Engineering Applications of Artificial Intelligence 109 (2022) 104599

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Differential exponential entropy-based multilevel threshold selection methodology for colour satellite images using equilibrium-cuckoo search optimizer

Monorama Swain [a], Tanmaya Tapaswini Tripathy [b], Rutuparna Panda [b,*], Sanjay Agrawal [b], Ajith Abraham [c]

[a] *Department of Elect. & Comm. Engg., Silicon Institute of Technology, Bhubaneswar 751024, India*
[b] *Department of Elect. & Telecomm. Engg., Veer Surendra Sai University of Tech., Burla 768018, India*
[c] *Machine Intelligence Research Labs, WA, USA, Center for Artificial Intelligence, Innopolis University, Innopolis, Russia*

ARTICLE INFO

Keywords:
Artificial intelligence
Entropy
Multilevel thresholding
Colour satellite image
Equilibrium Optimizer
Cuckoo Search

ABSTRACT

Recently, the entropic based multilevel threshold selection methods use 2D histogram, which is constructed using the local averages, leading to a loss of edges. Further, the computation of the entropy using the diagonal pixel values only leads to a loss of information. Nevertheless, traditional 2D histogram based multilevel thresholding methods suffer from efficiently retaining the spatial correlation information. In addition, the conventional entropy uses logarithmic function, which has inherent problems, thereby, reducing the accuracy at some situations. To solve these issues, a differential exponential entropy (DEE) -based multilevel threshold selection methodology is proposed. To suppress the high magnitude peaks in the 2D histogram, the normalized local variance is used while the construction. A novel objective function is suggested to compute the DEE. A new Equilibrium-Cuckoo Search Optimizer (ECSO) is suggested to maximize the DEE. For testing, standard benchmark functions are used. The results are compared with the physics-based Equilibrium Optimizer (EO) and the nature-inspired Cuckoo Search Algorithm (CSA). Different benchmark colour satellite images are acquired from the Landsat Image Gallery database for the experiment. The performances are compared with the state-of-the-art methods. Different metrics such as PSNR, SSIM and FSIM are used for the image quality assessment. A statistical analysis is presented in terms of the Box plots. Our proposed DEE-ECSO outperforms the other techniques. The suggested algorithm would be useful for segmentation of the brain MR images for biomedical engineering applications.

1. Introduction

The relevance of vision in humans was at the root of the incorporation of image processing into a variety of fields, as well as the growth of science and emerging topics in the area of the Artificial Intelligence (AI). Among these fields, one can mention remote sensing, which has seen an increase in interest in recent years, especially in all things related to satellite image segmentation and interpretation techniques. In today's AI image processing era, where the majority of satellite images are virtually captured using digital format. In addition, AI image processing helps us to combine intelligence with machine learning and computer vision to process large volumes of images very easily and speedily. Note that the image perception and analysis entail some level of digital processing (Awad and Chehdi, 2009) methods using AI. Digital image processing may include a variety of procedures such as data formatting and correction, digital enhancement to aid visual perception, and even computer-assisted classification of targets and features. Large quantities of satellite image data are generated due to the creation/presentation of numerous remote sensing platforms. As a result, efficient querying and browsing in these image databases are becoming increasingly important (Pandey et al., 2018). It is intended to get useful information from satellite imagery to get the advantages, making the effective utility of the data. Indeed, the detection and/or quantity of different goals in an image to obtain useful knowledge about them are a part of satellite imagery interpretation and analysis.

There are different types of image processing techniques to analyse the satellite images such as pre-processing, transformation, segmentation and analysis. Pre-processing techniques include a variety of operations that can be applied to satellite images, including information analysis and retrieval. As a result, these techniques adequately depict

[*] Corresponding author.
E-mail addresses: mswain@silicon.ac.in (M. Swain), tanmayatapaswinitripathy98@gmail.com (T.T. Tripathy), r_ppanda@yahoo.co.in (R. Panda), agrawals_72@yahoo.com (S. Agrawal), ajith.abraham@ieee.org (A. Abraham).

https://doi.org/10.1016/j.engappai.2021.104599
Received 19 September 2021; Received in revised form 12 November 2021; Accepted 3 December 2021
Available online xxxx

the sensor's measured, reflected or released radiation, as well as geometric distortion adjustment due to Sensor-Earth geometry changes. Segmentation techniques are used for classifications of different regions (Rekik et al., 2007). Image segmentation is the method in which various regions of an image are divided using similarity of the pixel values, intensity homogeneity, texture, and contrast (Deepika and Vishnu, 2016). Therefore, it is needed to investigate a profound segmentation method for analysis and interpretation of colour satellite images.

There are different types of image segmentation techniques. In this work, we use the thresholding-based image segmentation, because it makes the analysis of an image simpler. The thresholding-based segmentation is divided into two types such as bi-level thresholding (BTH) and multi-level thresholding (MTH). Especially, for colour satellite image segmentation, MTH-based segmentation technique is preferable, because it provides more information for analysis. There are two types of approaches based on both BTH and MTH methods such as: Parametric-based segmentation approach and Non-parametric-based segmentation approach. In parametric-based segmentation approach, every class's grey-level distribution is comprised of a probability density function which follow the Gaussian distribution (Pare et al., 2020). These methods attempt to estimate the distribution parameters, which match the given histogram data. It usually results in a nonlinear optimization problem with a time-consuming solution. The non-parametric-based segmentation approach determines the best threshold for separating the pixel areas of an image through different selective criteria such as: Otsu's between class variance (Otsu, 1979), Kapur's entropy (Kapur et al., 1985), Shannon's entropy (El-Sayed and Tarek Abd-El Hafeez, 2011), Renyi's entropy (Sahoo et al., 1997), Cross entropy (Pal, 1996), Tsallis entropy (Portes de Albuquerque et al., 2004), Masi entropy (Masi, 2005) etc. This has motivated the authors to investigate a new class of entropic-based methodology for MTH.

Mostly, recent entropic-based methods are extended to 2D. All these techniques explicitly use a 2D histogram. Usually, a 2D histogram is constructed utilizing the local averaging scheme, which fails to retain the edge (high frequency) information. Moreover, the logarithmic functions are used while defining the entropy mentioned above, which have inherent problems. It is observed that $-\log p_i = \infty \ with \ p_i = 0$, where p is the probability of occurrence of a pixel. This may lead to a reduced accuracy. On the contrary, the exponential entropy has certain merits over the Shannon related entropy functions used in MTH applications. To figure out — (i) it is always non-negative, (ii) its value at a given probability is higher than the Shannon entropy, (iii) the function is concave, (iv) well suited for MTH applications. This is the motivation behind the development of our proposal. In this context, we suggest novel differential exponential entropy (DEE) -based MTH method. A first hand objective function is proposed.

Now days, for reducing time and complexity of the image segmentation algorithm, to compute the optimal thresholds, various kinds of optimization techniques, i.e. Metaheuristic algorithms (MA) are used. The MAs maximize or minimize an objective function for getting the optimal solutions. The MAs are well-known for solving exhaustive search optimization problems, which cannot be solved using a deterministic method in a reasonable amount of time. There are two features of MA such as the exploration and exploitation. The exploration is the ability to search the globally associated solutions with an easy escape from the local optima. The exploitation is the ability to search the local best solutions for increasing their quality (Oliva, 2019).

It is noteworthy to mention here that Equilibrium Optimizer(EO) (Faramarzi et al., 2020) is a physics-based MA and Cuckoo Search Algorithm (CSA) (Yang and Deb, 2009) is a nature inspired MA. Generally, EO possesses both exploration and exploitation ability, whereas CSA is more efficient in exploration due to the use of Lévy flights concept. The cuckoo search optimizer has very few parameters to be tuned as compared to other metaheuristic algorithms, which results in a faster convergence. These features motivated the authors to use cuckoo search optimizer for the problem on hand. It is known that the optimization technique which has better exploration ability gives optimal solution. This has motivated us to propose a new optimization technique called Equilibrium-Cuckoo Search Optimizer (ECSO) by integrating the features of EO with the CSA. The proposed optimizer is applied to MTH-based colour satellite image segmentation problem by maximizing the objective functions such as: (i) the proposed, (ii) Otsu's between class variance, (iii) Kapur's entropy. The results are compared with the EO and the CSA for the performance evaluation (validation) of the ECSO. In summary, the main contributions of this article are manifold. To figure out, (i) a novel objective function named DEE is introduced, which is an original idea and may be very useful for solving different real world engineering problems; (ii) a new optimizer called ECSO is suggested, which may be useful for solving the optimization problems. Although, the EO and CS are reported in the literature, their combined approach seems to offer better performances in connection with the optimization. Therefore, our contribution is better in terms of the convergence and the accuracy; (iii) the experimental results on colour satellite image segmentation using the artificial intelligence (AI) may enrich the subject field of the image processing (Protasov and Khan, 2021).

The rest of the Sections are as follows. The literature survey of our work is presented in Section 2. Section 3 presents the preliminaries. Our newly proposed Equilibrium-Cuckoo Search Optimizer is presented in Section 4. Our suggested MTH technique is described in Section 5. Results and discussions are highlighted in Section 6. The conclusion of our work is drawn in Section 7.

2. Related work

Yang and Deb (2009) proposed a nature inspired Cuckoo Search Algorithm (CSA) via *Lévy* flight based on the cuckoo bird's brood parasitism. Interestingly, Cuckoo birds place their eggs in horde nests. When the horde bird ascertains eggs with a probability of p_a, it either discards the eggs or builds a new nest. Suresh and Lal (2016) have proposed an efficient CSA for MTH for satellite images by using Otsu between class variance, Kapur's entropy and Tsallis entropy as 3 different objective functions. For testing, the results are compared with PSO, Darwinian PSO, Artificial Bee Colony (ABC) and CSA. Otsu's method has shown better performance compared to Kapur's method and Tsallis method using *McCulloch's* scheme. Tsallis entropy-based multilevel thresholding by using the CSA is introduced by Agrawal et al. (2013) to perform image segmentation. From these papers, it is seen that CSA can be used in image segmentation application efficiently.

The authors in Bhandari et al. (2015) have proposed the CSA based MTH of colour satellite images by maximization of Tsallis entropy. Bhandari et al. (2016) have proposed the nature inspired algorithm based MTH of colour images by maximization of Otsu's method and Kapur's entropy in which Otsu's method outperforms better than Kapur's entropy. Jia et al. (2019) proposed Multi-Strategy Emperor Penguin Optimizer (MSEPO) based MTH of colour satellite image by maximization of Masi entropy in which Masi entropy based MSEPO has shown better results compared to Kapur's and Tsallis entropy based MSEPO. Another technique is proposed, that is, hybrid Harris Hawks Optimizer-Differential Evolution (HHO-DE) based MTH of colour images by maximization of 2 fitness functions namely Otsu's method and Kapur's entropy in which HHO-DE has shown better results compared to the HHO and DE (Bao et al., 2019). Shubham and Bhandari (2019) have proposed MTH of colour images by maximizing Masi entropy, which has shown better results compared to other techniques.

Faramarzi et al. (2020) proposed a novel physics-based optimization algorithm, that is, Equilibrium Optimizer (EO) which is inspired by estimation of both dynamic and equilibrium states using control volume mass balance models. A well-defined "generation rate" term has been shown to improve EO's ability in the exploration, exploitation while avoiding the local minima. EO balances both the exploration and exploitation ability in a better way. Wunnava et al. (2020) proposed a new Adaptive Equilibrium Optimizer (AEO) for MTH-based image segmentation. Interdependencies between the regions are minimized using the AEO. For experiment, images of the BSDS500 dataset were taken. For testing the effectiveness of the AEO, it is compared with different types of algorithms. For a quantitative analysis, popular IQAs namely PSNR, SSIM and FSIM are taken. They have claimed its future uses in engineering applications like soft computing techniques, which is the subfield of AI. Abdel-Basset et al. (2020) proposed MTH-based image segmentation by using the EO to find the optimal threshold values. Here Kapur's Entropy is used as an objective function. For testing the efficiency of EO, it is compared with different types of popular optimization algorithms. They have shown how to use the EO in MTH-based image segmentation.

An efficient Krill herd algorithm (EKH) was proposed by He and Huang (2020) based on the MTH of the colour images by maximizing 3 objective functions such as: Otsu's method, Kapur's entropy and Tsallis entropy. EKH shows good results than Krill Herd (KH) by maximizing Kapur's entropy. Naik et al. (2020) suggested a Normalized Square Difference (NSD) based MTH of multispectral colour satellite images by using the Leader Slime Mould Algorithm (LSMA), which outperforms other algorithms.

Naik et al. (2021) proposed the Opposition Equilibrium Optimizer (OEO) for MTH of colour satellite images by using a new objective function coined as Context-Sensitive Entropy Dependency (CSED). Their proposed CSED-OEO method outperforms other existing optimization algorithms. Rahaman and Sing (2021) proposed MTH of colour satellite image using adaptive cuckoo search (ACS) method by using 2 fitness functions namely Otsu and Tsallis entropy. They have compared with the CSA and *McCulloch's* method for *Lévy* flight. The ACS algorithm gives better results compared to the *McCulloch's* method. Anitha et al. (2021) proposed modified Whale Optimization Algorithm (MWOA) based MTH for colour images. Otsu's and Kapur's schemes are utilized as 2 different fitness functions. Maximizing Otsu's method based MTH using MWOA yields better performance than Kapur's method.

3. Preliminaries

3.1. Equilibrium optimizer (EO)

A new physics-based "Equilibrium Optimizer" is suggested by Faramarzi et al. (2020). The inspiration for the Equilibrium Optimizer came from a dynamic mass balance on a control volume. They have suggested a mass balance equation for approximating both the dynamic and the equilibrium states. More details on the concentration function is found in Faramarzi et al. (2020). The EO, like most MA, starts the optimization process with the initial population. The initial concentrations are calculated using the particles and dimensions in the search space, with uniform random initialization as follows:

$$\vec{C}_i^{initial}\,(iter = 1) = LB + rand_i\,(1, D) \times (UB - LB)\,, \quad i = 1, 2, \ldots, N \tag{1}$$

Here, $UB\ and\ LB$ represent upper bound and lower bound, respectively, D denotes the dimension of the problem, $rand_i$ is the random vector of ith search agent in the interval of 0 and 1, N is represented as the number of search agents.

Then the ith search agent's position for a control volume V in the EO is updated as:

$$\begin{aligned}\vec{C}_i\,(new) \ =\ & \vec{C}_{eq}\,(iter) + \left(\vec{C}_i\,(iter) - \vec{C}_{eq}\,(iter)\right) \times \vec{F}_i\,(iter) + \frac{\vec{G}_i\,(iter)}{\vec{\lambda}_i\,(iter) \times V} \\ & \times\ (1 - \vec{F}_i\,(iter))\end{aligned} \tag{2}$$

Here, $V\ =\ 1$, $\vec{C}_{eq}$ is an equilibrium candidate that is selected at random from an equilibrium pool $\vec{C}_{eq,pool}$ of four best search agents such as: $\vec{C}_{eq(1)}$, $\vec{C}_{eq(2)}$, $\vec{C}_{eq(3)}$, $\vec{C}_{eq(4)}$. Note that $\vec{C}_{eq(ave)}$ is the average of these four best search agents. Then $\vec{C}_{eq(1)}$, $\vec{C}_{eq(2)}$, $\vec{C}_{eq(3)}$, $\vec{C}_{eq(4)}$ are evaluated by using their fitness values $fit\left(\vec{C}_{eq(1)}\right)$, $fit\left(\vec{C}_{eq(2)}\right)$, $fit\left(\vec{C}_{eq(3)}\right)$ and $fit\left(\vec{C}_{eq(4)}\right)$. The search agents' fitness values are:

$$fit = \left(fit_1, fit_2, \ldots, fit_N\right) \tag{3}$$

The following values are listed in an ascending order:

$$[sorted\_fit, sort\_index] = sort\,(fit) \tag{4}$$

Here, the objective function values of the equilibrium candidates are defined by:

$$\begin{aligned} &fit\left(\vec{C}_{eq(1)}\right) = sorted\_fit(1) \text{ and } \vec{C}_{eq(1)} = \vec{C}\,(sort\_index(1)) \\ &fit\left(\vec{C}_{eq(2)}\right) = sorted\_fit(2) \text{ and } \vec{C}_{eq(2)} = \vec{C}\,(sort\_index(2)) \\ &fit\left(\vec{C}_{eq(3)}\right) = sorted\_fit(3) \text{ and } \vec{C}_{eq(3)} = \vec{C}\,(sort\_index(3)) \\ &fit\left(\vec{C}_{eq(4)}\right) = sorted\_fit(4) \text{ and } \vec{C}_{eq(4)} = \vec{C}\,(sort\_index(4)) \end{aligned} \tag{5}$$

The averaged candidate $\vec{C}_{eq(ave)}$ is calculated as follows:

$$\vec{C}_{eq(ave)} = \frac{\vec{C}_{eq(1)} + \vec{C}_{eq(2)} + \vec{C}_{eq(3)} + \vec{C}_{eq(4)}}{4} \tag{6}$$

Then the equilibrium pool can be expressed as:

$$\vec{C}_{eq,pool} = \left\{\vec{C}_{eq(1)}, \vec{C}_{eq(2)}, \vec{C}_{eq(3)}, \vec{C}_{eq(4)}, \vec{C}_{eq(ave)}\right\} \tag{7}$$

The exponential term $\vec{F}_i\,(iter)$ is a term that contributes to the key concentration updating rule and assist the EO for the exploration and exploitation ability for ith search agent in $iter$ iteration is formulated as:

$$\vec{F}_i\,(iter) = a_1 sign\left(\vec{r}_i\,(iter) - 0.5\right)\left[e^{-\vec{\lambda}_i(iter)\times t} - 1\right] \tag{8}$$

Here, t is represented as the time which is described as the function of iteration $(iter)$ and t is decreased with a decrement of the iteration $(iter)$ using the following formula:

$$t = (1 - \frac{iter}{\max\_iter})^{(a_2 \times \frac{iter}{\max\_iter})} \tag{9}$$

Here, a_1 and a_2 are utilized to control both exploration and exploitation abilities, respectively. The $sign$ is used for controlling direction of the search, which depends on a random vector $\vec{r}_i\,(iter)$ for the ith search agent in $iter$ iteration in the interval $[0, 1]$. Note that $iter$ denotes current iteration while $\max\_iter$ represents max number of iteration. Here, $\vec{\lambda}_i\,(iter)$ is the random vector in the interval $[0, 1]$. This is used for the ith search agent.

The generation term $\vec{G}_i\,(iter)$ is the most important term in the EO for getting exact solution by improving the exploitation phase for the ith search agent in $iter$ iteration and it is formulated as:

$$\vec{G}_i\,(iter) = \vec{G}_{0_i}\,(iter) \times \vec{F}_i\,(iter) \tag{10}$$

$\vec{G}_{0_i}\,(iter)$ and $G\vec{C}P_i\,(iter)$ are formulated as:

$$\vec{G}_{0_i}\,(iter) = G\vec{C}P_i\,(iter) \times (\vec{C}_{eq}\,(iter) - \left(\vec{\lambda}_i\,(iter) \times \vec{C}_i\,(iter)\right)) \tag{11}$$

$$G\vec{C}P_i\,(iter) = \begin{cases} 0.5r_1 & r_2 \geq GP \\ 0 & r_2 < GP \end{cases} \tag{12}$$

Here, $G\vec{C}P_i\,(iter)$ is the Generation rate Control Parameter for the ith search agent in $iter$ iteration, GP is Generation Probability. Note that r_1 and r_2 are the random variables in the interval $[0, 1]$. These are the overall steps of the EO for updating the concentration.

3.2. Cuckoo search algorithm (CSA)

The nature inspired Cuckoo Search Algorithm was proposed by Yang and Deb (2009). The cuckoo bird's aggressive reproductive technique is the inspiration for CS. During reproduction, these birds exhibit a sophisticated behaviour in which they delicately place eggs in other host birds' nest. The intelligence is ascribed to the female cuckoo's imitating of egg's colour/shape, as well as the time of egg-laying, with other horde birds. Three idealized laws underpin CSA as follow:

- Each cuckoo lay single egg at a time and deposits it in a nest that is chosen at random;
- The best nests with the best quality of eggs would be passed on to the next generation;
- The number of total host nests is set, and the host bird discovers the cuckoo's egg with a certain probability $p_a \in [0, 1]$. Surprisingly, the host bird has the choice of either throwing the egg or immediately surrendering the nest. New solutions are used to replace the number of nests "N" and "p_a".

The algorithm's goal is to find the best-egg-laying nest. The finest nests are the Cuckoo bird's fresh eggs. The objective function with the highest value is taken into consideration, others are eliminated. To explore the search space, they employed Levy steps and Mantegna's scheme. The new solution is formulated as:

$$X_i^{(t+1)} = X_i^{(t)} + \alpha \oplus Lévy(\lambda) \tag{13}$$

The idea of the levy flight is quite interesting and is defined here.

$$Lévy \sim u = t^{-\lambda}, (1 < \lambda \leq 3) \tag{14}$$

Note that 'u' is a variable while 'λ' is a constant, $X_i^{(t)}$ is the current search space at time t and represented as $X_i^{(t)} = x_{i_1}, \dots .x_{i_d} \dots .x_{i_n}$ and $\alpha > 0$ (the value of 1 is mostly chosen), $\oplus$ is the entry-wise multiplication while $Lévy(\lambda)$ is the random work through $Lévy$ flight. Here, t is represented by the number for a recent group $t = \left(1, 2, 3, \dots, t_{max}\right)$ and t_{max} is the pre-determined extreme cohort position. These are the overall process of the CSA.

Gonzalez et al. (2016) proposed optimization of the interval type-2 fuzzy logic using the CS algorithm. Angelova et al. (2018) have presented an efficient method for parameter identification using the CS optimizer. The functioning of the CS algorithm is well explained in Roeva et al. (2020). The authors have used the generalized nets to describe the functioning of the CS optimizer. The fuzzy inference systems are optimized using the CS optimizer. Recently, Guerrero-Luis et al. (2021) presented a review of the CSA. More details on the CS optimizer are found in this article. Readers can also refer the book published by the Springer Nature (Castillo, 2021), which highlights the features of the CS algorithm.

3.3. Otsu's between class variance method

The between class variance method, a non-parametric scheme, which is used to determine the optimal threshold values, was first developed by Otsu (1979) for both BTH and MTH method. This technique aims for maximizing between class (inter-class) variance while minimizing within-class (intra-class) variance measure between the pixel in each class. To segment the image, it uses the maximum variance value of the various classes as a criterion. The maximum variance is calculated when BTH method or MTH method segment an image into several classes. In this technique, the histogram of an image is the input. Note that the probability distribution of the grey level values is determined using the L intensity levels from an image intensity or each component of an image as follow:

$$Ph_i^c = \frac{h_i^c}{NP}, Ph_i^c \geq 0, \quad \sum_{i=1}^{L} Ph_i^c = 1,$$
$$c = \begin{cases} 1, 2, 3, & if\ RGB\ image \\ 1, & if\ Grayscale\ Image \end{cases} \tag{15}$$

Here, i represents a specific intensity level $(0 \leq i \leq L - 1)$. c is the image component that determines the image is Greyscale or RGB. The total number of pixels in the image is denoted by NP. The number of pixels that relate to the i intensity level in c is denoted as h_i^c (histogram). Inside a probability distribution Ph_i^c, the histogram is normalized.

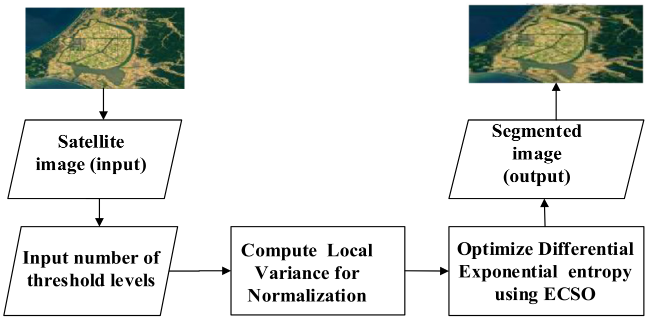

Satellite image (input)
Input number of threshold levels
Compute Local Variance for Normalization
Optimize Differential Exponential entropy using ECSO
Segmented image (output)

Fig. 1. Block diagram of the suggested method.

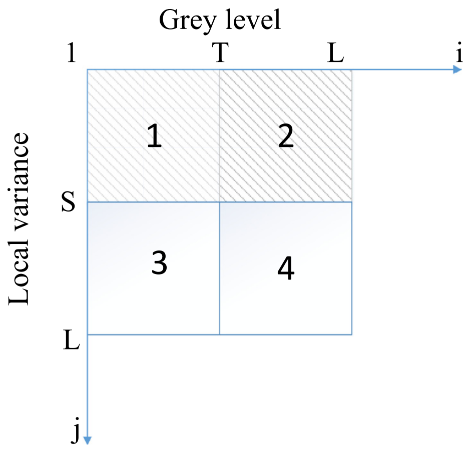


Fig. 2. 2D histogram for bi-level threshold selection.

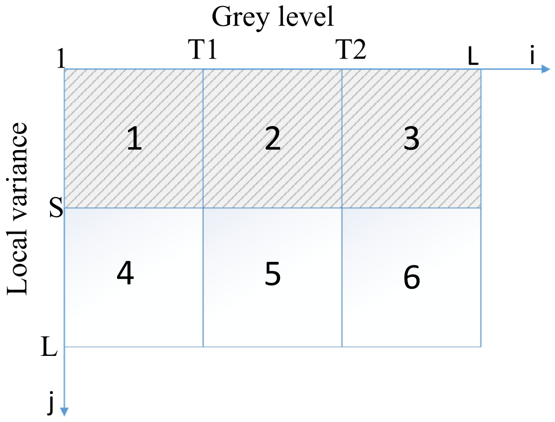


Fig. 3. 2D histogram for multilevel threshold selection.

For the BTH-based image segmentation method, the two classes are formulated as:

$$C_1 = \frac{Ph_1^c}{\omega_0^c(th)}, \dots, \frac{Ph_{th}^c}{\omega_0^c(th)}, \qquad C_2 = \frac{Ph_{th+1}^c}{\omega_1^c(th)}, \dots, \frac{Ph_L^c}{\omega_1^c(th)} \tag{16}$$

Here, C_1 and C_2 are two classes and $\omega_0(th)$ and $\omega_1(th)$ are probability distributions of these 2 classes respectively.

$$\omega_0^c(th) = \sum_{i=1}^{th} Ph_i^c, \qquad \omega_1^c(th) = \sum_{i=th+1}^{L} Ph_i^c \tag{17}$$

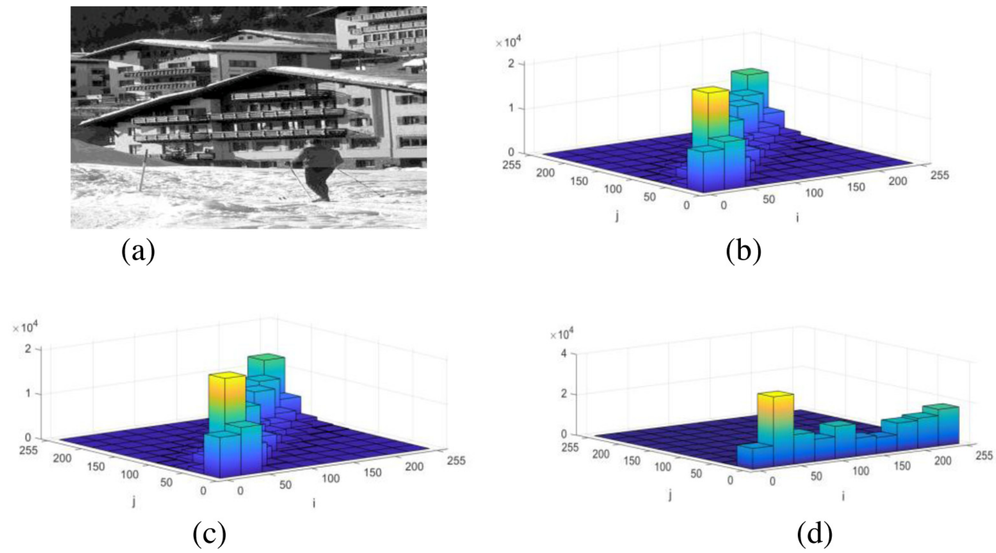


Fig. 4. Demonstration of 2D histogram (a) '#61086' of BSDS500 data set (b) Otsu technique (c) Kapur's technique (d) Proposed technique.

The mean levels μ_0^c and μ_1^c and the Otsu variance between classes σ_B^{2c} is determined as:

$$\mu_0^c = \sum_{i=1}^{th} \frac{iPh_i^c}{\omega_0^c(th)}, \qquad \mu_1^c = \sum_{i=th+1}^{L} \frac{iPh_i^c}{\omega_1^c(th)} \tag{18}$$

$$\sigma_B^{2c} = \sigma_1^{2c} + \sigma_2^{2c} \tag{19}$$

The type of the image is depended by c. Furthermore, σ_1^{2c} and σ_2^{2c} are the variance of C_1 and C_2 are formulated as follow:

$$\sigma_1^{2c} = \omega_0^c\left(\mu_0^c - \mu_T^c\right)^2, \quad \sigma_2^{2c} = \omega_1^c\left(\mu_1^c - \mu_T^c\right)^2 \tag{20}$$

Here, $\mu_T^c = \omega_0^c\mu_0^c + \omega_1^c\mu_1^c$ and $\omega_0^c + \omega_1^c = 1$. For BTH-based image segmentation technique the objective function is formulated as:

$$f_{Otsu}(th) = max\left(\sigma_B^{2c}(th)\right), \quad 0 \le th \le L-1,$$
$$c = \begin{cases} 1,2,3, & if\ RGB\ image \\ 1, & if\ Grayscale\ Image \end{cases} \tag{21}$$

For MTH-based image segmentation technique, the objective function is formulated as:

$$f_{Otsu}(TH) = max\left(\sigma_B^{2c}(TH)\right), \quad 0 \le th_i \le L-1,$$
$$i = 1,2,\ldots,k, \quad c = \begin{cases} 1,2,3, & if\ RGB\ image \\ 1, & if\ Grayscale\ Image \end{cases} \tag{22}$$

Here, $TH = \left[th_1, th_2, th_3 \cdots\cdots, th_{k-1}\right]$ is a vector which contains multiple threshold values. The variance is formulated as:

$$\sigma_B^{2c} = \sum_{i=1}^{k} \sigma_i^{2c} = \sum_{i=1}^{k} \omega_i^c\left(\mu_i^c - \mu_T^c\right)^2 \tag{23}$$

Here, i denotes a special class. The mean of a class and the probability of occurrence, respectively, are μ_i^c and ω_i^c. These values are computed as follows in the case of MTH:

The probability of occurrence values is expressed as:

$$\omega_0^c(th) = \sum_{i=1}^{th_1} Ph_i^c, \quad \omega_1^c(th) = \sum_{i=th_1+1}^{th_2} Ph_i^c, \ldots, \quad \omega_{k-1}^c(th) = \sum_{i=th_k+1}^{L} Ph_i^c \tag{24}$$

And the mean values are expressed as:

$$\mu_0^c = \sum_{i=1}^{th_1} \frac{iPh_i^c}{\omega_0^c(th_1)}, \quad \mu_1^c = \sum_{i=th_1+1}^{th_2} \frac{iPh_i^c}{\omega_1^c(th_2)}, \ldots, \quad \mu_{k-1}^c = \sum_{i=th_k+1}^{L} \frac{iPh_i^c}{\omega_{k-1}^c(th_k)} \tag{25}$$

These are the overall steps of Otsu's between class variance method for both BTH-based image segmentation technique and MTH-based image segmentation technique.

3.4. Kapur's entropy based method

Another non-parametric segmentation technique, which is used to determine the optimum threshold values, was first developed by Kapur et al. (1985) for both BTH and MTH. This technique is totally reliant on the entropy and probability distribution of the image histogram. This method aims to get optimal threshold by maximizing overall entropy. For the BTH-based image segmentation method, the objective function of the Kapur's entropy can be formulated as:

$$f_{Kapur} = H_1^c + H_2^c, \qquad c = \begin{cases} 1,2,3, & if\ RGB\ image \\ 1, & if\ Grayscale\ Image \end{cases} \tag{26}$$

where, H_1 and H_2 are two entropies, which are formulated as follow:

$$H_1^c = \sum_{i=1}^{th} \frac{Ph_i^c}{\omega_0^c} \ln\left(\frac{Ph_i^c}{\omega_0^c}\right), \qquad H_2^c = \sum_{i=th+1}^{L} \frac{Ph_i^c}{\omega_1^c} \ln\left(\frac{Ph_i^c}{\omega_1^c}\right) \tag{27}$$

Here, Ph_i^c is the probability distribution of the pixel values, $\omega_0^c(th)$ and $\omega_1^c(th)$ represent the two-probability distribution for 2 classes C_1 and C_2 respectively. The natural logarithm is indicated by $\ln(\cdot)$. The entropy-based approach, like Otsu's method, can be generalized to multiple thresholds; in this case, the image must be divided into k groups using the same number of thresholds. For the MTH-based image segmentation method, the fitness of the Kapur's entropy is formulated as:

$$f_{Kapur}(TH) = \sum_{i=1}^{k} H_i^c, \qquad c = \begin{cases} 1,2,3, & if\ RGB\ image \\ 1, & if\ Grayscale\ Image \end{cases} \tag{28}$$

Here, $TH = \left[th_1, th_2, \cdots\cdots, th_{k-1}\right]$ is a vector in which k units of threshold values are indicated by it. For one channel images, the calculation of k level optimum thresholds can be done as a k -dimensional problem, and for three channel images, it can be done as a $3k$ -dimensional optimization problem. Each entropy is calculated separately with its own threshold value; thus, to calculate k entropies, Eq. (29) is expanded as follows:

$$H_1^c = \sum_{i=1}^{th_1} \frac{Ph_i^c}{\omega_0^c} \ln\left(\frac{Ph_i^c}{\omega_0^c}\right), \qquad H_2^c = \sum_{i=th_1+1}^{th_2} \frac{Ph_i^c}{\omega_1^c} \ln\left(\frac{Ph_i^c}{\omega_1^c}\right), \ldots,$$
$$H_k^c = \sum_{i=th_k+1}^{th_2} \frac{Ph_i^c}{\omega_{k-1}^c} \ln\left(\frac{Ph_i^c}{\omega_{k-1}^c}\right) \tag{29}$$

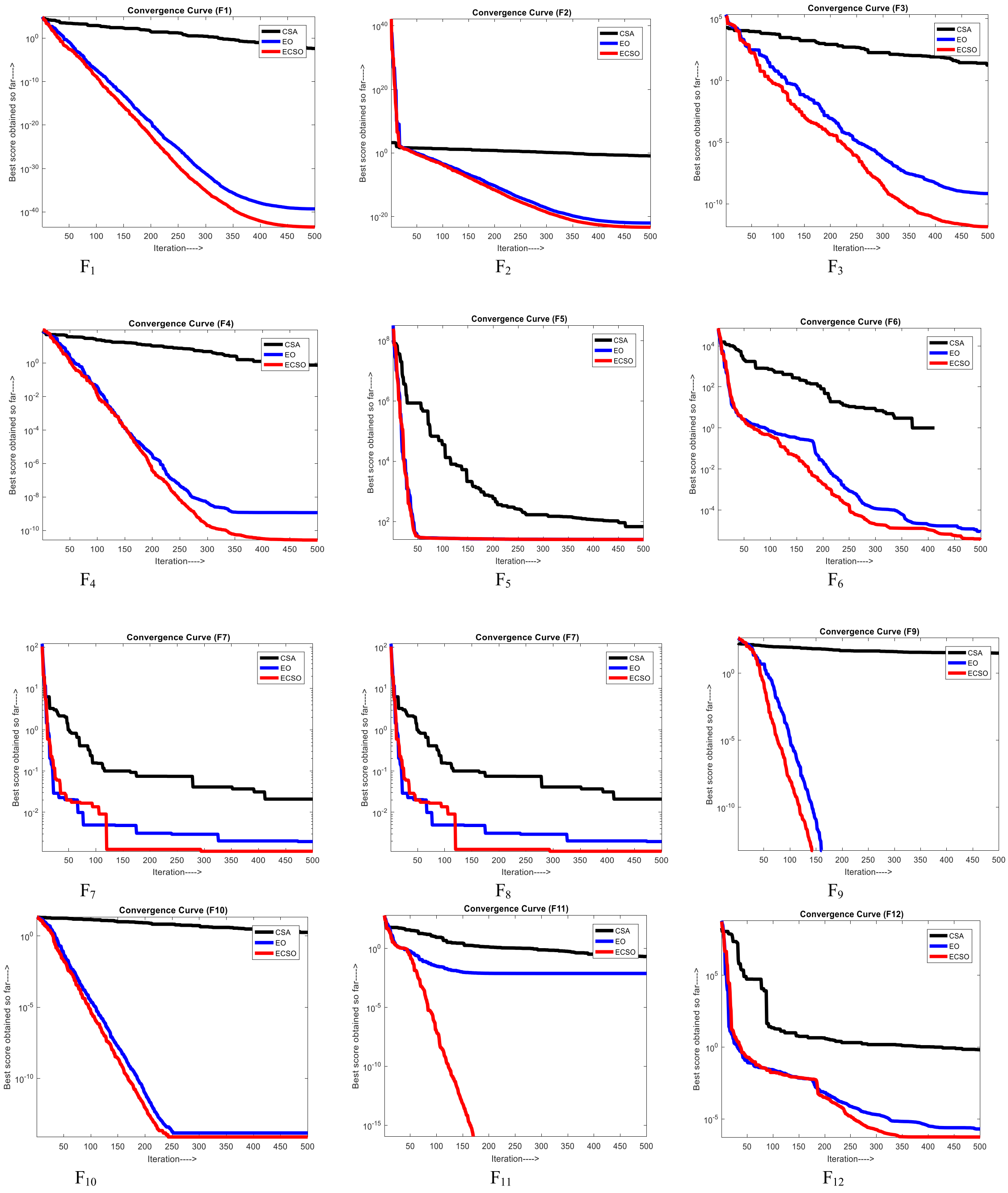


Fig. 5. The comparison of the convergence curves of the ECSO, EO and CSA.

where, Ph_i^c represents probability distribution, which can be obtained in Eq. (15) and $\left(\omega_0^c, \omega_1^c, \ldots\ldots, \omega_{k-1}^c\right)$ are represent the probability occurrence values for k units. Finally, Eq. (17) must be used to split the pixels into their respective classes. These are the overall steps of Kapur's entropy method for both BTH-based image segmentation technique and MTH-based image segmentation technique.

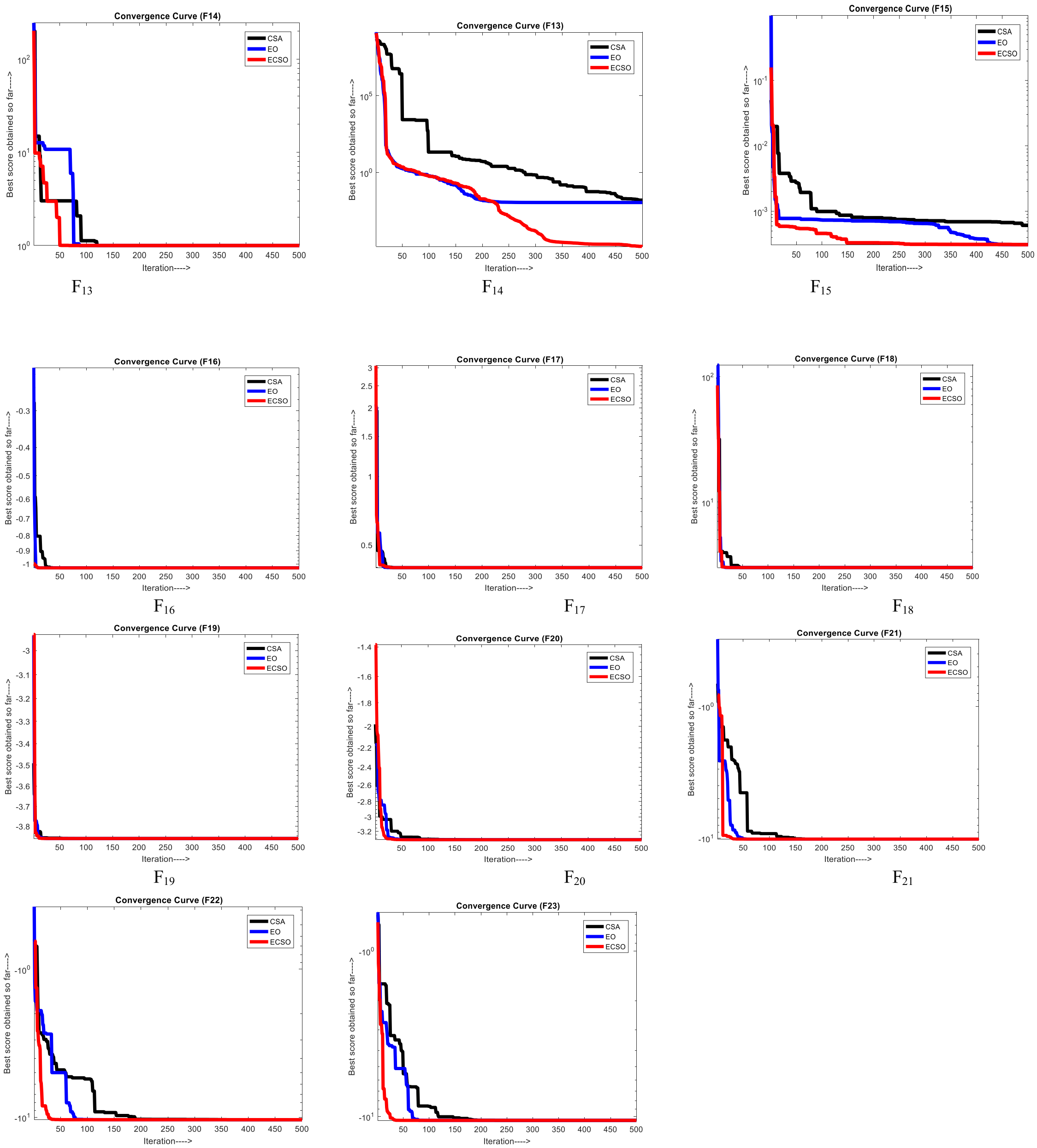
4. The proposed DEE based method

In this section, a novel DEE based MTH technique is proposed.

Fig. 1 illustrates the block diagram of our DEE method. Exponential entropy values are computed row wise from the 2D histogram. Thresholds are optimized to achieve the maximum entropy using the suggested optimizer. This could be made more accurate using the artificial intelligence (AI) coupled with the proposed ECSO. Most importantly, the image processing, computer vision, and Machine learning form an Artificial intelligence system. It is noteworthy to mention here that the image processing is the procedure of manipulating an image to either enhance the quality or extract relevant features from it. Recently, the AI image processing combines advanced algorithmic technology with machine learning and computer vision to process large volumes of data comfortably and speedily. In turn, it augments the machine intelligence in extracting optimal features that yields improvised accuracy and decision making. In this connection, our proposal may add strength to the field for image segmentation using the artificial intelligence.

Let I be the grey level image of size $M{\times}N$. Total of L grey levels are present in the image. Let $g = 1, 2, \ldots, \mathrm{L}$ be the pixel intensity value.

Let us assume that $g(x, y)$ is the grey level of the satellite image (I) with (x,y) as the coordinates. Let avg(x,y) be the local average grey



F13 F14 F15

F16 F17 F18

F19 F20 F21

F22 F23

Fig. 5. (*continued*).

value given by:

$$avg(x,y) = \left\lfloor \frac{1}{w\times w}\sum_{c=-l}^{l}\sum_{d=-l}^{l} f(x+c,y+d)\right\rfloor \tag{30}$$

where, $l = \left\lfloor \frac{w}{2} \right\rfloor$ with w as the window size. An odd number is often chosen for the window size.

In this proposal, an attempt is made to incorporate the differential values instead of directly taking the average pixel values, which is the case with the most of the existing methods.

The local variance $v(x, y)$ is computed by:

$$v(x,y) = (g(x,y) - avg(x,y))^2 \tag{31}$$

To suppress high magnitude peaks, it is normalized.

$$v_n(x,y) = \frac{(v(x,y) - v_{\min})\times K}{v_{\max} - v_{\min}} \tag{32}$$

where $v_{\max}$ and $v_{\min}$ are the max & min values of $v(x,y)$. The value of K is 256.

Here, $g(x,\ y) = i,\ v_n(x,\ y) = j$. Assume that the occurrence of the pair $(i,j) = q_{ij}$.

The probability of occurrence of *(i,j)* is

$$\text{The probability of occurrence of } (i,j) \text{ is } p_{ij} = \frac{q_{ij}}{M\times N} \; with\ 1 \le i,j \le L. \tag{33}$$

Fig. 2 displays the construction of 2D histogram to be used for bi-level threshold selection. When we consider only one threshold T, the histogram is partitioned into 4 quadrants. Quadrants 1 and 2 carry the edge (high frequency) info. In this Fig. 2, it is noteworthy to mention here that the 1st quadrant refers to the background (C_1) while the 2nd quadrant denotes to the foreground (C_2). Note that C_1 and C_2 are two different class assignments. Interestingly, in this construction, both the classes are found row wise, not diagonally.

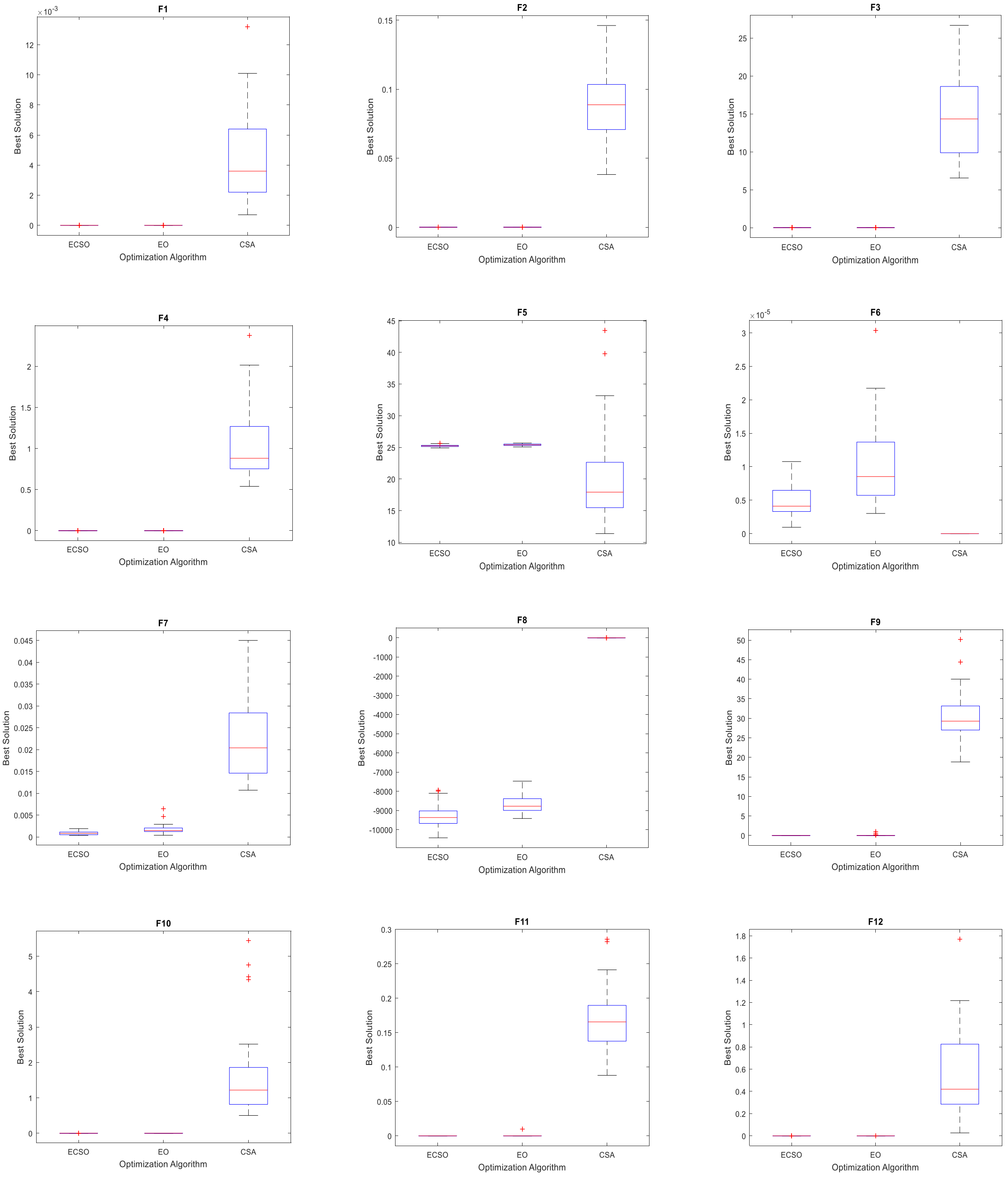


Fig. 6. Comparison of Boxplots of the ECSO with the EO and the CSA.

Probability distribution of class 1 is defined as:

$$P_1(C_1) = \sum_{i=1}^{S} \sum_{j=1}^{T} p_{ij}. \tag{34}$$

And probability distribution of class 2 is defined by:

$$P_2(C_2) = \sum_{i=1}^{S} \sum_{j=T+1}^{L} p_{ij}. \tag{35}$$

The class probabilities of $C_1 \& C_2$ are given as:

$$C_1 : \left\{ \frac{p_{ij}}{P_1}, i = 1, 2, \ldots, S; j = 1, 2, \ldots, T \right\} \text{ and}$$

$$C_2 : \left\{ \frac{p_{ij}}{P_2}, i = 1, 2, \ldots, S; j = T+1, \mathrm{T}+2, \ldots, \mathrm{L} \right\}$$

Proposed differential exponential entropy is defined by:

$$E_1(S,T) = -\sum_{i=1}^{S} \sum_{j=1}^{T} \left(\frac{p_{ij}}{P_1} \right) \exp\left(1 - \frac{p_{ij}}{P_1} \right) \tag{36}$$

and

$$E_2(S,T) = -\sum_{i=1}^{S} \sum_{j=T+1}^{L} \left(\frac{p_{ij}}{P_2} \right) \exp\left(1 - \frac{p_{ij}}{P_2} \right). \tag{37}$$

Using the additive property of the exponential entropy, we have:

$$E_{Total}(S,T) = E_1(S,T) + E_2(S,T) \tag{38}$$

Thus, our optimal fitness value is found through maximization as defined below:

$$(S_{opt}, T_{opt}) = \operatorname{argmax} \left\{ E_{Total}(S,T) \right\}. \tag{39}$$

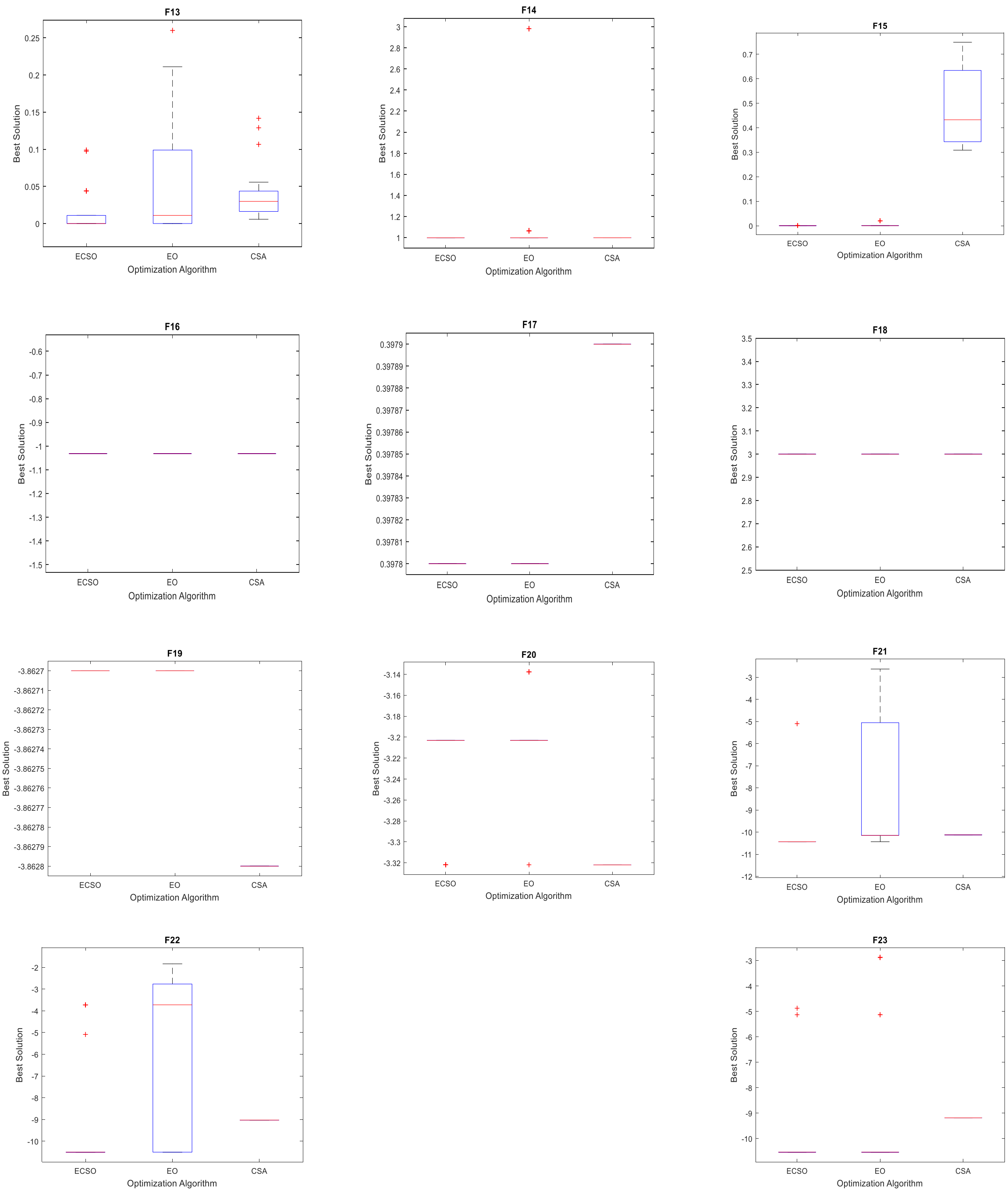


Fig. 6. (*continued*).

2D histogram for multilevel thresholding is displayed in Fig. 3. When we consider two threshold values, the histogram gets partitioned into six different regions. Interestingly, information is found along the first row. This facilitates a reduction of computations, because it is needed to compute an optimal value for ‘S’ only once. Whereas, the optimal values for T1 and T2 are to be computed two times separately.

Let us assume that there are ‘k’ classes. Then, the probability distribution of classes $C_1, C_2, \ldots, C_k$ are defined as:

$$P_1(C_1) = \sum_{i=1}^{S}\sum_{j=1}^{T_1} p_{ij}$$

$$P_2(C_2) = \sum_{i=1}^{S}\sum_{j=T_1+1}^{T_2} p_{ij}$$

$$\ldots$$

$$P_k(C_k) = \sum_{i=1}^{S}\sum_{j=T_{k-1}+1}^{L} p_{ij} \tag{40}$$

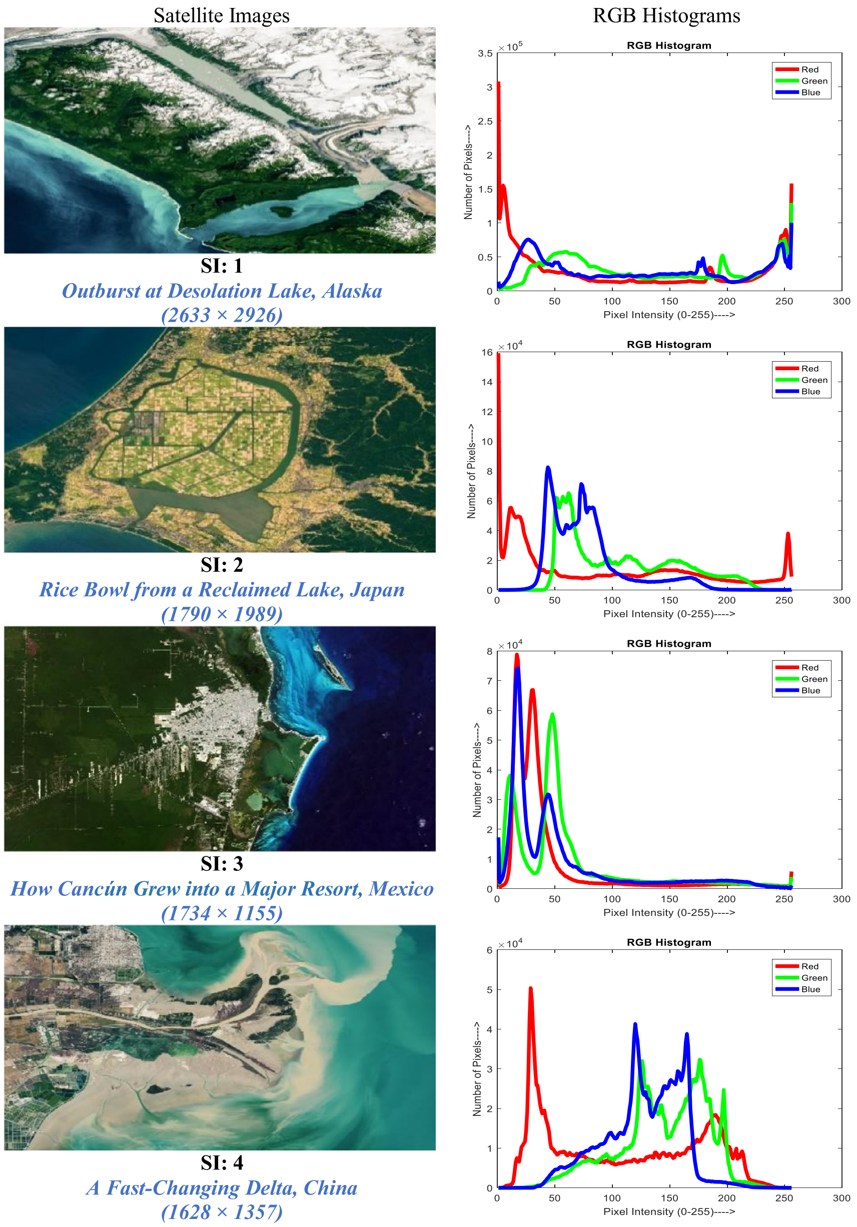
The exponential entropies are given as:

$$E_1(S,T_1) = -\sum_{i=1}^{S}\sum_{j=1}^{T_1}\left(\frac{p_{ij}}{P_1}\right)\exp\left(1-\frac{p_{ij}}{P_1}\right)$$

$$E_2(S,T_2) = -\sum_{i=1}^{S}\sum_{j=T_1+1}^{T_2}\left(\frac{p_{ij}}{P_2}\right)\exp\left(1-\frac{p_{ij}}{P_2}\right)$$

$$\ldots$$

$$E_k(S,T_{k-1}) = -\sum_{i=1}^{S}\sum_{j=T_{k-1}+1}^{L}\left(\frac{p_{ij}}{P_k}\right)\exp\left(1-\frac{p_{ij}}{P_k}\right) \tag{41}$$


Satellite Images

RGB Histograms

SI: 1

Outburst at Desolation Lake, Alaska
(2633 × 2926)

SI: 2

Rice Bowl from a Reclaimed Lake, Japan
(1790 × 1989)

SI: 3

How Cancún Grew into a Major Resort, Mexico
(1734 × 1155)

SI: 4

A Fast-Changing Delta, China
(1628 × 1357)

Fig. 7. Benchmark colour satellite images with the corresponding RGB histograms. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 1
Description of the Unimodal benchmark functions.

Functions	VN	Range	f_{min}
$F_1(x) = \sum_{i=1}^{n} x_i^2$	30	[−100, 100]	0
$F_2(x) = \sum_{i=1}^{n} \lvert x_i\rvert + \prod_{i=1}^{n} \lvert x_i\rvert$	30	[−10, 10]	0
$F_3(x) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j\right)^2$	30	[−100, 100]	0
$F_4(x) = max_i \left\{\lvert x_i\rvert, 1 \le i \le n\right\}$	30	[−100, 100]	0
$F_5(x) = \sum_{i=1}^{n-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + \left(x_i - 1\right)^2\right]$	30	[−30, 30]	0
$F_6(x) = \sum_{i=1}^{n} \left(\lvert x_i + 0.5\rvert\right)^2$	30	[−100, 100]	0
$F_7(x) = \sum_{i=1}^{n} ix_i^4 + random\,[0.1)$	30	[−1.28, 1.28]	0

On adding, the total exponential Entropy is written by:

$$E_{Total}(ST_1, ST_2, \dots, ST_{k-1}) = E_1(S, T_1) + E_2(S, T_2) + \dots + E_k(S, T_{k-1}) \quad (42)$$

Maximizing Eq. (42), the objective function is derived:

$$\left(S_{opt}T_{opt_1}, S_{opt}T_{opt_2}, \dots, S_{opt}T_{opt_{k-1}}\right) = \underset{1 \le ST_i \le k-1}{\operatorname{argmax}} \left\{E_{Total}(ST_1, ST_2, \dots, ST_{k-1})\right\} \quad (43)$$

Fig. 4 displays 2D histogram for three different methods. The original grey image is from BSDS 500, which is shown in Fig. 4(a). 2D histogram utilizing Otsu scheme is shown in Fig. 4(b). 2D histogram using Kapur and the proposed methods are displayed in Fig. 4(c) and (d), respectively. From Fig. 4(b) and (c), it is seen that the pixels from the diagonal regions are taken into consideration of the computation of the entropy both for Otsu and Kapur's methods. Hence, for a 2D setting, the time complexity is of the order of $O(L^{2M})$, where L is the total

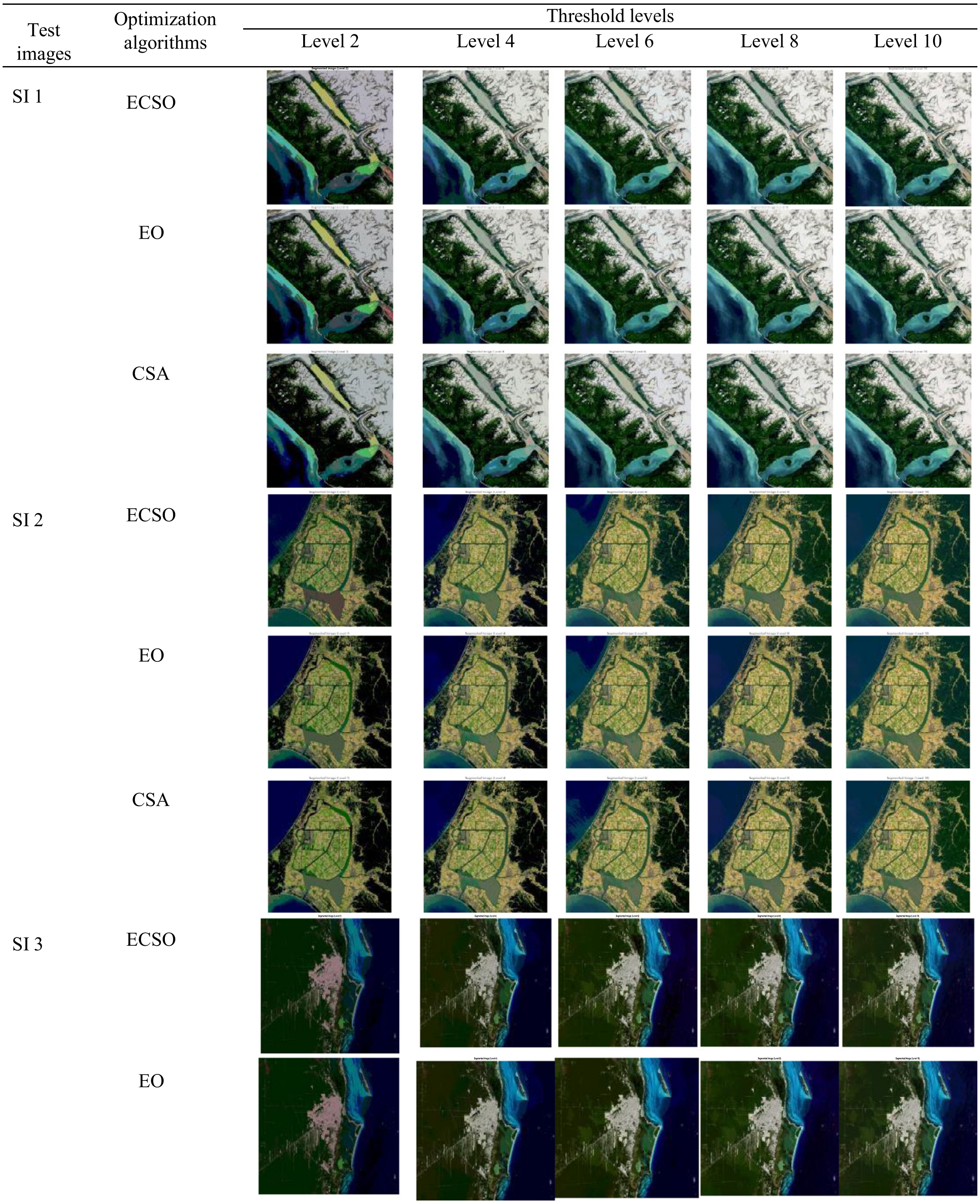


Fig. 8. Segmented satellite images using the proposed technique.

number of the grey level and M is the number of multilevel. Nonetheless, the entropic methods based on 2D histograms take more time for computation of the entropy. On the other hand, it is observed from Fig. 4(d) that the pixels found in a row are taken into consideration for the computation of the entropy while considering our approach. Interestingly, the computational complexity is reduced to $O(L^{M+1})$. This is also seen in Fig. 3. The optimal values for 'S' is computed only once. One may recall that the computational complexity for 1D histogram based methods is $O(L^{M})$. In our case, it is simply 'L' times the complexity of 1D method. Therefore, the proposed 2D construction idea is time efficient and, thus, useful for multilevel threshold selection.

It is noteworthy to mentions here that Eq. (43) is the objective function which is used for obtaining the optimal thresholds.

5. The proposed equilibrium-Cuckoo Search Optimizer (ECSO)

Artificial Intelligence (AI) aims to make machines intelligent. A computing model can be evolved to solve issues where human intelligence is desired. Such a computing model can easily handle the tolerance levels like — uncertainty, imprecision, partly available information, etc. The primary goal of the AI is the presence of knowledge to solve various real world engineering problems. In this connection, an effort is made here to suggest a new computing model called ECSO for solving the optimization problems. The EO algorithm is designed with a high level of exploratory and exploitative search mechanism to adjust the solution at random. The equilibrium pool plays a vital role for balancing both exploration and exploitation ability for finding optimal

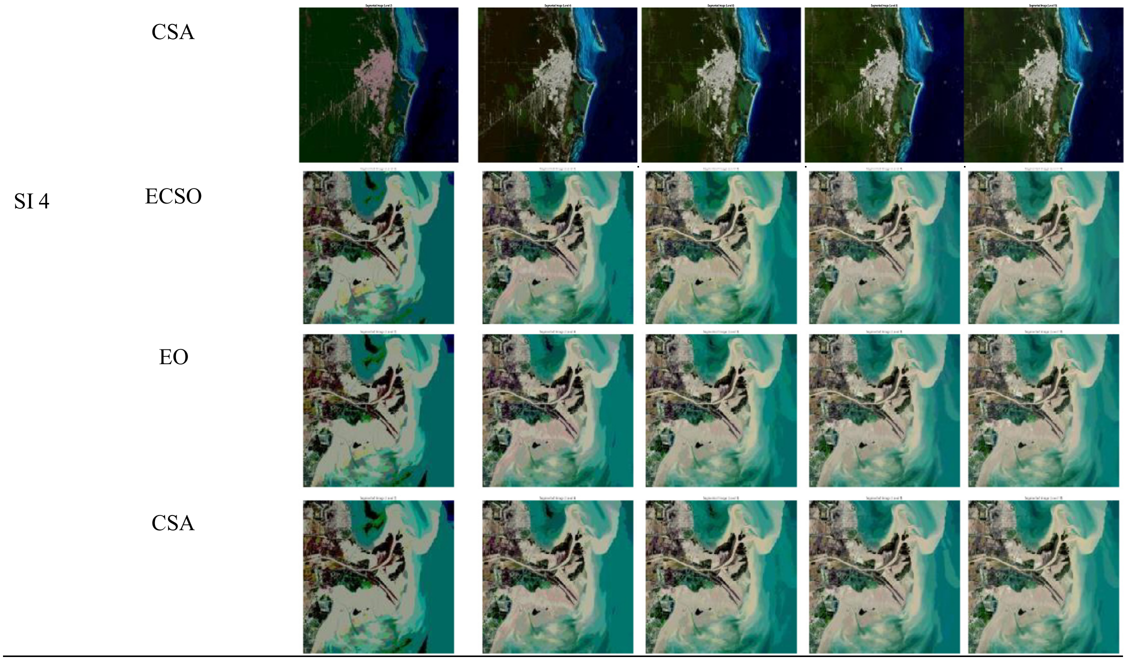


Fig. 8. (*continued*).

Table 2
Description of the Scalable Multimodal benchmark functions.

Functions	VN	Range	f_{min}
$F_8(x) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	30	$[-500, 500]$	-418.9829×5
$F_9(x) = \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$	30	$[-5.12, 5.12]$	0
$F_{10}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	30	$[-32, 32]$	0
$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600, 600]$	0
$F_{12}(x) = \frac{\pi}{n}\left\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2\left[1 + 10\sin^2(\pi y_{i+1})\right]\right.$ $\left. + (y_n - 1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4} u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m\ x_i > a \\ 0 - a < x_i < a \\ k(-x_i - a)^m\ x_i < -a \end{cases}$	30	$[-50, 50]$	0
$F_{13}(x) = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2\left[1 + \sin^2(3\pi x_i + 1)\right]\right.$ $\left. + (x_n - 1)^2\left[1 + \sin^2(2\pi x_n)\right]\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	$[-50, 50]$	0

solutions. In this work, the CSA is used to update the concentration of the best four candidates in the equilibrium pool. The CSA uses the random walk and Lévy flights to change the location of the nest. The search route may be long or short, with nearly equal probabilities, but the direction is highly random. As a result, it quickly transitions from one region to another. Also due to the Lévy flights concept, CSA is more efficient in exploration ability. Here, the CSA is integrated with the EO algorithm for balancing exploration and exploitation ability in a better way. It finds better optimal solutions, according to this feature of the CSA. Especially, Eq. (13), as discussed in the CSA Section 3.2, is utilized here to update the concentration candidates. The Pseudo code for the newly proposed ECSO is given below:

5.1. Pseudo code of the ECSO

Pseudo code of the ECSO is given in next page.

5.2. The performance evaluation of the ECSO

5.2.1. The test functions and the experimental setup

The efficiency of the ECSO, EO, and CSA, evolved in this study, is tested by formative 23 classic mathematical optimization problems. Generally, the classical benchmark functions can be classified into 3 groups, such as: Unimodal, Scalable Multimodal and Fixed-Dimensions Multimodal. For testing the exploitation ability, unimodal functions are

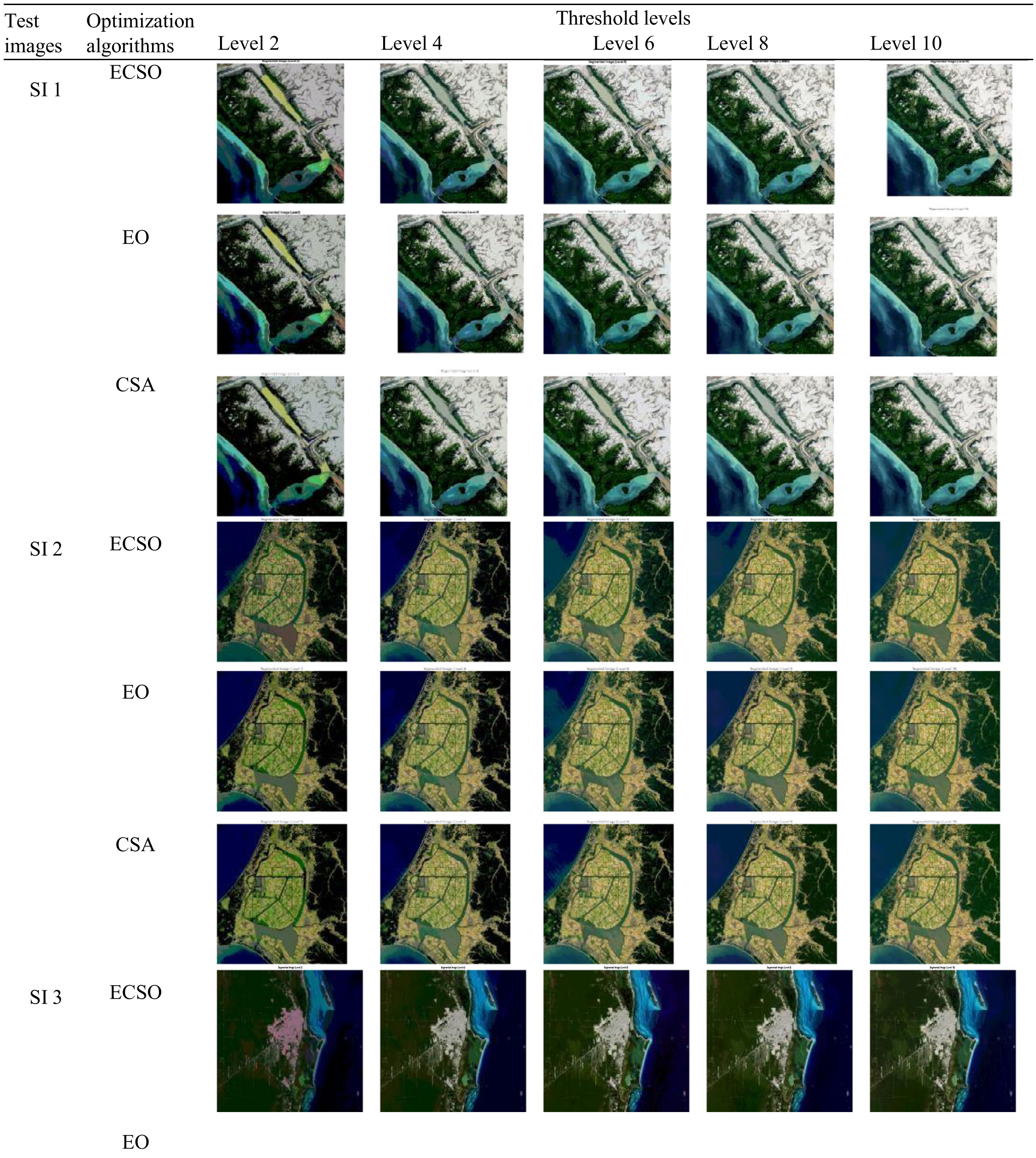


Fig. 9. Segmented satellite images using Otsu's between class variance.

used while for testing the exploration ability, Scalable multimodal functions are used. The ability for reaching the global best is signified by the enactment of the unimodal functions while the local optimum escaping is shown by the performance of the scalable multimodal functions. The 3 groups of mathematical benchmark functions (F_1 *to* F_{23}) are shown in Tables 1–3, with number of design variables which is denoted as 'VN', range and the minimum function value which is denoted as f_{min}.

For testing the effectiveness of the proposed ECSO, EO and CSA, the group of standard 23 mathematical benchmark functions such as unimodal, scalable multimodal and fixed-dimension multimodal are taken for solving optimization problems. The dimension size, the population size and the maximum iteration are 30, 30 and 500; respectively. All algorithms such as the ECSO, EO and CSA are implemented in MATLAB R2016a with Intel Core I5 2400 CPU clocked at 3.10 GHz with 8 GB RAM, AMD RX 550 GPU with 2 GB RAM using windows 10 operating system. The control parameters of our proposed ECSO, EO and CSA are presented in Table 4. From the extensive simulations, these parameters are decided as they exhibited the best performances. These parameters are also found in the related papers such as: the EO and the CSA. For a statistical analysis, the simulation results are presented in this Section. A detailed analysis is provided below for the readers.

5.2.2. The quantitative results comparison of the ECSO with the EO and the CSA

All the results including the Average (Ave.), Standard Deviation (Std.), Minimum (Min.) and Maximum (Max.) of thirty runs of each mathematical benchmark function are displayed in Table 5.

From Table 5, it is observed that the proposed ECSO performs better than the other two algorithms in most of the cases. Note that the boldface letters indicate the best results. The reason could be due to the inheritance of the properties of the CSA by the EO algorithm. Both the exploration and the exploitation properties of the ECSO help us to obtain better statistical results. Quite often the suggested ECSO algorithm exhibits better values (the minimum values) for the standard

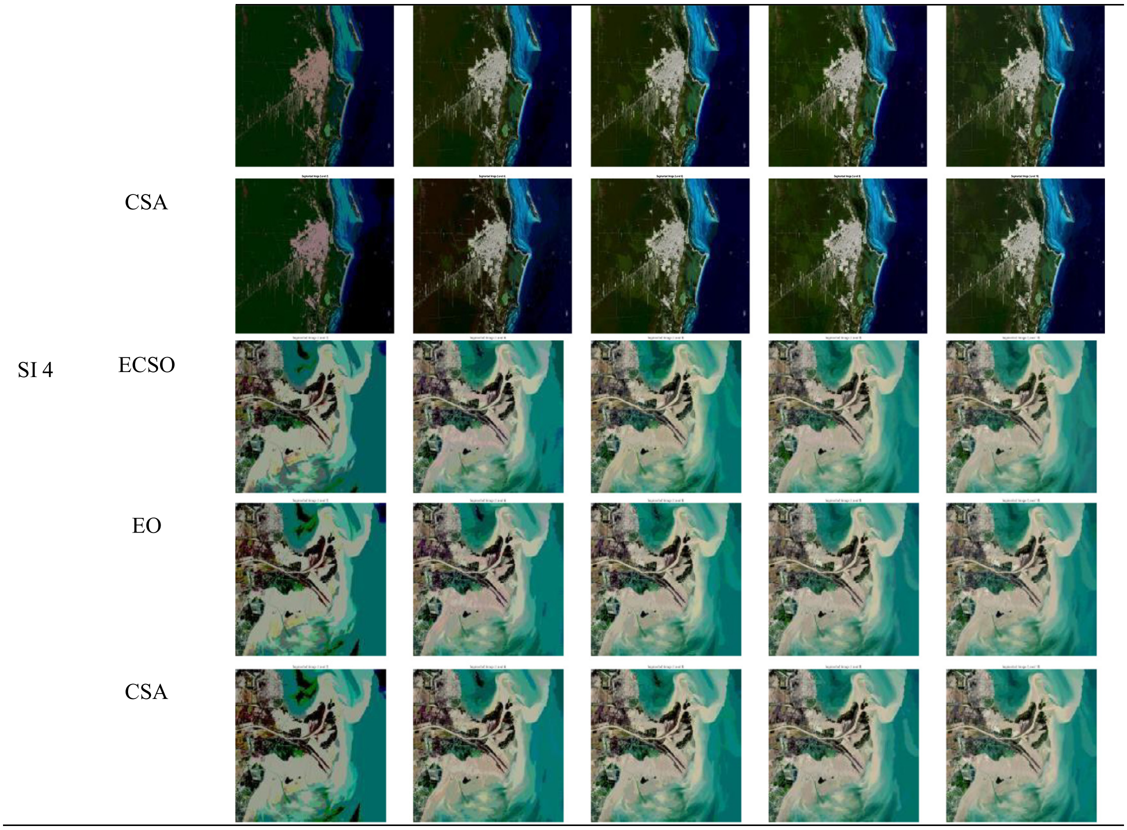


Fig. 9. (*continued*).

Pseudo code of our proposed ECSO

```
Initialize the particle's population $i = 1, 2, \ldots, N$
Assign equation candidates' fitness a large number
Assign free parameters $a_1 = 2$; $a_2 = 1$; $GP = 0.5$;
While $iter < max\_iter$
    For $i = 1$ : number of particles $(N)$
        Calculate fitness of $i^{th}$ particle
            If $fit(\vec{C}_i) < fit(\vec{C}_{eq(1)})$
                Replace $\vec{C}_{eq(1)}$ with $\vec{C}_i$ and $fit(\vec{C}_{eq(1)})$ with $fit(\vec{C}_i)$
            Elseif $fit(\vec{C}_i) > fit(\vec{C}_{eq(1)})$ & $fit(\vec{C}_i) < fit(\vec{C}_{eq(2)})$
                Replace $\vec{C}_{eq(2)}$ with $\vec{C}_i$ and $fit(\vec{C}_{eq(2)})$ with $fit(\vec{C}_i)$
            Elseif $fit(\vec{C}_i) > fit(\vec{C}_{eq(1)})$ & $fit(\vec{C}_i) < fit(\vec{C}_{eq(2)})$ & $fit(\vec{C}_i) < fit(\vec{C}_{eq(3)})$
                Replace $\vec{C}_{eq\,3}$ with $\vec{C}_i$ and $fit(\vec{C}_{eq\,3})$ with $fit(\vec{C}_i)$
            Elseif $fit(\vec{C}_i) > fit(\vec{C}_{eq(1)})$ & $fit(\vec{C}_i) < fit(\vec{C}_{eq(2)})$ & $fit(\vec{C}_i) < fit(\vec{C}_{eq(3)})$ & $fit(\vec{C}_i) < fit(\vec{C}_{eq(4)})$
                Replace $\vec{C}_{eq(4)}$ with $\vec{C}_i$ and $fit(\vec{C}_{eq(4)})$ with $fit(\vec{C}_i)$
            End
    End
    Update the concentration candidate $\vec{C}_{eq(1)}, \vec{C}_{eq(2)}, \vec{C}_{eq(3)}$ and $\vec{C}_{eq(4)}$ by using (13)
    $\vec{C}_{eq(ave)} = (\vec{C}_{eq(1)} + \vec{C}_{eq(2)} + \vec{C}_{eq(3)} + \vec{C}_{eq(4)}) / 4$
    Construct the equilibrium pool $\vec{C}_{eq,pool} = \{\vec{C}_{eq(1)}, \vec{C}_{eq(2)}, \vec{C}_{eq(3)}, \vec{C}_{eq(4)}, \vec{C}_{eq(ave)}\}$
    Accomplish memory saving $(if\ iter > 1)$
    Assign $t = (1 - \frac{iter}{\max\_iter})^{(a_2 \times \frac{iter}{\max\_iter})}$ )

    For $i = 1$: number of particles $(N)$
        Randomly choose one candidate from the equilibrium pool (Vector)
        Generate random vector of $\vec{\lambda}_i(iter)$ and $\vec{r}_i(iter)$
        Construct $\vec{F}_i(iter) = a_1 sign(\vec{r}_i(iter) - 0.5) \left[e^{-\vec{\lambda}_i(iter) \times t} - 1\right]$
        Construct $\overrightarrow{GCP}_i(iter) = \begin{cases} 0.5r_1 & r_2 \geq GP \\ 0 & r_2 < GP \end{cases}$
        Construct $\vec{G}_{0_i}(iter) = \overrightarrow{GCP}_i(iter) \times \left(\vec{C}_{eq}(iter) - \left(\vec{\lambda}_i(iter) \times \vec{C}_i(iter)\right)\right)$
        Construct $\vec{G}_i(iter) = \vec{G}_{0_i}(iter) \times \vec{F}_i(iter)$
        Update Concentrations using (2)
    End
    $iter = iter + 1$
End
```

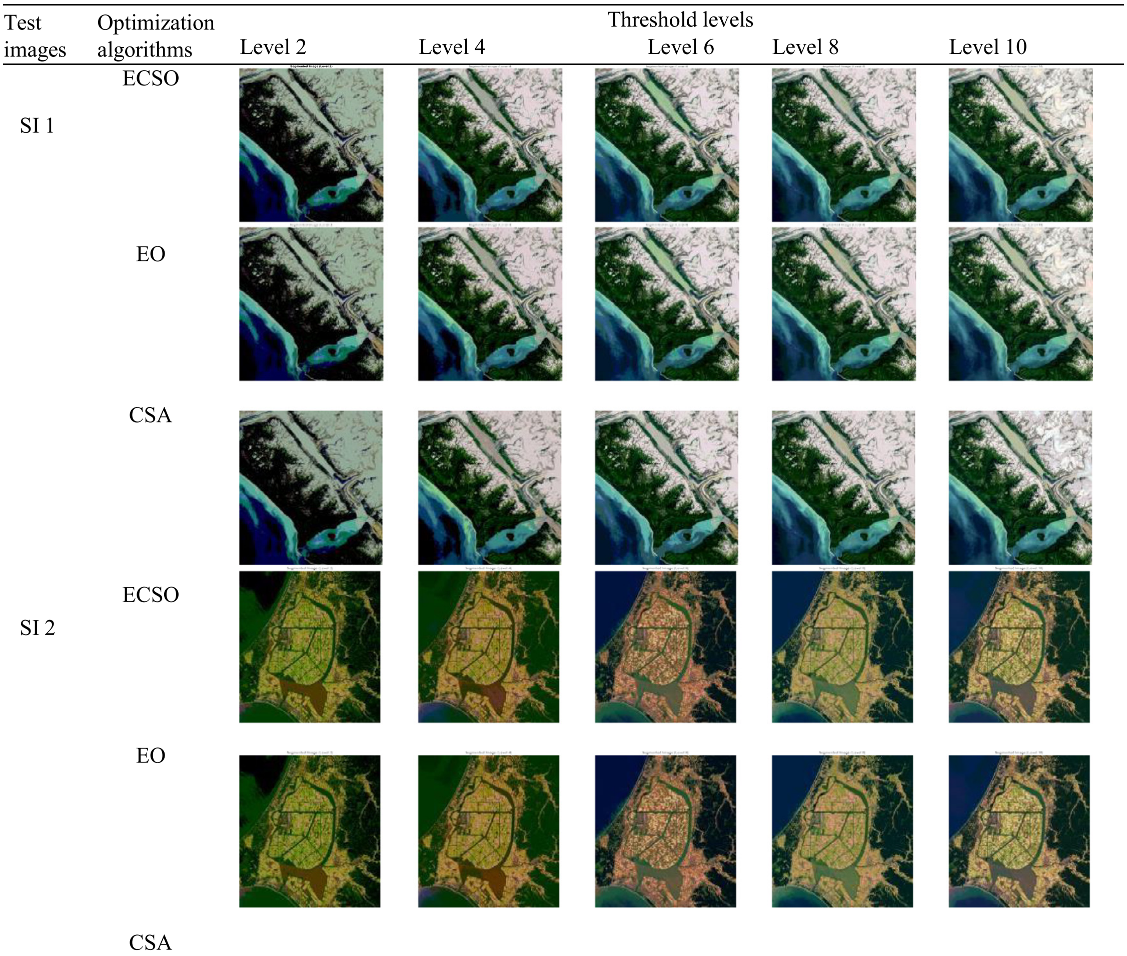


Fig. 10. Segmented satellite images using Kapur's entropy.

Table 3
Description of the Fixed-Dimension Multimodal benchmark functions.

Functions	VN	Range	f_{min}
$F_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6}\right)^{-1}$	2	$[-65, 65]$	1
$F_{15}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1 (b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]^2$	4	$[-5, 5]$	0.00030
$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]$	−1.0316
$F_{17}(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	2	$[-5, 5]$	0.398
$F_{18}(x) = \left[1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2)\right] \times \left[30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2)\right]$	2	$[-2, 2]$	3
$F_{19}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij} (x_j - p_{ij})^2\right)$	3	$[1, 3]$	−3.86
$F_{20}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij} (x_j - p_{ij})^2\right)$	6	$[0, 1]$	−3.32
$F_{21}(x) = -\sum_{i=1}^{5} \left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	4	$[0, 10]$	−10.1532
$F_{22}(x) = -\sum_{i=1}^{7} \left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	4	$[0, 10]$	−10.4028
$F_{23}(x) = -\sum_{i=1}^{10} \left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	4	$[0, 10]$	−10.5363

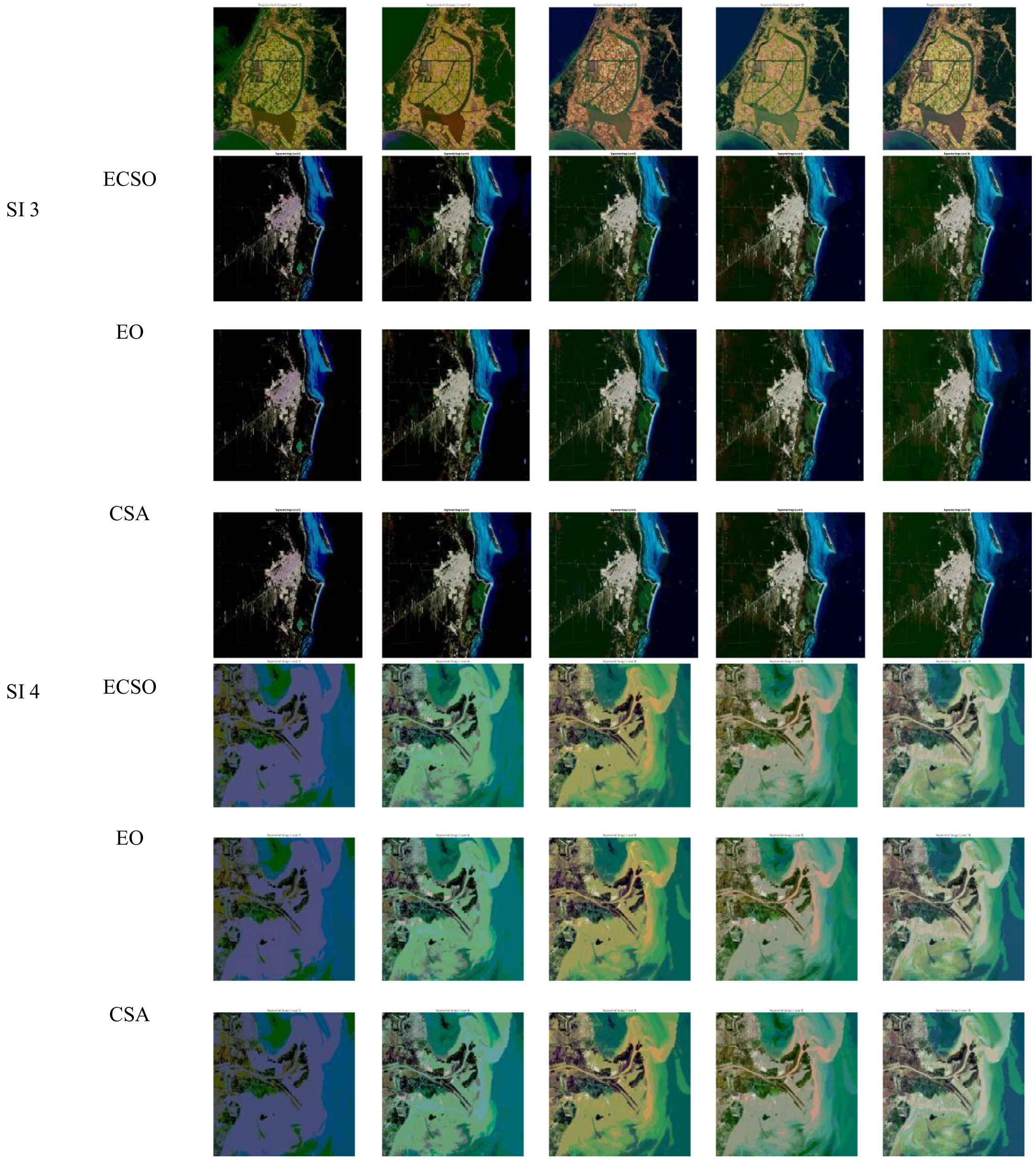


Fig. 10. (*continued*).

deviation, which evokes its superiority over other two well-known algorithms. It may be noted that the suggested ECSO technique is more consistent than others, which is implicit from Table 5.

5.2.3. The qualitative results comparison of the ECSO with the EO and the CSA

For a better analysis, the boxplots of the ECSO, EO and CSA are implemented to check the efficiency of the ECSO for all mathematical test functions. The convergence curves and the boxplots are shown in Figs. 5 and 6, respectively.

From Fig. 5, it is reflected that the suggested ECSO algorithm converges at a faster rate than the other two. Thus, the convergence is guaranteed for all the test functions. In this study, the number of iterations considered is 500. The convergence curves, iteration vs. best score obtained so far, are plotted in Fig. 5. The population size is 30 for each case. The descent convergence curves are noticed with our suggested method, which is desirable. It is implicit in Fig. 5. From the convergence curves of F9 and F11, it is noticed that the objective function values of the ECSO converged to zero (which is also reflected in Table 5) around 150 iterations only. Whereas the other optimizers are unable to compete. This study further evokes us many interesting findings. For other benchmark functions also, our suggested ECSO optimizer converges quicker than others.

An in depth statistical analysis is carried out and presented in the form of box plots, which are displayed in Fig. 6. It is depicted from Fig. 6 that our suggested ECSO algorithm performs well compared to other two optimizers. The best solutions are obtained in our case. The reason could be due to the best exploration together with its best ability to exploit the best opportunities in the selection of the optimum solutions. It may be reiterated that the EO has a better exploring capability while CSO has a better exploitation skill. The fusion of these capabilities led to the development of a quality optimizer. The performances rendered by the suggested optimizer ECSO are explicitly visible from Fig. 6.

Table 4
Parameter setting for the ECSO, EO and CSA.

Optimization algorithms	Control parameters	Values of parameters
ECSO	N	30
	Max_iter	500
	Run_no	30
	a_1	2
	a_2	1
	GP	0.5
	$Iter$	0
	V	1
	p_a	0.25
	α	1
	γ	1.5
	λ	0.5
EO	N	30
	Max_iter	500
	Run_no	30
	a_1	2
	a_2	1
	GP	0.5
	$Iter$	0
	V	1
CSA	N	30
	Max_iter	500
	p_a	0.25
	α	1
	γ	1.5
	λ	0.5

5.2.4. Discussion on the statistical results of the ECSO

The performance evaluation of the proposed optimizer ECSO is carried out in terms of the different statistical parameters. The ‘Ave.’ stands for the average fitness. The word ‘Std.’ represents the standard deviation. It signifies the deviation from its ideal value. The less is the value, better is the method. The reason is that the deviation is the least compared to other methods. The word ‘Min”. is used to denote the minimum value while the word ‘Max.’ is utilized to represent the maximum value. These notations are utilized for an in-depth analysis of the statistical results.

In this work, a new ECSO is introduced. To measure the effectiveness of this proposed algorithm, it is implemented and compared with the EO and the CSA. In Table 5, the bold values represent the best results. By analysing the results of three optimization algorithms, it is claimed that the proposed ECSO gives better results compared to the EO and the CSA by taking unimodal and scalable multimodal mathematical function as the objective function. For a better analysis, all three algorithms are implemented for knowing about which algorithm gives better convergence rate. From Fig. 5, it is observed that the ECSO converges at a faster rate compared to the other 2 algorithms. From the Box plots shown in Fig. 6, we claim that the ECSO gives better results compared to other two algorithms in terms of the statistical analysis. From this study, by analysing all tables, figures of convergence curves and figures of boxplots, it is claimed that the ECSO outperforms the EO and the CSA. However, the main limitation is that the proposed ECSO does not perform well by taking fixed dimension multimodal benchmark function as the objective function.

6. Results and discussion

This section presents all the results of MTH-based image segmentation (discussed in Section 4) of 4-benchmark colour satellite images, which are acquired from NASA Landsat image dataset (Landsat Image Gallery, 2020), using 3 different optimization algorithms such as: the proposed ECSO, EO and CSA. The test images with the corresponding RGB histograms are displayed in Fig. 7. It may be noted that 2 objective functions, such as Otsu and Kapur, are also implemented for a comparison. The maximum number of iterations and the population size of each algorithm are taken as 500 and 30, respectively. The upper bound (UB) and the lower bound (LB) are taken as 0 and 255. The performances of these methods are computed. Note that the outputs are obtained from 30 runs, for all colour satellite images. The optimal threshold values for all R (red), G(green) and B(blue) components are computed. The image quality metrics such as: PSNR, SSIM and FSIM for all 5 levels such as: 2, 4, 6, 8 and 10 are also computed for the experimental analysis. All the simulations of experiments is performed in MATLAB R2016a supported by Intel Core I5 2400 CPU clocked at 3.10 GHz processor with 8 GB RAM, AMD RX 550 GPU processor with 2 GB RAM running on windows 10 operating system. All the results are shown in tabular form, image form and boxplots. In tabular form, the best results are represented as bold values. In the image form, both the original and segmented images for 5 levels are represented. For the statistical result analysis, box plots are presented.

6.1. Image quality assessments

Image quality assessment is used to check the quality of the image. There are several techniques used to measure the quality of the segmented image. Here, for our work, we use three popular image quality metrics which are explained below.

6.1.1. Peak signal to noise ratio (PSNR)

PSNR is a quality metric of the image, which is the ratio of a signal’s maximum potential power to the power of corrupting noise that affects the fidelity of its representation (Jia et al., 2019). It is normally expressed as a logarithmic quantity using the decibel (dB) scale. PSNR is formulated as:

$$PSNR = 10\log_{10}\left(\frac{L_{max}^2}{MSE}\right) = 20\log_{10}\left(\frac{L_{max}}{\sqrt{MSE}}\right) \\ = 20\log_{10}\left(L_{max}\right) - 10\log_{10}\left(MSE\right) \tag{44}$$

Here, L_{max} is the image’s maximum possible pixel value which is 255. Also MSE is denoted by Mean Square Error, which is computed as follow:

$$MSE = \frac{\sum_{i=1}^{R}\sum_{j=1}^{C}\left[I_{ref}(i,j) - I_{seg}(i,j)\right]^2}{R \times C} \tag{45}$$

Here, $I_{ref}(i,j)$ and $I_{seg}(i,j)$ are Grey value of coordinates at original input or reference image and segmented output image respectively, $R \times C$ represents the size of image.

6.1.2. Structural similarity index (SSIM)

SSIM is a metric for evaluating image quality that is dependent on structure similarity (Wang et al., 2004). It compares images based on brightness, contrast, and structure. [0, 1] is its value range. The image distortion is lower if the value is closer to 1. The SSIM is formulated as follow:

$$SSIM\left(I_{ref}, I_{seg}\right) = \frac{\left(2\mu_{I_{ref}}\mu_{I_{seg}} + C_1\right)\left(2\sigma_{I_{ref}I_{seg}} + C_2\right)}{\left(\mu_{I_{ref}}^2 + \mu_{I_{seg}}^2 + C_1\right)\left(\sigma_{I_{ref}}^2 + \sigma_{I_{seg}}^2 + C_2\right)} \tag{46}$$

Here, $\mu_{I_{ref}}$ and $\mu_{I_{seg}}$ are represented as average Grey values of the original reference image and segmented output image respectively, $\sigma_{I_{ref}}^2$ and $\sigma_{I_{seg}}^2$ are represented as variance of the original reference image and segmented output image respectively, C_1 and C_2 are two constants that are used for maintaining the stability. The 2 constants C_1 and C_2 are computed as $C_1 = \left(0.01L_{max}\right)^2$ and $C_2 = \left(0.03L_{max}\right)^2$. For our work, L_{max} was taken as 255. Thus, the values of C_1 and C_2 are 6.5025 and 58.5225 respectively. The term $\sigma_{I_{ref}I_{seg}}$ is a standard deviation of the original reference image and segmented output image, which is formulated as follow:

$$\sigma_{I_{ref}I_{seg}} = \frac{1}{N-1}\sum_{i=1}^{N}\left(I_{ref_i} - \mu_{I_{ref}}\right)\left(I_{seg_i} - \mu_{I_{seg}}\right) \tag{47}$$

Table 5
Comparison of statistical performance analysis of ECSO with EO and CSA.

Function names	Test functions	Metrics	Optimization algorithms		
			ECSO	EO	CSA
Unimodal test functions	F_1	Ave.	**2.292624E−41**	3.768115E−41	0.0036
		Std.	**3.069319E−41**	4.397874E−41	0.0022
		Min.	**1.581065E−44**	8.223556E−43	5.6419E−04
		Max.	**1.095311E−40**	1.840133E−40	0.0103
	F_2	Ave.	**7.418274E−23**	1.056652E−22	0.0979
		Std.	**7.299525E−23**	1.500899E−22	0.0396
		Min.	**6.394402E−24**	1.559412E−23	0.0451
		Max.	**2.717719E−22**	7.675547E−22	0.2001
	F_3	Ave.	**2.644936E−09**	1.223669E−08	12.7573
		Std.	**7.416349E−09**	6.462921E−08	4.2900
		Min.	**7.337925E−14**	8.306347E−13	2.4210
		Max.	**3.385222E−08**	3.544007E−07	20.5091
	F_4	Ave.	**3.182616E−10**	3.305541E−10	1.2319
		Std.	**6.394724E−10**	6.878226E−10	0.7596
		Min.	**2.797141E−12**	2.083585E−11	0.5024
		Max.	**3.020686E−09**	3.445036E−09	4.7015
	F_5	Ave.	**25.351**	25.414	19.9339
		Std.	**0.18076**	0.23178	6.9211
		Min.	25.041	25.042	**11.2018**
		Max.	25.858	25.897	42.3580
	F_6	Ave.	**7.8709E−06**	9.731711E−06	0
		Std.	**6.6286E−06**	7.137965E−06	0
		Min.	**1.0252E−06**	1.084340E−06	0
		Max.	**2.6353E−05**	3.321895E−05	0
	F_7	Ave.	**0.0010387**	0.0011909	0.0212
		Std.	**0.00042925**	0.00067191	0.0084
		Min.	**8.2504E−05**	0.0003234	0.0072
		Max.	**0.0020863**	0.0028132	0.0385
Scalable multimodal test functions	F_8	Ave.	**−8865.997**	−8806.061	−4.649422E+03
		Std.	**582.8793**	627.6424	172.7994
		Min.	**−10216.85**	−9832.297	−4.972054E+03
		Max.	**−7565.471**	−7435.565	−4.332463E+03
	F_9	Ave.	**0**	0.03316539	33.1134
		Std.	**0**	0.1816543	5.7171
		Min.	**0**	0	23.1109
		Max.	**0**	0.9949616	48.1210
	F_{10}	Ave.	**8.467301E−15**	8.822572E−15	2.0221
		Std.	**2.224220E−15**	2.233983E−15	0.8338
		Min.	**4.440892E−15**	7.993606E−15	0.3782
		Max.	1.509903E−14	1.509903E−14	3.7094
	F_{11}	Ave.	**0.0003288231**	0.001064097	0.1658
		Std.	**0.001801038**	0.005828299	0.0422
		Min.	0	0	0.0916
		Max.	**0.009864694**	0.03192291	0.2405
	F_{12}	Ave.	**4.669719E−07**	0.003456304	0.5734
		Std.	**5.174856E−07**	0.01892726	0.4131
		Min.	**8.116455E−08**	5.174856E−07	0.0599
		Max.	**2.844230E−06**	0.1036696	1.7576
	F_{13}	Ave.	**0.01858217**	0.02175677	0.0204
		Std.	0.04174209	0.040016	0.0112
		Min.	3.616917E−06	2.212902E−06	0.0052
		Max.	0.195854	0.1413271	0.0486
Fixed dimension multimodal test functions	F_{14}	Ave.	**0.9980038**	1.064141	0.9981
		Std.	**1.797289E−16**	0.3622457	4.7997E−16
		Min.	**0.9980038**	0.9980038	0.9981
		Max.	**0.9980038**	2.982105	0.9981
	F_{15}	Ave.	0.001653312	0.004390945	**4.0188E−04**
		Std.	0.005085961	0.008125923	**9.7156E−05**
		Min.	0.000307487	0.0003074863	**3.0846E−04**
		Max.	0.02036334	0.02036334	**6.9488E−04**
	F_{16}	Ave.	**−1.031628**	−1.031628	−1.0316
		Std.	**4.320845E−16**	6.11579E−16	5.0499E−16
		Min.	**−1.031628**	−1.031628	−1.0336
		Max.	**−1.031628**	−1.031628	−1.0346
	F_{17}	Ave.	**0.3973274**	0.3978874	0.3979
		Std.	**0**	0	6.7089E−14

(*continued on next page*)

Table 5 (*continued*).

Function names	Test functions	Metrics	Optimization algorithms		
			ECSO	EO	CSA
		Min.	**0.3971874**	0.3978874	0.3979
		Max.	**0.3971174**	0.3978874	0.3979
	F_{18}	Ave.	**3**	3	3
		Std.	**1.121649E−15**	8.205193E−16	1.6534E−15
		Min.	**3**	3	3
		Max.	**3**	3	3
	F_{19}	Ave.	**−3.862782**	−3.862782	−3.8628
		Std.	**2.339074E−15**	2.553763E−15	2.4795E−15
		Min.	**−3.862782**	−3.862782	−3.7629
		Max.	**−3.862782**	−3.862782	−3.7638
	F_{20}	Ave.	**−3.320284**	−3.240039	−3.2220
		Std.	**0.06015335**	0.06920909	1.9711E−07
		Min.	**−3.321995**	−3.321995	−3.3220
		Max.	**−3.197383**	−3.137642	−3.3220
	F_{21}	Ave.	**−10.233594**	−9.052768	−10.1265
		Std.	0.074059	2.275893	**3.8737E−06**
		Min.	**−10.2335**	−10.0469	−10.1265
		Max.	−5.055198	−2.630472	−10.1265
	F_{22}	Ave.	**−10.402965**	−10.003145	−9.0263
		Std.	0.367541	1.531972	**1.1059E−05**
		Min.	**−10.402965**	−10.402945	−9.0263
		Max.	−3.885487	−3.7243	−9.0263
	F_{23}	Ave.	**−10.536482**	−9.772065	−9.1857
		Std.	0.2164	2.342062	4.1884E−04
		Min.	**−10.536482**	−10.53641	−9.1857
		Max.	−2.504575	−2.421734	−9.1857

6.1.3. Feature similarity index (FSIM)

FSIM (Zhang et al., 2011) is a quality metric, which is measured the similarity between of the original reference image and segmented output image by using phase congruency (PC) and gradient magnitude (GM). Thus, the FSIM index between I_{ref} and I_{seg} is computed as:

$$FSIM\left(I_{ref}, I_{seg}\right) = \frac{\sum_{x\in\Omega} S_L(x) \cdot PC_m(x)}{\sum_{x\in\Omega} PC_m(x)} \tag{48}$$

Here, Ω is represented as the whole image spatial domain and $PC_m(x)$ and $S_L(x)$

$$PC_m(x) = max\left(PC_{I_{ref}}, PC_{I_{seg}}\right) \tag{49}$$

$$S_L(x) = \left[S_{PC}(x)\right]^{\alpha} \cdot \left[S_G(x)\right]^{\beta} = S_{PC}(x) \cdot S_G(x) \quad for\ \alpha = \beta = 1 \tag{50}$$

Here, $S_{PC}(x)$ and $S_G(x)$ are similarity measure of phase consistency and similarity measure of gradient magnitude which are formulated as:

$$S_{PC}(x) = \frac{2PC_{I_{ref}}(x) \cdot PC_{I_{seg}}(x) + T_1}{PC^2_{I_{ref}}(x) + PC^2_{I_{seg}}(x) + T_1} \tag{51}$$

$$S_G(x) = \frac{2G_{I_{ref}}(x) \cdot G_{I_{seg}}(x) + T_2}{G^2_{I_{ref}}(x) + G^2_{I_{seg}}(x) + T_2} \tag{52}$$

Here, T_1 and T_2 are positive constants which help for increasing the stability.

The results of our implementations are shown in Tables 6–8. The PSNR values are displayed in Table 6. The SSIM values are presented in Table 7 and the FSIM values are highlighted in Table 8.

From the results tabulated above, it is seen that our suggested technique offers better performance indices compared to the state-of-the-art methods. Note that PSNR is a ratio of signal to noise. The higher is the PSNR value, more is the information and less is the noise in the output result. In Table 6, it is implicit that our suggested DEE-ECSO method outperforms others. It is observed that with an increase in the threshold levels M, we achieve better results. The incorporation of ECSO in order to maximize the fitness function DEE yields the best results. The high PSNR value is a clear indication of the fact that DEE preserves the maximum edge information than the conventional 2-D Otsu and Kapur methods. The possible reason could be the concave nature of the fitness function, which is always non-negative.

It is noteworthy to mention that SSIM takes into consideration the structure and edge information from the input images. Hence, the higher is the SSIM value, better is the result. Table 7 presents the SSIM values. The results indicate that DEE-ECSO yields higher SSIM values than other techniques. As expected, the SSIM values increase with an increase in the number of thresholds. On further analysis, the superiority of DEE-ECSO is established, because it offers better exploration and exploitation capabilities. The reason may be attributed to the preservation of edge information in the DEE-ECSO method. Further, the non-negative and concave nature of the suggested DEE objective function prevents the truncation of very high values.

In Table 8, the FSIM values of the result images (outputs) are presented. A high FSIM value is needed for a good segmentation. It is observed that with an increase in the threshold levels, the FSIM values increase, which is expected. Our method yields better results consistently. The reason may be that the information is pushed to the edges, enhancing the gradient magnitude (GM). Further, the normalization step incorporated in our method preserves all the information with a greater efficiency.

The optimal thresholds, achieved utilizing the proposed DEE method, are displayed in Table 9. The optimal thresholds, by Otsu's scheme, are highlighted in Table 10. The optimal thresholds, with Kapur's technique, are displayed in Table 11. It is noteworthy to mention here that these optimal thresholds are finally used to get the thresholded outputs. To be precise, the accuracy of a method solely depends on these optimal threshold values.

Note that computing the appropriate value of the threshold(s) is crucial in image segmentation. The key factors in guiding the choice of the threshold(s) are — the separation between peaks in the image histogram; the relative size of objects and background; the uniformity of the pixel intensities; the uniformity of the reflectance; and inherent image characteristics. The accuracy of segmentation method depends on these threshold values. Therefore, there is a strong need to develop an efficient method to compute optimal threshold values. The statistical

Table 6
The comparison of PSNR values.

Test images	Levels	Optimization algorithms	Objective functions		
			Proposed technique	Otsu’s between class variance	Kapur’s entropy
SI 1	2	ECSO	**14.076641**	14.064985	13.519067
		EO	14.070842	14.051943	13.431726
		CSA	14.042773	13.954837	13.207700
	4	ECSO	**18.956367**	18.895481	18.619060
		EO	18.820891	18.797101	18.199735
		CSA	18.590645	18.433486	17.649838
	6	ECSO	**21.761947**	21.701047	20.969002
		EO	21.713465	21.550492	20.889969
		CSA	21.601635	21.515812	20.610802
	8	ECSO	**24.004153**	23.926208	22.901482
		EO	23.993219	23.887724	22.729411
		CSA	23.849746	23.763408	22.435156
	10	ECSO	**25.729944**	25.676438	24.542264
		EO	25.695366	25.643133	24.455505
		CSA	25.556198	25.186696	24.004484
SI 2	2	ECSO	**15.415855**	15.288448	13.906079
		EO	15.213215	15.032633	13.871325
		CSA	15.127159	14.942757	13.603601
	4	ECSO	**18.405196**	18.208968	15.442324
		EO	18.262014	18.082944	15.182156
		CSA	17.947901	17.902664	15.094871
	6	ECSO	**22.437641**	21.799513	18.135054
		EO	21.554595	20.979288	17.919374
		CSA	20.129045	19.724046	17.715656
	8	ECSO	**24.312598**	23.516957	21.782951
		EO	23.197120	22.226629	21.384187
		CSA	22.145689	21.790867	20.964203
	10	ECSO	**26.609594**	26.296628	22.681315
		EO	26.533160	25.046306	22.196791
		CSA	24.798072	24.253691	22.032893
SI 3	2	ECSO	**17.649824**	17.533148	15.634178
		EO	17.623815	17.518399	15.612621
		CSA	17.551789	17.368111	15.592576
	4	ECSO	**21.709927**	21.632492	18.291431
		EO	21.650170	21.617851	18.112370
		CSA	20.904613	20.479081	18.016088
	6	ECSO	**23.802265**	23.740845	20.860612
		EO	23.790096	23.716751	20.801690
		CSA	23.656026	23.322666	20.779763
	8	ECSO	**25.527215**	25.453020	22.445147
		EO	25.425095	25.206069	22.310290
		CSA	25.358918	25.130726	22.208263
	10	ECSO	**27.636622**	27.313789	23.839304
		EO	27.390669	27.150860	23.460544
		CSA	27.172351	26.840161	23.283364
SI 4	2	ECSO	**16.043974**	15.997929	11.624734
		EO	16.016712	15.876325	11.502583
		CSA	15.900836	15.717933	11.291352
	4	ECSO	**20.182642**	19.984895	16.110352
		EO	20.037795	19.824630	15.998756
		CSA	19.830895	19.572677	15.793564
	6	ECSO	**22.016620**	21.818761	17.163397
		EO	21.717502	21.441500	16.903493
		CSA	21.498040	21.144517	16.667974
	8	ECSO	**24.496508**	24.401182	19.372555
		EO	24.135127	23.761123	19.123871
		CSA	23.724824	23.172264	18.810373
	10	ECSO	**26.258232**	25.878042	22.068767
		EO	25.796744	25.631068	21.726971
		CSA	25.266190	25.107525	21.231406

analysis on results ultimately depends on the output thresholded (segmented) images, which are obtained using these optimal values. These values are displayed in Tables 9–11.

For a visual comparison, the output (segmented) images are shown in Figs. 8–10. One can also analyse the performance of the proposed method from the following results. An in-depth result analysis is provided below.

From the output results shown in Fig. 8, it is seen that our method performs well for satellite image segmentation. These colour satellite images are high dimensional in nature. The qualitative analysis is

Table 7
The comparison of SSIM values.

Test images	Levels	Optimization algorithms	Objective functions		
			Proposed technique	Otsu's between class variance	Kapur's entropy
SI 1	2	ECSO	**0.442222**	0.439531	0.412222
		EO	0.440131	0.424697	0.411097
		CSA	0.419280	0.417043	0.409376
	4	ECSO	**0.602017**	0.597052	0.568747
		EO	0.594988	0.580480	0.551287
		CSA	0.593507	0.579041	0.542825
	6	ECSO	**0.707017**	0.697847	0.665222
		EO	0.706412	0.696824	0.658690
		CSA	0.695811	0.686646	0.657338
	8	ECSO	**0.761798**	0.750692	0.734949
		EO	0.756290	0.749883	0.733538
		CSA	0.749246	0.736466	0.726893
	10	ECSO	**0.804352**	0.792824	0.766536
		EO	0.804286	0.790709	0.760863
		CSA	0.802855	0.775285	0.752535
SI 2	2	ECSO	**0.450757**	0.445157	0.348138
		EO	0.431578	0.418214	0.342851
		CSA	0.424197	0.409126	0.330549
	4	ECSO	**0.590909**	0.580457	0.473346
		EO	0.580636	0.568986	0.463496
		CSA	0.564225	0.561268	0.459705
	6	ECSO	**0.724818**	0.700905	0.643334
		EO	0.694711	0.672036	0.633365
		CSA	0.656728	0.639429	0.624900
	8	ECSO	**0.800311**	0.773993	0.727478
		EO	0.791348	0.769516	0.714680
		CSA	0.774124	0.756476	0.704760
	10	ECSO	**0.845229**	0.842915	0.793458
		EO	0.836851	0.790460	0.787371
		CSA	0.801934	0.783451	0.779746
SI 3	2	ECSO	**0.354279**	0.351053	0.141434
		EO	0.350745	0.349243	0.140288
		CSA	0.337107	0.300808	0.139536
	4	ECSO	**0.530347**	0.509737	0.253479
		EO	0.526722	0.507169	0.245465
		CSA	0.522579	0.490245	0.240537
	6	ECSO	**0.604971**	0.594792	0.464804
		EO	0.581993	0.564931	0.453261
		CSA	0.578450	0.561054	0.449175
	8	ECSO	**0.647741**	0.622843	0.501081
		EO	0.640804	0.617542	0.489886
		CSA	0.616243	0.599002	0.481070
	10	ECSO	**0.791522**	0.765312	0.534502
		EO	0.767171	0.740928	0.530888
		CSA	0.739151	0.706129	0.512840
SI 4	2	ECSO	**0.636897**	0.632638	0.530945
		EO	0.631655	0.622152	0.528357
		CSA	0.627142	0.618168	0.520468
	4	ECSO	**0.721273**	0.715121	0.658395
		EO	0.720602	0.712893	0.654122
		CSA	0.718099	0.707851	0.647489
	6	ECSO	**0.761458**	0.754865	0.745884
		EO	0.756979	0.750490	0.730075
		CSA	0.754082	0.745664	0.728924
	8	ECSO	**0.825388**	0.820703	0.768009
		EO	0.819501	0.806756	0.764446
		CSA	0.808544	0.795162	0.755672
	10	ECSO	**0.873593**	0.863822	0.838369
		EO	0.843555	0.838432	0.836747
		CSA	0.834896	0.829119	0.826778

Table 8
The comparison of FSIM values.

Test images	Levels	Optimization algorithms	Objective functions		
			Proposed technique	Otsu's between class variance	Kapur's entropy
SI 1	2	ECSO	**0.895251**	0.894185	0.846149
		EO	0.894594	0.884283	0.842533
		CSA	0.882598	0.873911	0.832092
	4	ECSO	**0.949518**	0.949321	0.946508
		EO	0.948922	0.947924	0.945345
		CSA	0.942696	0.941575	0.928594
	6	ECSO	**0.976078**	0.975874	0.967147
		EO	0.975975	0.974873	0.965917
		CSA	0.970211	0.969866	0.956543
	8	ECSO	**0.979901**	0.979650	0.970734
		EO	0.979692	0.979026	0.970127
		CSA	0.978828	0.978503	0.966395
	10	ECSO	**0.988336**	0.988260	0.985839
		EO	0.988274	0.987971	0.985079
		CSA	0.986033	0.985995	0.974383
SI 2	2	ECSO	**0.950146**	0.949662	0.931239
		EO	0.942259	0.942182	0.930893
		CSA	0.936832	0.935715	0.923106
	4	ECSO	**0.962723**	0.961809	0.940061
		EO	0.961686	0.960858	0.939566
		CSA	0.958443	0.958364	0.936513
	6	ECSO	**0.978290**	0.974367	0.946050
		EO	0.971303	0.968793	0.944181
		CSA	0.963080	0.962965	0.942158
	8	ECSO	**0.992838**	0.987961	0.961130
		EO	0.988251	0.987883	0.959636
		CSA	0.986779	0.986527	0.956083
	10	ECSO	**0.995713**	0.995457	0.981043
		EO	0.995617	0.995335	0.980410
		CSA	0.995268	0.995223	0.979710
SI 3	2	ECSO	**0.882374**	0.881147	0.849982
		EO	0.880691	0.879286	0.848047
		CSA	0.878254	0.877412	0.845655
	4	ECSO	**0.955057**	0.954950	0.927180
		EO	0.954895	0.953996	0.926033
		CSA	0.953890	0.952898	0.924809
	6	ECSO	**0.976223**	0.975825	0.960089
		EO	0.975019	0.974860	0.959674
		CSA	0.974652	0.974407	0.956388
	8	ECSO	**0.983270**	0.980671	0.970653
		EO	0.982363	0.980077	0.967448
		CSA	0.981052	0.979901	0.963074
	10	ECSO	**0.988288**	0.987728	0.980363
		EO	0.987369	0.987272	0.979815
		CSA	0.986279	0.985967	0.977650
SI 4	2	ECSO	**0.857360**	0.855731	0.756533
		EO	0.856522	0.854827	0.747525
		CSA	0.853261	0.852384	0.736437
	4	ECSO	**0.925245**	0.922447	0.894031
		EO	0.921738	0.920887	0.886888
		CSA	0.920209	0.919909	0.877392
	6	ECSO	**0.951495**	0.949909	0.911721
		EO	0.944200	0.943386	0.902507
		CSA	0.941822	0.939974	0.898888
	8	ECSO	**0.966490**	0.964969	0.937865
		EO	0.963953	0.961566	0.933913
		CSA	0.957915	0.954346	0.928560
	10	ECSO	**0.976323**	0.975412	0.944564
		EO	0.974624	0.973831	0.939774
		CSA	0.973059	0.971264	0.935197

presented here. The results obtained by Otsu's technique are presented in Fig. 9. The segmented outputs obtained using Kapur's entropy are displayed in Fig. 10. Five different threshold levels ($M = 2, 4, 6, 8, 10$) are considered in this experiment. Different colour levels are used for different output images. It is reiterated that for the threshold level *M,* there are *M + 1* classes. For instance, when M = 4, there are 5 classes (segments) in the output thresholded image. It may be recalled that we deal with three planes R, G and B. Therefore, a total of 5 x 5 x 5 = 125 colour levels. Similarly, a total of 729 colour levels are used when $M = 8$. To provide the visual information, all these output images are presented. More quality is observed with the proposed method (results displayed in Fig. 8). The reason could be the inherent feature of DEE-ECSO method to retain edges even for the high dimensional images like satellite images. The differential exponential entropy plays the key

Table 9
Optimal threshold values using the proposed technique.

Test images	Colours	Levels	Optimization algorithms		
			ECSO	EO	CSA
SI 1	R	2	65,180	65,180	72,174
		4	40,93,156,219	37,90,151,215	35,88,150,211
		6	15,50,94,133,179,223	13,55,94,136,179,223	21,52,91,134,177,219
		8	15,41,68,98,135,169,204,235	17,41,68,98,135,169,204,235	19,39,66,96,133,167,202, 233
		10	16,35,55,77,102,133,162,190,218,240	16,35,55,77,102,133,162,194,218,240	14,33,51,75,100,131,161,192,216,238
	G	2	69,177	71,183	99,183
		4	53,115,172,223	57,110,168,217	62,108,162,215
		6	35,72,109,147,185,230	35,72,109,147,185,230	46,72,105,142,181,221
		8	34,65,87,115,148,178,211,236	38,65,87,115,148,178,211,236	40,63,85,113,146,176,209,234
		10	31,64,83,103,131,156,181,200,220,237	31,64,83,105,131,156,179,200,219,238	39,62,81,103,129,154,177,197,217,234
	B	2	82,188	80,185	91,185
		4	43,110,153,213	50,108,152,210	49,97,150,206
		6	41,80,116,160,195,230	41,80,116,160,195,230	40,76,116,154,191,227
		8	31,55,78,110,142,170,201,234	31,55,78,110,142,170,201,234	33,53,76,108,140,168,199,232
		10	22,45,70,96,119,142,166,187,215,237	22,45,70,96,117,141,166,187,211,235	25,45,68,94,115,139,164,185,209,233
SI 2	R	2	81,193	75,170	75,173
		4	37,93,146,204	37,93,146,204	39,91,144,202
		6	25,65,103,140,177,219	28,62,100,140,177,219	26,60,99,138,175,218
		8	15,36,68,96,125,157,190,230	17,36,68,96,125,157,190,230	15,36,66,94,123,155,188,227
		10	18,30,60,88,115,140,163,185,211,240	20,28,55,88,113,138,160,182,209,235	20,34,63,89,110,135,157,178,205,230
	G	2	67,150	88,155	91,148
		4	73,108,142,180	73,108,142,180	75,106,140,178
		6	55,84,112,142,169,193	58,84,112,138,166,193	63,82,110,136,164,194
		8	49,83,107,126,138,162,183,200	32,83,107,126,138,162,183,200	30,81,105,124,136,160,181,198
		10	48,75,95,105,128,140,160,183,197,210	48,73,93,110,132,148,166,185,199,210	45,70,90,108,130,147,169,180,200,236
	B	2	60,120	65,119	67,118
		4	48,75,104,143	50,80,106,144	57,78,104,142
		6	42,69,87,101,131,158	45,70,85,100,129,156	52,68,83,99,127,154
		8	44,67,82,100,118,140,163,175	43,66,81,98,117,138,160,186	49,64,79,96,115,136,158,186
		10	37,52,70,86,110,139,157,175,215,230	37,50,71,89,111,141,160,171,210,230	46,57,68,82,95,119,149,170,195,230
SI 3	R	2	58,158	60,155	65,152
		4	28,62,120,187	28,62,120,185	26,60,119,183
		6	17,36,75,117,164,211	24,42,72,115,162,209	25,41,70,113,160,207
		8	19,41,55,82,117,150,187,225	19,37,53,80,115,148,185,223	23,35,51,78,113,146,183,221
		10	18,31,44,59,74,98,132,154,192,223	20,34,42,57,72,96,130,152,190,221	23,32,40,55,71,94,128,151,188,219
	G	2	34,120	35,118	36,116
		4	34,72,125,186	32,73,125,185	17,63,123,183
		6	38,60,86,123,165,208	32,58,84,121,163,206	29,56,82,119,161,204
		8	23,43,55,83,110,146,183,221	35,48,60,81,108,144,181,219	28,46,58,79,106,142,179,217
		10	13,37,54,67,80,106,133,161,189,224	15,35,52,65,78,104,131,159,187,222	17,33,50,63,76,102,129,157,185,220
	B	2	39,127	40,124	43,122
		4	29,67,127,181	32,67,118,177	35,65,116,175
		6	40,57,83,116,158,198	34,55,81,114,156,196	30,53,79,112,154,194
		8	23,43,63,93,123,155,186,211	25,47,67,91,121,153,184,207	27,45,65,89,119,151,181,205
		10	10,30,43,58,80,104,132,159,185,211	12,26,41,56,79,102,130,157,183,209	14,24,39,54,77,100,128,155,181,207
SI 4	R	2	69,155	74,153	78,151
		4	49,103,146,185	54,100,144,182	54,98,142,180
		6	45,85,121,153,181,205	49,83,119,151,179,203	49,81,117,149,177,201
		8	31,63,88,113,137,166,186,205	33,61,86,111,135,164,184,203	36,59,84,109,133,162,182,201
		10	25,58,77,98,124,150,167,187,196,213	32,56,75,96,122,148,165,180,194,211	35,54,73,93,120,146,163,178,192,209
	G	2	92,160	101,157	105,155
		4	78,121,154,185	83,119,151,182	83,117,149,179
		6	65,110,134,155,174,192	76,107,131,152,173,190	77,105,129,150,170,187
		8	59,92,112,126,138,153,165,187	64,90,113,140,152,173,188,215	68,88,111,137,151,171,186,213
		10	46,80,105,124,137,155,168,182,199,223	52,84,103,122,135,153,166,180,197,221	57,82,100,120,133,151,164,178,194,219
	B	2	90,144	96,141	98,139
		4	70,113,138,162	73,110,136,159	79,108,134,157
		6	60,106,127,131,158,185	71,104,125,141,156,183	75,102,123,137,154,180
		8	60,95,112,126,138,153,165,187	66,93,109,124,136,151,162,183	68,92,107,122,134,149,160,181
		10	58,91,107,120,132,144,153,164,186,211	60,88,105,118,130,142,150,161,184,210	65,86,103,116,128,141,148,159,182,207

role in improvising the quality. Problems in the loss of information are avoided by using our method. The output segmented images obtained using DEE-ECSO technique look more like the original images when the number of thresholds are more, for instance, $M = 10$. In this experiment, Otsu's method using the proposed ECSO proved to be the second contestant. To justify our claim in a more statistical sense, the Box plot analysis is provided below.

From the box plots, it is seen that the suggested method outperforms other techniques. The plots are given in Fig. 11. These box plots find it easy to track the performances of a method. It is also utilized to compare with other methods. Fig. 11 gives the box plot of the average PSNR, SSIM and FSIM values for 2D methods. Our method seems to be the best experimentally. The more PSNR value is seen with the DEE-ECSO technique. Similarly, the better SSIM and FSIM values are

Table 10
Optimal threshold values using Otsu's between class variance method.

Test images	Colours	Levels	Optimization algorithms		
			ECSO	EO	CSA
SI 1	R	2	67,182	67,180	74,176
		4	45,95,158,221	44,92,153,217	33,90,152,213
		6	14,52,96,135,181,225	9,57,96,138,181,225	19,54,93,136,179,221
		8	16,43,70,100,137,171,206,237	15,43,70,100,137,171,206,237	21,41,68,98,135,169,204,235
		10	18,37,57,79,104,135,164,192,220,242	19,37,57,79,104,135,164,196,220,242	16,35,53,77,102,133,163,194,218,240
	G	2	71,179	95,185	107,185
		4	55,117,174,225	59,112,170,219	64,110,164,217
		6	37,74,111,149,187,232	37,74,111,149,187,232	48,74,107,144,183,223
		8	36,67,89,117,150,180,213,238	40,67,89,117,150,180,213,238	42,65,87,115,148,178,211,236
		10	33,66,85,105,133,158,183,202,222,239	33,66,85,107,133,158,181,202,221,240	41,64,83,105,131,156,179,199,219,236
	B	2	77,182	80,180	83,185
		4	39,112,155,215	54,110,154,212	60,95,152,208
		6	45,82,118,162,197,232	45,82,118,162,197,232	45,75,118,156,193,229
		8	33,57,80,112,144,172,203,236	33,57,80,112,144,172,203,236	35,55,78,110,142,170,201,234
		10	24,47,72,98,121,144,168,189,217,239	24,47,72,98,119,143,168,189,213,237	33,47,70,96,117,141,166,187,211,235
SI 2	R	2	83,195	77,172	77,175
		4	39,95,148,206	39,95,148,206	40,93,146,204
		6	27,67,105,142,179,221	30,64,102,142,179,221	28,62,101,140,177,220
		8	17,38,70,98,127,159,192,232	16,38,70,98,127,159,192,232	17,38,68,96,125,157,190,229
		10	18,30,62,90,117,142,165,187,213,242	22,30,57,90,115,140,162,184,211,237	22,36,65,91,112,137,159,180,207,232
	G	2	69,156	90,157	93,150
		4	75,110,144,182	75,110,144,182	77,108,142,180
		6	57,86,114,144,171,195	60,86,114,140,168,195	65,84,112,138,166,196
		8	51,85,109,128,140,164,185,202	34,85,109,128,140,164,185,202	32,83,107,126,138,162,183,200
		10	48,75,99,107,130,142,162,185,199,212	48,75,95,112,134,150,168,187,201,212	47,72,92,110,132,149,171,182,202,238
	B	2	60,125	70,121	70,105
		4	50,77,106,145	52,82,108,146	56,80,106,144
		6	44,71,89,103,133,160	47,72,87,102,131,158	54,70,85,101,129,156
		8	46,68,84,102,120,142,165,177	50,68,83,100,119,140,162,188	53,66,81,98,117,138,160,188
		10	33,54,74,90,112,141,157,177,217,232	44,52,73,91,113,143,162,173,213,232	48,59,70,84,97,121,151,171,197,232
SI 3	R	2	38.160	62,157	67,155
		4	30,64,122,189	28,64,122,187	28,62,121,185
		6	18,38,77,119,166,213	26,44,74,117,164,211	27,40,72,115,162,209
		8	25,37,53,80,115,148,185,223	21,39,55,82,117,150,187,225	25,37,53,80,115,148,185,223
		10	22,34,42,57,73,96,130,153,190,221	22,36,44,59,74,98,132,154,192,223	25,34,42,57,73,96,130,153,190,221
	G	2	34.106	37,120	38,118
		4	36,74,127,188	34,75,127,187	14,65,125,185
		6	38,62,88,125,167,210	34,60,86,123,165,208	31,58,84,121,163,206
		8	30,43,60,81,108,144,181,219	37,50,62,83,110,146,183,221	30,48,60,81,108,144,181,219
		10	12,35,52,65,78,104,131,159,187,222	17,37,53,67,80,106,133,161,189,224	19,35,52,65,78,104,131,159,187,222
	B	2	45.130	40,110	57,125
		4	31,81,130,183	40,65,117,177	40,68,120,177
		6	40,60,85,118,160,200	36,57,83,116,158,198	32,45,99,114,160,196
		8	23,40,67,91,121,153,183,207	32,49,69,93,123,155,186,209	29,47,67,91,121,153,183,207
		10	14,26,41,56,79,102,130,157,183,209	14,28,43,58,88,104,132,156,185,211	16,26,41,59,79,102,130,157,183,209
SI 4	R	2	71,157	76,155	80,152
		4	51,105,148,187	56,102,146,184	56,100,144,182
		6	47,87,123,155,183,207	51,85,121,153,181,205	51,83,119,151,180,203
		8	33,65,90,115,139,168,188,207	35,63,88,113,137,166,187,205	38,61,86,111,135,164,185,203
		10	27,60,79,100,126,152,169,189,198,215	33,60,77,98,124,150,167,182,196,213	35,56,75,95,122,148,165,180,194,211
	G	2	94,162	103,160	107,157
		4	80,123,156,187	85,121,153,185	85,120,152,181
		6	67,112,136,157,176,194	78,109,133,154,175,192	79,107,131,152,173,189
		8	61,94,114,128,140,155,167,189	66,92,115,142,154,175,190,217	70,91,113,139,153,174,188,215
		10	48,82,107,126,139,157,170,184,201,225	54,86,105,124,137,155,168,182,199,223	55,84,102,122,135,153,166,175,196,219
	B	2	92,143	98,145	102,141
		4	72,115,140,158	75,112,138,161	81,112,128,159
		6	62,108,129,133,160,187	73,106,127,143,158,185	77,105,126,139,156,159
		8	62,97,114,128,140,155,168,189	68,95,111,126,138,153,164,185	70,94,100,124,136,151,162,183
		10	60,93,109,122,134,146,155,166,188,213	55,80,110,120,132,144,155,163,186,212	66,87,105,120,130,143,150,161,184,209

observed with the suggested DEE-ECSO scheme. It is observed that for most of the cases, the proposed method is the best compared to others. Its inherent mechanism of normalization of the local variance could be one of the possible reasons. Further, the differential exponential entropy provides accurate information regarding the partition of the high density images. Furthermore, the exploring the search space and exploiting the best solutions are also useful to get the optimal threshold values. The results of the EO are also very competitive. However, the suggested DEE-ECSO captures the high frequency information from the images under consideration. Moreover, the in-built mechanisms of the ECSO out beat other soft computing methods. Most importantly, the proposed DEE-ECSO method is a powerful method in terms of retaining most of the information, thereby increasing the quality of

Table 11
Optimal threshold values using Kapur's entropy.

Test images	Colours	Levels	Optimization Algorithms		
			ECSO	EO	CSA
SI 1	R	2	82,156	82,154	80,148
		4	46,117,160,227	50,117,160,221	53,115,156,212
		6	43,83,95,145,190,232	43,83,94,153,190,232	40,80,90,150,182,230
		8	22,57,77,117,132,183,204,235	22,57,77,117,132,183,204,235	20,55,75,115,130,180,200,230
		10	22,50,70,89,115,130,155,181,214,245	24,50,70,89,115,130,155,179,214,245	24,50,70,89,115,130,155,181,216,250
	G	2	113,179	113,177	111,170
		4	50,109,165,210	61,110,160,205	61,110,140,200
		6	24,65,90,150,191,219	24,65,90,150,191,219	21,63,88,153,182,217
		8	32,53,97,119,143,178,201,220	32,53,97,119,143,178,201,220	30,50,95,117,140,175,199,217
		10	25,55,76,91,117,133,158,187,222,249	26,55,76,94,120,134,158,188,220,249	26,55,76,92,119,131,159,188,205,247
	B	2	72,159	72,156	70,152
		4	72,113,153,215	77,110,145,210	80,105,135,201
		6	39,78,93,140,187,225	43,78,93,140,187,225	41,76,91,149,170,223
		8	30,69,88,125,140,177,198,225	30,72,90,127,140,181,198,225	36,70,88,125,138,179,196,220
		10	21,57,76,99,118,134,156,195,212,248	23,57,76,99,127,135,156,197,212,248	23,53,71,92,125,133,154,195,210,245
SI 2	R	2	78,171	80,169	70,158
		4	41,94,140,193	43,94,140,193	41,91,140,191
		6	29,81,111,155,169,224	27,79,108,153,167,222	25,77,106,150,165,220
		8	39,62,89,117,145,176,203,254	37,60,87,115,143,174,201,252	35,58,85,113,141,172,199,250
		10	18,54,69,94,116,138,163,187,210,233	18,52,67,92,114,136,161,185,208,231	20,50,65,90,112,134,159,183,206,229
	G	2	59,143	60,145	62,160
		4	39,52,133,251	39,52,133,251	39,47,131,249
		6	18,74,89,190,204,223	14,72,87,188,202,221	10,71,85,185,200,219
		8	18,34,81,103,114,135,155,253	16,32,79,101,112,132,146,255	14,30,77,99,110,130,144,255
		10	41,85,150,163,179,191,214,219,220,239	33,83,148,162,177,189,212,217,218,237	31,77,146,161,175,187,210,215,216,235
	B	2	105,190	107,195	103,196
		4	90,144,188,230	97,142,188,230	95,140,191,220
		6	26,62,125,148,195,242	24,60,123,146,193,240	22,57,121,144,191,239
		8	36,69,100,148,164,184,200,230	32,67,100,146,162,182,197,227	30,65,99,144,160,180,195,225
		10	27,58,76,102,123,145,170,199,217,227	27,56,74,100,121,143,168,197,215,230	27,54,72,98,119,141,166,193,213,233
SI 3	R	2	117,204	119,202	120,200
		4	51,111,162,210	53,109,160,208	55,107,157,206
		6	43,79,118,152,187,224	47,73,116,150,185,222	55,79,114,148,183,219
		8	34,62,87,114,145,174,203,231	36,60,85,112,143,172,201,229	38,58,83,110,141,170,199,227
		10	35,65,88,109,130,154,176,198,220,243	37,63,86,107,128,152,174,196,218,241	40,61,84,105,125,149,172,193,216,239
	G	2	118,221	120,219	123,217
		4	63,119,167,216	67,117,165,212	69,115,163,210
		6	25,65,113,147,184,223	28,71,111,145,182,221	30,71,109,143,180,219
		8	25,69,96,121,149,178,204,232	24,67,94,119,147,176,202,230	23,65,92,117,145,174,200,228
		10	29,43,74,96,117,137,165,189,221,235	17,41,72,94,115,135,163,187,219,233	20,39,69,92,112,132,161,185,217,231
	B	2	96,170	99,167	100,165
		4	54,108,159,210	56,106,157,208	58,104,155,205
		6	20,88,127,162,195,227	22,92,125,160,193,225	22,89,123,157,191,223
		8	26,63,90,115,145,174,205,234	27,61,88,113,143,172,203,232	28,58,86,111,141,170,201,230
		10	21,50,80,107,132,155,176,200,223,242	23,58,79,105,130,153,174,197,218,240	26,56,77,103,128,151,172,195,216,237
SI 4	R	2	77,254	75,251	70,250
		4	77,106,222,250	79,104,220,246	81,102,218,241
		6	33,48,89,104,155,203	36,46,87,102,153,200	35,44,85,100,150,200
		8	33,53,58,76,83,99,144,195	34,51,56,74,81,97,142,198	36,49,54,72,79,95,140,200
		10	24,51,68,70,75,92,110,136,187,255	23,49,66,68,72,90,108,134,185,248	26,49,64,65,70,87,106,132,190,255
	G	2	80,247	77,243	76,240
		4	79,112,160,173	80,110,158,169	83,107,156,162
		6	75,97,143,152,157,204	79,95,141,150,155,200	79,93,139,147,153,210
		8	37,64,89,116,130,154,219,253	38,62,87,114,128,152,217,252	40,60,85,112,126,150,215,253
		10	50,74,87,135,148,169,174,186,217,250	49,74,81,135,146,167,172,184,215,250	56,72,79,133,144,165,170,182,212,250
	B	2	107,126	105,125	106,121
		4	70,119,168,244	73,117,166,240	77,115,163,236
		6	20,74,81,89,194,228	22,72,79,87,192,225	20,69,77,85,190,220
		8	87,132,200,204,205,219,242,254	88,130,198,202,203,217,240,253	90,128,196,200,201,215,237,254
		10	43,74,89,95,109,155,166,198,238,255	42,72,89,97,107,153,164,192,236,250	35,70,87,95,105,152,161,190,234,255

the segmented outputs. From the above analysis, it is seen that Otsu's method using the suggested ECSO is the second contestant.

7. Conclusion

In this research, an efficient methodology named DEE is suggested. A new optimizer called ECSO is also proposed. The ECSO outperforms its predecessor EO while maintaining the same level of complexity. On well-known 23 benchmark functions, the ECSO's qualitative and quantitative tests are compared to those of other MAs such as: EO and CSA. The ECSO's efficacy in obtaining the best or near-optimum solutions by properly managing the exploration and exploitation capabilities is demonstrated by the results. In AI field, the ECSO can be used to solve the optimization problems and find the best solutions. The experiments

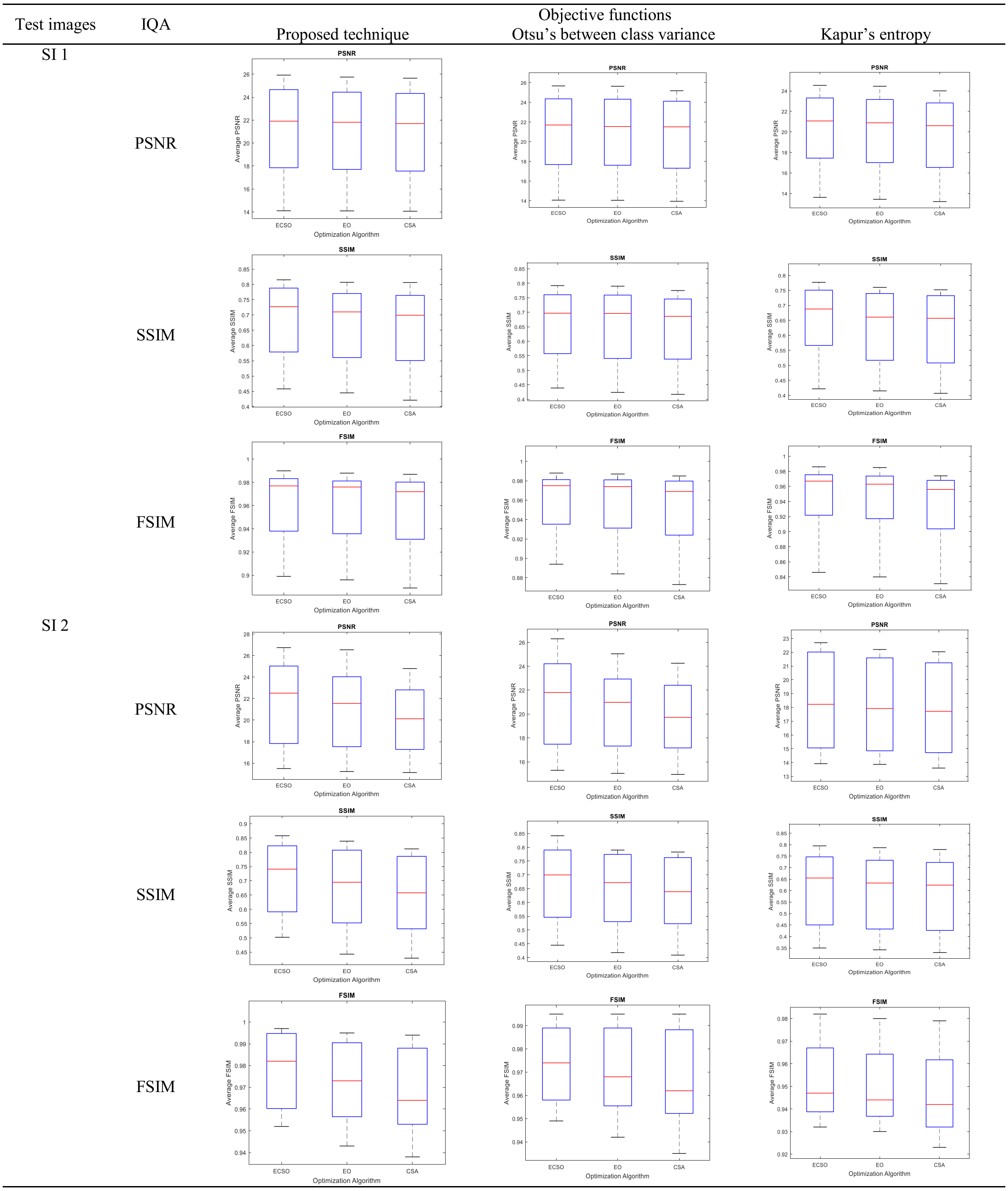


Fig. 11. The Box plots in terms of PSNR, SSIM and FSIM values.

are conducted to determine the efficiency of the DEE-ECSO in multilevel thresholding, an essential activity of computer visualization, a subfield of AI, for understanding and interpreting data in the real world. This multilevel thresholding is created by the maximization of the newly suggested objective function. The proposed ECSO outperformed the EO and the CSA. Our method DEE-ECSO outperformed both Otsu's and Kapur's methods. The reason is that our method inherently includes mechanisms for retaining more edge information, enhanced exploration of the search space, reduced computation complexity etc. The regions of interest are found in the first row only.

The future scope of the work is discussed here. The suggested ECSO may be expanded to handle the multi-objective problems. The possibility of expanding the ECSO includes a scheme of crossover-mutation, levels based on chaos, and learning based on opposition bodes well for the upcoming research. More scope is also there to enhance its capability to achieve the global best solution by incorporating the chance factor. Our proposal, i.e., the suggested threshold selection methodology would be useful for different applications like — brain magnetic resonance image segmentation, breast cancer thermogram image segmentation, and other natural grey-scale or RGB image segmentations by maximizing the proposed objective function.

CRediT authorship contribution statement

Monorama Swain: Methodology, Data handling, Programming, Draft. **Tanmaya Tapaswini Tripathy:** Implementation, Data handling, Programming, Draft. **Rutuparna Panda:** Conceptualization, Guidance, Data analysis, Revision of draft. **Sanjay Agrawal:** Implementation, Programming, Editing, Draft. **Ajith Abraham:** Analysis, Supervision.

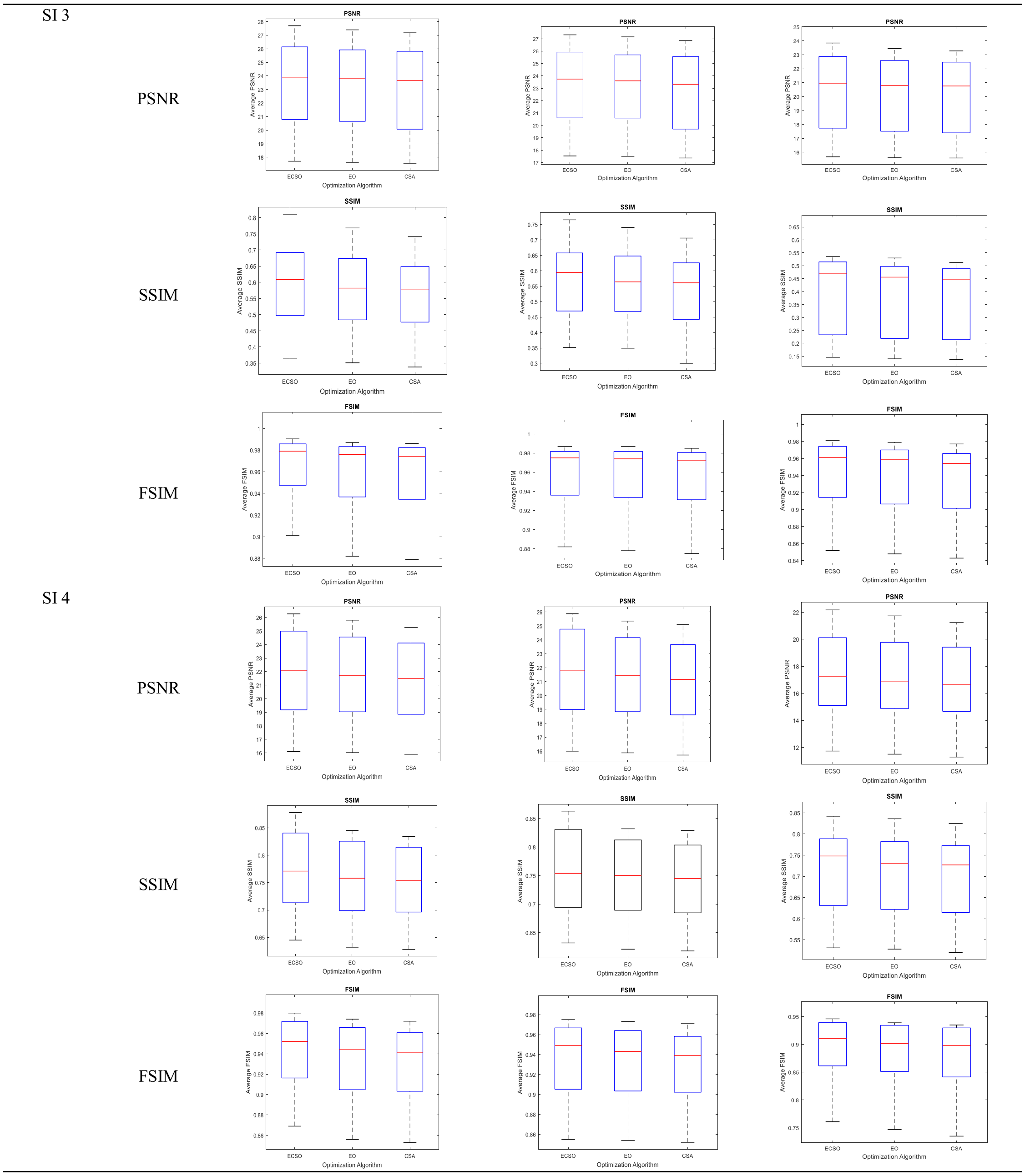


Fig. 11. (*continued*).

Acknowledgement

This research has been financially supported by The Analytical Center for the Government of the Russian Federation (Agreement No. 70-2021-00143 dd. 01.11.2021, IGK000000D730321P5Q0002).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Abdel-Basset, M., Chang, V., Mohamed, R., 2020. A novel Equilibrium Optimization algorithm for multi-thresholding image segmentation problems. In: Neural Computing and Applications. Springer, London.

Agrawal, S., Panda, R., Bhuyan, S., Panigrahi, B.K., 2013. Tsallis entropy based optimal multilevel thresholding using cuckoo search algorithm. Swarm Evol. Comput. 11, 16–30.

Angelova, M., Roeva, O., Pencheva, T., 2018. Cuckoo search algorithm for parameter identification of fermentation process model. In: International Conference on Numerical Methods and Applications. Springer, Cham, pp. 39–47.

Anitha, J., Immanuel Alex Pandian, S., Akila Agnes, S., 2021. An efficient multilevel color image thresholding based on modified whale optimization algorithm. Expert Syst. Appl. 178, 115003.

Awad, M.M., Chehdi, K., 2009. Satellite image segmentation using hybrid variable genetic algorithm. Int. J. Imaging Syst. Technol. 19 (3), 199–207.

Bao, X., Jia, H., Lang, C., 2019. A novel hybrid harris hawks optimization for color image multilevel thresholding segmentation. IEEE Access 7, 76529–76546.

Bhandari, A.K., Kumar, A., Chaudhary, S., Singh, G.K., 2016. A novel color image multilevel thresholding based segmentation using nature inspired optimization algorithms. Expert Syst. Appl. 63, 112–133.

Bhandari, A.K., Kumar, A., Singh, G.K., 2015. Tsallis entropy based multilevel thresholding for colored satellite image segmentation using evolutionary algorithms. Expert Syst. Appl. 42 (22), 8707–8730.

Castillo, O., 2021. Fuzzy Logic Hybrid Extensions of Neural and Optimization Algorithms: Theory and Applications, Vol. 940. Springer Nature.

Deepika, N.P., Vishnu, K., 2016. Different techniques for satellite image segmentation. In: IC-GET 2015 - Proceedings of 2015 Online International Conference on Green Engineering and Technologies. pp. 0–5.

El-Sayed, M.A., Tarek Abd-El Hafeez, 2011. New edge detection technique based on the Shannon entropy in grey level images. Int. J. Comput. Sci. Eng. (IJCSE) 3 (6), 2224–2232.

Faramarzi, A., Heidarinejad, M., Stephens, B., Mirjalili, S., 2020. Equilibrium optimizer: A novel optimization algorithm. Knowl.-Based Syst. 191.

Gonzalez, C.I., Melin, P., Castro, J.R., Castillo, O., Mendoza, O., 2016. Optimization of interval type-2 fuzzy systems for image edge detection. Appl. Soft Comput. 47, 631–643.

Guerrero-Luis, M., Valdez, F., Castillo, O., 2021. A Review on the Cuckoo Search Algorithm. Springer, Nature, pp. 113–124.

He, L., Huang, S., 2020. An efficient krill herd algorithm for color image multilevel thresholding segmentation problem. Appl. Soft Comput. 89, 106063.

Jia, H., Sun, K., Song, W., Peng, X., Lang, C., Li, Y., 2019. Multi-strategy emperor penguin optimizer for RGB histogram-based color satellite image segmentation using masi entropy. IEEE Access 7, 134448–134474.

Kapur, J.N., Sahoo, P.K., Wong, A.K.C., 1985. A new method for grey-level picture thresholding using the entropy of the histogram. Comput. Vis. Graph. Image Process. 29 (3), 273–285.

Landsat Image Gallery, 2020. https://landsat.visibleearth.nasa.gov/.

Masi, M., 2005. A step beyond Tsallis and Rényi entropies. Phys. Lett. A 338 (3–5), 217–224.

Naik, M.K., Panda, R., Abraham, Ajith, 2020. Normalized square difference based multilevel thresholding technique for multispectral images using leader slime mould algorithm. J. King Saud University-Comput. Inform. Sci. http://dx.doi.org/10.1016/j.jksuci.2020.10.030.

Naik, M.K., Panda, R., Abraham, A., 2021. An opposition equilibrium optimizer for context-sensitive entropy dependency based multilevel thresholding of remote sensing images. Swarm Evol. Comput. 65 (May), 100907.

Oliva, D., 2019. Metaheuristic Algorithms for Image Segmentation : Theory and Applications.

Otsu, N., 1979. A threshold selection method from grey-level histograms. IEEE Trans. Syst. Man Cybern. 20 (1), 62–66.

Pal, N.R., 1996. On minimum cross-entropy thresholding. Pattern Recognit. 29 (4), 575–580.

Pandey, B.N., Shrivastava, A.K., Rana, A., 2018. A literature survey of optimization techniques for satellite image segmentation. In: 2018 International Conference on Advanced Computation and Telecommunication, ICACAT 2018.

Pare, S., Kumar, A., Singh, G.K., Bajaj, V., 2020. Image segmentation using multilevel thresholding: A research review. Iran. J. Sci. Technol. - Trans. Electr. Eng. 44 (1).

Portes de Albuquerque, M., Esquef, I.A., Gesualdi Mello, A.R., Portes de Albuquerque, M., 2004. Image thresholding using Tsallis entropy. Pattern Recognit. Lett. 25 (9), 1059–1065.

Protasov, S., Khan, A.M., 2021. Using proximity graph cut for fast and robust instance-based classification in large datasets. Complexity http://dx.doi.org/10.1155/2021/2011738.

Rahaman, J., Sing, M., 2021. An efficient multilevel thresholding based satellite image segmentation approach using a new adaptive cuckoo search algorithm. Expert Syst. Appl. 174, 114633.

Rekik, A., Zribi, M., Hamida, A. Ben, Benjelloun, M., 2007. Review of satellite image segmentation for an optimal fusion system based on the edge and region approaches. Int. J. Comput. Sci. Netw. Secur. 7 (10), 242–250.

Roeva, O., Zoteva, D., Atanassova, V., Atanassov, K., Castillo, O., 2020. Cuckoo search and firefly algorithms in terms of generalized net theory. Soft Comput. 24 (7), 4877–4898.

Sahoo, P., Wilkins, C., Yeager, J., 1997. Threshold selection using Renyi's entropy. Pattern Recognit. 30 (1), 71–84.

Shubham, S., Bhandari, A.K., 2019. A generalized masi entropy based efficient multilevel thresholding method for color image segmentation. Multimedia Tools Appl. 78 (12), 17197–17238.

Suresh, S., Lal, S., 2016. An efficient cuckoo search algorithm based multilevel thresholding for segmentation of satellite images using different objective functions. Expert Syst. Appl. 58, 184–209.

Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., 2004. Image quality assessment: From error visibility to structural similarity. IEEE Trans. Image Process. 13 (4), 600–612.

Wunnava, A., Naik, M.K., Panda, R., Jena, B., Abraham, A., 2020. A novel interdependence based multilevel thresholding technique using Adaptive Equilibrium Optimizer. Eng. Appl. Artif. Intell. 94.

Yang, X., Deb, S., 2009. Cuckoo Search via Levy Flights. IEEE, pp. 210–214.

Zhang, L., Zhang, L., Mou, X., Zhang, D., 2011. FSIM: A feature similarity index for image quality assessment. IEEE Trans. Image Process. 20 (8), 2378–2386.