977**	34.9685	35.1190	33.2548	34.6736	34.7786	33.0519	34.5806	32.0652	33.5967

Table 4. The AM values of fitness functions using MCE method.

Images	Levels	MDA	DA	SSA	SCA	ALO	HSO	BA	PSO	BDE	EOFPA
Bridge	4	**−620.2763**	**−620.2763**	**−620.2763**	−620.2542	**−620.2763**	−620.2648	−620.2578	**−620.2763**	−609.0790	−609.3784
	6	**−620.6749**	−620.6743	**−620.6749**	−620.6221	−620.6748	−620.6437	−620.6128	−620.6748	−609.6698	−609.7163
	8	**−620.8419**	−620.8385	−620.8414	−620.7237	−620.8097	−620.7891	−620.6825	−620.8393	−609.8519	−609.8320
	10	**−620.9233**	−620.9178	−620.9058	−620.8120	−620.9068	−620.8768	−620.8129	−620.9175	−609.9459	−610.0020
	12	**−620.9702**	−620.9608	−620.9477	−620.8790	−620.9668	−620.9398	−620.8730	−620.9541	−609.9813	−610.0675
Building	4	**−660.4584**	**−660.4584**	**−660.4584**	−660.4452	**−660.4584**	−660.4560	−660.4562	**−660.4584**	−658.6000	−658.7712
	6	**−660.8619**	−660.8616	**−660.8619**	−660.8236	−660.8014	−660.8492	−660.8435	**−660.8619**	−659.0710	−659.1223
	8	**−661.0323**	−661.0316	−661.0312	−660.8618	−661.0320	−660.9857	−660.7957	−661.0070	−659.3262	−659.3747
	10	**−661.1222**	−661.1219	−661.0910	−661.0239	−661.1093	−661.0670	−661.0161	−661.0894	−659.3922	−659.4202
	12	**−661.1753**	−661.1616	−661.1249	−661.0725	−661.1443	−661.1479	−661.0530	−661.1650	−659.5068	−659.5193
Cactus	4	**−375.3032**	**−375.3032**	**−375.3032**	−375.2890	**−375.3032**	−375.2894	−375.2985	**−375.3032**	−372.5073	−371.6678
	6	**−375.6282**	−375.6275	−375.6281	−375.5085	−375.5830	−375.6132	−375.5937	−375.6275	−372.9594	−373.0201
	8	**−375.7553**	−375.7483	−375.7532	−375.6164	−375.7541	−375.7250	−375.5836	−375.7529	−375.1600	−370.2542
	10	**−375.8189**	−375.8003	−375.8125	−375.7240	−375.8078	−375.7907	−375.7469	−375.8125	−372.1932	−372.3085
	12	**−375.8552**	−375.8445	−375.8338	−375.7616	−375.8434	−375.8263	−375.7679	−375.8484	−370.3258	−374.3484
Cow	4	**−700.7303**	**−700.7303**	**−700.7303**	**−700.7303**	−700.7134	**−700.7303**	−700.7263	−700.6034	−698.8509	−698.9133
	6	**−701.0632**	**−701.0632**	**−701.0632**	−700.9918	−701.0281	−701.0334	−701.0425	−701.0631	−699.1213	−699.1382
	8	**−701.1934**	−701.1928	−701.1932	−701.0718	−701.1763	−701.1654	−701.1646	−701.1924	−699.3032	−699.2943
	10	**−701.2660**	−701.2620	−701.2453	−701.1829	−701.2657	−701.2356	−701.1861	−701.2536	−699.3819	−699.4248
	12	**−701.3083**	−701.2998	−701.2872	−701.2130	−701.2949	−701.2783	−701.2162	−701.2998	−699.4799	−699.4939
Deer	4	**−394.8162**	**−394.8162**	**−394.8162**	−394.7204	**−394.8162**	−394.8098	−394.8158	−394.7272	−377.5073	−377.6678
	6	**−395.0601**	**−395.0601**	**−395.0601**	−394.9579	−395.0290	−394.9943	−394.9624	−395.0593	−377.9594	−378.0201
	8	**−395.1690**	−395.1521	−395.1171	−395.1094	−395.1303	−395.1088	−395.0924	−395.1305	−378.1600	−378.2542
	10	**−395.2250**	−395.2132	−395.1898	−395.1037	−395.1815	−395.1494	−395.1254	−395.1797	−378.1932	−378.3085
	12	**−395.2572**	−395.2394	−395.2311	−395.1551	−395.2271	−395.2057	−395.1679	−395.2235	−378.3258	−378.3484
Diver	4	**−130.2537**	**−130.2537**	**−130.2537**	−130.2490	**−130.2537**	−130.1758	−130.1763	**−130.2537**	−129.9030	−129.1653
	6	**−130.4961**	−130.4955	−130.4816	−130.4609	−130.4587	−130.4076	−130.4042	−130.4947	−128.4047	−126.3978
	8	**−130.6021**	−130.6007	−130.5604	−130.5324	−130.5841	−130.5097	−130.4255	−130.5598	−127.5298	−129.6061
	10	**−130.6536**	−130.6447	−130.6170	−130.5825	−130.6424	−130.5848	−130.5879	−130.6252	−128.6593	−129.7220
	12	**−130.6882**	−130.6721	−130.6341	−130.5951	−130.6744	−130.6113	−130.5267	−130.6643	−129.7131	−129.7822
Elephant	4	**−456.8572**	**−456.8572**	**−456.8572**	−456.8455	−456.8552	−456.8524	−456.8540	−456.8552	−4552665	−455.2189
	6	**−457.1171**	**−457.1171**	**−457.1171**	−457.0303	−457.0916	−457.0805	−457.0660	−457.1167	−455.4441	−455.7034
	8	**−457.2138**	−457.2028	−457.2017	−457.1199	−457.1564	−457.1792	−457.1036	−457.1707	−455.7360	−455.7811
	10	**−457.2615**	−457.2584	−457.2571	−457.1509	−457.2461	−457.2384	−457.2160	−457.2471	−455.8112	−455.9329
	12	**−457.2974**	−457.2750	−457.2645	−457.1782	−457.2761	−457.2625	−457.2034	−457.2755	−455.8923	−4559428
Horse	4	**−650.9465**	**−650.9465**	**−650.9465**	−650.9346	−650.9464	−650.9453	−650.8275	−650.9464	−638.8670	−639.0225
	6	**−651.2310**	−651.2305	**−651.2310**	−651.1613	−651.2309	−651.2223	−651.2226	−651.2308	−639.0824	−639.1809
	8	**−651.3496**	−651.3491	−651.3492	−651.2074	−651.3354	−651.3218	−651.3145	−651.3486	−639.2078	−639.2707
	10	**−651.4103**	−651.4075	−651.3959	−651.3230	−651.4090	−651.3884	−651.3445	−651.3985	−639.3381	−639.3124
	12	**−651.4451**	−651.4384	−651.4396	−651.3675	−651.4213	−651.4220	−651.3387	−651.4357	−639.3136	−639.3694
Kangaroo	4	**−441.3100**	**−441.3100**	**−441.3100**	−441.2870	−441.3099	−441.3033	−441.3051	**−441.3100**	−440.3940	−440.6180
	6	**−441.6207**	−441.6151	**−441.6207**	−441.5240	−441.6198	−441.5944	−441.5567	−441.6204	−440.8093	−440.8583
	8	**−441.7491**	−441.7469	−441.7401	−441.6495	−441.7466	−441.7079	−441.6409	−441.7284	−440.9658	−440.0692
	10	**−441.8182**	−441.8136	−441.7947	−441.7388	−441.7836	−441.7767	−441.6903	−441.7961	−440.1546	−440.1106
	12	**−441.8579**	−441.8372	−441.8412	−441.7623	−441.8426	−441.8145	−441.7252	−441.8418	−440.1620	−440.2211
Lake	4	**−812.1139**	**−812.1139**	**−812.1139**	−812.0987	**−812.1139**	−812.1079	−812.0152	**−812.1139**	−811.9079	−810.5752
	6	**−812.3819**	−812.3818	**−812.3819**	−812.2984	−812.3818	−812.3736	−812.3574	−812.3818	−811.7736	−810.3594
	8	**−812.4913**	−812.4849	**−812.4913**	−812.3919	−812.4845	−812.4776	−812.4708	−812.4777	−810.6776	−810.2308
	10	**−812.5476**	−812.5416	−812.5465	−812.4565	−812.5374	−812.5347	−812.4901	−812.5360	−810.8347	−811.4691
	12	**−812.5797**	−812.5663	−812.5738	−812.4903	−812.5752	−812.5588	−812.5076	−812.5747	−811.9588	−810.4976

Table 5. The STD values of fitness functions using Otsu's method.

Images	Levels	MDA	DA	SSA	SCA	ALO	HSO	BA	PSO	BDE	EOFPA
Bridge	4	**0.0000e + 00**	**0.0000e + 00**	**0.0000e + 00**	1.5052e + 00	6.9599e - 03	2.9240e - 01	1.0625e + 01	1.9536e - 02	1.2525e - 02	1.5216e - 02
	6	**4.1541e - 03**	1.0091e - 01	5.6543e - 03	1.3679e + 01	4.6615e + 00	1.7517e + 00	1.4785e + 01	4.5974e + 00	1.3759e + 01	4.4865e + 00
	8	**2.2692e - 03**	7.3258e - 01	1.1033e + 00	3.3881e + 00	3.8795e + 00	9.8235e - 01	3.1223e + 00	1.5591e - 01	3.3871e + 00	1.9732e - 01
	10	**5.3445e - 02**	1.8133e + 00	8.9802e - 01	4.4712e + 00	2.2772e + 00	7.2268e - 01	8.1170e + 00	1.4814e + 00	9.5750e + 00	3.4353e + 00
	12	**2.2612e - 01**	8.7240e - 01	1.1039e + 00	2.5169e + 00	1.6665e + 00	1.2467e + 00	1.8633e + 00	1.1796e + 00	2.9986e + 00	2.8652e + 00
Building	4	**0.0000e + 00**	5.2206e - 03	**0.0000e + 00**	1.0556e + 00	**0.0000e + 00**	2.0873e - 01	1.9321e - 01	4.9614e - 03	3.2768e - 01	5.9787e - 03
	6	2.7075e - 03	4.1034e - 01	**1.4030e - 03**	5.8629e + 00	3.7949e + 00	5.4196e - 01	4.5020e + 00	4.8013e + 00	5.9732e + 00	5.8364e + 00
	8	**1.5158e - 02**	7.8094e - 01	3.6351e + 00	4.4713e + 00	1.8527e + 00	1.0868e + 00	7.2455e + 00	1.8760e + 00	9.8648e + 00	7.8648e + 00
	10	**2.5611e - 02**	1.5275e + 00	2.2609e + 00	1.8080e + 00	1.8371e + 00	1.1496e + 00	2.8591e + 00	2.5056e + 00	6.8734e + 00	6.8642e + 00
	12	**2.6689e - 01**	6.5488e - 01	1.0901e + 00	1.5993e + 00	6.8305e - 01	1.0169e + 00	3.8292e + 00	9.4510e - 01	5.8743e + 00	9.9608e - 01
Cactus	4	**0.0000e + 00**	8.8624e - 03	**0.0000e + 00**	9.7805e - 01	1.9783e - 03	2.2157e - 01	9.0407e - 02	1.0212e + 01	8.8738e - 02	7.9492e + 01
	6	**0.0000e + 00**	1.4716e - 01	1.0193e - 02	6.3001e + 00	2.8521e + 00	1.2033e + 00	5.9922e + 00	3.6189e + 00	6.8438e + 00	7.9332e + 00
	8	**2.7104e - 03**	9.2525e - 01	4.1563e - 01	5.5874e + 00	1.8328e + 00	1.4973e + 00	5.7798e + 00	1.6235e + 00	6.6432e + 00	4.4932e + 00
	10	**7.5561e - 03**	7.4369e - 01	2.9334e - 01	3.4511e + 00	1.7996e + 00	8.5357e - 01	5.9010e + 00	1.3657e + 00	6.9754e + 00	5.9733e + 00
	12	**1.2812e - 01**	1.4887e + 00	4.1829e - 01	1.7774e + 00	8.5019e - 01	7.4279e - 01	1.6913e + 00	6.4976e - 01	5.4883e + 00	7.4934e - 01
Cow	4	**5.0842e - 13**	9.4847e - 06	6.1882e - 07	1.3402e + 00	5.1671e - 03	3.2173e - 01	1.4038e - 01	9.3752e - 03	3.9754e - 01	9.7353e - 03
	6	1.6545e - 01	1.9084e - 01	1.6323e - 01	5.9744e + 00	2.5197e + 00	1.3783e + 00	2.3592e + 00	**1.6166e - 01**	5.4873e + 00	2.9753e - 01
	8	**2.4108e - 02**	7.2264e - 01	9.2347e - 01	4.8122e + 00	2.4409e - 02	1.5244e + 00	6.3387e + 00	8.8607e - 01	7.8478e + 00	9.7743e - 01
	10	**2.5064e - 01**	3.3253e - 01	9.2932e - 01	3.1243e + 00	6.9114e - 01	7.6097e - 01	6.7771e + 00	1.2230e + 00	7.6436e + 00	8.9549e + 00
	12	**1.0103e - 01**	1.3010e + 00	1.0714e - 01	2.1099e + 00	4.9416e - 01	1.0799e + 00	5.4692e + 00	1.0640e + 00	6.7484e + 00	5.8622e + 00
Deer	4	**0.0000e + 00**	**0.0000e + 00**	5.4207e + 00	5.4024e + 00	7.4593e - 03	4.5577e - 01	6.2558e - 02	5.3883e + 00	7.6532e - 02	8.7353e + 00
	6	**0.0000e + 00**	2.3296e + 00	2.5645e + 00	3.9302e + 00	2.2381e + 00	2.2796e + 00	6.8111e + 00	4.7647e + 00	7.3752e + 00	6.8352e + 00
	8	**2.4978e - 02**	3.0141e + 00	1.5768e + 00	2.9499e + 00	3.4235e + 00	3.2422e + 00	3.7797e + 00	2.0554e + 00	6.7534e + 00	5.7534e + 00
	10	**4.0174e - 01**	1.5329e + 00	1.5146e + 00	2.1600e + 00	1.9536e + 00	1.6882e + 00	8.6796e + 00	3.0491e + 00	7.9333e + 00	6.8457e + 00
	12	**4.1945e - 01**	1.3304e + 00	1.0805e + 00	3.8898e + 00	9.9896e - 01	7.9462e - 01	5.3120e + 00	1.7347e + 00	6.4378e + 00	4.8363e + 00
Diver	4	1.6462e - 03	1.3852e - 02	**0.0000e + 00**	4.2953e - 01	1.0325e - 02	4.3669e - 01	1.8932e - 01	2.0237e + 00	4.7522e - 01	5.4656e + 00
	6	**0.0000e + 00**	4.5469e - 02	1.3710e - 03	2.1993e + 00	7.7053e - 01	1.6244e + 00	4.9161e + 00	3.5567e - 02	7.7453e + 00	6.8634e - 02
	8	**3.0942e - 02**	9.2514e - 01	4.5038e - 01	1.8318e + 00	4.4823e - 01	3.5167e - 01	2.0986e + 00	1.0432e + 00	3.7543e + 00	4.8652e + 00
	10	**8.3825e - 03**	7.0084e - 01	5.9167e - 01	1.9117e + 00	2.1269e - 01	9.7038e - 01	1.2839e + 00	7.0318e - 01	3.8658e + 00	8.7454e - 01
	12	**1.3694e - 01**	5.1430e - 01	4.0234e - 01	2.1263e + 00	2.6468e - 01	1.4983e - 01	1.9739e + 00	5.5533e - 01	5.4867e + 00	6.7543e - 01
Elephant	4	**0.0000e + 00**	6.4237e - 03	**0.0000e + 00**	6.0503e + 00	9.0498e - 04	1.6488e - 01	1.8436e - 01	7.2737e - 03	3.7523e - 01	8.7647e - 03
	6	**4.4492e - 02**	1.9454e + 00	2.3242e + 00	6.2803e + 00	1.7561e + 00	1.3165e + 00	3.0963e + 00	4.7712e + 00	5.8437e + 00	6.8783e + 00
	8	**5.5357e - 01**	1.2826e + 00	1.5730e + 00	3.1859e + 00	1.5738e + 00	7.7418e - 01	4.9522e + 00	7.3335e - 01	6.8773e + 00	8.6534e - 01
	10	**1.2384e + 00**	2.1375e + 00	1.2931e + 00	5.7291e + 00	6.7686e + 01	1.6294e + 00	1.0518e + 01	2.0065e + 00	3.8856e + 01	4.7436e + 00
	12	**4.3928e - 01**	5.2442e - 01	1.0687e + 00	2.3214e + 00	1.7015e + 00	1.1204e + 00	2.8496e + 00	1.7069e + 00	4.8753e + 00	5.8475e + 00
Horse	4	**0.0000e + 00**	**0.0000e + 00**	**0.0000e + 00**	2.9545e + 00	1.0649e - 03	2.2981e - 01	1.4238e + 01	1.8426e - 02	4.7583e + 01	3.8964e - 02
	6	**7.9497e - 04**	4.3199e - 02	4.3438e - 03	7.1120e + 00	1.2905e - 02	8.5581e - 01	6.9369e - 01	5.0399e + 00	7.7643e - 01	6.8353e + 00
	8	**4.5720e - 02**	2.3215e + 00	1.7100e + 00	6.0056e + 00	2.2284e + 00	6.7892e - 01	4.5581e + 00	2.1843e - 01	6.7654e + 00	5.8753e - 01
	10	**1.7780e - 02**	1.0456e + 00	1.0097e + 00	4.8101e + 00	2.8650e + 00	1.0801e + 00	7.7065e + 00	1.5548e + 00	8.8775e + 00	5.8753e + 00
	12	**3.6262e - 02**	9.1042e - 01	5.2105e - 01	2.7274e + 00	9.6076e - 01	1.3683e + 00	3.1956e + 00	1.2537e + 00	4.7535e + 00	4.7653e + 00
Kangaroo	4	**0.0000e + 00**	2.8080e - 03	**0.0000e + 00**	1.4197e + 00	**0.0000e + 00**	4.5233e - 01	1.5513e - 01	1.1710e - 02	2.8684e - 01	4.7833e - 02
	6	**7.0945e - 04**	3.3564e - 02	3.1934e - 03	8.2077e + 00	1.2418e - 02	1.0105e + 00	4.8869e + 00	3.0115e + 00	5.7536e + 00	4.8753e + 00
	8	**7.9087e - 03**	5.6464e - 02	3.6501e - 01	6.1105e + 00	1.2628e + 00	1.2967e + 00	5.9183e + 00	2.3696e + 00	6.8733e + 00	5.8757e + 00
	10	**2.5610e - 03**	4.6108e - 01	1.1588e + 00	4.0363e + 00	5.7158e - 01	8.9576e - 01	6.7443e + 00	1.3835e + 00	7.9864e + 00	4.8875e + 00
	12	**4.0049e - 02**	8.7691e - 01	1.1766e + 00	2.5601e + 00	3.8831e - 01	6.2671e - 01	3.8655e + 00	1.3033e + 00	5.8754e + 00	4.8684e + 00
Lake	4	**5.0842e - 13**	8.5440e - 07	4.4517e - 03	1.0043e + 01	1.4746e - 04	1.5678e - 01	2.4311e + 01	1.2714e - 02	4.9854e + 01	3.9685e - 02
	6	**1.3355e - 03**	1.0129e - 01	3.4139e + 00	6.2361e + 00	1.6624e - 02	1.4420e + 00	1.4072e + 01	3.6018e + 00	3.7987e + 01	4.8644e + 00
	8	**1.8655e - 01**	1.5349e + 00	1.5352e + 00	2.7270e + 00	1.5140e + 00	1.8072e + 00	7.3736e + 00	3.7370e + 00	9.3875e + 00	4.8743e + 00
	10	**5.2140e - 03**	1.3903e + 00	1.4982e + 00	3.8401e + 00	1.6502e + 00	1.7344e + 00	1.1550e + 01	2.2614e + 00	5.9486e + 01	3.7844e + 00
	12	**1.3347e - 01**	8.2871e - 01	6.0813e - 01	9.4868e - 01	1.0480e + 00	1.1963e + 00	7.1376e + 00	1.8657e + 00	8.7543e + 00	5.9883e + 00

Table 6. The STD values of fitness functions using Kapur's entropy.

Images	Levels	MDA	DA	SSA	SCA	ALO	HSO	BA	PSO	BDE	EOFPA
Bridge	4	**0.0000e + 00**	1.1889e - 03	4.5792e - 05	1.2654e - 02	2.5227e - 02	2.6238e - 03	2.3338e - 02	4.5792e - 05	3.8752e - 03	4.8758e - 02
	6	**1.0803e - 04**	1.1619e - 04	1.2647e - 04	7.0324e - 02	3.1088e - 04	1.7798e - 02	4.0157e - 02	1.3631e - 04	4.8775e - 02	6.8753e - 02
	8	**1.9295e - 04**	8.4380e - 03	1.3024e - 02	2.3574e - 01	2.5639e - 02	4.3985e - 02	1.3272e - 01	1.4655e - 02	5.8753e - 02	4.7533e - 01
	10	**6.3220e - 03**	5.5282e - 02	5.7422e - 02	2.0425e - 01	1.9574e - 02	4.6981e - 02	5.8561e - 01	3.1833e - 02	6.8735e - 02	7.8743e - 01
	12	**1.6956e - 02**	9.7999e - 02	4.5541e - 02	3.0201e - 01	3.6739e - 01	1.2307e - 01	6.4889e - 01	2.4119e - 02	3.9864e - 01	7.8743e - 01
Building	4	**0.0000e + 00**	**0.0000e + 00**	**0.0000e + 00**	7.0614e - 03	3.2018e - 05	1.2271e - 03	9.2528e - 04	**0.0000e + 00**	4.9863e - 03	9.8743e - 04
	6	**8.8020e - 03**	1.5824e - 02	4.2359e - 02	6.1177e - 02	9.7401e - 03	3.0400e - 02	1.7481e - 01	1.1671e - 02	4.8733e - 02	4.3734e - 01
	8	**2.8468e - 03**	3.2699e - 02	3.0477e - 03	1.7964e - 01	4.6111e - 02	3.0846e - 02	2.1943e - 01	3.3616e - 02	4.8753e - 02	5.9843e - 01
	10	**2.0785e - 02**	2.3636e - 02	2.1609e - 02	2.5603e - 01	8.2992e - 02	3.5160e - 02	5.6484e - 01	2.2881e - 02	4.8753e - 02	6.9886e - 01
	12	**5.2186e - 02**	9.7696e - 02	1.5118e - 01	5.0506e - 01	9.8895e - 02	1.6661e - 01	6.3697e - 01	1.7954e - 01	5.9864e - 01	7.4875e - 01
Cactus	4	**2.6599e - 05**	**2.5477e - 03**	**8.5579e - 05**	4.3927e - 03	5.8348e - 05	3.1864e - 03	2.1741e - 03	2.6752e - 05	5.3634e - 03	3.8767e - 03
	6	**1.4541e - 04**	9.5823e - 04	1.6053e - 04	7.4453e - 02	2.9505e - 04	1.9699e - 02	5.3930e - 02	4.3274e - 04	4.9863e - 02	6.8244e - 02
	8	**9.5675e - 04**	8.8883e - 03	2.7352e - 02	1.3299e - 01	3.2443e - 02	3.3815e - 02	1.8186e - 01	7.3041e - 03	5.8743e - 02	4.7863e - 01
	10	**4.3610e - 03**	3.9010e - 02	5.3608e - 03	2.6713e - 01	4.9142e - 02	3.2420e - 02	1.5062e - 01	3.3415e - 02	5.8674e - 02	4.9836e - 01
	12	**2.7111e - 03**	7.4828e - 02	3.2452e - 02	2.3010e - 01	1.8277e - 02	5.0186e - 02	6.4139e - 01	7.3479e - 03	6.8765e - 02	7.8754e - 01
Cow	4	**4.3875e - 04**	4.5348e - 04	**4.5512e - 03**	1.9697e - 02	2.1813e - 02	3.4052e - 02	7.4689e - 03	4.3942e - 04	4.6733e - 02	8.8735e - 03
	6	**2.8795e - 05**	8.4886e - 02	3.9295e - 02	5.9538e - 02	6.2887e - 02	1.5797e - 02	2.9128e - 01	4.9848e - 04	2.7535e - 02	3.8965e - 01
	8	**3.6982e - 02**	1.2156e - 01	4.4534e - 02	1.1567e - 01	7.5803e - 02	4.3787e - 02	4.7450e - 01	6.0485e - 02	5.8767e - 02	5.9886e - 01
	10	**4.3049e - 02**	1.2759e - 01	7.2200e - 02	2.1993e - 01	5.6487e - 02	4.4582e - 02	4.3961e - 01	5.2046e - 02	6.7854e - 02	5.9846e - 01
	12	**3.2045e - 02**	2.6235e - 01	7.0662e - 02	3.1597e - 01	1.1705e - 01	5.1689e - 02	5.3110e - 01	7.5080e - 02	7.8775e - 02	8.7654e - 01
Deer	4	**0.0000e + 00**	5.8974e - 03	5.9135e - 03	4.0585e - 03	7.2305e - 03	1.9331e - 02	6.9060e - 03	7.2004e - 03	6.7864e - 02	7.7435e - 03
	6	**2.2839e - 04**	1.4763e - 03	3.3548e - 04	8.1484e - 02	6.2308e - 04	1.9429e - 02	1.1446e - 02	8.4301e - 04	2.8863e - 02	3.9753e - 02
	8	**2.9669e - 03**	6.7418e - 03	3.2304e - 03	1.3820e - 01	2.3685e - 02	7.5112e - 02	3.6370e - 01	5.8357e - 03	9.8754e - 02	4.8963e - 01
	10	**4.9822e - 03**	2.8841e - 01	2.7250e - 01	6.5556e - 01	2.6518e - 01	1.9191e - 01	6.4420e - 01	2.8244e - 01	2.0793e - 01	7.8735e - 01
	12	**6.1415e - 02**	4.7209e - 01	2.0164e - 01	3.5708e - 01	4.9867e - 01	1.1798e - 01	4.9348e - 01	2.4177e - 01	2.9753e - 01	5.4383e - 01
Diver	4	**4.5390e - 04**	9.5888e - 04	7.7273e - 04	1.8401e - 02	6.0705e - 04	4.7672e - 03	1.4489e - 03	4.5784e - 04	5.8986e - 03	2.8633e - 03
	6	**0.0000e + 00**	2.0699e - 02	2.1247e - 02	2.0500e - 02	1.6986e - 02	2.0526e - 02	5.9176e - 02	2.1127e - 02	3.8646e - 02	6.9743e - 02
	8	**4.9174e - 04**	2.1986e - 02	2.7020e - 02	2.9803e - 01	2.7249e - 02	3.1078e - 02	8.7167e - 02	3.3923e - 01	5.8946e - 02	9.8763e - 02
	10	**5.5672e - 03**	2.1660e - 02	4.5160e - 02	2.5325e - 01	2.7837e - 01	9.0056e - 02	5.5104e - 01	2.7969e - 01	9.8534e - 02	6.8754e - 01
	12	**2.1843e - 02**	3.0838e - 01	2.8180e - 02	1.3301e - 01	4.2689e - 01	8.5895e - 02	5.6435e - 01	3.2203e - 01	9.8757e - 02	6.7654e - 01
Elephant	4	**0.0000e + 00**	1.9632e - 02	**0.0000e + 00**	1.6930e - 02	1.9460e - 02	3.0622e - 03	4.9727e - 03	1.5939e - 02	5.8754e - 03	5.8754e - 03
	6	**8.9460e - 04**	4.7949e - 03	7.4900e - 03	2.7495e - 02	3.9956e - 03	3.5767e - 02	1.0488e - 02	3.3018e - 03	4.8753e - 02	2.8785e - 02
	8	**1.2782e - 02**	1.4891e - 02	5.2655e - 02	3.0559e - 01	1.1137e - 01	7.2515e - 02	2.7681e - 01	1.8693e - 02	8.4875e - 02	3.8754e - 01
	10	**1.0683e - 02**	2.2650e - 02	7.8857e - 02	3.4112e - 01	3.1072e - 02	1.4351e - 01	8.6519e - 01	6.9635e - 02	2.8645e - 01	9.7643e - 01
	12	**3.2082e - 02**	1.5757e - 01	7.0852e - 02	4.7685e - 01	9.6941e - 02	1.2052e - 01	6.8057e - 01	3.7980e - 02	3.9865e - 01	7.9863e - 01
Horse	4	**0.0000e + 00**	4.3697e - 05	**0.0000e + 00**	5.7362e - 03	3.0031e - 05	1.1992e - 03	9.0646e - 04	**0.0000e + 00**	2.9864e - 03	9.7845e - 04
	6	**3.9001e - 05**	6.8235e - 04	6.4030e - 05	2.7168e - 02	1.8898e - 02	2.4707e - 02	3.9755e - 01	1.6253e - 04	3.8496e - 02	4.0379e - 01
	8	**1.0658e - 03**	1.5830e - 02	1.5773e - 02	1.7886e - 01	3.2528e - 01	2.0050e - 02	5.1045e - 02	1.8431e - 02	3.8564e - 02	6.8734e - 02
	10	**1.1899e - 03**	2.6765e - 02	1.0699e - 02	4.4824e - 01	2.0086e - 03	9.0610e - 02	1.0780e + 00	1.9035e - 02	9.9363e - 02	2.9747e + 00
	12	**1.0195e - 02**	9.3998e - 02	1.0905e - 02	1.4685e - 01	2.7702e - 01	1.3892e - 01	1.1139e + 00	1.6405e - 02	2.9875e - 01	3.9785e + 00
Kangaroo	4	**0.0000e + 00**	1.8494e - 03	**0.0000e + 00**	1.3075e - 02	7.9334e - 03	2.8633e - 03	3.8586e - 03	5.7127e - 05	3.4079e - 03	4.9445e - 03
	6	**2.9074e - 03**	3.7403e - 03	2.9915e - 03	7.1561e - 02	3.3148e - 03	5.2471e - 03	3.8709e - 02	3.0783e - 03	6.9856e - 03	4.9865e - 02
	8	**5.6649e - 04**	1.1019e - 02	5.4813e - 03	8.2995e - 02	5.6226e - 03	7.0540e - 03	3.1840e - 01	4.0150e - 03	8.4835e - 03	4.9864e - 01
	10	**3.7801e - 03**	1.1827e - 02	1.8489e - 02	3.4071e - 01	6.9291e - 03	3.8810e - 02	5.2978e - 01	8.3780e - 03	5.9836e - 02	6.9886e - 01
	12	**8.7743e - 03**	4.7216e - 02	1.6021e - 02	4.4485e - 01	2.2123e - 02	6.8302e - 02	8.7635e - 01	3.7581e - 02	7.9867e - 02	9.7643e - 01
Lake	4	**0.0000e + 00**	**0.0000e + 00**	**0.0000e + 00**	9.7684e - 03	6.1440e - 03	4.9956e - 03	2.4380e - 04	9.6789e - 05	5.6536e - 03	3.8464e - 04
	6	**1.1065e - 04**	3.8191e - 03	1.1223e - 04	5.6110e - 02	3.2872e - 04	2.1645e - 02	1.6446e - 02	2.1150e - 04	3.9865e - 02	2.4783e - 02
	8	**1.2889e - 03**	7.5427e - 03	2.1513e - 02	9.1336e - 02	2.0880e - 02	4.3782e - 02	5.4877e - 01	5.1575e - 03	5.4765e - 02	6.6584e - 01
	10	**6.1286e - 03**	4.2495e - 02	2.5681e - 02	3.4080e - 01	1.3618e - 02	1.2310e - 01	8.1195e - 01	1.3520e - 02	2.9864e - 01	9.8346e - 01
	12	**1.0719e - 02**	6.5413e - 02	2.2308e - 01	2.9771e - 01	6.4892e - 02	5.0581e - 02	6.3487e - 01	3.5103e - 02	6.5794e - 02	7.3875e - 01

Table 7. The STD values of fitness functions using MCE method.

Images	Levels	MDA	DA	SSA	SCA	ALO	HSO	BA	PSO	BDE	EOFPA
Bridge	4	**0.0000e + 00**	4.3555e - 05	**0.0000e + 00**	4.9764e - 03	1.9393e - 05	8.8101e - 04	1.0221e - 02	1.9393e - 05	3.0201e - 01	6.3697e - 01
	6	**1.6098e - 05**	2.2962e - 03	4.1531e - 05	2.2771e - 02	1.8929e - 02	8.7123e - 03	5.3511e - 02	3.2681e - 04	1.7954e - 01	2.2881e - 02
	8	**4.1976e - 05**	2.6942e - 03	1.6424e - 02	1.6160e - 02	9.5683e - 03	1.2381e - 02	5.2152e - 02	1.3810e - 02	7.4453e - 02	5.3608e - 03
	10	**4.5375e - 05**	5.3553e - 03	2.9144e - 02	1.7617e - 02	6.8743e - 03	1.0299e - 02	4.5361e - 02	6.4860e - 03	1.9697e - 02	2.1813e - 02
	12	**1.3560e - 03**	5.3794e - 03	1.3202e - 02	1.9366e - 02	1.4206e - 02	5.8245e - 03	2.5064e - 02	8.4871e - 03	2.9128e - 01	1.1567e - 01
Building	4	**0.0000e + 00**	**0.0000e + 00**	**0.0000e + 00**	8.4222e - 03	8.2640e - 02	1.8660e - 03	4.0474e - 03	3.7078e - 05	1.5062e - 01	5.9538e - 02
	6	**5.7722e - 06**	2.2725e - 02	1.8815e - 05	2.1296e - 02	1.3243e - 04	1.4934e - 02	6.4824e - 02	4.4130e - 05	6.4139e - 01	1.1567e - 01
	8	**8.1194e - 05**	4.6351e - 03	1.0580e - 02	2.0644e - 02	1.8725e - 02	9.6663e - 03	3.7373e - 02	8.9031e - 03	5.9538e - 02	2.4707e - 02
	10	**4.3296e - 03**	4.5957e - 03	1.4784e - 02	2.9631e - 02	1.4996e - 02	7.8380e - 03	4.1964e - 02	1.3119e - 02	1.1567e - 01	2.2050e - 02
	12	**2.9477e - 03**	4.7520e - 03	1.0995e - 02	1.4305e - 02	9.2023e - 03	3.3979e - 03	3.3865e - 02	3.3371e - 03	5.9538e - 02	9.0610e - 02
Cactus	4	**0.0000e + 00**	1.7572e - 06	**0.0000e + 00**	5.5546e - 03	1.4347e - 06	1.9111e - 03	1.9601e - 03	**0.0000e + 00**	1.1567e - 01	1.3892e - 01
	6	**0.0000e + 00**	2.2275e - 04	3.3031e - 05	2.0208e - 02	1.6240e - 02	7.5591e - 03	3.3427e - 03	1.8360e - 04	3.1104e - 03	1.3820e - 01
	8	**5.7902e - 05**	7.8170e - 03	1.6810e - 03	3.3547e - 02	7.3692e - 03	4.4322e - 03	6.7041e - 02	2.6121e - 03	2.7250e - 01	6.5556e - 01
	10	**1.3753e - 04**	7.0512e - 03	5.4453e - 03	5.7462e - 03	7.6790e - 03	7.0182e - 03	4.6264e - 02	1.3599e - 03	2.3624e - 01	3.468e - 01
	12	**1.7572e - 06**	1.9572e - 06	6.2473e - 05	5.5546e - 03	1.9347e - 06	1.9111e - 03	1.9601e - 03	3.9851e - 04	7.7243e - 03	1.8401e - 02
Cow	4	**0.0000e + 00**	3.5553e - 06	**0.0000e + 00**	4.3800e - 03	7.0152e - 06	1.4766e - 03	5.5669e - 02	2.1702e - 05	2.730e - 01	6.5556e - 01
	6	**8.3327e - 06**	3.3821e - 04	2.9309e - 05	2.9149e - 02	1.5862e - 02	1.1595e - 02	2.3744e - 02	5.9844e - 05	2.0164e - 01	3.5708e - 01
	8	**1.0541e - 05**	7.6901e - 03	8.9780e - 03	1.2516e - 02	7.2417e - 03	5.5736e - 03	3.6431e - 02	1.4694e - 02	7.7273e - 02	1.8401e - 02
	10	**4.4374e - 04**	7.8347e - 03	6.0441e - 03	2.7480e - 02	8.2380e - 03	6.4775e - 03	8.9388e - 03	7.9328e - 03	2.1247e - 02	2.0322e - 02
	12	**6.9515e - 04**	7.2237e - 03	4.2157e - 03	1.2538e - 02	5.5394e - 03	5.9790e - 03	2.4809e - 02	3.0575e - 03	2.7010e - 02	2.673e - 01
Deer	4	**0.0000e + 00**	2.8320e - 05	4.2927e - 02	3.5671e - 02	5.1525e - 02	6.7133e - 03	5.1398e - 02	3.0850e - 05	7.6374e - 03	7.6974e - 03
	6	**2.6894e - 05**	3.4557e - 05	2.5596e - 02	3.2448e - 02	1.6901e - 02	1.9843e - 02	5.0846e - 02	1.6964e - 02	3.8839e - 04	1.9763e - 03
	8	**2.8452e - 05**	1.2801e - 02	3.2649e - 02	2.4045e - 02	2.0842e - 02	9.6566e - 03	2.4055e - 02	1.7755e - 02	4.6669e - 03	2.7418e - 03
	10	**1.7530e - 04**	2.9018e - 02	1.2639e - 02	1.9853e - 02	1.1923e - 02	1.6688e - 02	5.3703e - 02	5.5481e - 03	5.6722e - 03	3.4841e - 01
	12	**2.1929e - 03**	1.2338e - 02	8.8961e - 03	9.8628e - 03	6.5436e - 03	6.7611e - 03	1.7855e - 02	1.0242e - 02	6.5415e - 02	6.5209e - 01
Diver	4	**0.0000e + 00**	**0.0000e + 00**	**0.0000e + 00**	4.1200e - 03	2.9214e - 02	2.5662e - 02	4.3891e - 02	3.1663e - 05	8.5668e - 04	6.3673e - 04
	6	**2.4028e - 07**	2.7901e - 04	2.3344e - 02	1.4257e - 02	1.0952e - 02	1.3559e - 02	7.2014e - 02	1.1899e - 02	3.0089e - 02	8.7357e - 02
	8	**7.2607e - 05**	7.7633e - 03	1.4945e - 02	1.9285e - 02	1.3145e - 02	1.6465e - 02	4.3331e - 02	9.7951e - 03	3.9863e - 02	4.6700e - 02
	10	**6.6864e - 04**	2.8358e - 03	1.1158e - 02	1.8721e - 02	1.1055e - 02	6.6527e - 03	3.3078e - 02	9.4649e - 03	5.9793e - 02	5.7864e - 02
	12	**3.0549e - 03**	7.9756e - 03	8.9144e - 03	7.0841e - 03	1.2907e - 02	1.3726e - 02	4.1452e - 02	1.1751e - 02	5.4598e - 01	3.7680e - 02
Elephant	4	**0.0000e + 00**	9.0077e - 04	9.0267e - 04	4.1081e - 02	1.1055e - 03	4.1184e - 04	4.9507e - 02	9.2036e - 04	2.7340e - 02	6.4674e - 02
	6	**1.4221e - 06**	8.0570e - 04	1.3606e - 02	2.4760e - 02	1.1116e - 02	1.4617e - 02	4.5924e - 02	3.0920e - 02	3.9863e - 02	4.5632e - 03
	8	**7.9685e - 04**	1.0640e - 02	1.2130e - 02	2.6000e - 02	1.0469e - 02	7.8799e - 03	2.8428e - 02	1.1566e - 02	9.8876e - 01	3.9450e - 01
	10	**2.3569e - 03**	2.6128e - 03	3.0860e - 03	1.8732e - 02	8.0987e - 03	8.4090e - 03	2.4145e - 02	6.6496e - 03	6.9973e - 01	4.0876e - 02
	12	**1.0095e - 03**	8.1323e - 03	5.5866e - 03	2.3722e - 02	1.2886e - 02	3.3161e - 03	3.9390e - 02	5.4535e - 03	5.6643e - 01	3.0267e - 02
Horse	4	**0.0000e + 00**	7.6146e - 06	**0.0000e + 00**	3.9869e - 02	**0.0000e + 00**	1.2334e - 03	3.3564e - 04	**0.0000e + 00**	8.6683e - 03	4.9931e - 05
	6	**6.9579e - 06**	2.9511e - 04	1.4611e - 05	3.9917e - 02	1.3604e - 02	1.9567e - 03	3.8756e - 02	1.0109e - 04	3.9825e - 02	1.9603e - 02
	8	**3.2793e - 05**	3.6051e - 03	5.9145e - 04	1.9079e - 02	3.1548e - 04	1.3885e - 02	3.6609e - 02	6.0967e - 03	2.1978e - 01	4.9930e - 01
	10	**2.6460e - 05**	2.5018e - 03	2.0397e - 03	1.4575e - 02	8.7068e - 03	6.2620e - 03	2.7941e - 02	4.9061e - 03	4.5722e - 01	4.9928e - 03
	12	**4.5934e - 04**	5.9886e - 03	5.6341e - 03	2.3385e - 02	5.5206e - 03	6.2517e - 03	1.9090e - 02	3.7609e - 03	2.7638e - 01	3.0145e - 01
Kangaroo	4	**0.0000e + 00**	6.9161e - 05	**0.0000e + 00**	4.5684e - 03	1.3734e - 05	3.7605e - 03	5.9342e - 02	7.9425e - 06	3.0024e - 02	1.5722e - 03
	6	**1.8491e - 06**	3.5289e - 06	7.4256e - 04	2.8424e - 02	5.6422e - 03	4.7666e - 03	6.5992e - 02	6.9616e - 05	5.7732e - 02	4.9731e - 03
	8	**4.1239e - 05**	1.3851e - 03	5.3166e - 03	2.3963e - 02	9.8772e - 03	4.3838e - 03	3.9993e - 02	7.9770e - 03	9.0244e - 02	6.0721e - 03
	10	**2.7729e - 04**	2.3215e - 03	6.1821e - 03	1.5739e - 02	8.2379e - 03	6.0833e - 03	2.9610e - 02	8.2685e - 03	3.5673e - 01	7.0421e - 03
	12	**3.2988e - 04**	6.1240e - 03	8.4453e - 03	9.4158e - 03	1.0962e - 02	5.6801e - 03	2.7494e - 02	3.0349e - 03	5.9722e - 01	6.8722e - 02
Lake	4	**0.0000e + 00**	2.0336e - 06	**0.0000e + 00**	8.3246e - 03	9.5161e - 06	4.1782e - 04	2.1471e - 03	2.0336e - 06	8.6322e - 03	6.2880e - 03
	6	**5.4704e - 06**	5.7846e - 04	1.3840e - 02	4.1747e - 02	1.8414e - 02	1.6083e - 03	4.1550e - 02	6.8613e - 05	8.9630e - 02	7.8652e - 04
	8	**2.5563e - 05**	6.9062e - 03	5.9416e - 03	2.7668e - 02	1.9965e - 02	4.8122e - 03	2.8436e - 02	1.1764e - 02	9.7383e - 02	5.8235e - 02
	10	**7.3581e - 05**	3.9563e - 03	3.9417e - 03	8.5210e - 03	1.1233e - 02	4.2182e - 03	2.3901e - 02	3.8337e - 03	4.8252e - 01	2.8433e - 02
	12	**9.4260e - 04**	6.5949e - 03	2.0884e - 03	1.2109e - 02	9.0033e - 03	5.1305e - 03	3.5508e - 02	2.7738e - 03	3.0211e - 01	5.9363e - 02

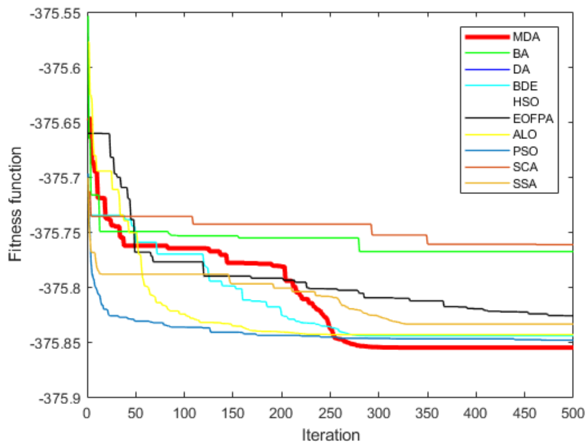

Figure 9. The convergence curves for fitness function using MCE method at 12 levels (Cactus).

In order to study the performance of MDA based method quantitatively, three indices namely PSNR, SSIM, and FSIM are used for all segmented images. Higher values of these indices, better visual similarity between the original image and the segmented image. The optimal PSNR values obtained by all existing methods based on Otsu, Kapur, and MCE are given in Table 8, Table 9, and Table 10 respectively. From the tables it is found that MDA based method gives higher values than other methods in general. For example, the PSNR values in case of Cactus image (for Levels = 12) using MCE are 31.8637, 26.1599, 25.7562, 26.2726, 28.2229, 28.9389, 29.0718, and 26.3566, 23.9993, 23.2377 for MDA, DA, SSA, SCA, ALO, HSO, BA, PSO, BDE, and EOFPA respectively. On comparing the SSIM and FSIM values, which are given in Tables 11–13 and Tables 14–16, the proposed MDA based method has again outperformed the other methods due its precise search ability. It is also seen from the tables above that, the values of these indices increase as the number of threshold levels increase. This indicates the segmentation accuracy improves as the number of threshold levels is increased. For visual analysis, the quantitative results of three indices above through Otsu, Kapur, and MCE method are statistically shown in Figures 10–12, Figures 13–15, and Figures 16–18. It can be seen that the proposed MDA method and other methods have given the similar results with a small number of threshold levels (such as Levels = 4). Whereas, the values obtained by MDA based method are higher than the other existing methods, which indicates the appropriate performance of proposed method.

Table 8. The PSNR values using Otsu's method.

Images	Levels	MDA	DA	SSA	SCA	ALO	HSO	BA	PSO	BDE	EOFPA
	4	**18.9590**	**18.9590**	**18.9590**	18.8034	**18.9590**	18.8983	18.8959	**18.9590**	17.3764	17.316
	6	**21.0920**	21.0911	21.0912	20.5635	**21.0920**	20.9637	20.6669	21.0909	19.1881	18.6372
Bridge	8	**22.5471**	22.3377	22.4018	22.3838	22.5463	22.4298	21.9278	22.4010	22.0039	21.0101
	10	23.4904	23.3166	23.9987	23.0423	23.4885	**24.0388**	23.1488	23.5456	23.1010	22.3231
	12	**27.7081**	24.5108	27.6512	23.5212	24.8332	27.6248	27.6485	24.7378	24.0274	24.7898
	4	**17.4155**	**17.4155**	**17.4155**	16.9523	**17.4155**	17.4062	**17.4155**	**17.4155**	16.9011	16.9837
	6	**19.8902**	19.8187	19.8827	19.2310	19.8816	19.5492	19.0128	19.8782	18.9328	18.9880
Building	8	**22.6625**	22.5599	22.5680	19.5148	22.3500	22.6063	22.0638	22.4596	21.8258	22.4735
	10	**27.1866**	24.2853	24.1301	22.4138	23.9861	24.8874	23.0124	24.2530	22.9203	23.4174
	12	**27.1983**	26.3422	24.7812	24.3360	25.2812	25.9322	24.9121	25.0993	23.8136	25.8439
	4	**20.3428**	**20.3428**	**20.3428**	20.3309	**20.3428**	19.8275	20.3239	**20.3428**	18.5723	18.7159
	6	**22.6678**	22.6661	22.6656	22.6482	22.6643	22.6390	22.5184	22.6668	19.2167	19.3623
Cactus	8	**23.9827**	23.7898	23.9062	23.9712	23.9422	23.5290	23.2659	23.9202	21.3695	21.8403
	10	**24.8346**	24.4556	24.6361	24.4523	24.7911	24.5695	24.7745	24.8199	22.4557	23.5664
	12	**25.3709**	24.8899	25.0184	24.5561	25.2555	24.8233	25.2156	25.2255	22.8350	22.6485
	4	**19.7023**	**19.7023**	**19.7023**	19.0298	19.6970	**19.7023**	19.6131	**19.7023**	16.7087	17.7800
	6	**21.8735**	21.8689	21.6575	21.2382	21.6519	21.5546	21.7785	21.6414	18.8470	19.6007
Cow	8	**23.6239**	23.2169	23.4659	23.0928	23.5503	23.4506	22.9317	23.5084	21.8301	21.2347
	10	**24.4648**	24.4371	24.4203	24.3740	24.3462	23.7822	23.2376	24.2446	21.5213	23.1898
	12	**25.0056**	24.5718	24.8910	24.5712	24.5871	24.6769	23.8856	24.9461	23.2105	24.1131
	4	**17.2462**	**17.2462**	**17.2462**	17.1630	**17.2462**	17.1945	**17.2462**	16.8531	17.2167	17.2285
	6	**22.3158**	21.1255	21.1255	21.2796	21.0306	20.9120	21.0078	21.0168	19.9878	22.3087
Deer	8	**26.4239**	23.5746	23.5002	24.0895	24.6123	26.3803	25.5392	23.9308	20.6895	22.7359
	10	**27.1472**	26.4163	26.8848	27.1317	27.1133	26.5453	26.8428	26.8889	22.3587	24.5996
	12	**27.9602**	27.1372	27.5484	27.8531	26.8076	26.9325	27.5315	27.3095	24.1497	26.6583
	4	**24.4620**	**24.4620**	**24.4620**	24.1907	**24.4620**	24.3989	24.3677	24.3073	22.1893	22.9514
	6	**26.9548**	26.9170	26.9448	26.7044	26.9499	26.6182	26.8265	26.9420	23.9825	24.7704
Diver	8	**29.5492**	29.0513	28.6783	29.0786	28.7387	28.3422	29.2312	28.3347	25.0701	25.4638
	10	**29.8856**	29.4112	29.3481	29.1587	29.8131	29.2198	29.7836	29.8342	25.1395	27.5492
	12	**31.1721**	30.5343	29.7237	31.0997	31.0803	30.9819	30.5211	30.6319	26.5864	27.9708
	4	**18.4590**	**18.4590**	**18.4590**	18.0512	**18.4590**	18.4539	18.3943	18.4092	17.9796	18.4096
	6	**20.9780**	20.5021	20.7348	20.9025	20.4230	20.9187	19.9182	20.5901	19.6514	20.2232
Elephant	8	**24.6549**	23.6814	23.3417	23.1786	22.8950	24.5622	23.5645	24.3392	21.5350	24.2511
	10	**25.7650**	25.4617	25.5042	25.0738	25.6656	25.1711	25.4416	25.6173	22.8253	23.9163
	12	**28.5499**	26.3839	27.1607	27.3927	26.9031	28.4144	28.2370	26.5514	23.4172	25.3764
	4	**18.5055**	**18.5055**	**18.5055**	18.1434	**18.5055**	18.4145	18.3600	18.4552	17.2096	17.1942
	6	**22.6464**	21.6461	21.6464	21.7744	21.6383	21.6707	21.8331	21.4744	20.7108	21.6493
Horse	8	**27.1218**	23.7877	23.7854	25.8693	24.0577	25.3926	24.7503	23.8225	22.3332	22.7768
	10	**28.6567**	24.9734	26.1591	28.5737	28.3542	28.0356	26.0373	26.8562	23.3827	22.5028
	12	**30.1088**	25.9198	28.5245	28.8863	29.9344	29.6778	29.5050	28.9135	25.1314	24.9332
	4	**19.3419**	**19.3419**	**19.3419**	18.9806	19.3351	19.2601	19.2313	19.3312	18.5756	19.2651
	6	**25.2386**	24.3138	24.3100	22.7301	24.3100	24.6060	24.8281	24.7980	21.1692	22.5045
Kangaroo	8	30.1938	28.9547	29.4461	26.8673	28.4724	**30.1943**	26.9309	28.0531	21.6472	22.6528
	10	**33.3271**	32.0741	32.5754	31.3008	31.8186	31.2343	30.6409	31.7392	23.6473	25.961
	12	**34.2483**	33.3702	33.5729	28.8392	33.8080	34.2136	31.7286	33.0561	24.8493	25.4618
	4	17.8079	17.8079	17.8079	17.8948	17.8079	17.8154	**17.8994**	17.7555	15.1330	16.6247
	6	**23.7148**	20.0906	20.2511	23.6525	22.1280	20.5707	20.5545	20.1912	16.8079	16.4567
Lake	8	**25.1894**	23.7439	22.1234	24.4846	22.2729	23.2811	23.1193	24.2111	21.4674	18.8651
	10	**27.3522**	24.2819	23.3769	25.8174	23.2683	26.8299	26.3802	25.8904	22.2032	21.4547
	12	**29.8302**	25.2634	29.3626	29.1830	29.6264	29.0032	27.3752	26.0006	23.9913	23.8048

Table 9. The PSNR values using Kapur's entropy.

Images	Levels	MDA	DA	SSA	SCA	ALO	HSO	BA	PSO	BDE	EOFPA
Bridge	4	**20.3166**	**20.3166**	20.2807	20.1875	18.3087	20.0478	20.0579	**20.3166**	18.0357	18.6645
	6	**24.3943**	23.2706	23.2631	23.9862	23.3301	24.0850	22.8633	23.3948	19.5943	20.5940
	8	**25.9201**	25.3943	25.5275	25.7371	25.5862	25.4035	25.6526	25.6077	21.1853	22.0672
	10	**28.8445**	27.5984	27.0704	28.3361	27.2035	27.9440	27.2823	27.4710	22.4667	23.2884
	12	**32.1186**	27.8596	30.0963	31.4096	30.0612	31.8332	28.2530	30.9551	23.3027	24.6414
Building	4	**17.4224**	17.4155	17.415	17.0523	17.4155	17.4142	17.4155	17.4155	17.4116	17.4155
	6	**19.9023**	19.8637	19.8907	19.2310	19.8816	19.5492	19.7828	19.8072	19.7734	19.7631
	8	**22.9857**	22.5599	22.5680	21.3148	22.3500	22.6063	22.0638	22.6296	20.1984	20.5919
	10	**27.2576**	26.8043	24.1301	22.3738	23.9861	24.8874	23.0124	24.4590	22.2514	22.0813
	12	**28.4124**	27.3672	26.2312	24.8230	25.67 12	25.8392	24.9071	25.0989	24.2399	24.4493
Cactus	4	**17.2477**	**17.2477**	**17.2477**	17.2083	**17.2477**	17.1240	17.2076	**17.2477**	16.4696	16.7799
	6	**25.9105**	24.0561	23.9672	21.5683	21.5404	24.9770	22.9863	22.2332	18.0968	19.4701
	8	**28.7324**	28.6092	28.5111	25.5115	28.6082	28.5553	27.6544	28.4325	19.1565	20.9022
	10	**30.6897**	30.4082	30.4321	28.3328	30.6687	30.4376	27.8268	30.5039	20.9462	22.4891
	12	**32.4563**	31.8106	32.0824	29.9900	32.1917	32.2054	29.6553	31.6696	22.3502	21.3577
Cow	4	**19.3706**	19.3456	**19.3706**	19.1520	**19.3706**	19.2890	19.3563	**19.3706**	16.3414	18.2875
	6	**24.6314**	23.7690	23.9292	23.9305	22.0374	24.1317	23.3235	23.8404	18.4244	21.1759
	8	**27.9669**	26.7169	26.5471	27.1888	26.0817	27.2073	27.0670	25.9695	21.8036	21.3845
	10	**30.8466**	28.8963	28.4620	28.5871	29.1743	28.6751	27.6186	29.6997	23.6779	23.4505
	12	**32.6043**	32.2767	32.4801	28.9822	32.1704	32.5000	30.2576	31.5353	21.9672	23.9536
Deer	4	**21.9086**	**21.9086**	**21.9086**	21.8579	**21.9086**	21.8656	21.8624	**21.9086**	15.8959	18.8904
	6	**25.8662**	25.7468	25.7976	25.2465	25.7708	25.7644	25.8643	25.8127	21.3477	20.6717
	8	**29.1490**	28.3340	28.0174	28.3990	28.1621	28.5526	28.3437	28.4946	22.1424	22.4715
	10	**30.9920**	30.4913	30.9573	28.9004	30.0481	30.8749	28.8843	30.1549	23.4554	23.8936
	12	**32.9202**	32.0876	30.9895	29.7392	32.7632	31.7944	29.8111	32.4031	25.5468	25.6665
Diver	4	**21.6563**	20.9671	21.4636	21.3132	20.9636	20.9593	21.4310	20.9636	20.3982	21.2676
	6	**23.8743**	22.3082	22.3113	22.9759	22.3083	22.3378	22.1613	22.3090	23.0791	21.7095
	8	**27.2237**	24.4619	24.4493	24.4387	24.4538	24.6248	25.8560	24.4246	22.6522	25.1251
	10	**29.4785**	26.0826	26.3010	26.3667	26.4421	26.5201	28.3006	26.7262	25.0380	26.7153
	12	**29.5071**	27.2349	27.1253	26.9334	26.8751	26.6722	28.5156	27.0078	24.8993	24.2954
Elephant	4	**18.6558**	18.6551	**18.6558**	18.6533	**18.6558**	18.5722	18.4352	**18.6558**	17.9303	18.2680
	6	**23.4967**	20.8588	20.8588	22.2481	21.3148	20.8610	20.2596	21.7136	18.8900	20.7315
	8	**26.2198**	23.1849	23.4237	23.4877	24.1624	24.5724	23.1158	23.3720	20.5137	21.8626
	10	**28.9322**	27.6131	25.2279	27.0446	27.7051	25.3211	25.1289	25.3938	20.8363	23.6299
	12	**29.8636**	29.4535	26.7054	28.6767	29.4309	26.3023	29.2218	29.8395	24.1360	22.5109
Horse	4	**19.5685**	**19.5685**	**19.5685**	19.5569	19.5616	19.5208	19.2442	19.5037	17.6152	18.1643
	6	**24.0011**	23.0457	22.9153	23.1092	22.9575	23.1314	22.7394	23.0037	20.2580	21.0483
	8	**27.3558**	26.6826	26.9406	26.3270	27.1482	27.2741	26.7321	26.9479	22.8904	21.8187
	10	**29.3890**	28.0878	28.9131	28.8581	29.3174	29.3410	27.3223	29.1691	23.5858	21.4618
	12	**31.6166**	29.1202	30.4900	29.8254	30.7306	30.7994	30.0701	30.8250	24.5833	23.3923
Kangaroo	4	**19.3602**	19.3419	19.3419	18.9906	19.3351	19.2890	19.2313	19.3451	17.6221	19.2606
	6	**25.3116**	24.3138	24.3120	22.8730	24.9068	24.6890	24.8608	24.3312	21.5133	21.5163
	8	**30.2043**	28.8512	29.4451	26.8932	28.8920	30.1893	26.9309	28.0531	21.7764	21.4465
	10	**33.3301**	32.1521	32.5122	31.4008	31.9123	31.2653	30.5950	31.7392	23.0558	21.7999
	12	**34.2579**	33.5667	33.5679	28.8232	33.9012	34.2716	31.7886	33.8661	24.1207	23.6892
Lake	4	**18.2514**	**18.2514**	**18.2514**	18.1857	18.0607	18.2348	18.2435	**18.2514**	16.4759	15.0601
	6	**23.6183**	22.9541	22.7654	22.8559	23.0808	23.0525	22.6119	22.9201	17.8063	18.2172
	8	**27.4202**	26.1891	26.0786	27.2324	26.0580	25.9331	25.6984	25.8055	20.0112	19.3166
	10	**31.8548**	29.0040	29.1402	28.3226	28.8951	28.6493	28.0904	28.9814	22.3835	22.3242
	12	**33.5569**	30.7909	30.4219	29.9866	31.1864	32.9733	28.4370	30.7989	23.3754	22.4949

Table 10. The PSNR values using MCE method.

Images	Levels	MDA	DA	SSA	SCA	ALO	HSO	BA	PSO	BDE	EOFPA
	4	**19.2861**	**19.2861**	**19.2861**	19.2055	**19.2861**	19.2229	19.2480	**19.2861**	16.9146	17.8353
	6	**22.1149**	21.6223	21.6743	21.8527	21.6255	21.6285	21.5617	21.6775	20.6040	19.7740
Bridge	8	**27.0817**	23.1912	23.2581	23.7577	23.3021	23.7125	23.7922	23.0945	21.9409	21.0794
	10	**28.0991**	24.4551	24.6218	25.4063	26.6986	24.3297	24.5463	24.3360	23.2507	21.9060
	12	**29.8725**	25.1136	29.8542	27.2696	26.9861	25.5679	24.7926	29.3463	23.9524	24.4563
	4	**17.4155**	**17.4155**	**17.4155**	17.3512	**17.4155**	17.2362	**17.4155**	**17.4155**	**17.4155**	**17.4155**
	6	**19.9001**	19.8187	19.8827	19.0280	19.8816	19.5492	19.0638	19. 4596	19.8014	19.5000
Building	8	**22.7921**	22.5599	22.6780	19.5148	22.3500	22.6063	22.0128	22.4512	21.8680	22.1245
	10	**27.2061**	24.2853	24.1301	22.2338	23.8961	24.9474	23.3224	24.6730	22.5496	23.0076
	12	**27.4723**	27.0421	25.0232	24.8906	25.6701	25.8344	24.9589	25.1773	24.5120	24.9875
	4	**21.0442**	**21.0442**	**21.0442**	**21.0422**	21.0126	21.0118	21.0333	**21.0442**	18.2742	19.3560
	6	**24.9529**	23.3510	23.3674	23.3678	24.0485	23.5364	23.4052	23.3895	20.2090	20.6766
Cactus	8	**28.5911**	24.5707	24.5779	24.5286	24.5989	24.5600	26.5850	24.6217	21.9887	22.1574
	10	**29.9778**	25.3832	25.3712	25.4471	27.9011	25.6677	27.6291	25.6254	22.0914	23.7324
	12	**31.8637**	26.1599	25.7562	26.2726	28.2229	28.9389	29.0718	26.3566	23.9993	23.2377
	4	**19.7030**	**19.7030**	19.6961	**19.7030**	19.6629	**19.7030**	19.6343	19.2385	18.4964	17.9766
	6	**24.1672**	21.9280	21.9130	21.3218	21.9130	21.8494	21.7807	21.9032	20.3522	19.2911
Cow	8	**27.3862**	24.0409	24.0290	22.8483	24.0287	23.9099	23.6381	23.9633	20.2832	22.2579
	10	**28.9123**	25.1526	25.0983	24.6953	25.1431	26.3286	25.5584	28.8959	20.6888	21.3226
	12	**30.6430**	30.5606	29.4939	27.4681	26.0056	27.0059	29.7306	29.9207	24.3489	23.7499
	4	**15.6122**	**15.6122**	**15.6122**	15.0935	**15.6122**	15.5409	15.3976	15.6084	**15.6122**	**15.6122**
	6	**24.6247**	19.1380	19.1380	17.9988	22.2128	23.7311	19.1380	19.4118	21.0277	22.7668
Deer	8	**28.0177**	23.9777	23.6379	21.3379	24.2505	25.7888	22.3424	25.0594	21.7233	23.8596
	10	**30.4749**	27.9028	24.8091	23.3179	28.4935	30.3411	28.8443	27.0434	24.3608	25.5655
	12	**32.3489**	31.2369	25.0884	26.1129	29.9859	31.2170	31.0402	30.0412	26.2164	26.8012
	4	**22.2354**	**22.2354**	**22.2354**	22.2239	**22.2354**	22.2020	22.1763	22.2124	22.1863	**22.2354**
	6	**27.8765**	25.6318	26.8252	25.4321	27.6780	26.0988	26.4248	26.3044	24.2075	24.2863
Diver	8	**29.7961**	27.7268	27.6574	27.5539	28.5651	29.6757	26.8094	27.5598	26.0722	24.1376
	10	**31.3786**	29.0860	28.7092	28.1075	28.5887	30.6830	30.3389	31.3065	25.7920	27.7836
	12	**31.8544**	30.3199	30.1751	29.0378	30.8514	31.1978	31.4286	31.3641	27.3253	27.4796
	4	**18.3463**	18.3336	**18.3463**	18.1708	17.4852	17.3805	17.5702	17.4852	**18.3463**	18.2466
	6	**22.1628**	20.4135	20.4535	21.9961	21.7783	20.4135	20.6891	20.3317	20.1564	21.0282
Elephant	8	**24.7220**	23.3686	23.7688	20.9605	24.5748	22.0756	22.7301	24.5871	23.1261	21.9422
	10	**26.9608**	25.5516	25.8022	22.5515	25.4095	25.9089	24.3794	26.5956	23.0379	23.8127
	12	**29.3330**	27.9280	28.9816	24.8952	27.1191	28.2273	25.7960	28.3370	23.8743	23.5579
	4	**20.1345**	19.6709	19.6709	19.6709	19.6763	19.7642	20.0307	19.6716	18.1976	19.3682
	6	**24.6958**	23.0096	23.1941	23.2140	23.1452	23.2670	22.9247	23.1408	20.7912	21.0884
Horse	8	**27.2155**	25.6084	25.4453	25.5795	25.4993	25.5279	24.7919	25.7945	21.5042	22.3496
	10	**28.8068**	26.7811	27.7569	27.8577	27.2524	27.9344	27.2989	28.1843	24.2632	23.8468
	12	**31.2694**	28.2212	28.6299	29.7647	31.1706	29.1077	30.2376	29.5208	24.7799	25.3405
	4	**18.3519**	17.4067	17.4067	17.4067	17.4103	17.0564	17.5407	17.4078	18.2768	18.3300
	6	**23.1569**	21.4058	20.6824	21.1593	21.0394	20.6824	22.5463	20.6663	20.7331	21.1320
Kangaroo	8	**30.0907**	26.0815	26.7610	25.4635	26.1473	24.6898	23.6899	26.0395	22.3714	23.3637
	10	**32.6099**	30.3705	27.9116	26.5561	30.5514	32.0864	29.5856	30.0876	25.4257	25.1888
	12	**34.1968**	34.1044	31.2554	27.3589	32.5094	33.6201	31.1735	33.9460	25.4637	27.0112
	4	**20.0431**	18.9654	18.9611	18.5261	18.9611	18.5723	18.9711	18.9611	15.2170	17.5212
	6	**23.1596**	21.9103	21.8344	21.9101	21.8345	21.7848	21.6807	21.9074	19.9245	19.0321
Lake	8	**26.8396**	22.9554	23.4779	23.4811	24.2708	24.7097	22.7091	23.7849	19.4923	21.2751
	10	**28.9053**	25.3040	25.3547	25.6594	26.8900	26.4456	24.8679	24.7697	21.2984	23.3053
	12	**30.8523**	25.6771	28.9978	27.3762	29.6128	28.9323	28.0416	27.6008	23.7859	23.7204

Table 11. The SSIM values using Otsu's method.

Images	Levels	MDA	DA	SSA	SCA	ALO	HSO	BA	PSO	BDE	EOFPA
Bridge	4	**0.7113**	**0.7113**	**0.7113**	0.7043	**0.7113**	0.7110	0.7090	**0.7113**	0.6573	0.7026
	6	**0.8056**	**0.8056**	0.8055	0.8003	0.8055	0.7992	0.7921	**0.8056**	0.8047	0.7817
	8	**0.8567**	0.8522	0.8544	0.8441	0.8566	0.8461	0.8505	0.8543	**0.8567**	0.8424
	10	**0.8835**	0.8792	0.8717	0.8715	0.8826	0.8756	0.8777	0.8824	0.8762	0.8763
	12	**0.9028**	0.8903	0.9003	0.8828	0.8944	0.8980	0.8969	0.8935	0.8888	0.9020
Building	4	**0.7372**	**0.7372**	**0.7372**	0.7313	**0.7372**	**0.7372**	0.7365	**0.7372**	0.6592	0.6868
	6	**0.8052**	0.8043	0.8030	**0.8052**	0.8031	0.8031	0.8002	0.8024	0.7017	0.7455
	8	**0.8389**	0.8382	0.8384	0.8353	0.8382	0.8346	0.8303	0.8387	0.7955	0.8048
	10	**0.8737**	0.8637	0.8635	0.8558	0.8639	0.8545	0.8531	0.8633	0.8005	0.8252
	12	**0.8831**	0.8813	0.8826	0.8751	0.8824	0.8823	0.8826	0.8824	0.8461	0.8729
Cactus	4	**0.5798**	**0.5798**	**0.5798**	0.5785	**0.5798**	0.5599	**0.5798**	**0.5798**	0.5737	0.5700
	6	**0.6815**	0.6812	0.6813	0.6790	**0.6815**	**0.6815**	0.6705	0.6813	0.6729	0.6498
	8	**0.7351**	0.7275	0.7320	0.7287	0.7349	0.7151	0.7087	0.7344	0.7325	0.7321
	10	**0.7690**	0.7541	0.7611	0.7678	0.7654	0.7551	0.7681	0.7689	0.7650	0.7628
	12	**0.7888**	0.7668	0.7747	0.7767	0.7823	0.7693	0.7724	0.7828	0.7851	0.7677
Cow	4	**0.7231**	**0.7231**	**0.7231**	0.7081	0.7229	0.7225	0.7219	**0.7231**	0.7038	0.7067
	6	0.8050	0.7960	0.8045	0.7888	0.8044	0.7897	0.7975	**0.8057**	0.7858	0.7798
	8	**0.8455**	0.8403	0.8446	0.8430	0.8447	0.8436	0.8362	0.8448	0.8170	0.8453
	10	**0.8739**	0.8688	0.8730	0.8651	0.8723	0.8706	0.8675	0.8706	0.8031	0.8469
	12	**0.8887**	0.8751	0.8818	0.8835	0.8842	0.8768	0.8778	0.8841	0.8701	0.8823
Deer	4	**0.6480**	**0.6480**	**0.6480**	0.6434	**0.6480**	0.6432	**0.6480**	0.6326	0.6397	0.6463
	6	**0.8121**	0.7898	0.7898	0.8034	0.7901	0.7874	0.7656	0.7923	0.7591	0.8120
	8	**0.8749**	0.8566	0.8554	0.8606	0.8595	0.8667	0.8642	0.8653	0.7847	0.8208
	10	**0.9043**	0.8987	0.8931	0.8869	0.8971	0.9015	0.9016	0.9042	0.8587	0.8950
	12	**0.9291**	0.9203	0.9190	0.9248	0.9171	0.9161	0.9087	0.9281	0.8958	0.9057
Diver	4	**0.3451**	**0.3451**	**0.3451**	0.3404	**0.3451**	0.3437	0.3436	0.3450	0.3446	0.3411
	6	**0.4352**	0.4349	**0.4352**	0.4132	**0.4352**	0.4053	**0.4352**	0.4314	0.4327	0.4324
	8	**0.5743**	0.5517	0.4710	0.4917	0.4702	0.4425	0.5405	0.4771	0.5675	0.5233
	10	0.6073	0.5922	0.4985	0.6026	0.6024	0.4838	**0.6380**	0.5744	0.4822	0.6048
	12	**0.6730**	0.6262	0.5055	0.6130	0.6159	0.6107	0.6456	0.5841	0.6068	0.6632
Elephant	4	**0.6080**	**0.6080**	**0.6080**	0.5992	**0.6080**	0.6073	0.6041	**0.6080**	0.6065	0.6042
	6	**0.7149**	0.7073	0.6950	0.7079	0.7084	0.6944	0.6911	0.6945	0.6990	0.7145
	8	**0.7957**	0.7560	0.7446	0.7685	0.7692	0.7543	0.7834	0.7588	0.7497	0.7838
	10	**0.8447**	0.8029	0.7844	0.8421	0.8229	0.7777	0.8314	0.8229	0.8102	0.8029
	12	**0.8776**	0.8164	0.8458	0.8775	0.8257	0.8376	0.8431	0.8309	**0.8776**	0.8657
Horse	4	**0.7462**	**0.7462**	**0.7462**	0.7304	**0.7462**	0.7370	0.7428	0.7450	0.6384	0.6560
	6	**0.8736**	0.8433	0.8436	0.8582	0.8436	0.8573	0.8427	0.8350	0.7952	0.8313
	8	**0.9207**	0.8900	0.8900	0.9195	0.8928	0.9059	0.8915	0.8913	0.8388	0.8528
	10	**0.9475**	0.9069	0.9332	0.9399	0.9426	0.9334	0.9128	0.9283	0.8639	0.8476
	12	**0.9552**	0.9232	0.9479	0.9457	0.9503	0.9458	0.9424	0.9447	0.9002	0.8971
Kangaroo	4	**0.7130**	**0.7130**	**0.7130**	0.7042	0.7126	0.7116	0.7086	0.7129	0.6375	0.6732
	6	**0.8414**	0.8413	**0.8414**	0.8207	**0.8414**	0.8363	0.8220	0.8338	0.7698	0.8036
	8	**0.9101**	0.9014	0.9015	0.8687	0.9007	0.8978	0.8672	0.8976	0.7547	0.8233
	10	**0.9352**	0.9266	0.9286	0.9069	0.9333	0.9211	0.9014	0.9256	0.8290	0.8893
	12	**0.9494**	0.9378	0.9394	0.9219	0.9492	0.9454	0.9192	0.9436	0.8724	0.8834
Lake	4	**0.6677**	0.6676	0.6676	0.6624	0.6676	0.6672	**0.6677**	0.6662	0.6199	0.6635
	6	**0.7982**	0.7646	0.7673	0.7979	0.7853	0.7719	0.7699	0.7663	0.6870	0.7027
	8	**0.8781**	0.8408	0.8277	0.8589	0.8290	0.8388	0.8212	0.8698	0.8205	0.7837
	10	**0.8929**	0.8620	0.8630	0.8812	0.8618	0.8846	0.8778	0.8804	0.8681	0.8331
	12	**0.9186**	0.8836	0.9057	0.8897	0.9070	0.9044	0.8912	0.8888	0.8515	0.8652

Table 12. The SSIM values using Kapur's entropy.

Images	Levels	MDA	DA	SSA	SCA	ALO	HSO	BA	PSO	BDA	EOFPA
	4	**0.6942**	**0.6942**	0.6930	0.6352	0.6889	0.6941	0.6851	**0.6942**	0.6645	0.6772
	6	**0.8170**	0.7948	0.7945	0.8051	0.7958	0.8073	0.7890	0.7969	0.7594	0.8127
Bridge	8	**0.8558**	0.8437	0.8490	0.8370	0.8509	0.8425	0.8443	0.8509	0.8557	0.8551
	10	**0.8984**	0.8837	0.8770	0.8780	0.8798	0.8880	0.8859	0.8803	0.8841	0.8857
	12	**0.9379**	0.8904	0.9126	0.9359	0.9135	0.9236	0.8982	0.9221	0.9241	0.9100
	4	**0.7384**	**0.7384**	**0.7384**	0.7370	**0.7384**	0.7321	0.7379	**0.7384**	0.7100	0.6493
	6	**0.8268**	0.8242	0.8214	0.8194	0.8061	0.8216	0.8199	0.8160	0.7319	0.7376
Building	8	**0.8690**	0.8660	0.8666	0.8503	0.8671	0.8685	0.8462	0.8656	0.7095	0.7367
	10	**0.8960**	0.8930	0.8915	0.8649	0.8943	0.8907	0.8694	0.8924	0.8010	0.8069
	12	**0.9167**	0.9124	0.9134	0.8945	0.9137	0.9147	0.8866	0.9089	0.8416	0.8599
	4	**0.6220**	**0.6220**	**0.6220**	0.6211	0.6207	0.6198	0.6201	**0.6220**	0.4588	0.4650
	6	**0.8041**	0.7119	0.7130	0.7132	0.7546	0.7120	0.7118	0.7138	0.5973	0.6548
Cactus	8	**0.8772**	0.7557	0.7570	0.7589	0.7612	0.7453	0.8131	0.7586	0.6363	0.7261
	10	**0.9072**	0.7797	0.7837	0.7903	0.8430	0.7859	0.8569	0.7907	0.7089	0.7736
	12	**0.9187**	0.8078	0.7949	0.8164	0.8462	0.8472	0.8668	0.8164	0.7656	0.7188
	4	**0.6847**	0.6827	**0.6847**	0.6837	**0.6847**	0.6833	0.6838	**0.6847**	0.6793	0.6830
	6	**0.8151**	0.7801	0.7823	0.7823	0.7913	0.7838	0.7701	0.7804	0.7421	0.8128
Cow	8	**0.8684**	0.8558	0.8543	0.8584	0.8404	0.8499	0.8451	0.8386	0.8317	0.8199
	10	**0.9059**	0.8786	0.8922	0.8798	0.9045	0.8955	0.8481	0.8948	0.8544	0.8746
	12	**0.9252**	0.9176	0.9218	0.9106	0.9209	0.9234	0.8943	0.9140	0.8741	0.8815
	4	**0.6468**	**0.6468**	**0.6468**	0.6385	**0.6468**	0.6467	0.6424	**0.6468**	0.4978	**0.6468**
	6	**0.7928**	0.7918	0.7907	0.7868	0.7923	0.7846	0.7902	**0.7928**	0.7919	0.7580
Deer	8	**0.8663**	0.8649	0.8542	0.8222	0.8581	0.8598	0.8544	0.8650	0.7811	0.8139
	10	**0.9069**	0.9031	0.9063	0.8389	0.8937	0.8933	0.8675	0.8977	0.8291	0.8549
	12	**0.9338**	0.9251	0.9138	0.8882	0.9322	0.9164	0.8742	0.9272	0.8947	0.8984
	4	**0.4393**	**0.4393**	**0.4393**	0.4298	**0.4393**	0.4095	0.4202	**0.4393**	0.4383	0.4321
	6	**0.6108**	0.6060	0.5739	0.5831	0.5682	0.4672	0.5918	0.5862	0.6001	0.5501
Diver	8	**0.8221**	0.6708	0.6074	0.6724	0.6627	0.6000	0.7211	0.6304	0.7962	0.8003
	10	**0.9111**	0.6921	0.6520	0.7011	0.6873	0.6278	0.7519	0.6820	0.8130	0.8131
	12	**0.9183**	0.7537	0.6933	0.7615	0.7566	0.6303	0.7791	0.7146	0.8281	0.8814
	4	**0.5266**	0.5258	**0.5266**	0.5253	**0.5266**	0.5212	0.5212	**0.5266**	**0.5266**	0.5195
	6	**0.6878**	0.6103	0.6103	0.6105	0.6192	0.6520	0.5864	0.6332	0.6492	0.6845
Elephant	8	**0.8032**	0.6783	0.6942	0.6976	0.7281	0.7224	0.7094	0.6978	0.7229	0.7631
	10	**0.8242**	0.8025	0.7513	0.7701	0.7996	0.7534	0.7247	0.7594	0.7079	0.8019
	12	**0.8759**	0.8403	0.7942	0.7859	0.8411	0.8463	0.8182	0.8505	0.8187	0.7491
	4	**0.7496**	**0.7496**	**0.7496**	0.7483	0.7492	0.7476	0.7425	0.7482	0.6587	0.6769
	6	**0.8645**	0.8532	0.8503	0.8535	0.8507	0.8557	0.8493	0.8526	0.7706	0.7907
Horse	8	**0.9231**	0.9168	0.9198	0.9082	0.9218	0.9203	0.9186	0.9199	0.8587	0.8251
	10	**0.9455**	0.9306	0.9434	0.9357	0.9452	0.9387	0.9252	0.9445	0.8553	0.7948
	12	**0.9602**	0.9453	0.9562	0.9434	0.9581	0.9579	0.9409	0.9586	0.8730	0.8409
	4	**0.6355**	0.6345	0.6347	0.6264	0.5800	0.6354	0.6266	**0.6355**	0.6344	0.6335
	6	**0.7745**	0.7558	0.7647	0.7499	0.7700	0.7661	0.7377	0.7698	0.7686	0.7738
Kangaroo	8	**0.8555**	0.8487	0.8514	0.8133	0.8493	0.8333	0.8335	0.8468	0.7650	0.7709
	10	**0.8983**	0.8874	0.8851	0.8398	0.8927	0.8880	0.8412	0.8963	0.8167	0.7521
	12	**0.9235**	0.9170	0.9193	0.8954	0.9143	0.9114	0.8838	0.9208	0.8294	0.8155
	4	**0.6625**	**0.6625**	**0.6625**	0.6610	0.6599	0.6615	0.6613	**0.6625**	0.6609	0.6377
	6	**0.7980**	0.7970	0.7900	0.7864	0.7975	0.7964	0.7871	0.7923	0.7442	0.7460
Lake	8	**0.8682**	0.8641	0.8630	0.8551	0.8649	0.8635	0.8531	0.8610	0.8138	0.7938
	10	**0.9156**	0.9018	0.9065	0.8733	0.9016	0.9025	0.8876	0.8998	0.8369	0.8575
	12	**0.9390**	0.9275	0.9258	0.9034	0.9287	0.9323	0.8979	0.9220	0.8718	0.8648

Table 13. The SSIM values using MCE method.

Images	Levels	MDA	DA	SSA	SCA	ALO	HSO	BA	PSO	BDE	EOFPA
Bridge	4	**0.7366**	**0.7366**	**0.7366**	0.7355	**0.7366**	0.7348	0.7356	**0.7366**	0.6931	0.6918
	6	**0.8266**	0.8214	0.8224	0.8199	0.8212	0.8226	0.8202	0.8221	0.7918	0.7982
	8	**0.8829**	0.8694	0.8703	0.8635	0.8710	0.8679	0.8739	0.8672	0.8516	0.8550
	10	**0.9071**	0.8966	0.9070	0.8893	0.9052	0.9002	0.8795	0.8957	0.8976	0.8873
	12	**0.9345**	0.9101	0.9180	0.9195	0.9204	0.9212	0.8978	0.9238	0.9022	0.9265
Building	4	**0.7553**	**0.7553**	**0.7553**	0.7537	**0.7553**	0.7549	0.7551	**0.7553**	0.6365	0.6991
	6	**0.8221**	0.8185	0.8176	0.8131	0.8181	0.8185	0.8172	0.8175	0.7506	0.7318
	8	**0.8521**	0.8468	0.8466	0.8239	0.8491	0.8511	0.8390	0.8476	0.8003	0.8024
	10	**0.8820**	0.8639	0.8645	0.8540	0.8726	0.8724	0.8571	0.8735	0.8023	0.8297
	12	**0.9048**	0.8922	0.8802	0.8765	0.8966	0.8811	0.8957	0.8786	0.8527	0.8656
Cactus	4	**0.4681**	**0.4681**	**0.4681**	0.4671	**0.4681**	0.4637	0.4673	**0.4681**	0.4657	0.4629
	6	**0.6581**	0.5566	0.5548	0.5573	0.5546	0.5960	0.5702	0.5560	0.6454	0.6295
	8	**0.7983**	0.7213	0.7949	0.7707	0.6305	0.6478	0.7329	0.6397	0.7855	0.7888
	10	**0.8446**	0.7609	0.8330	0.7738	0.7436	0.8322	0.7820	0.7121	0.8344	0.8363
	12	**0.8672**	0.7861	0.8619	0.8372	0.7952	0.8634	0.8492	0.8624	0.8513	0.8209
Cow	4	**0.7327**	**0.7327**	0.7320	**0.7327**	0.7285	**0.7327**	0.7319	0.7140	0.7252	**0.7327**
	6	**0.8227**	0.8206	0.8206	0.8137	0.8133	0.8149	0.8217	0.8199	0.7943	0.8213
	8	**0.8712**	0.8604	0.8603	0.8610	0.8568	0.8455	0.8677	0.8583	0.8489	0.8318
	10	**0.8921**	0.8895	0.8848	0.8830	0.8912	0.8865	0.8896	0.8889	0.8156	0.8744
	12	**0.9139**	0.9108	0.9043	0.9131	0.9119	0.9075	0.8956	0.9097	0.8822	0.8922
Deer	4	**0.6399**	**0.6399**	**0.6399**	0.6170	**0.6399**	0.6304	0.6345	0.6323	0.6381	0.6330
	6	**0.8098**	0.7790	0.7790	0.7627	0.7956	0.7790	0.8026	0.7845	0.7389	0.7832
	8	**0.8807**	0.8621	0.8671	0.8443	0.8553	0.8544	0.8305	0.8628	0.7842	0.8523
	10	**0.9130**	0.9108	0.8948	0.8491	0.9035	0.8997	0.8915	0.8886	0.8737	0.8860
	12	**0.9362**	0.9336	0.9174	0.8651	0.9227	0.9218	0.9191	0.9235	0.8922	0.9079
Diver	4	**0.1592**	0.1351	0.1579	0.1516	0.1351	0.1570	0.1337	0.1351	0.1448	0.1413
	6	**0.2508**	0.1743	0.1743	0.2117	0.1743	0.1736	0.1667	0.1743	0.2476	0.2455
	8	**0.5062**	0.2698	0.2715	0.2716	0.2717	0.2864	0.3493	0.2697	0.4995	0.4941
	10	**0.6209**	0.3446	0.3557	0.5907	0.3566	0.3814	0.3595	0.3819	0.6119	0.6123
	12	**0.6299**	0.5336	0.3968	0.6005	0.3868	0.3936	0.5492	0.3971	0.6141	0.6193
Elephant	4	**0.5256**	0.5208	**0.5256**	0.5253	0.5036	0.5212	0.5092	0.5191	0.5189	0.5169
	6	**0.6787**	0.6093	0.6001	0.6605	0.6192	0.6760	0.5864	0.6332	0.6716	0.6679
	8	**0.7902**	0.6983	0.6984	0.6876	0.7281	0.7204	0.7254	0.6868	**0.7902**	0.7803
	10	**0.8189**	0.8015	0.7845	0.7861	0.7866	0.7214	0.7357	0.7414	0.8141	0.8042
	12	**0.8670**	0.8303	0.7712	0.7989	0.8311	0.8543	0.8212	0.8435	0.8125	0.8098
Horse	4	**0.7878**	0.7769	0.7769	0.7769	0.7791	0.7783	0.7855	0.7689	0.6849	0.7398
	6	**0.8841**	0.8679	0.8700	0.8700	0.8687	0.8673	0.8647	0.8691	0.7936	0.8095
	8	**0.9249**	0.9118	0.9106	0.9139	0.9117	0.9069	0.9032	0.9144	0.8239	0.8532
	10	**0.9413**	0.9313	0.9354	0.9320	0.9336	0.9339	0.9389	0.9398	0.8851	0.8743
	12	**0.9601**	0.9451	0.9459	0.9537	0.9596	0.9529	0.9504	0.9536	0.8939	0.9144
Kangaroo	4	**0.7084**	0.6978	0.6978	0.6958	0.6985	0.6881	0.6919	0.6980	0.6295	0.7071
	6	**0.8353**	0.8252	0.8211	0.8024	0.8256	0.8211	0.8106	0.8203	0.7677	0.7520
	8	**0.8978**	0.8934	0.8936	0.8646	0.8916	0.8878	0.8445	0.8867	0.7731	0.8215
	10	**0.9297**	0.9289	0.9222	0.8966	0.9190	0.9228	0.8905	0.9224	0.8699	0.8722
	12	**0.9479**	0.9457	0.9471	0.9050	0.9434	0.9398	0.9086	0.9471	0.8763	0.9062
Lake	4	**0.6875**	0.6871	**0.6875**	0.6807	**0.6875**	0.6783	0.6788	**0.6875**	0.6346	0.6589
	6	**0.7906**	0.7904	0.7892	0.7882	0.7895	0.7856	0.7800	0.7902	**0.7906**	0.7692
	8	**0.8579**	0.8360	0.8441	0.8470	0.8487	0.8473	0.8280	0.8432	0.7952	0.8246
	10	**0.8919**	0.8798	0.8813	0.8785	0.8835	0.8833	0.8612	0.8740	0.8399	0.8827
	12	**0.9200**	0.8950	0.9176	0.9049	0.9090	0.9093	0.8971	0.9092	0.8726	0.8929

Table 14. The FSIM values using Otsu's method.

Images	Levels	MDA	DA	SSA	SCA	ALO	HSO	BA	PSO	BDE	EOFPA
Bridge	4	**0.7972**	**0.7972**	**0.7972**	0.7925	**0.7972**	**0.7972**	**0.7972**	**0.7972**	0.7684	0.7886
	6	**0.8669**	0.8667	**0.8669**	0.8525	**0.8669**	0.8642	0.8576	**0.8669**	**0.8669**	0.8365
	8	**0.8995**	0.8988	0.8993	0.8795	**0.8995**	0.8973	0.8916	0.8989	0.8826	0.8775
	10	**0.9183**	0.9177	0.9164	0.9099	0.9175	0.9126	0.9086	0.9181	0.9039	0.8923
	12	**0.9383**	0.9274	0.9378	0.9103	0.9289	0.9318	0.9292	0.9288	0.9120	0.9240
Building	4	**0.7762**	**0.7762**	**0.7762**	0.7722	**0.7762**	**0.7762**	**0.7762**	**0.7762**	0.7234	0.7362
	6	**0.8413**	**0.8413**	0.8409	0.8370	0.8408	**0.8413**	0.8387	0.8403	0.7659	0.7850
	8	**0.8752**	0.8749	0.8747	0.8575	0.8739	0.8731	0.8660	0.8744	0.8161	0.8417
	10	**0.9075**	0.8981	0.8975	0.8827	0.8968	0.8923	0.8857	0.8979	0.8348	0.8574
	12	**0.9161**	0.9156	0.9021	0.9036	0.9130	0.9130	0.9049	0.9122	0.8883	0.8905
Cactus	4	**0.7771**	**0.7771**	**0.7771**	0.7747	**0.7771**	0.7740	0.7764	**0.7771**	0.7770	**0.7771**
	6	**0.8458**	0.8456	**0.8458**	0.8371	0.8456	0.8444	0.8402	**0.8458**	0.8454	0.8464
	8	**0.8812**	0.8760	0.8794	0.8805	**0.8812**	0.8746	0.8650	0.8810	0.8804	0.8733
	10	**0.9012**	0.8957	0.8993	0.9011	0.9001	0.8971	0.8790	0.9009	0.8968	**0.9012**
	12	**0.9133**	0.9064	0.9101	0.9078	0.9129	0.9044	0.9099	0.9113	0.9013	0.9053
Cow	4	**0.7983**	**0.7983**	**0.7983**	0.7909	**0.7983**	0.7982	0.7975	**0.7983**	0.7813	0.7834
	6	**0.8698**	0.8657	0.8695	0.8455	0.8694	0.8612	0.8607	0.8690	0.8434	0.8397
	8	**0.9022**	0.8990	0.9017	0.8830	0.9015	0.9008	0.8961	0.9015	0.8810	0.8885
	10	0.9212	0.9207	**0.9214**	0.9086	0.9103	0.9190	0.9129	0.9196	0.8495	0.8888
	12	**0.9329**	0.9307	0.9313	0.9178	0.9136	0.9203	0.9186	0.9312	0.9076	0.9279
Deer	4	**0.7449**	**0.7449**	**0.7449**	0.7422	**0.7449**	0.7434	0.7439	0.7410	0.7158	0.7332
	6	**0.8517**	0.8423	0.8423	0.8367	0.8420	0.8466	0.8307	0.8507	0.7969	0.8551
	8	**0.9086**	0.8875	0.8861	0.8840	0.8921	0.9033	0.9022	0.8940	0.8153	0.8513
	10	**0.9346**	0.9203	0.9233	0.9097	0.9212	0.9322	0.9223	0.9292	0.8827	0.9027
	12	**0.9481**	0.9349	0.9416	0.9466	0.9363	0.9382	0.9323	0.9445	0.9009	0.9307
Diver	4	**0.7798**	**0.7798**	**0.7798**	0.7776	**0.7798**	0.7739	0.7754	0.7783	**0.7798**	**0.7798**
	6	0.8331	0.8329	0.8331	0.8303	**0.8332**	0.8212	0.8327	0.8331	0.8324	0.8301
	8	**0.8648**	0.8599	0.8627	0.8631	0.8617	0.8471	0.8498	0.8641	0.8614	0.8614
	10	**0.8877**	0.8819	0.8798	0.8870	0.8853	0.8717	0.8562	0.8643	0.8804	0.8739
	12	**0.9114**	0.9021	0.8913	0.8982	0.8979	0.8879	0.8746	0.8910	**0.9114**	**0.9114**
Elephant	4	**0.7582**	**0.7582**	**0.7582**	0.7516	**0.7582**	0.7564	0.7568	**0.7582**	0.7924	**0.7582**
	6	**0.8248**	0.8217	0.8139	0.8245	0.8216	0.8130	0.8150	0.8136	0.8228	0.8209
	8	**0.8582**	0.8514	0.8477	0.8533	0.8552	0.8515	0.8574	0.8539	0.8463	**0.8582**
	10	**0.8855**	0.8775	0.8723	0.8854	0.8848	0.8683	0.8849	0.8828	0.8738	0.8778
	12	**0.9051**	0.8915	0.9023	0.9036	0.8947	0.8943	0.8851	0.8866	0.8802	0.9008
Horse	4	**0.8207**	**0.8207**	**0.8207**	0.8194	**0.8207**	0.8206	0.8181	0.8200	0.7313	0.7576
	6	**0.8887**	0.8786	0.8787	0.8881	0.8783	0.8867	0.8760	0.8736	0.8576	0.8862
	8	**0.9303**	0.9079	0.9078	0.9298	0.9102	0.9215	0.9069	0.9070	0.8814	0.8808
	10	**0.9489**	0.9252	0.9363	0.9466	0.9393	0.9386	0.9169	0.9370	0.9081	0.8875
	12	**0.9568**	0.9349	0.9392	0.9479	0.9559	0.9559	0.9501	0.9466	0.9103	0.9323
Kangaroo	4	**0.7456**	**0.7456**	**0.7456**	0.7387	0.7455	0.7448	0.7425	0.7452	0.7207	**0.7456**
	6	**0.8534**	0.8529	0.8529	0.8250	0.8529	0.8515	0.8468	0.8498	0.8340	0.8528
	8	**0.9214**	0.9148	0.9186	0.8776	0.9117	0.9213	0.8827	0.9086	0.8286	0.8690
	10	**0.9522**	0.9457	0.9471	0.9279	0.9453	0.9375	0.9224	0.9434	0.9075	0.9026
	12	**0.9685**	0.9557	0.9567	0.9392	0.9607	0.9628	0.9356	0.9554	0.9239	0.9292
Lake	4	**0.7694**	**0.7694**	**0.7694**	0.7684	0.7654	0.7677	0.7682	0.7689	0.7399	**0.7694**
	6	**0.8582**	0.8370	0.8376	0.8560	0.8478	0.8401	0.8389	0.8369	0.7859	0.7923
	8	**0.9063**	0.8804	0.8730	0.8902	0.8738	0.8790	0.8666	0.8936	0.8610	0.8461
	10	**0.9137**	0.8929	0.8936	0.9132	0.8933	0.9120	0.9074	0.9103	0.8870	0.8808
	12	**0.9399**	0.9054	0.9357	0.9210	0.9384	0.9317	0.9166	0.9189	0.8835	0.8849

Table 15. The FSIM values using Kapur's entropy.

Images	Levels	MDA	DA	SSA	SCA	ALO	HSO	BA	PSO	BDE	EOFPA
	4	**0.7954**	**0.7954**	0.7947	0.7941	0.7856	0.7943	0.7915	**0.7954**	0.7262	0.7361
	6	**0.8787**	0.8701	0.8695	0.8651	0.8700	0.8749	0.8641	0.8705	0.7359	0.8764
Bridge	8	**0.9081**	0.9008	0.9043	0.8956	0.9052	0.9004	0.8941	0.9051	0.8783	0.8948
	10	**0.9274**	0.9240	0.9220	0.9095	0.9227	0.9244	0.9225	0.9221	0.8928	0.9056
	12	**0.9549**	0.9308	0.9409	0.9471	0.9434	0.9505	0.9276	0.9451	0.9054	0.9201
	4	**0.7776**	**0.7776**	**0.7776**	0.7771	**0.7776**	0.7757	0.7769	**0.7776**	0.7401	0.7354
	6	**0.8574**	0.8564	0.8547	0.8472	0.8428	0.8547	0.8502	0.8507	0.8022	0.7879
Building	8	**0.9017**	0.9009	0.9000	0.8703	0.9004	0.9001	0.8757	0.8987	0.8136	0.7887
	10	**0.9283**	0.9257	0.9258	0.8887	0.9274	0.9228	0.8992	0.9260	0.8481	0.8528
	12	**0.9444**	0.9423	0.9440	0.9193	0.9439	0.9440	0.9149	0.9394	0.8840	0.8803
	4	**0.7766**	**0.7766**	**0.7766**	0.7756	0.7761	0.7762	0.7762	**0.7766**	0.7304	0.7284
	6	**0.8489**	0.8488	0.8482	0.8470	0.8464	0.8483	0.8448	0.8484	0.8011	0.8056
Cactus	8	**0.8981**	0.8840	0.8847	0.8834	0.8835	0.8827	0.8728	0.8846	0.8344	0.8758
	10	**0.9266**	0.9045	0.9070	0.9181	0.9054	0.9049	0.8886	0.9066	0.8683	0.8960
	12	**0.9390**	0.9196	0.9165	0.9187	0.9271	0.9262	0.9203	0.9161	0.8800	0.8806
	4	**0.7726**	0.7708	**0.7726**	0.7722	**0.7726**	0.7709	0.7717	**0.7726**	0.7703	0.7703
	6	**0.8635**	0.8522	0.8538	0.8442	**0.8635**	0.8548	0.8418	0.8524	0.8304	0.8547
Cow	8	**0.9024**	0.8964	0.9014	0.8891	0.8984	0.8959	0.8834	0.8973	0.8765	0.8779
	10	**0.9311**	0.9223	0.9256	0.9038	0.9201	0.9298	0.8881	0.9251	0.8951	0.9120
	12	**0.9473**	0.9428	0.9453	0.9282	0.9453	0.9471	0.9114	0.9412	0.9017	0.9080
	4	**0.7496**	**0.7496**	**0.7496**	0.7470	**0.7496**	0.7355	0.7456	**0.7496**	0.6511	0.7429
	6	**0.8617**	0.8607	0.8600	0.8511	0.8613	0.8559	0.8580	0.8612	0.8397	0.8291
Deer	8	**0.9122**	0.9114	0.9061	0.8735	0.9078	0.9083	0.8997	0.9118	0.8417	0.8613
	10	**0.9393**	0.9352	0.9387	0.8841	0.9323	0.9337	0.9115	0.9322	0.8806	0.9047
	12	**0.9573**	0.9509	0.9440	0.9275	0.9568	0.9468	0.9145	0.9531	0.9168	0.9195
	4	**0.7295**	0.7093	0.7275	0.7159	0.7092	0.7268	0.7066	0.7092	0.7037	0.7242
	6	**0.7528**	**0.7528**	**0.7528**	0.7527	0.7526	0.7511	0.7387	**0.7528**	0.7446	0.7525
Diver	8	**0.8116**	0.7948	0.7945	0.8045	0.7949	0.7982	0.7944	0.7925	0.8115	**0.8116**
	10	**0.8470**	0.8129	0.8234	0.8263	0.8251	0.8399	0.8329	0.8461	0.8464	0.8456
	12	**0.8776**	0.8625	0.8548	0.8551	0.8512	0.8410	0.8359	0.8547	0.8679	0.8685
	4	**0.7151**	0.7148	**0.7151**	0.7149	**0.7151**	0.7117	0.7115	**0.7151**	0.7115	0.7120
	6	**0.8016**	0.7707	0.7707	0.7708	0.7711	0.7921	0.7577	0.7799	0.8007	0.8008
Elephant	8	**0.8543**	0.8104	0.8221	0.8246	0.8426	0.8313	0.8093	0.8240	0.8244	0.8451
	10	**0.8813**	0.8782	0.8601	0.8423	0.8738	0.8616	0.8153	0.8640	0.8425	0.8786
	12	**0.9057**	0.8965	0.8835	0.9036	0.9007	0.8978	0.8531	0.8806	0.8882	0.8725
	4	**0.8264**	**0.8264**	**0.8264**	0.8243	0.8262	0.8254	0.8231	**0.8264**	0.7714	0.7843
	6	**0.8900**	0.8897	0.8883	0.8890	0.8891	0.8891	0.8847	0.8892	0.8358	0.8393
Horse	8	**0.9355**	0.9316	0.9324	0.9197	0.9339	0.9341	0.9316	0.9326	0.8989	0.8731
	10	**0.9558**	0.9457	0.9518	0.9495	0.9531	0.9541	0.9390	0.9520	0.9011	0.8674
	12	**0.9707**	0.9536	0.9632	0.9565	0.9647	0.9655	0.9538	0.9638	0.9147	0.9060
	4	**0.7364**	0.7362	0.7355	0.7275	0.6934	0.7359	0.7347	**0.7364**	0.7349	0.7364
	6	**0.8485**	0.8388	0.8427	0.8298	0.8470	0.8450	0.8203	0.8468	0.8388	0.8445
Kangaroo	8	**0.9153**	0.9116	0.9113	0.8906	0.9113	0.9058	0.9020	0.9110	0.8460	0.8402
	10	**0.9450**	0.9403	0.9393	0.8920	0.9421	0.9386	0.9072	0.9431	0.8831	0.8697
	12	**0.9612**	0.9576	0.9589	0.9295	0.9575	0.9545	0.9288	0.9587	0.8962	0.9052
	4	**0.7464**	**0.7464**	**0.7464**	0.7430	0.7408	0.7452	0.7449	**0.7464**	0.7430	0.7381
	6	**0.8443**	0.8371	0.8366	0.8276	0.8387	0.8371	0.8312	0.8370	0.7971	0.8113
Lake	8	**0.8988**	0.8948	0.8936	0.8888	0.8920	0.8912	0.8827	0.8894	0.8458	0.8448
	10	**0.9441**	0.9268	0.9297	0.9017	0.9258	0.9259	0.9165	0.9263	0.8775	0.8727
	12	**0.9620**	0.9465	0.9441	0.9274	0.9477	0.9582	0.9249	0.9436	0.8995	0.8989

Table 16. The FSIM values using MCE method.

Images	Levels	MDA	DA	SSA	SCA	ALO	HSO	BA	PSO	BDE	EOFPA
Bridge	4	**0.7948**	**0.7948**	**0.7948**	0.7940	**0.7948**	0.7945	0.7938	**0.7948**	0.7881	0.7891
	6	**0.8652**	0.8573	0.8570	0.8453	0.8556	0.8554	0.8517	0.8570	**0.8652**	0.8600
	8	**0.9101**	0.8985	0.8963	0.8728	0.8951	0.9015	0.8927	0.8921	0.8914	0.8619
	10	**0.9293**	0.9182	0.9176	0.8977	0.9235	0.9266	0.8966	0.9138	0.9065	0.8803
	12	**0.9501**	0.9314	0.9302	0.9335	0.9348	0.9422	0.9128	0.9406	0.9101	0.9289
Building	4	**0.7772**	**0.7772**	**0.7772**	0.7754	**0.7772**	0.7762	0.7770	**0.7772**	0.7277	0.7619
	6	**0.8460**	0.8425	0.8419	0.8348	0.8451	0.8420	0.8408	0.8418	0.8029	0.7832
	8	**0.8811**	0.8726	0.8731	0.8416	0.8740	0.8731	0.8637	0.8783	0.8221	0.8537
	10	**0.9105**	0.8907	0.8904	0.8741	0.8984	0.9025	0.8800	0.9016	0.8490	0.8660
	12	**0.9328**	0.9205	0.9060	0.8996	0.9244	0.9063	0.9173	0.9042	0.8928	0.8911
Cactus	4	**0.7340**	**0.7340**	**0.7340**	0.7335	**0.7340**	0.7312	0.7331	**0.7340**	0.7244	0.7325
	6	**0.8110**	0.7972	0.7966	0.8018	0.7965	0.7976	0.8020	0.7975	0.7917	0.7965
	8	**0.8765**	0.8619	0.8742	0.8745	0.8400	0.8475	0.8658	0.8430	0.8577	0.8673
	10	**0.9094**	0.8887	0.9039	0.8903	0.8934	0.9035	0.8925	0.8775	0.8754	0.8922
	12	**0.9270**	0.9015	0.9246	0.8909	0.9257	0.9078	0.9009	0.9241	0.9117	0.9067
Cow	4	**0.7975**	**0.7975**	0.7973	**0.7975**	0.7949	**0.7975**	0.7968	0.7774	0.7974	0.7935
	6	**0.8715**	0.8713	0.8712	0.8594	0.8669	0.8688	0.8675	0.8712	0.8690	0.8675
	8	**0.9066**	0.9062	0.9063	0.8869	0.9035	0.8981	0.9029	0.9052	0.8815	0.8796
	10	**0.9263**	0.9253	0.9225	0.9161	0.9258	0.9207	0.9156	0.9261	0.8709	0.9013
	12	**0.9424**	0.9395	0.9338	0.9311	0.9395	0.9373	0.9187	0.9379	0.9107	0.9175
Deer	4	**0.7485**	**0.7485**	**0.7485**	0.7480	**0.7485**	0.7455	0.7466	0.7476	0.7156	0.7482
	6	**0.8607**	0.8601	0.8590	0.8531	0.8593	0.8579	0.8581	0.8602	0.7937	0.8604
	8	**0.9119**	0.9109	0.9034	0.8856	0.9098	0.9103	0.9097	0.9068	0.8382	0.8882
	10	**0.9376**	0.9312	0.9307	0.8956	0.9311	0.9354	0.9178	0.9302	0.8935	0.9048
	12	**0.9478**	0.9419	0.9422	0.9301	0.9437	0.9429	0.9181	0.9428	0.9151	0.9192
Diver	4	**0.8083**	**0.8083**	**0.8083**	0.8073	**0.8083**	0.8008	0.7961	0.8082	0.7115	0.7290
	6	**0.8422**	0.8399	0.8393	0.8396	0.8325	0.8352	0.8321	0.8403	0.7426	0.7332
	8	**0.8815**	0.8813	0.8620	0.8664	0.8776	0.8525	0.8497	0.8646	0.8036	0.7981
	10	**0.9033**	0.8998	0.8890	0.8826	0.8987	0.8729	0.8690	0.8925	0.8219	0.8381
	12	**0.9100**	0.9096	0.9007	0.8908	0.9040	0.8870	0.8756	0.9070	0.8686	0.8585
Elephant	4	**0.7657**	0.7653	**0.7657**	0.7609	0.7640	0.7651	0.7648	0.7650	0.6993	0.7016
	6	**0.8294**	0.8279	0.8278	0.8219	0.8293	0.8279	0.8178	0.8278	0.7987	0.8000
	8	**0.8733**	0.8585	0.8582	0.8586	0.8586	0.8569	0.8431	0.8536	0.8355	0.8476
	10	**0.8816**	0.8808	0.8803	0.8758	0.8811	0.8757	0.8638	**0.8816**	0.8678	0.8794
	12	**0.9084**	0.8932	0.9069	0.8901	0.8964	0.8952	0.8953	0.8969	0.8796	0.8808
Horse	4	**0.8266**	**0.8266**	**0.8266**	0.8257	**0.8266**	0.8264	0.8202	**0.8266**	0.7711	0.8128
	6	**0.8953**	0.8862	0.8871	0.8875	0.8871	0.8910	0.8842	0.8866	0.8491	0.8596
	8	**0.9317**	0.9201	0.9186	0.9081	0.9186	0.9165	0.9070	0.9214	0.8657	0.8913
	10	**0.9498**	0.9342	0.9368	0.9373	0.9374	0.9432	0.9451	0.9399	0.9153	0.9017
	12	**0.9654**	0.9472	0.9500	0.9520	0.9653	0.9516	0.9555	0.9545	0.9279	0.9324
Kangaroo	4	**0.7364**	0.7229	0.7229	0.7229	0.7232	0.7146	0.7214	0.7229	0.7259	0.7212
	6	**0.8423**	0.8269	0.8184	0.8057	0.8239	0.8184	0.8222	0.8180	0.8174	0.7918
	8	**0.9212**	0.8954	0.8977	0.8613	0.8958	0.8841	0.8489	0.8879	0.8367	0.8795
	10	**0.9469**	0.9361	0.9204	0.8916	0.9283	0.9452	0.9122	0.9247	0.9001	0.8817
	12	**0.9623**	0.9486	0.9606	0.9088	0.9502	0.9582	0.9217	0.9615	0.9056	0.9228
Lake	4	**0.7765**	0.7764	**0.7765**	0.7727	**0.7765**	0.7731	0.7758	**0.7765**	0.7306	0.7314
	6	**0.8451**	0.8450	0.8443	0.8448	0.8445	0.8422	0.8372	0.8449	0.8418	0.8311
	8	**0.8945**	0.8768	0.8812	0.8885	0.8849	0.8831	0.8704	0.8810	0.8474	0.8564
	10	**0.9134**	0.9047	0.9051	0.9115	0.9064	0.9079	0.8901	0.8994	0.8758	0.9036
	12	**0.9370**	0.9158	0.9347	0.9337	0.9248	0.9314	0.9226	0.9257	0.8956	0.9162

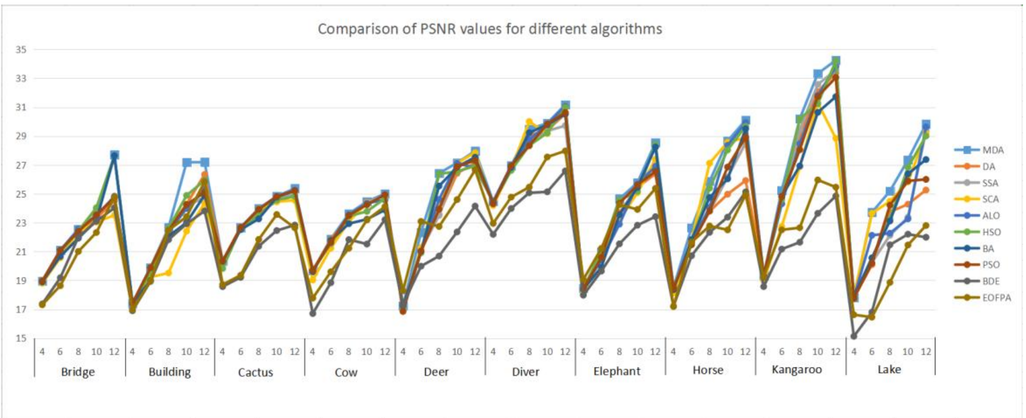

Figure 10. Comparison of PSNR values for different algorithms using Otsu's method at five levels.

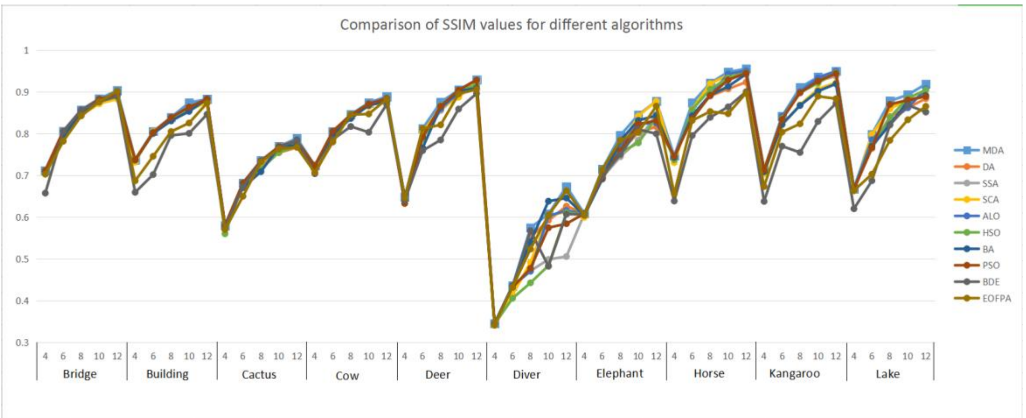

Figure 11. Comparison of SSIM values for different algorithms using Otsu's method at five 12 levels.

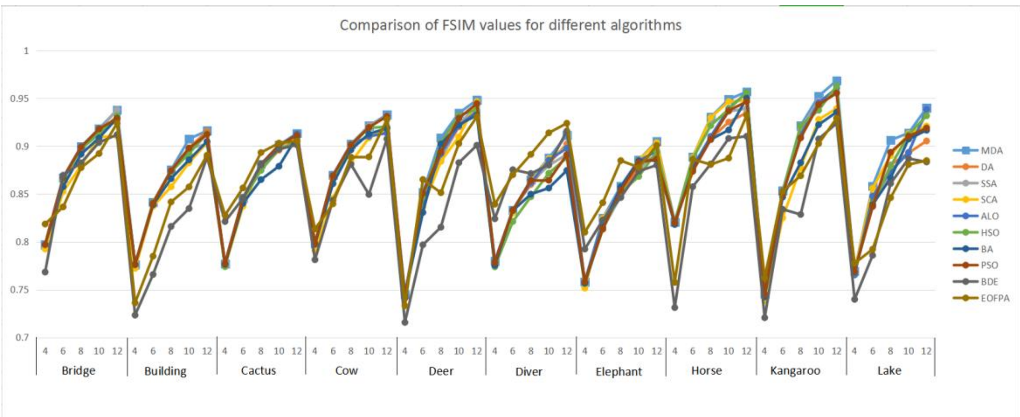

Figure 12. Comparison of FSIM values for different algorithms using Otsu's method at five levels.

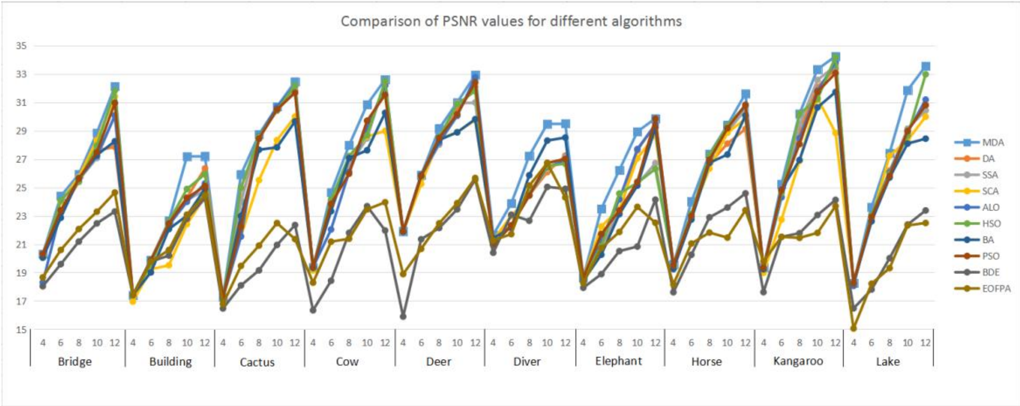

Figure 13. Comparison of PSNR values for different algorithms using Kapur's entropy at five levels.

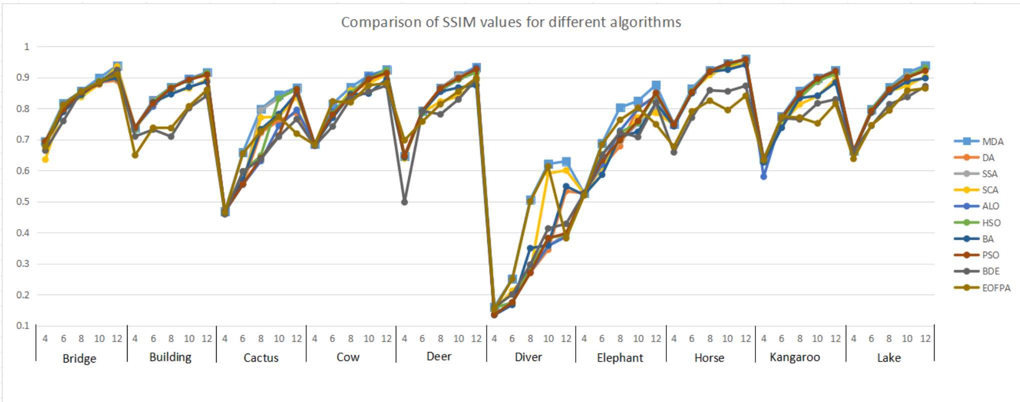

Figure 14. Comparison of SSIM values for different algorithms using Kapur's entropy at five levels.

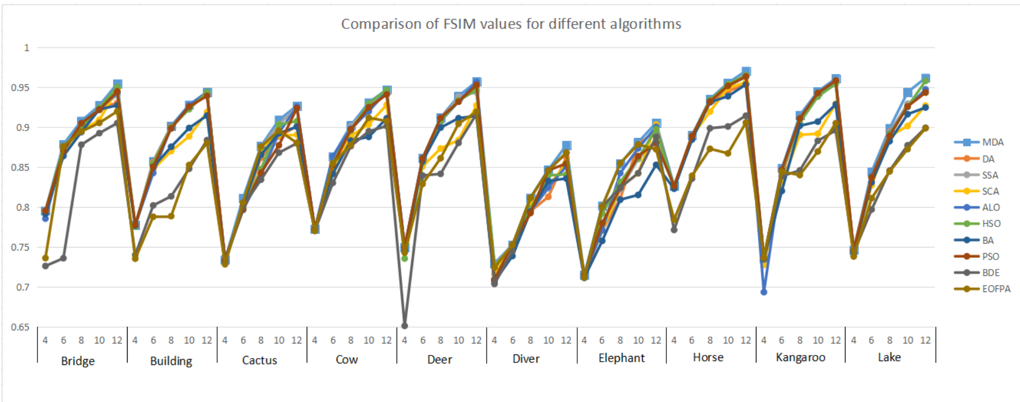

Figure 15. Comparison of FSIM values for different algorithms using Kapur's entropy at five levels.

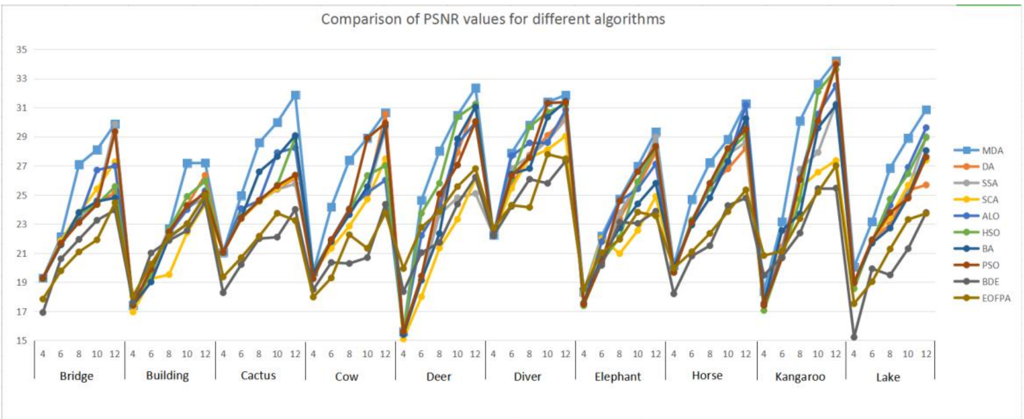

Figure 16. Comparison of PSNR values for different algorithms using MCE method at five levels.

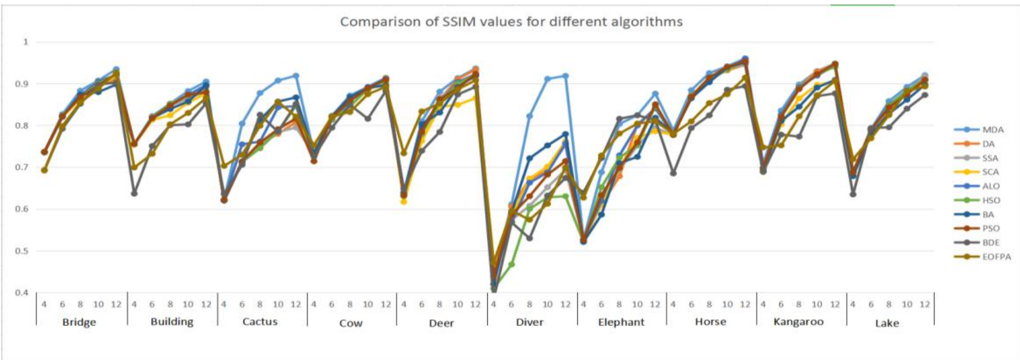

Figure 17. Comparison of SSIM values for different algorithms using MCE method at five levels.

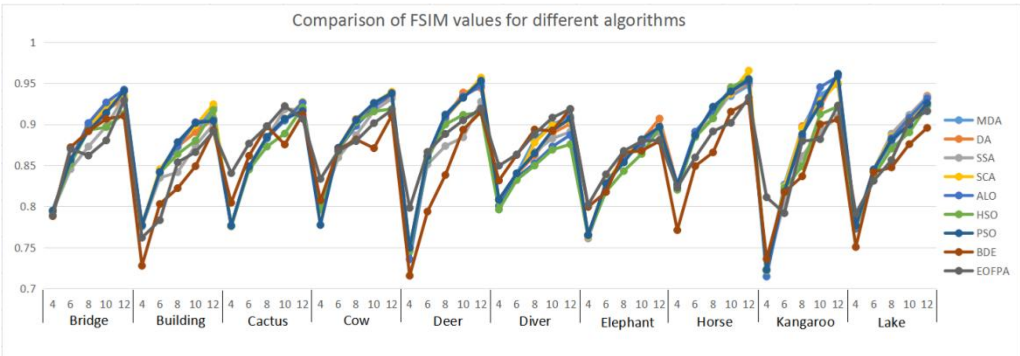

Figure 18. Comparison of FSIM values for different algorithms using MCE method at 4,6,8,10 and 12 levels.

6.1.2. Statistical analysis

In order to add further analysis to the results given above, a non-parametric Wilcoxon's rank sum test for 30 cases of experiment (10 images through 3 different thresholding approaches) has been conducted, which is performed at significance level 5%. The fitness function values of MDA based method are compared with other existing methods. The null hypothesis considered as there is no significant difference between the two compared methods. $h = 1$ means the null hypothesis can be rejected at 5% significance level. On the other hand, $h = 0$ means the null hypothesis cannot be rejected. p is statistical probability. If a value of $p < 0.05$, it means that there is strong evidence against the null hypothesis. Table 17 gives the p and h values of all cases at 12 threshold levels. From the table it is found that MDA based method gives the satisfied values in general. For example, in the experiment using Otsu method, MDA based method produces better results in 89 out of 90 cases (10 images and 9 compared algorithms) when compared with other existing methods. Besides, in the experiment using Kapur method, MDA based method gives satisfied results in 88 out of 90 cases when compared with other existing methods. Additionally, in the experiment using MCE method, MDA based method gives satisfied results in 88 out of 90 cases when compared with other existing methods. From the experimental results above it is evident that MDA based method not only obtains higher quality segmented images, but also verifies its superior performance in a statistically meaningful way.

Table 17. Statistical analysis for the results of experiments on three methods.

Threshold Methods	Images	MDA vs DA		MDA vs SSA		MDA vs SCA		MDA vs ALO		MDA vs HSO		MDA vs BA		MDA vs PSO		MDA vs BDE		MDA vs EOFPA	
		p	h	p	h	p	h	p	h	p	h	p	h	p	h	p	h	p	h
Otsu	Bridge	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1
	Building	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1
	Cactus	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1
	Cow	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1
	Deer	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1
	Diver	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1
	Elephant	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1
	Horse	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1
	Kangaroo	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1
	Lake	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	**0.0540**	**0**	<0.05	1	<0.05	1
Kapur	Bridge	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1
	Building	<0.05	1	**0.1496**	**0**	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1
	Cactus	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1
	Cow	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	**0.0740**	**0**	<0.05	1	<0.05	1
	Deer	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1
	Diver	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1
	Elephant	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1
	Horse	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1
	Kangaroo	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1
	Lake	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1
MCE	Bridge	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1
	Building	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1
	Cactus	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1
	Cow	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1
	Deer	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1
	Diver	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1
	Elephant	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1
	Horse	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1
	Kangaroo	<0.05	1	**0.0731**	**0**	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1
	Lake	<0.05	1	<0.05	1	<0.05	1	**0.2401**	**0**	<0.05	1	<0.05	1	<0.05	1	<0.05	1	<0.05	1

Table 18. Comparison of PSNR, SSIM and FSIM values obtained by MDA.

Images	Levels	PSNR			SSIM			FSIM		
		Otsu	Kapur	MCE	Otsu	Kapur	MCE	Otsu	Kapur	MCE
Bridge	4	18.9590	**20.3166**	19.2861	0.7113	0.6942	**0.7366**	**0.7972**	0.7954	0.7948
	6	21.0920	**24.3943**	22.1149	0.8056	0.8170	**0.8266**	0.8669	**0.8787**	0.8652
	8	22.5471	25.9201	**27.0817**	0.8567	0.8558	**0.8829**	0.8995	0.9081	**0.9101**
	10	23.4904	**28.8445**	28.0991	0.8835	0.8984	**0.9071**	0.9183	0.9274	**0.9293**
	12	27.7081	**32.1186**	29.8725	0.9028	**0.9379**	0.9345	0.9378	**0.9549**	0.9501
Building	4	17.4155	**17.4224**	17.4155	0.7372	0.7384	**0.7553**	0.7762	**0.7776**	0.7772
	6	19.8902	**19.9023**	19.9001	0.8052	**0.8268**	0.8221	0.8413	**0.8574**	0.8460
	8	22.6625	**22.9857**	22.7921	0.8389	**0.8690**	0.8521	0.8752	**0.9017**	0.8811
	10	27.1866	**27.2576**	27.2061	0.8737	**0.8960**	0.8820	0.9075	**0.9283**	0.9105
	12	27.1983	**28.4124**	27.4723	0.8826	**0.9167**	0.9048	0.9161	**0.9444**	0.9328
Cactus	4	20.3428	17.2477	**21.0442**	0.5798	**0.6220**	0.4681	**0.7771**	0.7766	0.7340
	6	22.6678	**25.9105**	24.9529	0.6815	**0.8041**	0.6581	0.8458	**0.8489**	0.8110
	8	23.9827	**28.7324**	28.5911	0.7351	**0.8772**	0.7983	0.8812	**0.8981**	0.8765
	10	24.8346	**30.6897**	29.9778	0.7690	**0.9072**	0.8446	0.9012	**0.9266**	0.9094
	12	25.3709	**32.4563**	31.8637	0.7888	**0.9187**	0.8672	0.9133	**0.9390**	0.9270
Cow	4	19.7023	19.3706	**19.7030**	0.7231	0.6847	**0.7327**	**0.7983**	0.7726	0.7975
	6	21.8735	**24.6314**	24.1672	0.8050	0.8151	**0.8227**	0.8698	0.8635	**0.8715**
	8	23.6239	**27.9669**	27.3862	0.8455	0.8684	**0.8712**	0.9022	0.9024	**0.9066**
	10	24.4648	**30.8466**	28.9123	0.8739	**0.9059**	0.8921	0.9212	**0.9311**	0.9263
	12	25.0056	**32.6043**	30.6430	0.8887	**0.9252**	0.9139	0.9329	**0.9473**	0.9424
Deer	4	17.2462	**21.9086**	15.6122	**0.6480**	0.6468	0.6399	0.7449	**0.7496**	0.7485
	6	22.3158	**25.8662**	24.6247	**0.8121**	0.7928	0.8098	0.8517	**0.8617**	0.8607
	8	26.4239	**29.1490**	28.0177	0.8749	0.8663	**0.8807**	0.9086	**0.9122**	0.9119
	10	27.1472	**30.9920**	30.4749	0.9043	0.9069	**0.9130**	0.9346	**0.9393**	0.9376
	12	27.9602	**32.9202**	32.3489	0.9291	0.9338	**0.9362**	0.9481	**0.9573**	0.9478
Diver	4	**24.4620**	21.6563	22.2354	0.3451	**0.4393**	0.1592	0.7798	0.7295	**0.8083**
	6	26.9548	23.8743	**27.8765**	0.4352	**0.6108**	0.2508	0.8331	0.7528	**0.8422**
	8	29.4786	27.2237	**29.7961**	0.5743	**0.8221**	0.5062	0.8648	0.8116	**0.8815**
	10	29.8856	29.4785	**31.3786**	0.6073	**0.9111**	0.6209	0.8877	0.8470	**0.9033**
	12	31.1721	29.5071	**31.8544**	0.6730	**0.9183**	0.6299	**0.9114**	0.8776	0.9100
Elephant	4	18.4590	**18.6558**	18.3463	**0.6080**	0.5266	0.5256	0.7582	0.7151	**0.7657**
	6	20.9780	**23.4967**	22.1628	**0.7149**	0.6878	0.6787	0.8248	0.8016	**0.8294**
	8	24.6549	**26.2198**	24.7220	0.7957	**0.8032**	0.7902	0.8582	0.8543	**0.8733**
	10	25.7650	**28.9322**	26.9608	**0.8447**	0.8242	0.8189	**0.8855**	0.8813	0.8816
	12	28.5499	**29.8636**	29.3330	**0.8776**	0.8759	0.8670	0.9051	0.9057	**0.9084**
Horse	4	18.5055	19.5685	**20.1345**	0.7462	0.7496	**0.7878**	0.8207	0.8264	**0.8266**
	6	22.6464	24.0011	**24.6958**	0.8736	0.8645	**0.8841**	0.8887	0.8900	**0.8953**
	8	25.8693	**27.3558**	27.2155	0.9207	0.9231	**0.9249**	0.9303	**0.9355**	0.9317
	10	28.6567	**29.3890**	28.8068	**0.9475**	0.9455	0.9413	0.9489	**0.9558**	0.9498
	12	30.1088	**31.6166**	31.2694	0.9552	**0.9602**	0.9601	0.9568	**0.9707**	0.9654
Kangaroo	4	19.3419	**19.3602**	18.3519	**0.7130**	0.6355	0.7084	**0.7456**	0.7364	0.7364
	6	25.2386	**25.3116**	23.1569	**0.8414**	0.7745	0.8353	**0.8534**	0.8485	0.8423
	8	30.1938	**30.2043**	30.0907	**0.9101**	0.8555	0.8978	**0.9214**	0.9153	0.9212
	10	33.3271	**33.3301**	32.6099	**0.9352**	0.8983	0.9297	**0.9522**	0.9450	0.9469
	12	34.2483	**34.2579**	34.1968	**0.9494**	0.9235	0.9479	**0.9685**	0.9612	0.9623
Lake	4	17.8079	18.2514	**20.0431**	0.6677	0.6625	**0.6875**	0.7694	0.7464	**0.7765**
	6	**23.7148**	23.6183	23.1596	**0.7982**	0.7980	0.7906	**0.8582**	0.8443	0.8451
	8	25.1894	**27.4202**	26.8396	**0.8781**	0.8682	0.8579	**0.9063**	0.8988	0.8945
	10	27.3522	**31.8548**	28.9053	0.8929	**0.9156**	0.8919	0.9137	**0.9441**	0.9134
	12	29.8302	**33.5569**	30.8523	0.9186	**0.9390**	0.9200	0.9399	**0.9620**	0.9370

6.2. Comparison of Otsu method, Kapur's entropy, and MCE based on MDA algorithm

In order to verify the performance of MDA based method using various thresholding techniques, we made a comparison between the Otsu's method, Kapur's entropy, and MCE using MDA algorithm. The optimal PSNR, SSIM, and FSIM values obtained by the three MDA based methods are given in Table 18. It can be found that PSNR values obtained by MDA based method using Otsu produces better result in 2 out of 50 cases (5 thresholds and 10 test images), the Kapur's entropy based method gives better result in 38 out of 50 cases, and the MCE based method gives better result in 10 out of 50 cases. On comparing the SSIM values, these three thresholding techniques give better results in 14 out of 50 cases, 21 out of 50 cases, and 15 out of 50 cases respectively. Additionally, it can be evidently seen that Otsu based method produces better result in 12 out of 50 cases, Kapur's entropy based method gives better result in 23 out of 50 cases, and MCE based method gives better result in 15 out of 50 cases. To sum up, the frequency of getting better results through these three thresholding methods are 28 out of 150 cases, 82 out of 150 cases, and 40 out of 150 cases respectively, which consider all performance measures above. From the analysis above, we can find that MDA based method using Kapur's entropy has comprehensively outperformed the MDA based method using Otsu and MCE in terms of PSNR, SSIM, and FSIM values. Just like the no free lunch (NFL) theorem states, the proposed method may obtain better results than other multilevel thresholding techniques on several segmentation problems which have not been solved yet [24,60]. Therefore, the application of MDA based method for various color image segmentation is potential and meaningful.

7. Conclusion

In this paper, MDA based multilevel thresholding method has been presented to determine the optimal thresholds values for color image segmentation. The proposed method is tested on the various color images from Berkley segmentation data set. The performance of proposed method is then compared with other nine algorithms.

The main contributions of this paper are: (1) the improvement of standard DA through the techniques such as chaotic maps, EOBL strategy and DE algorithm. (2) the application of three MDA based multilevel thresholding techniques for color image segmentation, namely Kapur's entropy, MCE method and Otsu method. In order to verify the effectiveness of proposed method, a comprehensive set of experimental series have been performed through several measures such as, AM, STD, PSNR, SSIM, and FSIM. Meanwhile, a non-parametric Wilcoxon's rank sum test has also been conducted in the paper for statistical analysis. From the experimental results we can find that MDA based method performs better than the compared methods in general. Thus, these promising results motivate the utilization of MDA based entropy method to solve various image segmentation problems. In the future, the MDA based method will be investigated for plant canopy image segmentation using other multilevel thresholding techniques such as Tsallis entropy, fuzzy entropy, and Renyi's entropy.

Funding

This research was supported by the Fundamental Research Funds for the Central Universities (No. 2572019BF04), the National Nature Science Foundation of China (No. 31470714), the Northeast Forestry University Horizontal Project (No. 43217002, No. 43217005, No. 43219002).

Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive comments and suggestions.

Conflicts of interest

The authors declare no conflict of interest.

References

1. C. Jung, M. Jian, J. Liu, et al., Interactive image segmentation via kernel propagation, *Pattern Recognit.*, **47** (2014), 2745–2755.
2. S. H. Lee, H. I. Koo and N. I. Cho, Image segmentation algorithms based on the machine learning of features, *Pattern Recognit. Lett.*, **31** (2010), 2325–2336.
3. W. Chen, H. Yue, J. Wang, et al., An improved edge detection algorithm for depth map inpainting, *Opt. Lasers Eng.*, **55** (2014), 69–77.
4. J. Ye, G. Fu and U. P. Poudel, High-accuracy edge detection with Blurred Edge Model, *Image Vision Comput.*, **23** (2005), 453–467.
5. G. Zhang, H. Zhu and N. Xu, Flotation bubble image segmentation based on seed region boundary growing, *Min. Sci. Technol.*, **21** (2011), 239–242.
6. K. Liu, L. Guo, H. Li, et al., Fusion of Infrared and Visible Light Images Based on Region Segmentation, *Chin. J. Aeronaut.*, **22** (2009), 75–80.
7. N. Otsu, A threshold selection method from gray-level histograms, *IEEE Trans. Syst. Man Cybern.*, **9** (1979), 62–66.
8. 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, *Comput. Vision Graphics Image Process.*, **29** (1985), 273–285.
9. C. H. Li and C. K. Lee, Minimum cross entropy thresholding, *Pattern Recognit.*, **26** (1993), 617–625.
10. M. A. E. Aziz, A. A. Ewees and A. E. Hassanien, Whale Optimization Algorithm and Moth-Flame Optimization for multilevel thresholding image segmentation, *Expert Syst. Appl.*, **83** (2017), 242–256.
11. G. Sun, A. Zhang, Y. Yao, et al., Multilevel thresholding using grey wolf optimizer for image segmentation, *Expert Syst. Appl.*, **86** (2017), 64–76.
12. E. Cuevas, D. Zaldivar and M. Pérez-cisneros, A novel multi-threshold segmentation approach based on differential evolution optimization, *Expert Syst. Appl.*, **37** (2010), 5265–5271.
13. G. Sun, A. Zhang, Y. Yao, et al., A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding, *Appl. Soft Comput.*, **46** (2016), 703–730.
14. A. K. Bhandari, V. K. Singh, A. Kumar, et al., Cuckoo search algorithm and wind driven optimization based study of satellite image segmentation for multilevel thresholding using Kapur's entropy, *Expert Syst. Appl.*, **41** (2014), 3538–3560.
15. A. K. Bhandari, V. K. Singh, A. Kumar, et al., A novel technique for multilevel optimal magnetic resonance brain image thresholding using bacterial foraging, *Meas.*, **41** (2008), 1124–1134.

16. S. Ouadfel and A. Taleb-Ahmed, Social spiders optimization and flower pollination algorithm for multilevel image thresholding: A performance study, *Expert Syst. Appl.*, **55** (2016), 566–584.
17. U. Mlakar, B. Potočnik and J. Brest, A hybrid differential evolution for optimal multilevel image thresholding, *Expert Syst. Appl.*, **65**(2016), 221–232.
18. P. D. Sathya and R. Kayalvizhi, Modified bacterial foraging algorithm based multilevel thresholding for image segmentation, *Eng. Appl. Artif. Intell.*, **24** (2011), 595–615.
19. W. A. Hussein, S. Sahran and S. N. H. S. Abdullah, A fast scheme for multilevel thresholding based on a modified bees algorithm, *Knowled. Based Syst.*, **101** (2016), 114–134.
20. H. S. Gill, B. S. Khehra, A. Singh, et al., Teaching-learning-based optimization algorithm to minimize cross entropy for Selecting multilevel threshold values, *Egypt. Inform. J.*, (2018).
21. K. P. B. Resma and S. N. Madhu, Multilevel thresholding for image segmentation using Krill Herd Optimization algorithm, *J. King Saud Univ. Comput. Inf. Sci.*, (2018).
22. S. Pare, A. K. Bhandari, A. Kumar, et al., A new technique for multilevel color image thresholding based on modified fuzzy entropy and Lévy flight firefly algorithm, *Comput. Electr. Eng.*, **70** (2018), 476–495.
23. R. A. Ibrahim, M. A. Elaziz and S. Lu, Chaotic opposition-based grey-wolf optimization algorithm based on differential evolution and disruption operator for global optimization, *Expert Syst. Appl.*, **108** (2018), 1–27.
24. S. Mirjalili, Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, *Neural Comput. Appl.*, **27** (2016), 1053–1073.
25. F. Wilcoxon, Individual comparison by ranking methods, *Biom. Bull.*, **1** (1945), 80–83.
26. C. Fan, H. Ouyang, Y. Zhang, et al., Optimal multilevel thresholding using molecular kinetic theory optimization algorithm, *Appl. Math. Comput.*, **239** (2014), 391–408.
27. S. Manikandan, K. Ramar, M. W. Iruthayarajan, et al., Multilevel thresholding for segmentation of medical brain images using real coded genetic algorithm, *Meas.*, **47** (2014), 558–568.
28. M. Horng, A multilevel image thresholding using the honey bee mating optimization, *Appl. Math. Comput.*, **215** (2010), 3302–3310.
29. L. Cao, P. Bao and Z. Shi, The strongest schema learning GA and its application to multilevel thresholding, *Image Vision Comput.*, **26** (2008), 716–724.
30. A. Bouaziz, A. Draa and S. Chikhi, Artificial bees for multilevel thresholding of iris images, *Swarm Evol. Comput.*, **21** (2015), 32–40.
31. A. K. Bhandari, A. Kumar and G. K. Singh, Tsallis entropy based multilevel thresholding for colored satellite image segmentation using evolutionary algorithms, *Expert Syst. Appl.*, **42** (2015), 8707–8730.
32. S. Pare, A. Kumar, V. Bajaj, et al., An efficient method for multilevel color image thresholding using cuckoo search algorithm based on minimum cross entropy, *Appl. Soft Comput.*, **61** (2017), 570–592.
33. K. Price, Differential evolution: A fast and simple numerical optimizer, *Fuzzy Inf. Process. Soc.*, (1996), 524–527.
34. H. R. Tizhoosh, Opposition-based learning: A new scheme for machine intelligence, *Int. Conf. Computat. Intell. Modell.*, **1** (2005), 695–701.
35. T. Xiang, X. Liao and K. Wong, An improved particle swarm optimization algorithm combined with piecewise linear chaotic map, *Appl. Math. Comput.*, **190** (2007), 1637–1645.

36. G. Kaur and S. Arora, Chaotic whale optimization algorithm, *J. Comput. Des. Eng.*, **5** (2018), 275–284.
37. M. Kohli and S. Arora, Chaotic grey wolf optimization algorithm for constrained optimization problems, *J. Comput. Des. Eng.*, **5** (2018), 458–472.
38. H. Wang, Z. Wu, S. Rahnamayan, et al., Enhancing particle swarm optimization using generalized opposition-based learning, *Inf. Sci.*, **181** (2011), 4699–4714.
39. H. Wang, Z. Wu and S. Rahnamayan, Enhanced opposition-based differential evolution for high-dimensional optimization problems, *Soft Comput.*, **15** (2011), 2127–2140.
40. H. Wang, S. Rahnamayan, H. Sun, et al., Gaussian bare-bones differential evolution, *IEEE Trans. Cybern.*, **43** (2013), 634–647.
41. H. Zorlu, Optimization of weighted myriad filters with differential evolution algorithm, *AEU Int. J. Electron. Commun.*, **77** (2017), 1–9.
42. U. Yüzgeç and M. Eser, Chaotic based differential evolution algorithm for optimization of baker's yeast drying process, *Egypt. Inf. J.*, **19** (2018), 151–163.
43. R. P. Parouha and K. N. Das, Economic load dispatch using memory based differential evolution, *Int. J. Bioinspired. Comput.*, **11** (2018), 159–170.
44. H. Wang, Z. Wu and S. Rahnamayan, Differential evolution based on node strength, *Int. J. Bioinspired. Comput.*, **11** (2018), 34–45.
45. S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, et al., Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems, *Adv. Eng. Software.*, **14** (2017), 163–191.
46. S. Mirjalili, SCA: A Sine Cosine Algorithm for solving optimization problems, *Knowled. Based Syst.*, **96** (2016), 120–133.
47. N. Singh and S. B. Singh, A novel hybrid GWO-SCA approach for optimization problems, *Eng. Sci. Technol. Int. J.*, **20** (2017), 1586–1601.
48. S. Mirjalili, The Ant Lion Optimizer, *Adv. Eng. Software.*, **83**(2015), 80–98.
49. E. Emary, H. M. Zawbaa and A. E. Hassanien, Binary ant lion approaches for feature selection, *Neurocomput.*, **213** (2016), 54–65.
50. D. Manjarres, I. Landa-Torres, S. Gil-Lopez et al., A survey on applications of the harmony search algorithm, *Eng. Appl. Artif. Intell.*, **26** (2013), 1818–1831.
51. A. H. Gandomi and X. Yang, Chaotic bat algorithm, *Int. J. Comput. Sci. Eng. Int.*, **5** (2014), 224–232.
52. Z. Ye, M. Wang, W. Liu, et al., Fuzzy entropy based optimal thresholding using bat algorithm, *Appl. Soft Comput.*, **31** (2015), 381–395.
53. N. S. M. Raja, S. A. Sukanya and Y. Nikita, Improved PSO based multi-level thresholding for cancer infected breast thermal images using otsu, *Procedia Comput. Sci*., **48** (2015), 524–529.
54. A. Chander, A. Chatterjee and P. Siarry, A new social and momentum component adaptive PSO algorithm for image segmentation, *Expert Sys. Appl.*, **38** (2011), 4998–5004.
55. A. K. Bhandari, A novel beta differential evolution algorithm-based fast multilevel thresholding for color image segmentation, *Neural Comput. Applic.*, (2018), 1–31. DOI: 10.1007/s00521-018-3771-z.
56. M. Abdel-Baset, H. Wu, Y. Zhou, et al., Elite opposition-flower pollination algorithm for quadratic assignment problem, *J. Intell. Fuzzy Syst.*, **33** (2017), 901–911. DOI: 10.3233/jifs-162141.

57. C. Li and A. C. Bovik, Content-partitioned structural similarity index for image quality assessment, *Signal Process. Image Commun.*, **25** (2010), 517–526.
58. J. John, M. S. Nair, P. R. A. Kumar, et al., A novel approach for detection and delineation of cell nuclei using feature similarity index measure, *Biocybern. Biomed. Eng.*, **36** (2016), 76–88.
59. S. Pare, A. K. Bhandari, A. Kumar, et al., An optimal color image multilevel thresholding technique using grey-level co-occurrence matrix, *Expert Syst. Appl.*, **87** (2017), 335–362.
60. D. H. Wolpert and W. G. Macready, No free lunch theorems for optimization, *Evol. Comput. IEEE Trans.*, **1** (1997), 67–82.