



Dominant color component and adaptive whale optimization algorithm for multilevel thresholding of color images

Sanjay Agrawal^a, Rutuparna Panda^{a,*}, Pratiksha Choudhury^a, Ajith Abraham^b

^a Department of Elect. & Telecomm. Engg., Veer Surendra Sai Univ. of Tech., Burla 768018, India

^b Machine Intelligence Research Labs, WA, USA

ARTICLE INFO

Article history:

Received 11 August 2021

Received in revised form 4 January 2022

Accepted 4 January 2022

Available online 10 January 2022

Keywords:

Multilevel color image segmentation

Dominant color component

Knowledge based systems

Adaptive whale optimization algorithm

Entropy

Edge magnitude

ABSTRACT

Optimal multilevel thresholding for image segmentation got much importance in recent years. Several entropic and non-entropic objective functions with evolutionary computing algorithms have been successfully implemented to get the optimal multilevel thresholds for gray scale images. The problem of multilevel thresholding becomes complex for color images. Because, the basic color components (red, blue, green) of the color image are extracted and the multiple optimum threshold values are calculated for each of the components separately. This makes the methods computationally intensive and inaccurate. Further, the required color information is not retained in the thresholded output. To solve these problems, an efficient technique is proposed in this paper, extracting only the dominant color component (DCC) of an image, for optimal thresholding. A novel segmentation score is introduced to justify the methodology. The optimum threshold values are obtained using a newly suggested evolutionary computing technique named adaptive whale optimization algorithm (AWOA). The main contributions are – (i) a novel DCC approach is introduced, (ii) an efficient optimizer AWOA is proposed, (iii) a new segmentation score is introduced, (iv) experimental results on standard test color images are explored. The outcomes are compared with all existing method's approaches (using all the RGB components) on color image thresholding. Its performance analysis using standard metrics is deliberated in detail. Statistical analysis is also performed. From the outcomes, it is perceived that the suggested DCC-AWOA concept yields high quality segmented images. The work may encourage further research to explore its high dimensional applications.

© 2022 Elsevier B.V. All rights reserved.

1. Introduction

As we know, image segmentation separates an image into distinct regions as per some inherent characteristic features. The various image segmentation techniques are classified based on – edge-based approach, region oriented approach, clustering based algorithms, histogram based thresholding, and so on [1]. One of the simplest and most commonly utilized techniques of segmentation is thresholding because of its accuracy and simplicity. The thresholding approaches are classified into bi-level and multilevel thresholding. In the bi-level case, only one threshold value is used to partition the image into two regions. This is because, it is believed that, the image comprises only two regions – background and object. However, real life images have multiple regions. So in multilevel thresholding, multiple threshold values are required to divide the image into multiple classes. The

implementation of multilevel thresholding yields more accurate results than bi-level thresholding due to the use of the different number of gray levels to denote the regions of the image. Hence, the problem of multilevel image thresholding is of prime concern among researchers working in the area of image processing.

There are two methods to find the optimum thresholds in the multilevel thresholding problem: parametric and non-parametric. The parametric method considers the distribution of probability density to define each class of the image. This approach is computationally intensive. On the other hand, the non-parametric approaches utilize discriminated procedures to divide the pixels into homogeneous regions [4]. Then a statistical criterion such as entropy, variance, or edge magnitude is used to obtain the thresholds. These criteria could also be used as optimization functions because of their accuracy. In recent years, many such criteria are proposed using between class variance (BCV) [5], minimum cross entropy (MCE) [6], Shannon's entropy [7], Tsallis entropy [8], Kapur's entropy [9], Renyi's entropy [10], Edge magnitude [11], Masi entropy [12], etc. A detailed survey on different thresholding techniques is undertaken in [13].

As we know, the computational complexity with these approaches increases exponentially as per the increase in threshold

* Corresponding author.

E-mail addresses: agrawals_72@yahoo.com (S. Agrawal), r_ppanda@yahoo.co.in (R. Panda), choudhurypratiksha@gmail.com (P. Choudhury), ajith.abraham@ieee.org (A. Abraham).

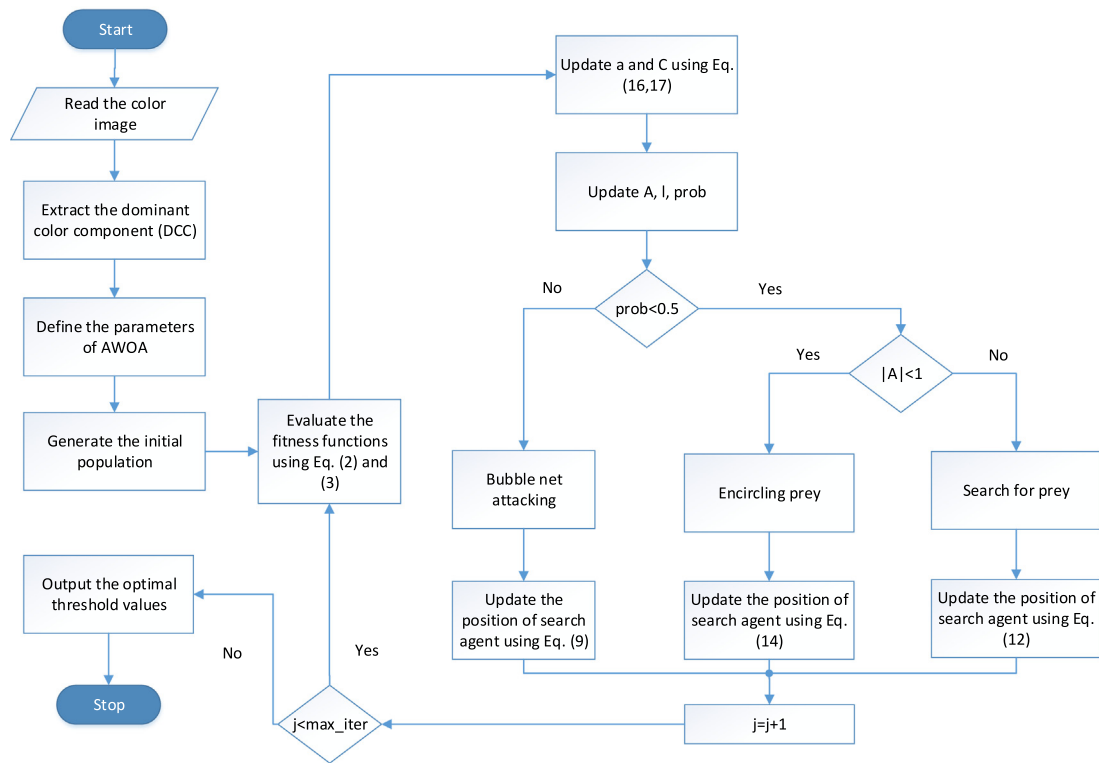


Fig. 1. Flow chart of the suggested multilevel thresholding method using DCC-AWOA.



Fig. 2. Original test color images (a) Lena, (b) 35 049, (c) 126 007, (d) 124 084, (e) 143 090, (f) 65 033, (g) 12 084, (h) 176 019, (i) 300 091, (j) 92 059, (k) a9 (TEST4) Ref. [2].

levels. Thus, different optimization techniques such as gravitational search algorithm (GSA) [14], differential evolution (DE)

[15], Darwinian particle swarm optimization (DPSO) [16], krill herd optimization (KHO) [17], bird mating optimization (BMO)

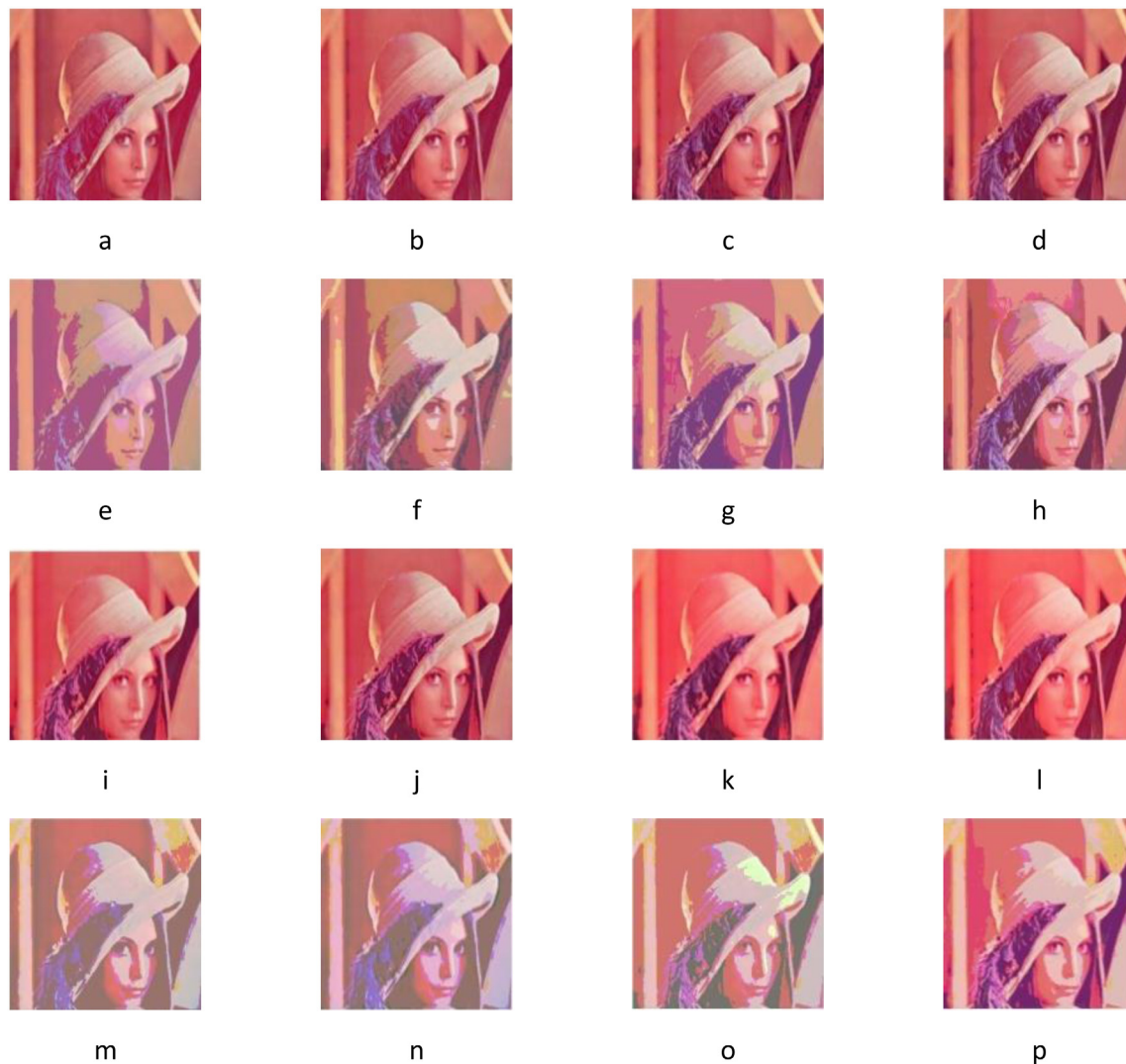


Fig. 3. Multilevel thresholded outputs of Lena image for level $M = 2, 3, 4$, and 5 , images (a)–(d) using KET on DCC, (e)–(h) using KET on RGB components, (i)–(l) using EMBT on DCC, (m)–(p) using EMBT on RGB components.

[18], dragonfly optimization (DFO) [19], spherical search optimization (SSO) [20], etc. are utilized to overcome these limitations. The optimal threshold values are obtained using different optimization techniques based on different fitness functions stated above.

Apart from the above optimization methods, Whale Optimization Algorithm (WOA) [21] is also utilized to solve the problem on hand [22]. The WOA as a modern and competitive population-based optimization procedure outclasses some other biologically inspired procedures from a simplicity and efficiency perspective. It is inspired by the social conduct of the humpback whales. The authors in [21] have claimed that WOA performs better in terms of accuracy and stability as compared to the other optimization methods when tested on standard benchmark functions.

This has inspired us to use WOA for the problem on hand. However, for global optimization problems, WOA can get trapped in local optima and degrade precision. Hence, an Adaptive Whale Optimization Algorithm (AWOA) for solving these problems is proposed in this paper. The investigation outcomes show the enhanced performance of AWOA w.r.t. solution precision and convergence speed in comparison with WOA. The suggested AWOA may be useful in the world of knowledge based systems.

It is well recognized that in practical engineering problems, a vast majority of images are color images, which are often complicated and have much detail. Hence, the problem of multilevel thresholding becomes more complex in the case of color images. Over the years, many works have been published in the area of multilevel thresholding of color images, suggesting a reduction in the computational complexity, for searching the optimal thresholds [23–32]. The authors in [1] proposed a modified firefly algorithm (MFA) for color image thresholding using all three components. They used BCV, MCE, and Kapur's entropy as their fitness functions. Their results were compared with various variations of the firefly technique. They used the chaotic map to initialize the firefly population for enhancing diversification. Further, the global search feature of PSO was used in the movement phase of fireflies.

The authors in [2] suggested Masi entropy based norm for color satellite image thresholding. They compared their suggested method with other entropy-based techniques such as Kapur's, Tsallis and Renyi's entropy. Besides satellite images, the authors also tested their method with various color images involving higher threshold levels. The authors in [33] proposed an efficient krill herd (EKH) method for color image thresholding utilizing all the RGB (R (red), G (green), B (blue)) color components. They used Tsallis, Kapur's entropy and Otsu's criteria as the cost functions.



Fig. 4. Multilevel thresholded outputs of 35049 image for level $M = 2, 3, 4$, and 5 , images (a)–(d) using KET on DCC, (e)–(h) using KET on RGB components, (i)–(l) using EMBT on DCC, (m)–(p) using EMBT on RGB components.

The authors used seven more algorithms for comparison. Pare et al. [34] proposed their approach using Renyi's entropy as the cost function and bat algorithm for optimization. The authors in [35] proposed a multilevel thresholding technique for color images in which minimum class variance thresholding and the Otsu technique are utilized to find the thresholds. Hussein et al. [36] suggested a Patch-Levy-based Bees Algorithm (PLBA) based fast procedure for multilevel thresholding. They used Kapur's entropy and Otsu's criteria as the cost functions and compared their results with bacteria foraging and quantum inspired schemes. Xing [37] suggested an improved emperor penguin optimization (EPO) based multi-threshold color image segmentation technique. They also used Kapur's entropy as the fitness function for computing the multiple thresholds. Anitha et al. [3] suggested a modified whale optimization algorithm (MWOA) for color image thresholding. They used Kapur's entropy and Otsu's criteria as the fitness function.

The color image contains three primary color components R (red), G (green), and B (blue). It is observed from the above discussions that all the three primary color components are used for thresholding in all the methods. All the above mentioned approaches search the optimal thresholds for each of the RGB color components separately and results are computed based on their average values. This increases the system complexity and

the computational load three times. Nevertheless, it is an injustice to say that the earlier approaches are inaccurate because other color components carry insignificant information required by the entropic/non-entropic functions utilized for solving the multilevel thresholding problem. In this context, it becomes a challenging task to quickly obtain the required information from a multicolor histogram.

This has inspired us to present an efficient method to meet the requirements of color image segmentation. We propose the dominant color component (DCC) based multilevel thresholding technique using AWOA. In the suggested method, only the dominant color component is extracted from the color image, which is a novel concept in this domain. The optimum threshold values are computed for the dominant component only using the suggested AWOA. This results in a significant decrease in the computation time together with better segmented images. In summary, the proposal ensures better accuracy and speed.

The proposed idea of DCC-AWOA is validated using (i) an entropic objective function such as Kapur's entropy thresholding (KET), (ii) a non-entropic objective function such as the edge magnitude based thresholding (EMBT). A comprehensive experiment is conducted with images from the BSDS500 dataset [38]. The outcomes are compared with all the state-of-the-art method's approaches on color image thresholding employing the RGB color

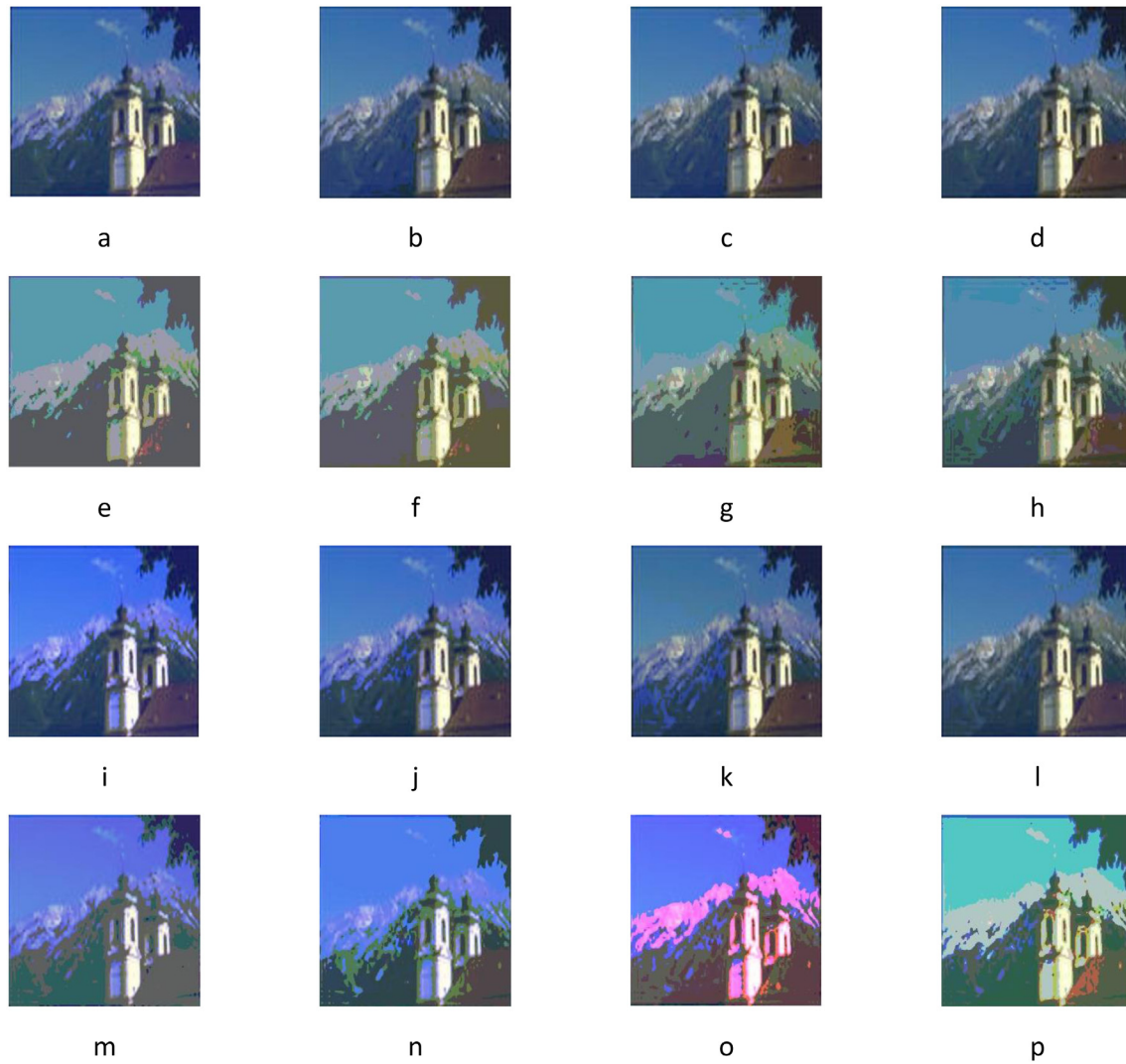


Fig. 5. Multilevel thresholded outputs of 126007 image for level $M = 2, 3, 4$, and 5 , images (a)–(d) using KET on DCC, (e)–(h) using KET on RGB components, (i)–(l) using EMBT on DCC, (m)–(p) using EMBT on RGB components.

components separately. It is observed that thresholding the DCC only using DCC-AWOA yields high quality segmented images as compared to the RGB approach.

The contributions of this work are: (i) An efficient DCC based method for multilevel thresholding of color images is proposed resulting in improved accuracy. The original color information is retained in the thresholded image. (ii) A novel segmentation score to justify the contribution is investigated. (iii) The optimization capability of WOA is enhanced by introducing Levy flight feature and updating the coefficient vectors using self-adaptive strategy based on the fitness function value to obtain the optimal thresholds. (iii) The adaptability of the proposed AWOA to color image segmentation is investigated. (iv) The effectiveness of the proposal is justified through extensive experimental results. The novelties are: (i) For the first time the DCC based approach for color image thresholding is proposed. (ii) A new parameter – segmentation score is defined to compare the DCC based approach with the RGB based approach. (iii) The WOA is improved by introducing Levy flight feature and adaptive coefficient vectors for enhanced convergence and faster computation.

The structure of the paper is: Section 2 presents the material and methods. Section 3 illustrates the suggested methodology.

The numerical and the statistical results of the suggested technique followed by a detailed discussion are presented in Section 4. Lastly, conclusive remarks with the future scope of the work are drawn in Section 5.

2. Material and methods

2.1. Entropic method: Kapur's entropy thresholding (KET)

Kapur's entropy is one of the widely used entropy based cost functions for multilevel thresholding. It has pulled the attention of many researchers and is widely utilized for solving the multilevel thresholding problem. Let $[T_1, T_2, \dots, T_n]$ represents the n optimal thresholds that partition the image into different classes. The relationship between the multiple thresholds and Kapur's entropy criterion is represented as:

$$H(T_1, T_2, \dots, T_n) = H_0 + H_1 + \dots + H_n \quad (1)$$

$$\text{where } H_0 = - \sum_{g=0}^{T_1-1} \frac{p(g)}{\omega_0} \log \frac{p(g)}{\omega_0}, \quad \omega_0 = \sum_{g=0}^{T_1-1} p(g);$$

$$H_1 = - \sum_{g=T_1}^{T_2-1} \frac{p(g)}{\omega_1} \log \frac{p(g)}{\omega_1}, \quad \omega_1 = \sum_{g=T_1}^{T_2-1} p(g);$$

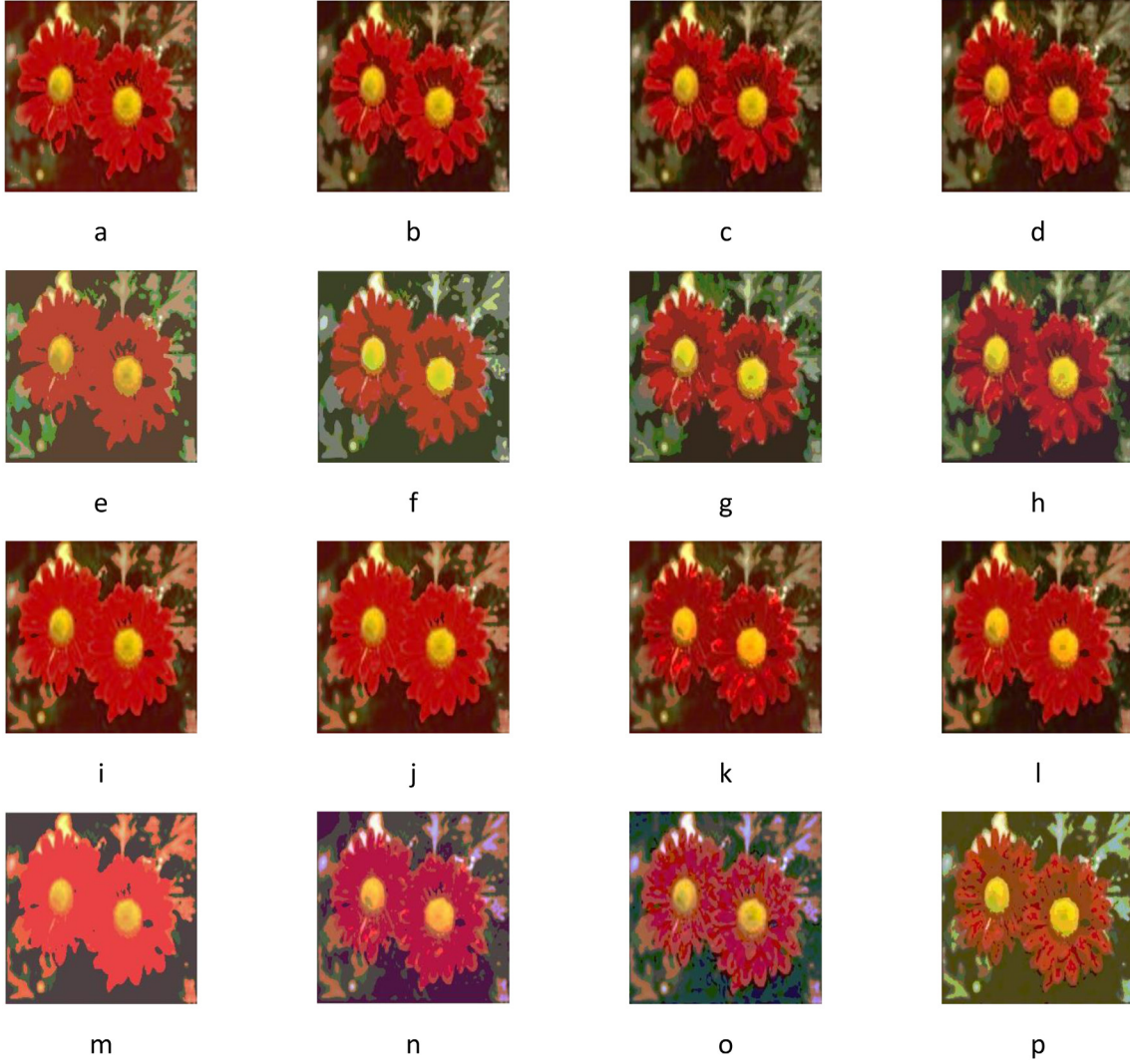


Fig. 6. Multilevel thresholded outputs of 124084 image for level $M = 2, 3, 4$, and 5 , images (a)–(d) using KET on DCC, (e)–(h) using KET on RGB components, (i)–(l) using EMBT on DCC, (m)–(p) using EMBT on RGB components.

\vdots

$$H_n = - \sum_{g=T_n}^{L-1} \frac{p(g)}{\omega_n} \log \frac{p(g)}{\omega_n}, \quad \omega_n = \sum_{g=T_n}^{L-1} p(g).$$

Here $H_0 + H_1 + \dots + H_n$ represents the entropies of the different classes, $\omega_0, \omega_1, \dots, \omega_n$ denotes the probabilities of each class. The optimum thresholds are found when Eq. (2) is maximized [9].

$$f_{KET}(T_1, T_2, \dots, T_n) = \arg \max \{H(T_1, T_2, \dots, T_n)\} \quad (2)$$

2.2. Non-entropic method: Edge magnitude based thresholding (EMBT)

One of the popular non-entropic objective functions utilized for multilevel thresholding is edge magnitude based. The idea of edge magnitude was discussed in [39] for bi-level thresholding of gray scale images. The extension to multilevel thresholding was implemented in [11]. The edge magnitude q calculated from the gray level co-occurrence matrix (GLCM) is considered for optimal thresholding. It is calculated from the position of the pixel pair. The image is optimally thresholded when the total edge information is maximum. Here also $[T_1, T_2, \dots, T_n]$ represents the n

optimal thresholds that partition the image into different classes. The threshold values are obtained as:

$$T_1 = \arg \max \left(\frac{1}{\eta_1} \sum_{i=0}^{q_1} \sum_{j=q_1+1}^{q_2} \frac{i+j}{2} G(i, j) \right);$$

$$T_2 = \arg \max \left(\frac{1}{\eta_2} \sum_{i=q_1+1}^{q_2} \sum_{j=q_2+1}^{q_3} \frac{i+j}{2} G(i, j) \right);$$

\vdots

$$T_n = \arg \max \left(\frac{1}{\eta_n} \sum_{i=q_{n-2}+1}^{q_{n-1}} \sum_{j=q_{n-1}+1}^{q_n} \frac{i+j}{2} G(i, j) \right),$$

$$\text{where } \eta_1 = \sum_{i=0}^{q_1} \sum_{j=q_1+1}^{q_2} G(i, j);$$

$$\eta_2 = \sum_{i=q_1+1}^{q_2} \sum_{j=q_2+1}^{q_3} G(i, j);$$

$$\eta_n = \sum_{i=q_{n-2}+1}^{q_{n-1}} \sum_{j=q_{n-1}+1}^{q_n} G(i, j).$$

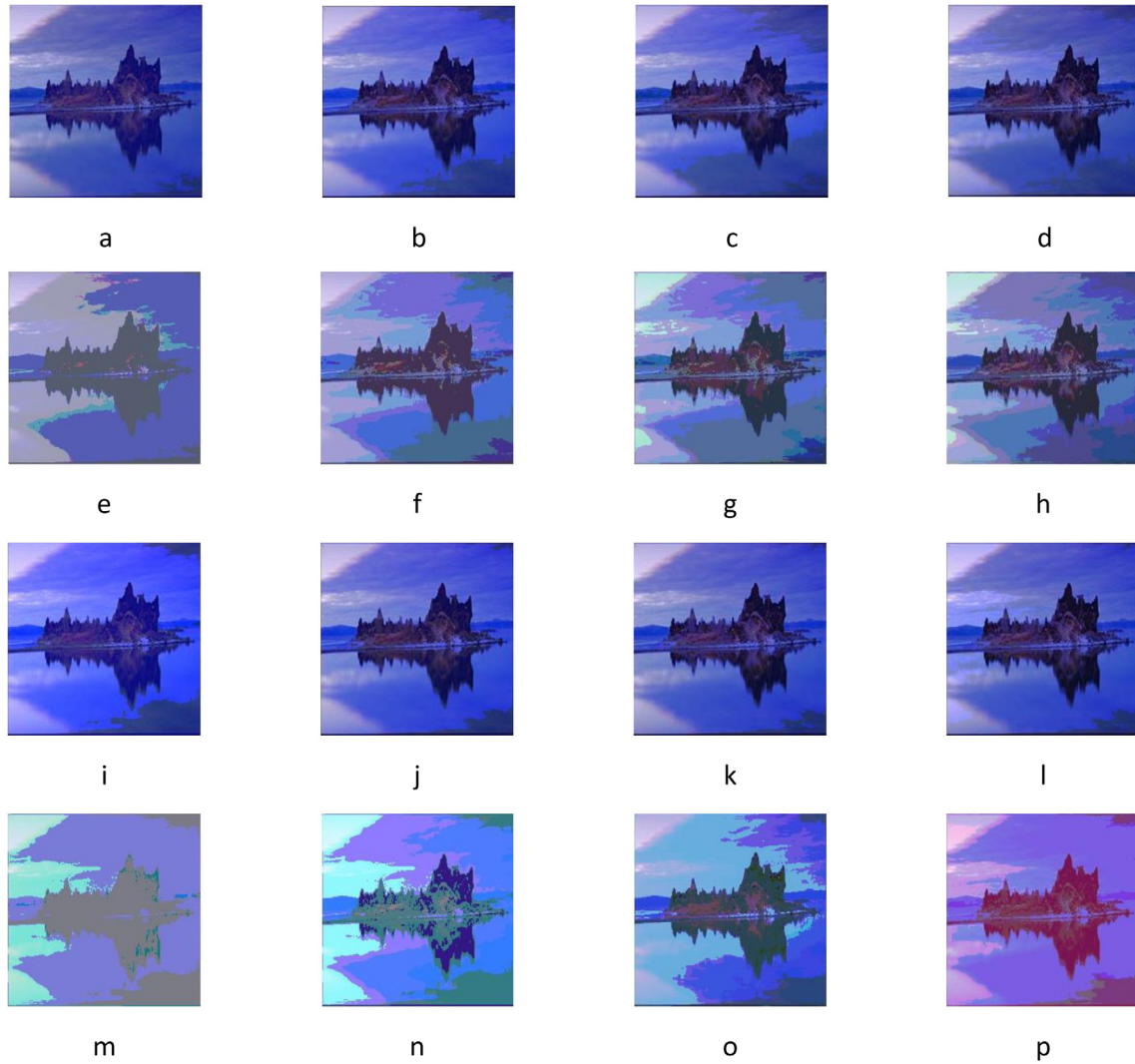


Fig. 7. Multilevel thresholded outputs of 143090 image for level $M = 2, 3, 4$, and 5 , images (a)–(d) using KET on DCC, (e)–(h) using KET on RGB components, (i)–(l) using EMBT on DCC, (m)–(p) using EMBT on RGB components.

The variable η signifies the count of pixel pairs in the GLCM whose magnitude is $\geq q$. The variable G represents the overall GLCM obtained from the average of all GLCMs. The proper choice of q is vital for the thresholding results. The thresholds obtained are optimum when the objective function given in Eq. (3) is maximized [11].

$$f_{EMBT} [T_1, T_2, \dots, T_n] = \arg \max \{f(q_1, q_2, \dots, q_n)\} \quad (3)$$

2.3. Whale optimization algorithm (WOA)

The WOA suggested in [21] is a nature-inspired optimization procedure. It is a relatively new technique that imitates the humpback whales chasing strategy. To hunt, the whales make spiral bubbles, then follow the trajectory to reach the prey. The whales can identify the position of the target and enclose them. There are two phases of the optimization process: (i) exploitation phase includes encircling the prey and bubble net attacking and (ii) exploration phase includes a random search for the target.

In the first phase (exploitation stage), the whales find the current best position as the target (global optimum) and change their locations towards the target using the given equations below.

$$\vec{D} = \left| \vec{C} \cdot \vec{X}_b(t) - \vec{X}(t) \right| \quad (4)$$

$$\vec{X}(t+1) = \vec{X}_b(t) - \vec{A} \cdot \vec{D} \quad (5)$$

where the variable t indicates the current iteration, the parameters \vec{A} and \vec{C} are the coefficient vectors, the variable \vec{X}_b is the location vector of the best solution found so far, $|\cdot|$ denotes the absolute value operation, \cdot denotes element-by-element multiplication, and the variable \vec{X} is the location vector of the whale [21]. It is to be noted that \vec{X}_b needs to be updated in each iteration if a better solution is reached. The parameters \vec{A} and \vec{C} are computed as follows:

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \quad (6)$$

$$\vec{C} = 2\vec{r}_2 \quad (7)$$

where values of \vec{a} linearly decreases from two to zero throughout the iterations (in both exploitation and exploration stages). The variables \vec{r}_1, \vec{r}_2 signify random vectors within $[0, 1]$. The parameters \vec{A} and \vec{C} signify the distance between the optimal location and the updated location. In the bubble net attack method, the whales attack their target utilizing the bubble net strategy. This behavior is mathematically represented as:

$$\vec{D}' = \left| \vec{X}_b(t) - \vec{X}(t) \right| \quad (8)$$

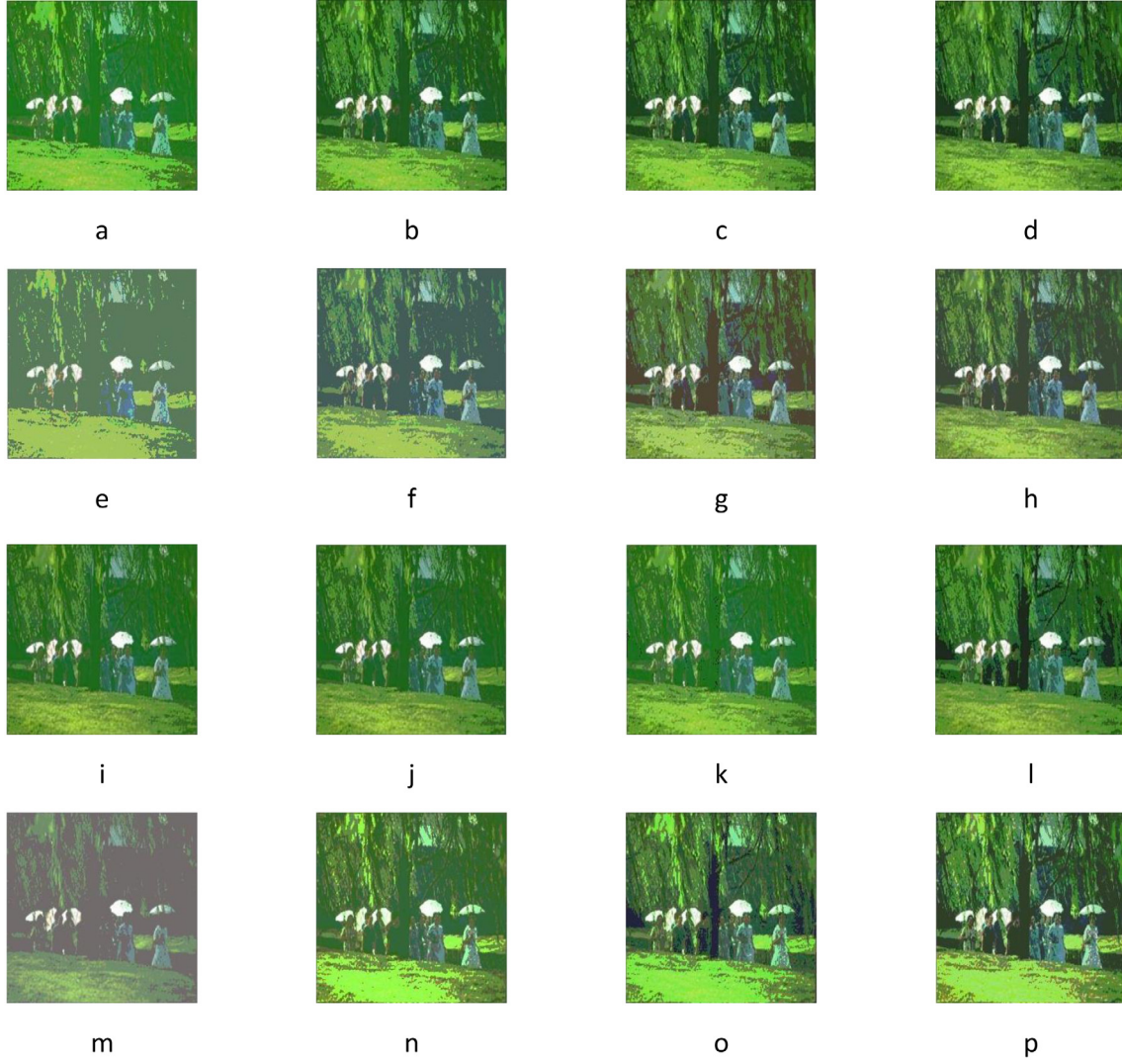


Fig. 8. Multilevel thresholded outputs of 65033 image for level $M = 2, 3, 4$, and 5 , images (a)–(d) using KET on DCC, (e)–(h) using KET on RGB components, (i)–(l) using EMBT on DCC, (m)–(p) using EMBT on RGB components.

$$\vec{X}(t+1) = \vec{D}' \cdot e^{pl} \cdot \cos(2\pi l) + \vec{X}_b(t) \quad (9)$$

where \vec{D}' indicates the location of the current whale location from the target. The variable p is a constant. The variable l signifies a random quantity in the range $[-1, 1]$. It is worth noting that, as the whales capture their prey, the encircling prey step and the bubble net attacking step is performed with a 50 percent likelihood. Hence, the location of the search agents in the exploitation phase is updated utilizing the given equation below.

$$\vec{X}(t+1) = \begin{cases} \vec{X}_b(t) - \vec{A} \cdot \vec{D} & \text{prob} < 0.5 \\ \vec{D}' \cdot e^{pl} \cdot \cos(2\pi l) + \vec{X}_b(t) & \text{prob} \geq 0.5 \end{cases} \quad (10)$$

here 'prob' is a random variable in the range $[0, 1]$.

In the second phase (exploration stage), a global search to improve the exploration capability is built. The whales hunt for the target randomly. The location of the search agents is updated as per a randomly selected agent. The equation for the random search is:

$$\vec{D} = |\vec{C} \cdot \vec{X}_{rand} - \vec{X}| \quad (11)$$

$$\vec{X}(t+1) = \vec{X}_{rand} - \vec{A} \cdot \vec{D} \quad (12)$$

where \vec{X}_{rand} denotes a random location vector. Further, it is to be noted that to update the location, the coefficient vector $|\vec{A}| > 1$ or $|\vec{A}| < -1$ is used to select either the exploration or the exploitation phase.

3. Proposed methodology

In this work, a novel method to enhance the computational efficiency and produce high quality segmented images using the DCC based optimal multilevel thresholding and AWOA is suggested. Kapur's entropy and edge magnitude are utilized as the fitness functions to find the optimal thresholds. Only the dominant component of the color image is thresholded without the need to threshold the remaining two color components.

3.1. Adaptive whale optimization algorithm (AWOA)

It is worth noting here that WOA quickly transitions between the phases of exploration and exploitation, based on a single parameter. While WOA is good compared to some conventional optimization algorithms mentioned above, it still has some drawbacks in solving global optimization problems like low precision and early convergence. This section suggests an adaptive WOA

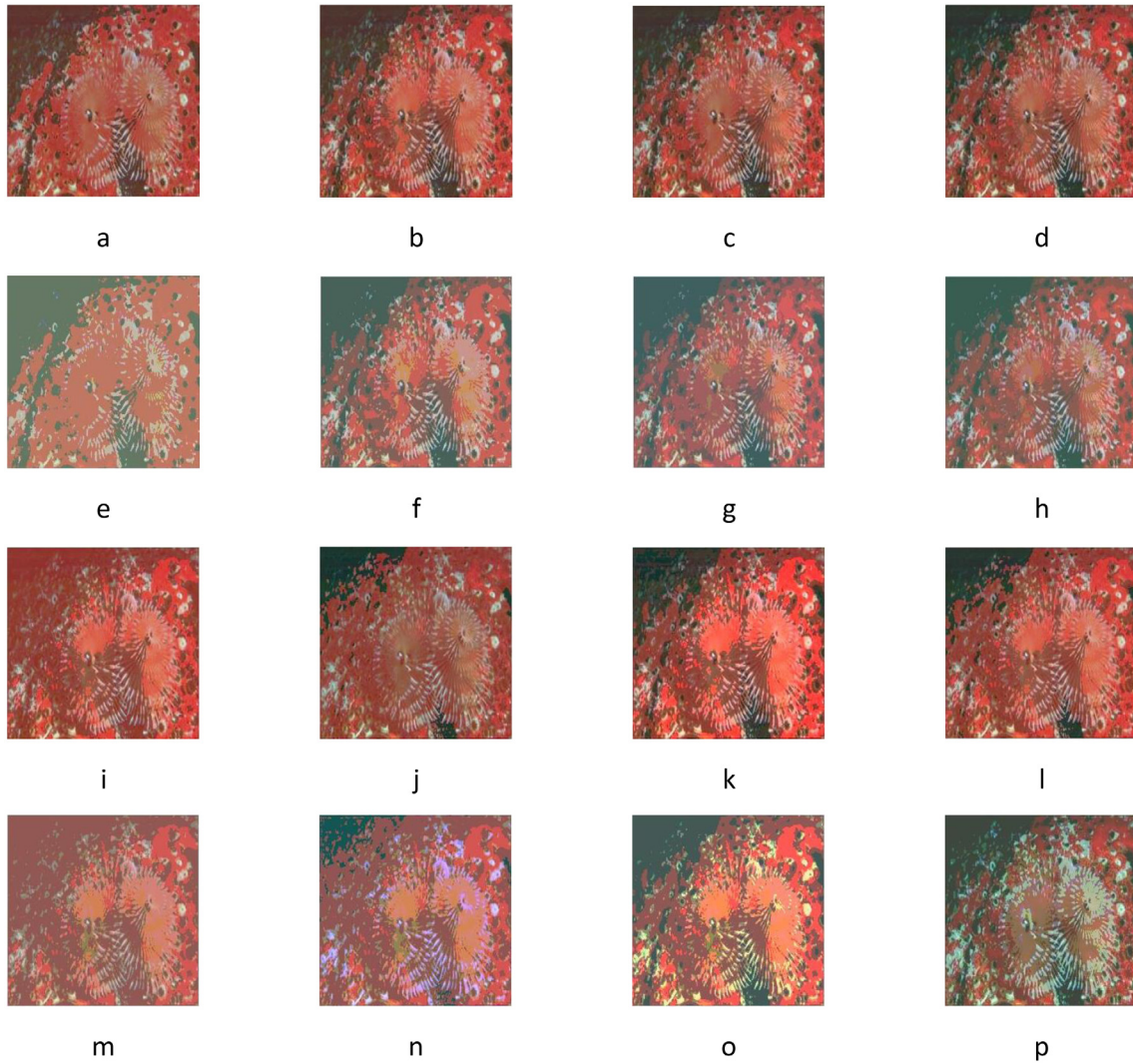


Fig. 9. Multilevel thresholded outputs of 12084 image for level $M = 2, 3, 4$, and 5 , images (a)–(d) using KET on DCC, (e)–(h) using KET on RGB components, (i)–(l) using EMBT on DCC, (m)–(p) using EMBT on RGB components.

(AWOA) inspired by Lévy flight and self-adaptive coefficient vectors for global optimization problems. Owing to the frequent search stage for short distances and sometimes the search stage for longer distances, the Lévy flight enables WOA to hop out of the local optima. Further, the self-adaptive coefficient vectors increase the precision of the solution by improving its exploitability. The following segment explains the adaptive WOA with a stronger trade-off between exploitation and exploration.

It is presented in the literature that most of the optimization techniques converge early towards local optima because of the quick decline of diversity. This behavior is also seen in WOA. Many authors have used the Lévy flight feature [40–42] to accelerate the convergence speed. Therefore, Lévy flight is utilized here also to enhance the convergence [43]. The authors in [43] presented the Lévy distribution as:

$$L(\lambda) \sim |\lambda|^{-1-\beta}, 0 < \beta \leq 2 \quad (13)$$

Here, β denotes an index, the parameter λ signifies the step size of the Lévy distribution. It is seen from the literature that Mantegna's algorithm is used to compute the value of λ . Because of the infinite variation of the Lévy distribution, the Lévy flight conducts the long-distance movement periodically to encourage the exploration capability, while the short-distance movement is performed to improve the exploitation capability. In AWOA,

the shrinking encircling process is substituted by Lévy flight for more effective exploration. The new location is revised as per the amendment below.

$$\vec{X}(t+1) = \vec{X}_b(t) + \text{sign}(r - 0.5) \oplus \text{Levy} \quad (14)$$

where

$$\text{Levy} = \text{rand}(\text{size}(D)) \oplus L(\beta) \sim 0.01\mu/|v|^{1/\beta}(\vec{X}_i(t) - \vec{X}_b(t)).$$

Here, $\text{size}(D)$ indicates the scale of the problem, \oplus indicates pointwise multiplication, X_i denotes the i th solution vector, t represents the current iteration, $\text{sign}(\cdot)$ represents the sign function having three values only $[-1, 0, 1]$ resulting in a more random search, r indicates a random number within $[0, 1]$. The rest of the variables carry the same meaning as defined in [44]. Now Eq. (10) is redefined as follows in AWOA.

$$\vec{X}(t+1) = \begin{cases} \vec{X}_b(t) + \text{sign}(r - 0.5) \oplus \text{Levy} & \text{prob} < 0.5 \\ \vec{D} \cdot e^{pl} \cdot \cos(2\pi l) + \vec{X}_b(t) & \text{prob} \geq 0.5 \end{cases} \quad (15)$$

It is observed that the authors in WOA have varied the value of \vec{A} for achieving exploration and exploitation. As stated above, $|\vec{A}| > 1$ takes the algorithm to search space and $|\vec{A}| \leq 1$ takes it for attacking the target. Note that the parameter \vec{a} is reduced from two to zero to vary $|\vec{A}|$. But this cannot adjust

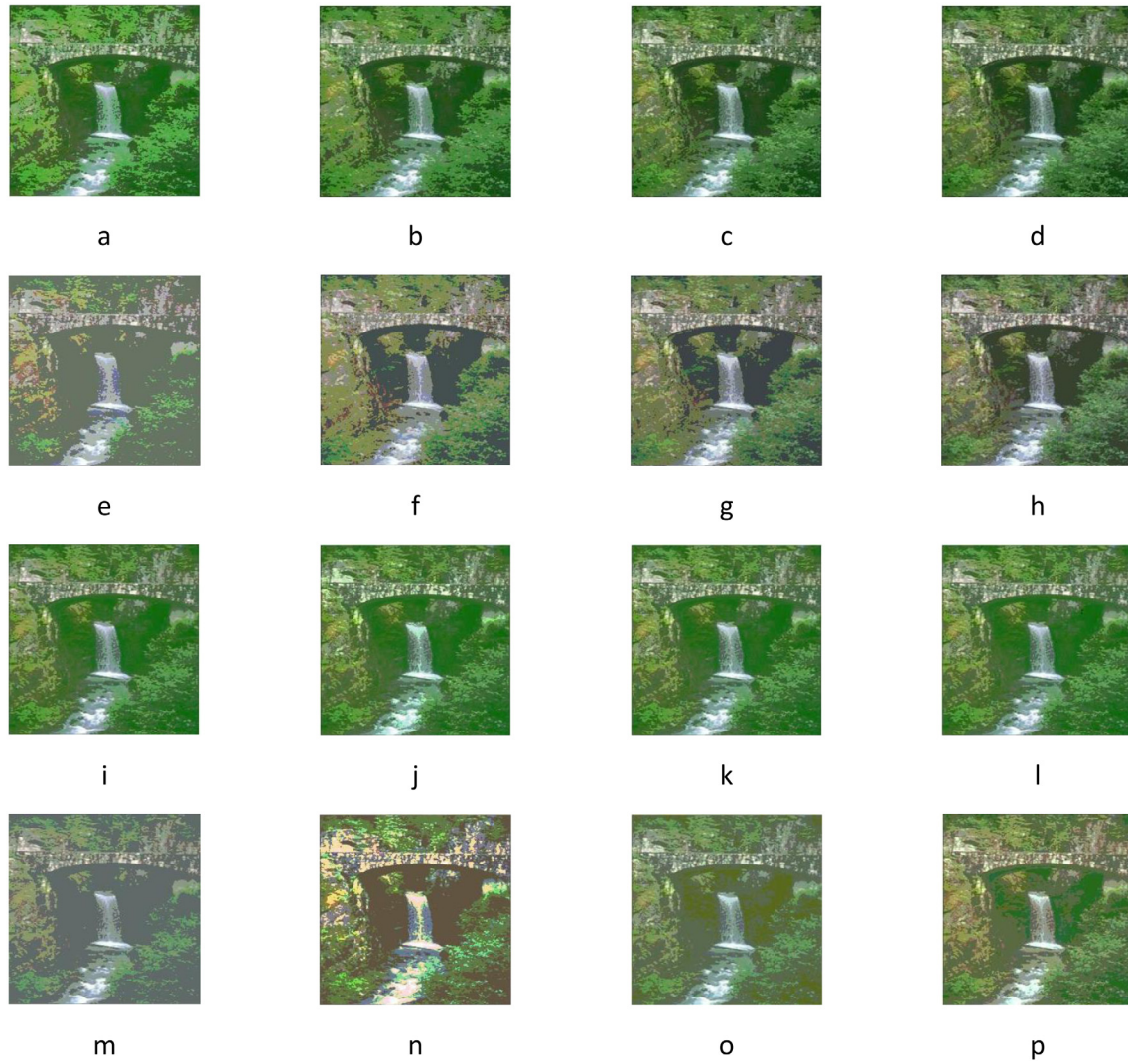


Fig. 10. Multilevel thresholded outputs of 176019 image for level $M = 2, 3, 4$, and 5 , images (a)–(d) using KET on DCC, (e)–(h) using KET on RGB components, (i)–(l) using EMBT on DCC, (m)–(p) using EMBT on RGB components.

to a complicated and non-linear search process. The proposed AWOA includes features with less parameter dependency. The value of the control parameter \vec{a} and the coefficient vector \vec{C} is changed adaptively according to the objective function value over the course of an iteration. So adaptive technique requires less execution time to reach an optimum solution resulting in faster convergence. Here the parameters \vec{a} in Eq. (6) and \vec{C} in Eq. (7) are computed as:

$$\vec{a} = \frac{f_{\max} - f_{\min}}{f_{\max} + f_{\min}} \times \frac{\vec{r}_1}{t} \quad (16)$$

$$\vec{C} = \frac{f_{\max} - f_{\min}}{f_{\max} + f_{\min}} \times \frac{\vec{r}_2}{t} \quad (17)$$

where f_{\max} is the maximum fitness value and f_{\min} is the minimum fitness value of the search agents. The parameter t signifies the recent iteration number. In this context, during the early stages when t is small, a large range of search space can be explored whereas a smaller range will be explored at a later stage when t is large. Note that AWOA inherits all other features from WOA.

The parts of WOA which have changed are (i) A Lévy-flight strategy for aiding the algorithm leap out of the local optima is adopted [43,44], (ii) A self-adaptive strategy based on the fitness

function value is proposed for updating the coefficient vectors to enrich the exploration and exploitation capabilities.

3.2. Proposed dominant color component (DCC) model

There are many ways of finding the dominant color in a color image [45]. All the pixels of individual color components (RGB) are counted. Then the index of the maximum count corresponding to a particular color gives the dominant color. An alternative way is to compute the average intensity values of each of the color components of the input image. The highest average value among them is considered as the dominant color. In this paper, the latter approach is used to determine the dominant color.

To justify our claim that the suggested technique is better than the RGB based approaches, a novel segmentation score is introduced. Let $I \in \mathbb{R}^N$ be a multimodal input image with N pixels, having intensity values from 0 to 255, to be segmented. Let us compute the optimal threshold values of the segmentation process that, when applied to I , maximize the optimization criteria with respect to the other areas (regions). Let us consider bi-level thresholding of the input image I as $S \in \{0, 1\}^N$ using Otsu's optimization criteria. The optimal threshold T obtained from Otsu's criteria is used to divide the image I into two classes. Class 0 contains the pixels from 0 to T , Class 1 contains the pixels

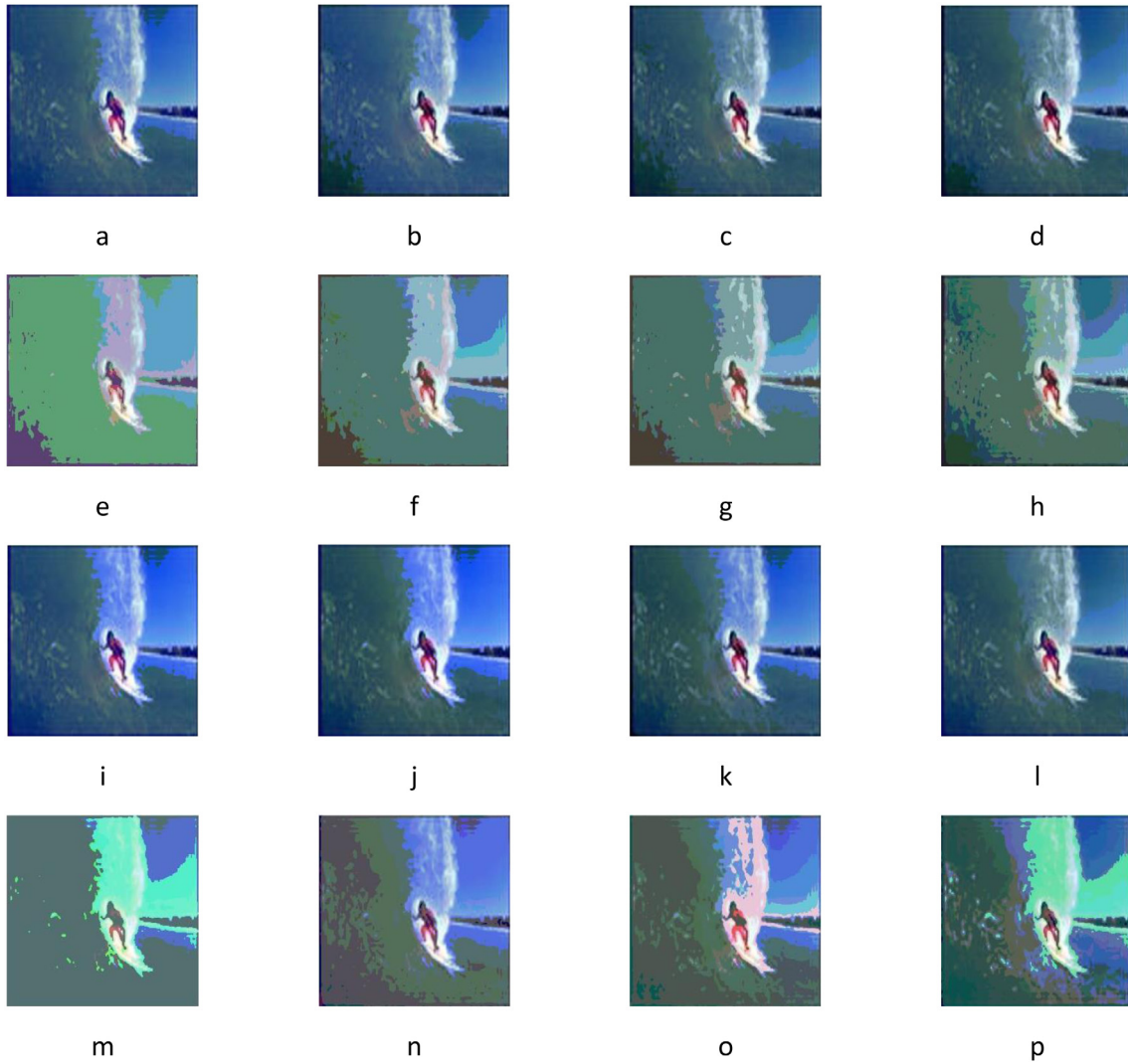


Fig. 11. Multilevel thresholded outputs of 300091 image for level $M = 2, 3, 4$, and 5 , images (a)–(d) using KET on DCC, (e)–(h) using KET on RGB components, (i)–(l) using EMBT on DCC, (m)–(p) using EMBT on RGB components.

from $T + 1$ to 255 . Note that S_k decides whether the k th pixel belongs to class 0 or 1. The sum of the squared error for the segmented image S is given as:

$$J = \sum_{S_k=0} (I_k - \mu_0)^2 + \sum_{S_k=1} (I_k - \mu_1)^2 \quad (18)$$

where I_k denotes the intensity of the k th pixel in class 0 or 1, $\mu_0 = \sum_{S_k=0} \frac{I_k}{n_S}$, $\mu_1 = \sum_{S_k=1} \frac{I_k}{(N-n_S)}$ represents the mean values of class 0 and 1 respectively, and n_S represents the number of pixels in class 0. Thus, the segmentation error is given as:

$$E = \|I\|_2^2 - \frac{1}{n_S} \left(\sum_{S_k=0} I_k \right)^2 - \frac{1}{N - n_S} \left(\sum_{S_k=1} I_k \right)^2 \quad (19)$$

To further simplify Eq. (19), we assume the second and the third term to be equal, then it can be represented as:


$$E = \|I\|_2^2 - \frac{N}{n_S(N - n_S)} \left(\sum_{S_k=0} I_k \right)^2 \quad (20)$$

The error in segmentation in Eq. (20) can be minimized by maximizing the second term. Now we define the segmentation score

as:

$$\psi_{I,T} = \max_{S \in \{0,1\}^N} \frac{N}{n_S(N - n_S)} \left(\sum I_k \right) \quad (21)$$

It is essential to note here that the segmentation score is determined by the threshold value. Interestingly, the segmentation score is more for the dominant color component based segmentation as compared to the RGB components based segmentation methods. Simple computations, for example standard color image, are shown below.

Image 35 049	Dominant color	Segmentation score	
		DCC based	RGB based
	GREEN	667.2330	362.4446

The image 35 049 is taken here for demonstration. The dominant color is green. The segmentation score for the DCC based

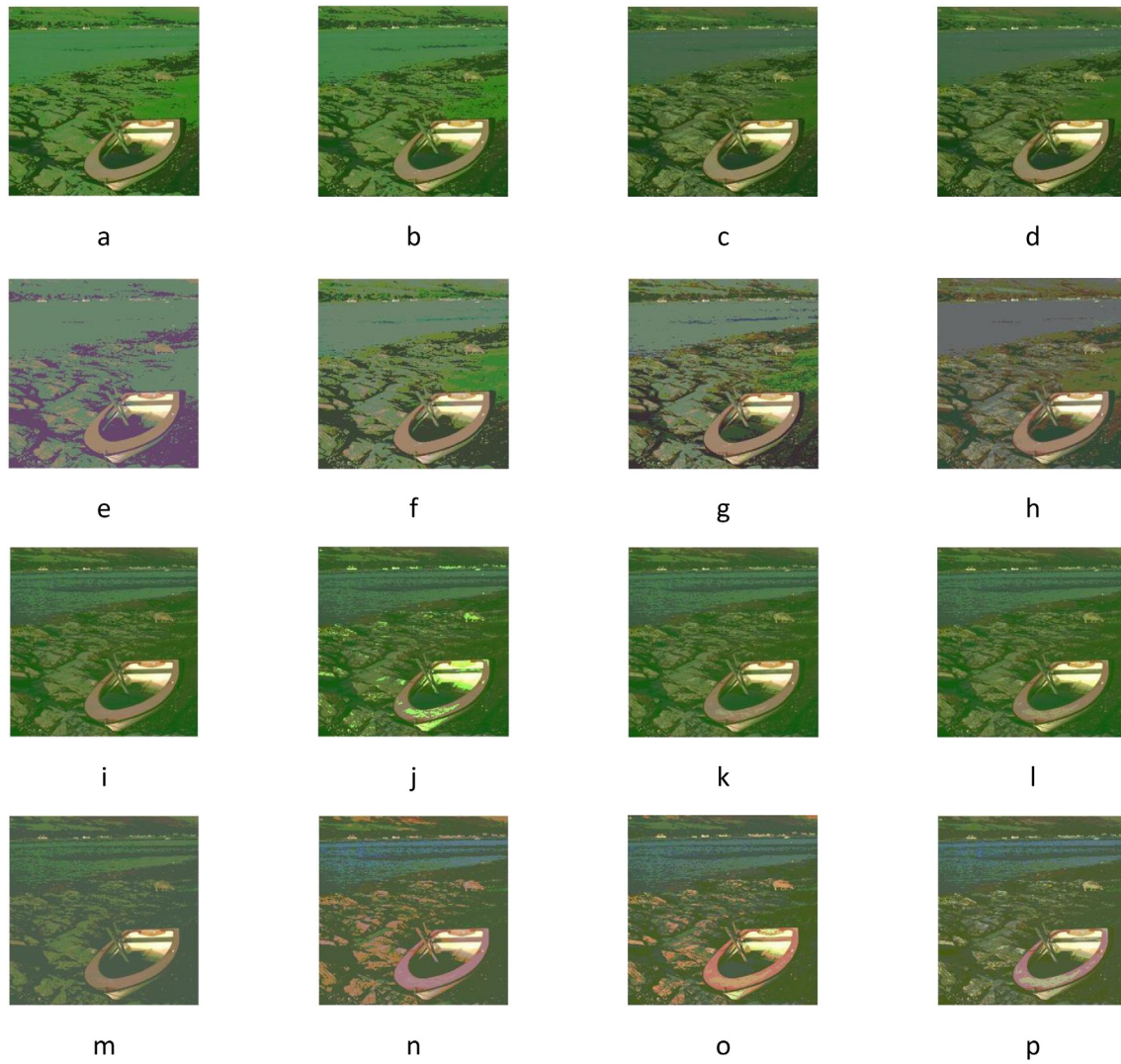


Fig. 12. Multilevel thresholded outputs of 92 059 image for level $M = 2, 3, 4$, and 5 , images (a)–(d) using KET on DCC, (e)–(h) using KET on RGB components, (i)–(l) using EMBT on DCC, (m)–(p) using EMBT on RGB components.

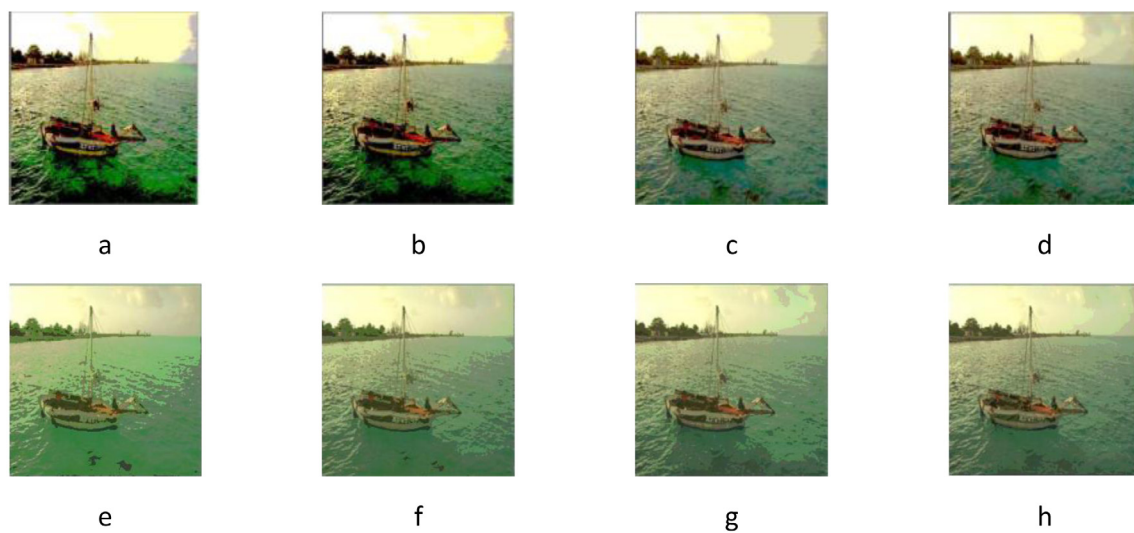


Fig. 13. Multilevel thresholded outputs of a9 (TEST4) image for level $M = 3, 5, 8$, and 12 , images (a)–(d) using Kapur's entropy in Ref. [2], (e)–(h) using proposed method.

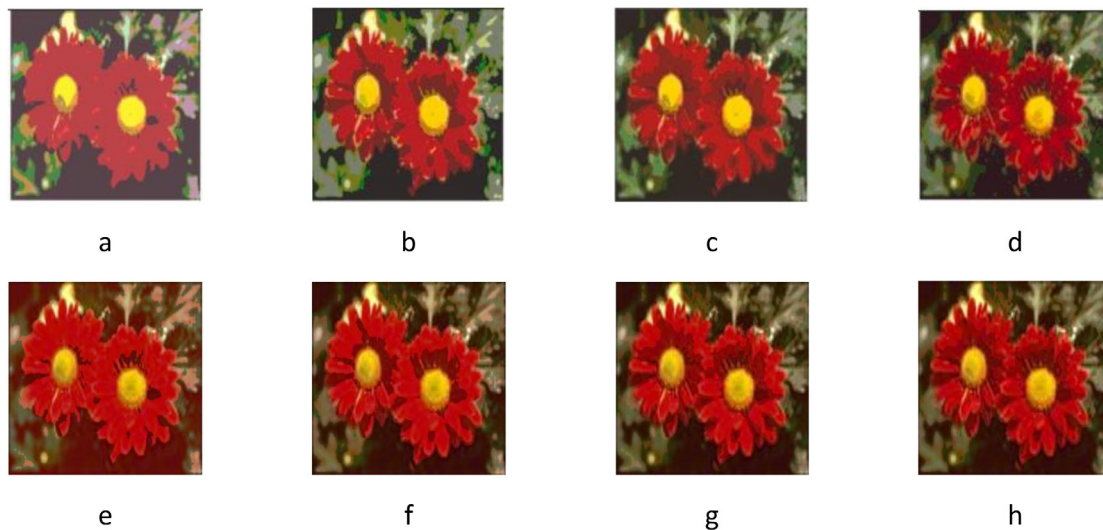


Fig. 14. Multilevel thresholded outputs of *flower* image for level $M = 2, 3, 4$, and 5 , images (a)–(d) using Kapur's entropy in Ref. [3], (e)–(h) using proposed method.

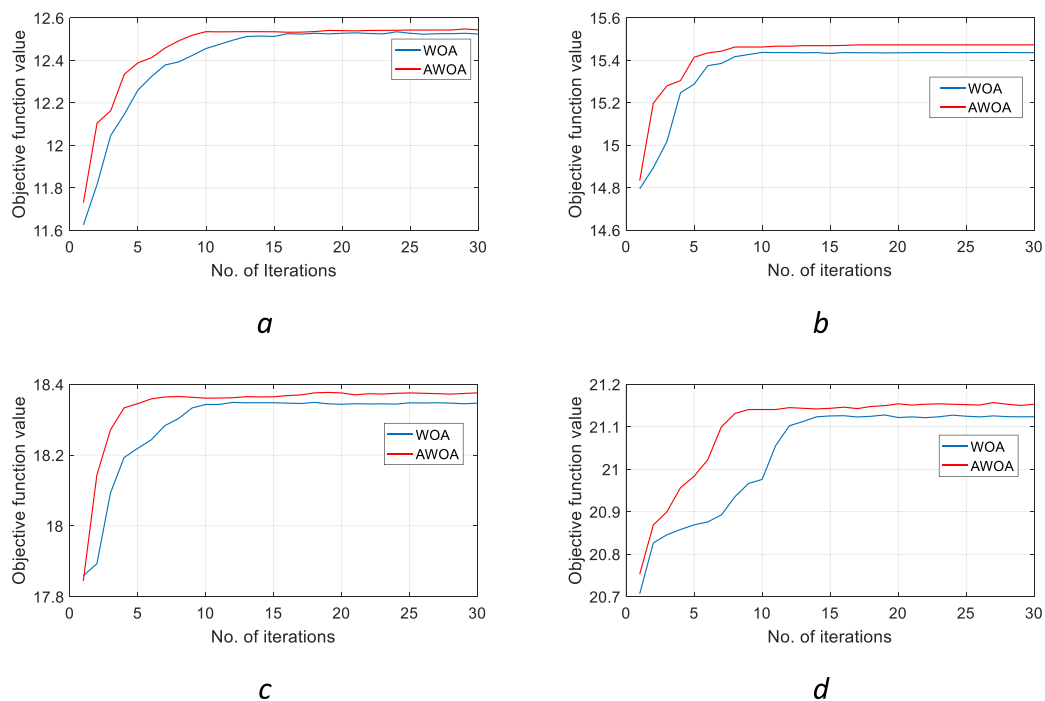


Fig. 15. Convergence curves for DCC with KET for different threshold levels. (a) $M = 2$, (b) $M = 3$, (c) $M = 4$, (d) $M = 5$.

approach is found to be 667.2330, whereas it is 362.4446 for the RGB based approach. Note that the segmentation score for the RGB based approach is computed by averaging the scores of all the three components R, G, B individually. These values implicit the expected research outcomes.

3.3. The suggested multilevel thresholding technique

The suggested multilevel thresholding process is investigated to get the optimal thresholds for color image segmentation. The technique uses the objective functions defined in Eqs. (2) and (3). The objective functions are maximized using the proposed AWOA. Conventionally, the optimal thresholds are searched for each component. However, the proposed method extracts only the dominant component. The optimal thresholds are searched for this component only. The flow diagram of the suggested method

is presented in Fig. 1. It is interesting to note that the computational complexity of the suggested method is reduced from the complexity of $O(3L^{2M})$ to $O(L^{2M})$ for M -level thresholding.

The pseudo-code for the proposed procedure is shown below:

4. Results and discussions

Segmentation of images or thresholding of a digital image can be subjective. To claim that one approach provides a better image of quality may vary from person to person. For this purpose, quantitative/empirical measures must be developed to compare the effects of image segmentation algorithms on image quality. Utilizing the same set of test images, various image segmentation algorithms can be systematically compared to determine whether a given algorithm is producing better results. If we can prove that

Pseudo-code for the suggested DCC-AWOA**Begin**

Read the color image

Compute the dominant color

Input: Population, dimension, Number of iterations (max_iter)Set the whale location \vec{X}_i Let \vec{X}_b = the best search agent**while** ($j < max_iter$)**for each** search agent

Compute the objective function value utilizing Eqn. (2) and (3)

 Update \vec{a} and vector \vec{C} using Eqn. (16, 17), Update \vec{A} , l , $prob$ **if1** ($prob < 0.5$) **if2** ($|\vec{A}| < 1$)

Update the position of the current search agent utilizing Eqn. (14)

else if2 ($|\vec{A}| \geq 1$) Pick a random search agent \vec{X}_{rand}

Update the position of the current search agent using Eqn. (11, 12)

endif2 **elseif1** ($prob \geq 0.5$)

Update the position of the current search agent using Eqn. (8, 9)

end if1**end for**

Test for search space violation and correct it

 Update \vec{X}_b if there is a better solution $j=j+1$ **end while****Output:** \vec{X}_b

Set the threshold values linking to the best positions.

Threshold the dominant color component image and replace that component only in the original RGB image

keeping other color component images unchanged

End

a degraded known image can be improved by an algorithm or group of algorithms to closely resemble the original image, then we can conclude more confidently that it is a better method. Here, four performance metrics: peak signal to noise ratio (PSNR) [45], root mean square error (RMSE) [45], structured similarity index (SSIM) [46] and feature similarity index (FSIM) [47] are used for a comparison. A detailed discussion on these metrics is provided in [Appendix A](#).

4.1. Experimental setup

4.1.1. Test images

Two groups of images are used for the experiment as shown in [Fig. 2](#). The first group contains ten standard test color images in [Fig. 2\(a–j\)](#) from publicly available Berkeley Segmentation Dataset, BSDS500 [38]. All the images used are true color RGB of dimension 256×256 . The second group contains one image in [Fig. 2\(k\)](#) taken from Ref. [2] having high threshold levels for comparison. The experiments are conducted using an Intel core i-7 processor

Table 1
Parameters for AWOA.

Parameter	Value
Number of Search agents	30
Number of Iterations	30
Lower bound	1
Upper bound	255

having 4 GB RAM, running under Windows 10 OS. The algorithms are developed using MATLAB.

The choice of parameters is crucial. The parameters chosen for AWOA from a pre-simulation study are presented in Table 1. Each input image is processed utilizing an individual algorithm 30 times for checking stability.

4.1.2. Experiment 1: Comparison with the existing approaches

In this experiment, the results are obtained utilizing DCC-AWOA with one entropic objective function i.e. Kapur's entropy thresholding (KET) and one non-entropic objective function i.e. edge magnitude based thresholding (EMBT). The suggested technique is compared with our implementations of the existing approaches of all the methods, where the thresholding is applied to all three components of the color image separately. Four threshold levels are considered as $M = 2, 3, 4, 5$. The optimal thresholds for the dominant component as well as each of the three color components R, G, B are found and displayed in Appendix B in Tables B.1 and B.2. The performance metrics RMSE, PSNR, SSIM and FSIM are calculated and shown in Tables 2 and 3. Note that the metrics computed in the case of RGB based approach are the average values of the three color components. The thresholded images for $M = 2, 3, 4, 5$ are shown in Figs. 3–12. The running time for DCC and RGB approaches is given in Table 4. Note that the bold numbers indicate the best in class results.

4.1.3. Experiment 2: Comparison with the method used in Ref. [2]

To further strengthen the claim, comparison with a recently published work in Ref. [2] on color image thresholding is done. In fact, the authors in [2] used Masi entropy based fitness function for color satellite image thresholding. They compared their method with other entropy-based techniques such as Kapur's, Tsallis and Renyi's entropy. Besides satellite images, they also tested their method with various color images. This approach is chosen because higher threshold levels ($M = 3, 5, 8, 12$) are considered in their experiment. The results utilizing the Kapur's entropy as the fitness function only are considered here for comparison. For the sake of consistency, the parameter setting is derived from the original article. The a9 (TEST4) image in [2] is considered for comparison. The authors have computed the results using each of the three color components separately. The PSNR, SSIM, FSIM and Entropy values for different threshold levels are given in Table 5. The corresponding thresholded images are shown in Fig. 13. Note that the dominant color, in this case, is Green.

4.2. Experimental results

4.2.1. Quantitative analysis

A detailed discussion on the quantitative results, displayed in the tables is provided here. From Tables B.1 and B.2, it is seen that the DCC based optimal threshold values are close to that obtained for the RGB based separately. The reason may be due to the inherent randomness of the optimization techniques used. These values are displayed because the performance of a multilevel thresholding procedure solely depends on the choice of optimal threshold values. The color images are thresholded using these values only.

In Table B.1, for the Lena image, the dominant color is red. The optimum thresholds obtained for the red component only for level 2 are 60 and 170. Now the red component of the Lena image is thresholded without thresholding the other two blue and green components. Finally, they are combined to get the output image. This approach does not disturb the original blue and green components of the input image thereby producing excellent output images. A similar process is followed for all the other threshold levels. To test the robustness of the suggested technique, different images with different dominant colors are used in this experiment. This is reflected in Tables B.1 and B.2. One can also visualize from the test images selected for the experiment that the dominant colors obtained (shown in Tables B.1 and B.2) are correct.

PSNR, RMSE, SSIM and FSIM values are compared and shown for different original and segmented images in Tables 2 and 3. These tables reflect the values obtained using the DCC based thresholded images and the traditional approach based on the RGB components. Note that the performance metric values are computed for each of the R, G, B components individually and the average value is displayed in Tables 2 and 3. It is noticeable from the tables that the RMSE and PSNR values obtained using DCC based approach with both Kapur's entropy and edge magnitude, have shown remarkable differences as compared to the RGB approach. From a close observation, it is evident that the improvement in PSNR values is quite encouraging while considering the proposed method. To be precise, an improvement of about 36%–38% is obtained. This highlights the usage of our proposed method. The reason behind this improvement is the color information retaining capability of the method, which is one of the main aims of this study. The introduction of adaptive coefficient vectors and Levy flight approaches in the proposed AWOA further help us to improve the selection of optimal solutions. Because the exploitation and the exploration capabilities are improved. A similar trend is perceived in the case of SSIM and FSIM values. It is observed from Tables 2 and 3 that an improvement of about 11%–12% is achieved in both cases. It also indicates that more color information is retained in the thresholded images.

In a nutshell, it is claimed that the DCC based approach using AWOA either with Kapur's entropy or with the edge magnitude produces better results compared to the conventional RGB based approaches. Therefore, the performance of the suggested method is found encouraging, irrespective of the colors, in all the cases. The quality further improves with an increase in the threshold level. This depicts the suitability of our approach to color image segmentation. Hence, we can claim that the DCC method is a better substitute for color image segmentation.

The results showing the running time in Table 4 indicate that the DCC based approach is faster than the conventional RGB based approach. It is seen that about a 12% reduction in computation time is achieved in the case of Lena image with Kapur's entropy at a threshold level $M = 2$. In the same way, about 30% reduction is achieved with edge magnitude. A similar trend is also observed while considering the case at the threshold level $M = 5$. In some other cases, for instance, the image 176019 could be processed with a noteworthy reduction in the computation time which is about 32% with the Kapur's entropy while 45% with the edge magnitude. An observation of the data in Table 4 indicates that the computation time increases proportionately with the increase in the threshold levels as expected.

Table 5 depicts the comparison of PSNR, SSIM, FSIM and entropy values for the proposed method and the method used in [2]. Note that the authors in [2] have used all the three primary color components for thresholding. This comparison is done because the method in [2] uses high threshold levels $M = 3$ to 12 using Kapur's entropy. Hence, we have considered our results of Kapur's

Table 2
Comparison of performance parameters obtained utilizing Kapur's entropy thresholding.

Image	M	DCC					RGB			
			RMSE	PSNR	SSIM	FSIM	RMSE	PSNR	SSIM	FSIM
Lena	2	R	16.7638	23.3687	0.9612	0.9856	36.1525	16.8293	0.8736	0.7930
	3		13.2264	26.5467	0.9779	0.9866	28.8715	18.2114	0.9322	0.7991
	4		11.1702	28.0767	0.9882	0.9870	22.9992	20.1892	0.9533	0.8417
	5		10.5706	29.8925	0.9933	0.9912	19.9484	21.6075	0.9718	0.8882
35 049	2	G	22.4533	20.4557	0.8235	0.9158	50.1777	14.2963	0.7118	0.8006
	3		21.3376	23.5340	0.8490	0.9200	39.0308	17.2440	0.7562	0.8305
	4		20.0105	25.4169	0.9379	0.9272	26.7446	19.8338	0.8655	0.8727
	5		10.6692	26.6183	0.9544	0.9300	24.7217	20.8690	0.8933	0.8820
126 007	2	B	22.2839	21.1710	0.9063	0.9807	38.1471	16.4743	0.8533	0.7620
	3		17.7577	23.1630	0.9439	0.9936	33.1753	17.3348	0.8927	0.7748
	4		13.3413	25.4408	0.9694	0.9943	30.6438	19.6931	0.9236	0.8218
	5		12.8527	26.1260	0.9697	0.9955	23.4760	20.3625	0.9492	0.8506
124 084	2	R	31.4896	18.0840	0.8705	0.9303	47.9009	14.4413	0.7807	0.7131
	3		20.8428	21.8296	0.9404	0.9508	44.3982	15.5072	0.8393	0.7312
	4		16.4089	24.1902	0.9630	0.9576	31.1302	17.4465	0.8819	0.8198
	5		15.0505	24.6572	0.9680	0.9651	28.6970	20.3852	0.9261	0.8535
143 090	2	B	19.8641	22.1704	0.9443	0.9956	37.9434	16.5270	0.9234	0.7458
	3		21.5176	20.8365	0.9428	0.9921	33.3886	17.3163	0.9225	0.8037
	4		14.7685	24.5755	0.9720	0.9929	27.0020	18.1641	0.9330	0.8163
	5		11.8618	26.1880	0.9798	0.9949	25.1655	20.7353	0.9587	0.8375
65 033	2	G	29.3244	18.7882	0.9234	0.8283	49.1610	14.3402	0.7794	0.7522
	3		19.6850	22.2156	0.9342	0.8852	37.4673	16.4976	0.8639	0.7924
	4		15.0480	24.3589	0.9605	0.9223	29.8252	18.6687	0.9014	0.8468
	5		11.9769	26.5149	0.9734	0.9431	22.9187	21.0586	0.9352	0.9000
12 084	2	R	30.3815	20.1052	0.8712	0.9420	44.2188	14.5918	0.8041	0.7148
	3		21.9141	21.2551	0.9074	0.9622	34.3875	17.2810	0.8881	0.8188
	4		17.9143	23.2359	0.9396	0.9733	28.0119	19.0734	0.9045	0.8869
	5		13.9637	25.1278	0.9593	0.9824	24.9017	20.4509	0.9376	0.9103
176 019	2	G	24.6535	20.3881	0.8837	0.8669	44.2448	15.1062	0.7745	0.7226
	3		19.5672	21.7387	0.9273	0.8975	35.2576	17.4291	0.8642	0.8049
	4		15.8436	23.1981	0.9541	0.9238	27.4277	19.5917	0.9075	0.8530
	5		13.5007	24.9262	0.9652	0.9421	22.8808	21.3464	0.9439	0.9069
30 091	2	B	27.7027	19.4419	0.8794	0.9878	56.2050	13.3573	0.8207	0.7582
	3		25.1903	20.2147	0.8980	0.9893	37.0486	16.9302	0.8771	0.7764
	4		17.4741	23.9750	0.9554	0.9920	29.5158	18.9646	0.9159	0.7933
	5		14.8680	24.6858	0.9664	0.9944	24.3004	20.3578	0.9431	0.8168
92 059	2	G	26.1177	19.3464	0.8752	0.8931	36.6647	13.3245	0.6633	0.8525
	3		20.8088	20.7563	0.8943	0.9432	30.0886	15.2345	0.8518	0.8734
	4		16.6766	23.3542	0.9592	0.9891	26.0467	18.3435	0.8688	0.8843
	5		11.4326	25.4563	0.9606	0.9921	22.8809	20.5321	0.8836	0.8892

entropy for a comparison. It is witnessed that most of the performance metric values are best for the proposed method. Even the PSNR value with our approach is better than that obtained at threshold level 12. In fact, the rest of the PSNR values are close. The entropy value obtained is much higher. In summary, the use of the suggested method is implicit.

Table 6 depicts the comparison of PSNR, SSIM, FSIM and entropy values for the proposed method and the method used in [3]. Note that the authors in [3] have used all the three primary color components RGB for thresholding. It is observed that all the performance metric values are best for the proposed method.

For statistical analysis, the t-test is conducted at a significance level of 0.05. The t-test considered the outcomes of finding the optimal thresholds on all the images used in the experiment as per all the threshold levels. The number of comparisons is as per 80 variables ($n \times v \times m = 10 \times 4 \times 2$; $n = 10$; $v = 4$; $m = 2$, where n signifies the number of images in the experiment, v denotes the number of performance metrics (RMSE, PSNR, SSIM, FSIM) used and m denotes the number of methods (KET, EMBT) used). The null hypothesis presumes no substantial variation between the two approaches. The alternate hypothesis presumes a significant

variation between the two approaches. The outcomes of the t-test are displayed in Table 7 where the suggested DCC approach clearly outperforms the RGB approach in all the cases.

4.2.2. Qualitative analysis

The results of thresholding on the test images are shown in Figs. 3–13. A detailed discussion on the visual results is provided here for qualitative analysis. In this experiment, images with different dominant colors are chosen deliberately for a fair and comprehensive study. The experiment covers all aspects of color image processing. These test images are considered in most of the color image segmentation experiments to validate different methods. The suggested method is implemented successfully using these images producing encouraging results. The results are provided for comparing the suggested method with RGB based approach. It is wise to reiterate that the images are obtained using the two different objective functions. The suggested DCC-AWOA is used to compute the optimum thresholds at four different levels $M = 2, 3, 4, 5$.

The Lena image in Fig. 2(a) has red color as the dominant component. The corresponding thresholded results are displayed in Fig. 3. The thresholded outputs using the suggested method

Table 3

Comparison of performance parameters obtained utilizing edge magnitude based thresholding.

Image	M	DCC					RGB			
			RMSE	PSNR	SSIM	FSIM	RMSE	PSNR	SSIM	FSIM
Lena	2	R	21.4295	21.5818	0.9499	0.9673	39.9962	15.9083	0.8732	0.8436
	3		20.8959	22.0867	0.9620	0.9688	35.5232	16.2917	0.8817	0.8580
	4		18.5809	22.3739	0.9630	0.9693	35.6693	17.4921	0.8627	0.8330
	5		15.5792	23.7085	0.9454	0.9677	34.0191	18.5774	0.9109	0.8453
35 049	2	G	41.8306	19.2599	0.9343	0.9059	49.7534	13.7059	0.8657	0.8497
	3		32.4962	23.4874	0.9351	0.9190	40.5771	14.9231	0.8406	0.8670
	4		25.6819	24.9716	0.9362	0.9203	35.2307	16.8002	0.8457	0.8617
	5		25.1279	26.5283	0.9382	0.9227	26.9394	19.2211	0.8575	0.8763
126 007	2	B	28.0294	20.1534	0.9053	0.9819	39.0346	14.0165	0.8451	0.8840
	3		27.6725	22.5473	0.9215	0.9899	35.5978	14.8867	0.8879	0.8576
	4		26.7315	23.5603	0.9409	0.9887	28.7259	16.1805	0.8746	0.8955
	5		19.3974	24.8578	0.9691	0.9937	24.6150	17.3549	0.8913	0.8931
124 084	2	R	37.3028	17.2623	0.8211	0.9203	42.9814	11.6205	0.6734	0.7182
	3		30.2130	18.8589	0.8427	0.9330	34.0573	13.3967	0.7439	0.7349
	4		28.5017	20.7084	0.8985	0.9435	30.0541	15.2898	0.7989	0.7561
	5		27.6570	22.3806	0.9567	0.9480	29.9304	16.0772	0.8270	0.7707
143 090	2	B	30.4731	17.6203	0.8750	0.9899	41.2662	15.9119	0.7516	0.7848
	3		29.1191	17.9854	0.9176	0.9860	39.5095	17.7167	0.7664	0.7901
	4		27.3914	19.2983	0.9421	0.9897	36.3373	18.5166	0.7706	0.8405
	5		25.2094	20.4368	0.9444	0.9913	35.2250	19.0178	0.8271	0.8453
65 033	2	G	31.7348	16.9828	0.9176	0.8167	47.1339	14.7402	0.8376	0.7348
	3		26.8410	19.2402	0.9256	0.8401	36.9892	17.2532	0.8678	0.7857
	4		24.2046	20.0492	0.9387	0.8637	34.0191	18.7896	0.8926	0.8029
	5		20.4460	22.7303	0.9574	0.8879	28.3161	20.1183	0.9051	0.8503
12 084	2	R	34.2754	16.3548	0.7661	0.9521	28.0666	14.7100	0.7620	0.6348
	3		26.8574	17.7985	0.8321	0.9569	38.0841	15.6370	0.7947	0.7168
	4		25.9571	18.9951	0.8789	0.9678	28.5556	16.0323	0.8331	0.7624
	5		24.7147	20.0058	0.8858	0.9746	28.1976	18.1970	0.8491	0.8089
176 019	2	G	34.0279	17.8556	0.9056	0.8450	40.5561	14.1392	0.7436	0.7806
	3		21.9860	21.3662	0.9184	0.8678	33.9175	15.9059	0.8137	0.8033
	4		21.3806	22.3292	0.9249	0.8863	30.5682	17.6146	0.8661	0.8367
	5		20.0047	22.9536	0.9256	0.8989	29.3551	18.9461	0.8948	0.8511
30 091	2	B	31.4551	17.4362	0.8252	0.9834	59.4794	12.5624	0.7840	0.7541
	3		27.2646	19.6422	0.8574	0.9867	48.4128	13.8763	0.8275	0.7643
	4		25.9858	22.4754	0.9078	0.9894	39.3294	16.7435	0.8427	0.7843
	5		13.3595	22.9566	0.9573	0.9906	30.5412	19.3756	0.8516	0.7989
92 059	2	G	20.4384	24.7516	0.9257	0.9065	23.5121	12.9050	0.8226	0.7730
	3		27.0358	22.9334	0.8935	0.8656	30.5339	14.2153	0.8352	0.7934
	4		24.2988	23.5343	0.9264	0.8934	27.2441	16.3215	0.8670	0.8132
	5		23.3131	24.2658	0.9499	0.9242	25.7806	18.5257	0.8814	0.8422

with KET at $M = 2, 3, 4, 5$ are displayed in Fig. 3(a–d). The images in Fig. 3(e–h) represent the thresholded outputs obtained using the RGB approach with Kapur's entropy (the same fitness function). Note that the thresholded outputs using the suggested method with EMBT at $M = 2, 3, 4, 5$ are also displayed in Fig. 3(i–l). The images in Fig. 3(m–p) represent the thresholded images got using the RGB approach with edge magnitude (the same objective function). It is visible from the figures that the quality of images obtained with the suggested technique is better than the images obtained with the conventional approaches. The reason is that the proposed approach optimally thresholds the red color component only. It may be noted that the traditional approach (RGB based) measurably fails to show the original color information.

Further, while thresholding using the suggested scheme, it is seen that the red color of the original image is maintained throughout the levels of thresholding from $M = 2$ to $M = 5$. In this context, the output obtained with the RGB based model fails to retain the color information, because all the three primary color components are thresholded individually. Then they are combined to get the thresholded image. This results in the introduction of new colors in the output image as seen in Fig. 3(e–h). More specifically, artifacts are introduced on the face and hat

portions of the images. A similar trend is observed in all the other figures. The original blue color is retained throughout, for instance, in Fig. 5. The edges are also nicely visible.

The images in Fig. 13 show the superiority of our approach over the method used in [2]. Note that the dominant green color is retained throughout in Fig. 13(e–h). But a greenish and black artifact is introduced in the water portion in Fig. 13(a–d). The reason being all the three color components are thresholded individually and then combined to get the final thresholded image in [2]. In summary, we can conclude that the suggested technique yields notably better performance as compared to the traditional methods using the RGB approach. It means, realistically, one would be ready for the practical implementations.

The images in Fig. 14 compares our approach over the method used in [3]. Note that the dominant red color is retained throughout in our method.

Keeping a method constant, AWOA is faster than the WOA, because the adaptive behavior of the proposed AWOA reduces the computation time. The coefficient vectors in WOA are made adaptive and the introduction of Levy flight improves the exploitation leading to a faster convergence. To justify this result, the convergence curves are presented in Fig. 15 for all the threshold levels $M = 2, 3, 4, 5$ with Kapur's entropy as the cost function.

Table 4

Comparison of running time (in secs) utilizing the proposed DCC-AWOA method.

Image	M	Kapur's entropy thresholding			Edge magnitude based thresholding		
			DCC	RGB		DCC	RGB
Lena	2	R	1.7315	2.5843	R	2.1795	3.1135
	3		1.7704	2.6036		2.1924	3.1320
	4		1.7866	2.6126		2.2075	3.1535
	5		1.7943	2.6183		2.2345	3.1922
35 049	2	G	1.7333	2.5831	G	2.0909	2.9870
	3		1.7631	2.5967		2.1032	3.0045
	4		1.7781	2.6070		2.1118	3.0168
	5		1.8052	2.6163		2.1331	3.0473
126 007	2	B	1.7627	2.5692	B	2.1638	3.0911
	3		1.7779	2.5902		2.2132	3.1617
	4		1.7903	2.6033		2.2287	3.1838
	5		1.8020	2.6271		2.3675	3.3822
124 084	2	R	1.6563	2.4617	R	2.5346	3.6208
	3		1.6965	2.4750		2.6279	3.7541
	4		1.7090	2.5197		2.6384	3.7691
	5		1.7414	2.5566		2.7236	3.8908
143 090	2	B	1.6492	2.4615	B	2.5161	3.5944
	3		1.6840	2.4765		2.5303	3.6147
	4		1.6979	2.5192		2.5796	3.6851
	5		1.7062	2.5466		2.5984	3.7120
65 033	2	G	1.6490	2.4612	G	2.4142	3.4489
	3		1.7075	2.4746		2.4601	3.5144
	4		1.7131	2.5193		2.4944	3.5634
	5		1.7243	2.5357		2.5119	3.5884
12 084	2	R	1.6155	2.4113	R	2.5756	3.6794
	3		1.7025	2.4675		2.6177	3.7396
	4		1.7293	2.5062		2.6758	3.8226
	5		1.7349	2.5367		2.7411	3.9159
176 019	2	G	1.6783	2.4612	G	2.4000	3.4285
	3		1.6830	2.4750		2.4425	3.4893
	4		1.7079	2.5193		2.4623	3.5176
	5		1.7375	2.5552		2.5196	3.5994
30 091	2	B	1.7387	2.5488	B	2.3498	3.3568
	3		1.7618	2.5679		2.3631	3.3758
	4		1.7758	2.5967		2.3755	3.3936
	5		1.7856	2.6168		2.4351	3.4787
92 059	2	G	1.7242	2.5568	G	1.9937	2.8482
	3		1.7341	2.5732		1.9982	2.8545
	4		1.7567	2.5921		2.0057	2.8653
	5		1.7918	2.6547		2.1795	2.8753

Table 5

Comparison of the proposed method with results from Ref. [2].

Image	DCC	M	Proposed method using KET				Ref. [2]			
			PSNR	SSIM	FSIM	Entropy	PSNR	SSIM	FSIM	Entropy
a9 (TEST4)	G	3	33.4646	0.9626	0.8887	15.6303	33.6538	0.6496	0.7776	3.2361
		5	32.8774	0.9865	0.9235	21.2682	33.0153	0.6791	0.8045	4.0360
		8	29.7196	0.9913	0.9590	28.2247	30.0213	0.8676	0.9115	5.0758
		12	32.1279	0.9950	0.9794	36.9869	28.8919	0.8901	0.9218	5.9003

Table 6

Comparison of the proposed method with results from Ref. [3].

Image	DCC	M	Proposed method using KET				Ref. [3]			
			PSNR	SSIM	FSIM	RMSE	PSNR	SSIM	FSIM	RMSE
Flower	R	2	18.0840	0.8705	0.9303	31.4896	13.2048	0.5105	0.7122	55.7568
		3	21.8296	0.9404	0.9508	20.8428	16.5358	0.6989	0.7489	37.9971
		4	24.1902	0.9630	0.9576	16.4089	19.0378	0.7799	0.8166	28.4874
		5	24.6572	0.9680	0.9651	15.0505	20.4588	0.7607	0.8336	24.1879

It is observed that AWOA converges faster than WOA in all the cases due to the introduction of Levy flight and adaptive coefficient vectors. Note that the results are given for the DCC based approach only. This behavior is seen across all the test images. Hence, the results are shown for one image (Lena) only for the

sake of completion and to avoid repetition. It is noteworthy to mention here that the convergence curves show the objective function values vs number of iterations, does not depend upon the input image. The characteristics of the curves do not change (excepting the objective function values) even if we change the

Table 7

Statistical analysis for all the images using KET and EMBT compared to DCC.

Image	t-test statistical analysis															
	KET								EMBT							
	RMSE		PSNR		SSIM		FSIM		RMSE		PSNR		SSIM		FSIM	
	<i>p</i>	<i>h</i>	<i>p</i>	<i>h</i>	<i>p</i>	<i>h</i>	<i>p</i>	<i>h</i>	<i>p</i>	<i>h</i>	<i>p</i>	<i>h</i>	<i>p</i>	<i>h</i>	<i>p</i>	<i>h</i>
Lena	0.0077	1	0.0003	1	0.0450	1	0.0049	1	0.0003	1	0.0001	1	0.0133	1	0.0002	1
35 049	0.0322	1	≤0.05	1	0.0048	1	0.0164	1	0.0283	1	0.0016	1	0.0007	1	0.0003	1
126 007	0.0020	1	0.0003	1	0.0109	1	0.0019	1	0.0424	1	0.0003	1	0.0079	1	0.0013	1
124 084	0.0046	1	0.0062	1	0.0088	1	0.0084	1	0.0356	1	0.0001	1	0.0022	1	0.0001	1
143 090	0.0024	1	0.0034	1	0.0115	1	0.0023	1	0.0001	1	0.0481	1	0.0015	1	0.0015	1
65 033	0.0038	1	0.0004	1	0.0429	1	0.0062	1	0.0068	1	0.0058	1	0.0041	1	0.0077	1
12 084	0.0007	1	0.0009	1	0.0473	1	0.0329	1	0.0169	1	0.0053	1	0.0434	1	0.0055	1
176 019	0.0084	1	0.0019	1	0.0476	1	0.0330	1	0.0036	1	0.0014	1	0.0246	1	0.0011	1
30 091	0.0388	1	0.0042	1	0.0267	1	0.0003	1	0.0079	1	0.0023	1	0.0366	1	0.0001	1
92 059	0.0003	1	0.0002	1	0.0117	1	0.0137	1	0.0008	1	0.0077	1	0.0063	1	0.0072	1

Table 8

Comparison of DCC based AWOA and WOA utilizing Kapur's entropy thresholding.

Image	M	AWOA				WOA					
		RMSE	PSNR (dB)	SSIM	FSIM	RMSE	PSNR (dB)	SSIM	FSIM		
Lena	2	R	281.0240	23.3687	0.9612	0.9856	R	299.3710	22.9671	0.9576	0.9830
	3		174.9387	26.5467	0.9779	0.9866		194.0635	26.1605	0.9771	0.9845
	4		124.7732	28.0767	0.9882	0.9870		147.3563	27.9165	0.9858	0.9850
	5		111.7372	29.8925	0.9933	0.9912		119.6966	28.1432	0.9870	0.9866
35 049	2	G	504.1529	20.4557	0.8235	0.9158	G	567.0070	19.8865	0.8139	0.9148
	3		455.2937	23.5340	0.8490	0.9200		492.2559	22.5410	0.8608	0.9218
	4		400.4185	25.4169	0.9379	0.9272		423.5262	24.6529	0.9363	0.9288
	5		113.8315	26.6183	0.9544	0.9300		155.6129	26.5397	0.9470	0.9375
126 007	2	B	496.5721	21.1710	0.9063	0.9807	B	500.1009	21.1641	0.9054	0.9921
	3		315.3372	23.1630	0.9439	0.9936		356.0332	23.0836	0.9430	0.9935
	4		177.9895	25.4408	0.9694	0.9943		207.5551	25.0101	0.9685	0.9945
	5		165.1928	26.1260	0.9697	0.9955		168.4557	25.1967	0.9691	0.9950
124 084	2	R	991.5922	18.0840	0.8705	0.9303	R	1012.500	18.0768	0.8691	0.9294
	3		434.4231	21.8296	0.9404	0.9508		441.4366	21.4515	0.9399	0.9491
	4		269.2522	24.1902	0.9630	0.9576		278.2697	23.9884	0.9619	0.9559
	5		226.5186	24.6572	0.9680	0.9651		244.6078	24.6242	0.9678	0.9610
143 090	2	B	394.5841	22.1704	0.9443	0.9956	B	395.4910	22.1593	0.9436	0.9955
	3		463.0054	20.8365	0.9428	0.9921		467.2150	21.4570	0.9415	0.9914
	4		218.1097	24.5755	0.9720	0.9929		237.8958	25.0345	0.9678	0.9920
	5		140.7033	26.1880	0.9798	0.9949		167.4018	26.0653	0.9793	0.9945
65 033	2	G	859.9218	18.7882	0.9234	0.8283	G	879.0748	18.6905	0.9172	0.8274
	3		387.4987	22.2156	0.9342	0.8852		390.3158	22.0963	0.9326	0.8848
	4		226.4411	24.3589	0.9605	0.9223		227.1466	23.8727	0.9581	0.9172
	5		143.4459	26.5149	0.9734	0.9431		156.6661	26.4897	0.9732	0.9392
12 084	2	R	923.0338	20.1052	0.8712	0.9420	R	933.0370	19.5014	0.8374	0.9413
	3		480.2261	21.2551	0.9074	0.9622		516.7960	21.1301	0.9024	0.9588
	4		320.9205	23.2359	0.9396	0.9733		327.1770	23.1340	0.9384	0.9690
	5		194.9842	25.1278	0.9593	0.9824		224.6521	25.1011	0.9556	0.9791
176 019	2	G	607.7952	20.3881	0.8837	0.8669	G	631.1178	20.2932	0.8773	0.8646
	3		382.8748	21.7387	0.9273	0.8975		400.6911	21.2737	0.9247	0.8923
	4		251.0201	23.1981	0.9541	0.9238		275.0466	22.9240	0.9535	0.9175
	5		182.2679	24.9262	0.9652	0.9421		223.4342	24.6513	0.9622	0.9365
30 091	2	B	767.4395	19.4419	0.8794	0.9878	B	778.9750	19.3725	0.8742	0.9872
	3		634.5520	20.2147	0.8980	0.9893		691.6706	20.1342	0.8808	0.9876
	4		305.3430	23.9750	0.9554	0.9920		310.9133	23.7571	0.9496	0.9908
	5		221.0567	24.6858	0.9664	0.9944		242.8530	23.9324	0.9652	0.9934
92 059	2	G	682.1321	19.3464	0.8752	0.8931	G	709.3324	19.2313	0.8714	0.8823
	3		433.0054	20.7563	0.8943	0.9432		487.2150	20.5421	0.8890	0.9331
	4		278.1097	23.3542	0.9592	0.9891		267.8958	23.2435	0.9455	0.9843
	5		130.7033	25.4563	0.9606	0.9921		167.4018	25.1346	0.9594	0.9912

input image. Ultimately, it is found that AWOA exhibits faster convergence compared to WOA.

To demonstrate the effectiveness of the improvement for the proposed algorithm, comparison with the DCC based WOA with Kapur's entropy and Edge magnitude as the fitness function, is shown in Tables 8 and 9. The performance parameters RMSE, PSNR, SSIM, and FSIM values are displayed. It is observed that

the proposed AWOA performed better than WOA using the DCC based approach.

Table 10 shows the comparison of running time between AWOA and WOA for both the objective functions. It is observed that AWOA needs lesser running time as compared to WOA in both the cases. Hence, it is confirmed that the introduction of Lévy-flight strategy and the self-adaptive strategy based on the fitness function value proposed for updating the coefficient

Table 9
Comparison of DCC based AWOA and WOA utilizing edge magnitude based thresholding.

Image	M	AWOA				WOA					
		RMSE	PSNR	SSIM	FSIM	RMSE	PSNR	SSIM	FSIM		
Lena	2	R	459.2228	21.5818	0.9499	0.9673	R	475.0767	21.4386	0.9473	0.9659
	3		436.6389	22.0867	0.9620	0.9688		464.6823	21.7296	0.9593	0.9649
	4		345.2494	22.3739	0.9630	0.9693		382.4914	22.1899	0.9361	0.9605
	5		242.7123	23.7085	0.9454	0.9677		276.6506	22.6588	0.9268	0.9591
35 049	2	G	1749.8000	19.2599	0.9343	0.9059	G	1844.2000	18.5555	0.9256	0.8987
	3		1056.0000	23.4874	0.9351	0.9190		1375.6000	22.1762	0.9339	0.8932
	4		659.5577	24.9716	0.9362	0.9203		775.7981	23.7778	0.9356	0.9037
	5		631.4099	26.5283	0.9382	0.9227		693.8117	24.4928	0.9379	0.8894
126 007	2	B	785.6464	20.1534	0.9053	0.9819	B	791.1350	19.8977	0.9033	0.9858
	3		765.7688	22.5473	0.9215	0.9899		872.1460	21.6414	0.8951	0.9884
	4		714.5726	23.5603	0.9409	0.9887		867.6896	22.6207	0.9353	0.9908
	5		376.2579	24.8578	0.9691	0.9937		385.3020	23.9017	0.9466	0.9898
124 084	2	R	1391.5000	17.2623	0.8211	0.9203	R	1910.0000	16.3016	0.8079	0.9002
	3		912.8238	18.8589	0.8427	0.9330		977.6509	18.5184	0.8194	0.9199
	4		812.3446	20.7084	0.8985	0.9435		839.2858	20.0998	0.8880	0.9356
	5		764.9071	22.3806	0.9567	0.9480		812.9682	21.2899	0.9471	0.9427
143 090	2	B	928.6094	17.6203	0.8750	0.9899	B	1033.2000	16.1929	0.8586	0.9888
	3		847.9235	17.9854	0.9176	0.9860		970.7710	17.7856	0.8937	0.9844
	4		750.2898	19.2983	0.9421	0.9897		800.5491	18.3177	0.9209	0.9878
	5		635.5137	20.4368	0.9444	0.9913		758.6847	19.5279	0.9277	0.9890
65 033	2	G	1007.1000	16.9828	0.9176	0.8167	G	1027.5000	15.9636	0.9128	0.8085
	3		720.4394	19.2402	0.9256	0.8401		784.1153	19.1019	0.9218	0.8284
	4		585.8623	20.0492	0.9387	0.8637		638.8717	19.8815	0.9321	0.8412
	5		418.0389	22.7303	0.9574	0.8879		547.4710	22.3889	0.9427	0.8707
12 084	2	R	1174.8000	16.3548	0.7661	0.9521	R	1294.6000	16.1909	0.7309	0.9482
	3		721.3206	17.7985	0.8321	0.9569		893.5697	17.6531	0.8370	0.9542
	4		673.7696	18.9951	0.8789	0.9678		797.2995	18.5722	0.8544	0.9623
	5		610.8175	20.0058	0.8858	0.9746		777.8105	19.7843	0.8676	0.9715
176 019	2	G	1157.9000	17.8556	0.9056	0.8450	G	1163.1000	17.4747	0.9000	0.8412
	3		483.3847	21.3662	0.9184	0.8678		496.4462	21.1411	0.9130	0.8632
	4		457.1312	22.3292	0.9249	0.8863		475.3765	21.6000	0.9198	0.8753
	5		400.1892	22.9536	0.9256	0.8989		411.9038	22.3199	0.9240	0.8934
30 091	2	B	989.4246	17.4362	0.8252	0.9834	B	1004.9000	16.3562	0.7932	0.9797
	3		743.3576	19.6422	0.8574	0.9867		839.2542	18.5231	0.8425	0.9837
	4		675.2643	22.4754	0.9078	0.9894		742.5166	21.7553	0.8967	0.9876
	5		578.4765	22.9566	0.9573	0.9906		601.8489	22.6574	0.9452	0.9900
92 059	2	G	417.7296	24.7516	0.9257	0.9065	G	467.3121	24.6712	0.8918	0.8931
	3		730.9370	22.9334	0.8935	0.8656		760.2421	22.6742	0.8929	0.8523
	4		590.4323	23.5343	0.9264	0.8934		505.3423	22.9832	0.9084	0.8834
	5		543.5021	24.2658	0.9499	0.9242		478.7321	23.3423	0.9276	0.9189

vectors enriches the algorithm's exploration and exploitation capabilities leading to faster convergence.

In summary, a significant speed improvement is achieved with the suggested approach, which is one of the main aims of the proposal. The consideration of dominant component only for thresholding purposes reduces the computation time because there is no need to compute the optimal thresholds for individual color components (R, G and B). Further, thresholding only the dominant color component in the image retains the original color information. It is noteworthy to mention here that the contribution of the information by non-dominant color components is retained.

Further, the proposed concept may be improved in future by including neural architecture search [48] for determining the optimal threshold values. The concept of compound rank-k projections for bilinear analysis [49] to reduce the dimension of the feature representation for better performance of the proposed method can also be explored in the future.

5. Conclusion

Unlike earlier methods reported for multilevel color image thresholding utilizing all the three color (RGB) components, our suggested technique uses the dominant color only, which is an

innovative idea for color image processing. Nevertheless, an exemplar idea is delivered for the color image thresholding. The justification behind the use of two competent objective functions, for the experiment, is for validation. The suggested segmentation score enriches the image processing literature. The stability and validity of the suggested technique are justified both qualitatively and quantitatively. The results with the proposed scheme show remarkable differences as compared to the conventional approaches and recently published research works. It is implicit from the outputs that the true color information is retained. Even more interesting is the significant reduction in computation time. Therefore, the suggested method is quite competent and enforces its application in color image processing. Its explicit use in the high threshold level applications is also highlighted. In this work, the method is implemented successfully for $M = 2$ to 12. The method may be implemented for higher threshold levels. The idea of neural architecture search may be used for determining the optimal thresholds. To improve the performance of the suggested method, the idea of compound rank-k projections for bilinear analysis to lower the dimension of the feature representation may be used. This needs to be explored in future studies. The method would be useful for the segmentation of biomedical images using data science methodologies for smart healthcare services. The proposed method may be useful for color images having low-contrast inhomogeneous visual features. This study may help

Table 10
Comparison of running time (in secs) between DCC-AWOA and DCC-WOA.

Image	M		Kapur's entropy thresholding			Edge magnitude based thresholding	
			AWOA	WOA		AWOA	WOA
Lena	2	R	1.7315	1.7422	R	2.1795	2.1890
	3		1.7704	1.7806		2.1924	2.2610
	4		1.7866	1.7969		2.2075	2.2686
	5		1.7943	1.8005		2.2345	2.3232
35 049	2	G	1.7333	1.7751	G	2.0909	2.1139
	3		1.7631	1.7943		2.1032	2.1550
	4		1.7781	1.8191		2.1118	2.1874
	5		1.8052	1.8465		2.1331	2.2103
126 007	2	B	1.7627	1.7866	B	2.1638	2.2377
	3		1.7779	1.7963		2.2132	2.2490
	4		1.7903	1.8234		2.2287	2.2791
	5		1.8020	1.8520		2.3675	2.3736
124 084	2	R	1.6563	1.6885	R	2.5346	2.6161
	3		1.6965	1.7177		2.6279	2.6940
	4		1.7090	1.7582		2.6384	2.7310
	5		1.7414	1.7785		2.7236	2.7765
143 090	2	B	1.6492	1.8522	B	2.5161	2.5403
	3		1.6840	1.8807		2.5303	2.5474
	4		1.6979	1.8945		2.5796	2.6418
	5		1.7062	1.9095		2.5984	2.7263
65 033	2	G	1.6490	1.8211	G	2.4142	2.4490
	3		1.7075	1.8482		2.4601	2.4928
	4		1.7131	1.8561		2.4944	2.5254
	5		1.7243	1.9009		2.5119	2.5339
12 084	2	R	1.6155	1.8224	R	2.5756	2.6373
	3		1.7025	1.8401		2.6177	2.6718
	4		1.7293	1.8616		2.6758	2.7453
	5		1.7349	1.8777		2.7411	2.7849
176 019	2	G	1.6783	1.6851	G	2.4000	2.4317
	3		1.6830	1.6994		2.4425	2.4637
	4		1.7079	1.7436		2.4623	2.5087
	5		1.7375	1.8433		2.5196	2.5320
30 091	2	B	1.7387	1.7551	B	2.3498	2.3638
	3		1.7618	1.7794		2.3631	2.3727
	4		1.7758	1.7837		2.3755	2.3830
	5		1.7856	1.8045		2.4351	2.5115
92 059	2	G	1.7242	1.7607	G	1.9937	1.9974
	3		1.7341	1.7833		1.9982	2.0050
	4		1.7567	1.7964		2.0057	2.0106
	5		1.7918	1.8385		2.1795	2.0184

researchers to explore further ideas in the field of multilevel color image thresholding.

CRediT authorship contribution statement

Sanjay Agrawal: Conceptualization, Methodology, Formatting, Writing – original draft. **Rutuparna Panda:** Review and editing, Supervision. **Pratiksha Choudhury:** Software, Investigation, Draft revision. **Ajith Abraham:** Supervision, Review.

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.

Appendix A

Performance metrics

RMSE and PSNR have significance in evaluating the operation of different techniques [45]. The term PSNR in an image is the ratio between the maximum possible value (power) and the strength of distorting noise which influences its representation

quality. Because of the wide dynamic range in the images, the PSNR is represented in dB (ratio between the largest and smallest possible values of a changeable quantity). The PSNR is computed as:

$$PSNR(in\ dB) = 20 \log_{10}(255/RMSE) \quad (A.1)$$

where

$$RMSE = \sqrt{\frac{\sum_{i=1}^M \sum_{j=1}^N (I(i,j) - I_t(i,j))^2}{MN}} \quad (A.2)$$

Here, MN is the image dimension, I denotes the input image and I_t represents the thresholded image. The reason behind the choice of PSNR as a performance metric is that it indicates the signal content, which ultimately depends on the thresholded image quality. PSNR and RMSE have an inverse relationship. So a larger value of PSNR means the (better) higher quality of output image. For a good segmented image, PSNR must increase and RMSE must decrease.

Only concerning the ground truth, PSNR computes the quality of a thresholded image. SSIM and FSIM are stronger measurement indices that measure the image structure. The SSIM and FSIM values close to '1' indicate the image is of good quality. Note that both are extensively used for quantitative analysis of

Table B.1
Comparison of optimal thresholds obtained utilizing Kapur's entropy thresholding.

Image	M	DCC	R	G	B
Lena	2	R	60,170	60,172	84,156
	3		60,147,196	60,152,201	65,141,203
	4		60,121,172,205	31,60,130,176	59,100,146,203
	5		20,58,127,172,208	33,60,110,170,204	50,90,137,168,203
35 049	2	G	91,172	93,170	89,163
	3		88,129,176	84,142,187	88,135,174
	4		57,90,131,173	70,113,142,187	95,138,170,208
	5		61,89,123,171,208	57,90,118,147,190	63,90,128,172,206
126 007	2	B	89,173	93,155	88,155
	3		62,118,174	84,142,197	87,155,209
	4		39,92,130,175	77,121,160,210	72,104,155,213
	5		60,103,137,173,206	52,90,128,160,207	72,103,140,175,216
124 084	2	R	95,187	96,187	66,149
	3		59,120,189	59,128,187	69,135,209
	4		47,94,142,191	52, 96, 141,191	41,88, 146,214
	5		49,94,137,180,217	43,86,133,180,217	42,86,124,168,214
143 090	2	B	109,180	83,155	65,150
	3		69,136,212	69,115,163	50,105,157
	4		75,125,171,214	41,83,124,163	60,120, 180,230
	5		66,97,133,174,215	33,84,130,157,188	48,111,153,190,230
65 033	2	G	121,199	90,164	125,199
	3		86,144,202	59,112,168	88,143,202
	4		75,111,157,205	63,105,169,213	67,132,168,205
	5		71,99,131,167,205	41,76,128,169,224	65,104,121,165,219
12 084	2	R	100,192	100,192	120,186
	3		84,141,203	74,126,204	104,149,197
	4		76,121,165,217	75,118,173,216	84,123,161,205
	5		62,100,136,180,222	44,86,131,177,222	39,81,126,158,216
176 019	2	G	117,181	104,171	98,172
	3		78,144,193	72,134,192	78,141,190
	4		71,119,161,213	56,105,153,204	69,117,161,207
	5		72,102,140,180,218	35,83,129,167,211	66,94,134,180,218
30 091	2	B	112,200	91,171	64,161
	3		54,123,200	76,137,194	63,118,183
	4		56,101,154,200	71,120,163,211	63,113,160,208
	5		45,94,139,174,211	33,73,121,162,211	41,69,108,162,208
92 059	2	G	71,134	103,162	70,134
	3		77,132,180	58,108,162	73,132, 175
	4		67,103,137,178	58,108,160,206	47,81,132,180
	5		66,103,134,170,204	47,101,136,162,198	74,105,132,173,208

color images. The SSIM is an index for measuring the observed quality of digital images. It is utilized to calculate the similarity between two images. It is a complete reference index i.e. image quality measurement or estimation depends on an original distortion-free image taken as a reference image. It is considered as upgrades to conventional approaches of comparison metrics like PSNR and RMSE. It is a perception-based metric that takes into account image deterioration as the structural information change. This also integrates important perceptual phenomena, including criteria for masking the luminance and the contrast parameters. It should be noted that structural information is the indication of strong interdependence of the spatially close pixels. These dependencies convey essential info about the object structure in the image. Luminance masking is a concept where image deteriorations are likely to be less visible in bright regions. At the same time, contrast masking is a process in which distortions are less apparent where the image includes texture. The SSIM between two images I and I_t is computed as [46]:

$$SSIM(I, I_t) = \frac{(2\mu_I\mu_{I_t} + c_1)(2\sigma_{II_t} + c_2)}{(\mu_I^2 + \mu_{I_t}^2 + c_1)(\sigma_I^2 + \sigma_{I_t}^2 + c_2)} \quad (A.3)$$

where μ_I denotes the mean value of I , μ_{I_t} denotes the mean value of I_t , σ represents the variance of both the images. σ_{II_t} represents the covariance of I and I_t . c_1 and c_2 are two variables determined by the dynamic range of the pixel values. For a color image, SSIM

is specified as $SSIM = \sum_c SSIM(x^c, y^c)$, where c represents the channel number. For an RGB image, it is 1, 2, 3. The symbol x represents the c th channel of the input image and y represents the c th channel of the thresholded image.

FSIM index is utilized to measure the segmentation performance based on low-level features. It uses two important elements: phase congruency (PC) and gradient magnitude (GM) as first and second features respectively. The PC indicates the significance of local structures. It is represented as [47]:

$$FSIM(x, y) = \frac{\sum_{x \in \Omega} S_L(x) \cdot PC_m(x)}{\sum_{x \in \Omega} PC_m(x)} \quad (A.4)$$

where the symbol Ω signifies the total image area, $S_L(x)$ is the similarity between the input and the thresholded output. A larger FSIM value indicates higher similarity. For an RGB image, it is represented as $FSIM = \sum_c (x^c, y^c)$, where the symbols carry the same meaning as defined for SSIM. The elaborate definitions of the performance indices are mentioned in the respective references.

Appendix B

See Tables B.1 and B.2.

Table B.2

Comparison of optimal thresholds obtained using edge magnitude based thresholding.

Image	M	DCC	R	G	B
Lena	2	R	145,225	150,225	115,184
	3		61,117,223	49,78,147	62,115,197
	4		4,81,146,246	78,98,134,255	52,63,114,188
	5		8,53,94,135,245	34,84,89,140,255	23,27,63,130,210
35 049	2	G	108,140	136,178	68,102
	3		68,105,188	84,108,136	68,101,202
	4		68,103,117,172	3,92,118,131	69,105,121,255
	5		68,97,136,154,239	9,23,80,106,139	42,61,105,131,143
126 007	2	B	95,195	31,93	71,95
	3		73,86,194	8,93,125	35,95,121
	4		69,88,174,225	31,41,82,195	8,56,95,189
	5		50,83,138,146,205	13,49,74,155,173	2,69,125,143,152
124 084	2	R	78,227	78,255	70,99
	3		78,160,219	43,78,195	4,66,97
	4		78,157,197,243	44,78,201,226	21,34,57,99
	5		27,42,78,151,162	78,169,204,208,251	4,20,23,62,97
143 090	2	B	133,218	1,122	123,249
	3		66,133,213	77,99,131	86,111,127
	4		33,62,133,216	1,50,93,213	15,86,118,147
	5		40,70,112,195,237	74,82,220,243,245	3,70,84,164,206
65 033	2	G	106,142	113,138	104,144
	3		112,157,225	94,111,138	16,106,142
	4		112,157,217,238	93,111,143,152	112,157,176,240
	5		27,112,157,170,211	37,93,135,181,233	60,116,132,211,255
12 084	2	R	145,247	145,236	88,132
	3		20,145,212	2,145,255	69,99,130
	4		9,68,144,231	59,83,143,255	3,31,86,133
	5		4,49,64,144,253	34,50,145,237,255	23,44,74,121,210
176 019	2	G	94,145	102,133	106,155
	3		30,106,155	94,134,158	98,152,167
	4		84,124,212,216	88,106,113,132	20,69,112,195
	5		41,106,155,198,244	29,77,113,197,223	106,155,179,183,207
30 091	2	B	112,237	83,110	111,239
	3		89,109,232	2,83,109	16,79,110
	4		17,81,113,215	6,76,108,233	86,111,140,198
	5		92,133,146,156,190	11,32,49,90,104	6,81,97,110,222
92 059	2	G	91,125	75,95	91,123
	3		91,123,245	74,92,171	91,123,138
	4		91,124,157,160	71,88,102,194	91,120,159,243
	5		91,122,154,185,247	74,93,117,172,193	91,120,178,197,199

References

- [1] L. He, S. Huang, Modified firefly algorithm based multilevel thresholding for color image segmentation, *Neurocomputing* 240 (2017) 152–174.
- [2] S. Shubham, A. Bhandari, A generalized Masi entropy based efficient multilevel thresholding method for color image segmentation, *Multimedia Tools Appl.* 78 (2019) 17197–17238.
- [3] J. Anitha, S.I.A. Pandian, S.A. Agnes, An efficient multilevel color image thresholding based on modified whale optimization algorithm, *Expert Syst. Appl.* 178 (2021) 115003.
- [4] J. Kapur, P. Sahoo, A. Wong, A new method for gray-level picture thresholding using the entropy of the histogram, *Comput. Vis. Graph. Image Process.* 29 (1985) 273–285.
- [5] N. Otsu, A threshold selection method from gray-level histograms, *IEEE Trans. Syst. Man Cybern.* 9 (1979) 62–66.
- [6] C. Li, C. Lee, Minimum cross entropy thresholding, *Pattern Recognit.* 26 (1993) 617–625.
- [7] D. Liu, Z. Jiang, H. Feng, A novel fuzzy classification entropy approach to image thresholding, *Pattern Recognit. Lett.* 27 (16) (2006) 1968–1975.
- [8] S. Agrawal, R. Panda, S. Bhuyan, B. Panigrahi, Tsallis entropy based optimal multilevel thresholding using cuckoo search algorithm, *Swarm Evol. Comput.* 11 (2013) 16–30.
- [9] D. Oliva, S. Hinojosa, V. Osuna-Enciso, E. Cuevas, M. Pérez-Cisneros, S.-A. G, Image segmentation by minimum cross entropy using evolutionary methods, *Soft Comput.* (2017) 1–20.
- [10] P. Sahoo, C. Wilkins, J. Yeager, Threshold selection using Renyi's entropy, *Pattern Recognit.* 30 (1997) 71–84.
- [11] R. Panda, S. Agrawal, S. Bhuyan, Edge magnitude based multilevel thresholding using Cuckoo search technique, *Expert Syst. Appl.* 40 (2013) 7617–7628.
- [12] F. Nie, P. Zhang, J. Li, D. Ding, A novel generalized entropy and its application in image thresholding, *Signal Process.* 134 (2017) 23–34.
- [13] M. Sezgin, B. Sankur, Survey over image thresholding techniques and quantitative performance evaluation, *J. Electron. Imaging* 13 (1) (2004) 146–166.
- [14] G. Sun, A. Zhang, Y. Yao, Z. Wang, A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding, *Appl. Soft Comput.* 46 (2016) 703–730.
- [15] S. Sarkar, S. Paul, R. Burman, S. Das, S. Chaudhuri, A fuzzy entropy based multi-level image thresholding using differential evolution, in: *Swarm, Evolutionary, and Memetic Computing*, Springer International Publishing, Cham, 2015, pp. 386–395.
- [16] P. Ghamisi, M. Couceiro, F. Martins, J. Benediktsson, Multilevel image segmentation based on fractional-order darwinian particle swarm optimization, *IEEE Trans. Geosci. Remote Sens.* 52 (2014) 2382–2394.
- [17] S. Bevi, M. Nair, G. Bindu, Automatic segmentation of cell nuclei using Krill Herd optimization based multi-thresholding and localized active contour model, *Biocybern. Biomed. Eng.* 36 (4) (2016) 584–596.
- [18] M. Ahmadi, K. Kazemi, A. Arabi, T. Niknam, M. Helfroush, Image segmentation using multilevel thresholding based on modified bird mating optimization, *Multimedia Tools Appl.* 78 (2019) 23003–23027.
- [19] R. Sambandam, S. Jayaraman, Self-adaptive dragonfly based optimal thresholding for multilevel segmentation of digital images, *J. King Saud Univ. - Comput. Inf. Sci.* 30 (2018) 449–461.
- [20] H. Alwerfali, M. Al-qaness, M. Elaziz, A. Ewees, D. Oliva, S. Lu, Multi-level image thresholding based on modified spherical search optimizer and fuzzy entropy, *Entropy* 22 (2020) 328.
- [21] S. Mirjalili, A. Lewis, The whale optimization algorithm, *Adv. Eng. Softw.* 95 (2016) 51–67.
- [22] M. Aziz, A. Ewees, A. Hassanien, Whale Optimization Algorithm and Moth-Flame Optimization for multilevel thresholding image segmentation, *Expert Syst. Appl.* 83 (2017) 242–256.
- [23] M. Elaziz, A. Ewees, D. Oliva, Hyper-heuristic method for multilevel thresholding image segmentation, *Expert Syst. Appl.* 146 (2020) 113201.

- [24] J. Gao, B. Wang, Z. Wang, Y. Wang, F. Kong, A wavelet transform-based image segmentation method, *Optik* 208 (2020) 164123.
- [25] J. Li, W. Tang, J. Wang, X. Zhang, A multilevel color image thresholding scheme based on minimum cross entropy and alternating direction method of multipliers, *Optik* 183 (2019) 30–37.
- [26] E. Rodriguez-Esparza, L. Zanella-Calzada, D. Oliva, A. Heidari, D. Zaldivar, M. Pérez-Cisneros, L. Foong, An efficient Harris hawks-inspired image segmentation method, *Expert Syst. Appl.* 155 (2020) 113428.
- [27] T. Kurban, P. Civicioglu, R. Kurban, E. Besdok, Comparison of evolutionary and swarm based computational techniques for multilevel color image thresholding, *Appl. Soft Comput.* 23 (2014) 128–143.
- [28] C. Lang, H. Jia, Kapur's entropy for color image segmentation based on a hybrid whale optimization algorithm, *Entropy* 21 (2019) 318.
- [29] B. Kucukugurlu, E. Gedikli, Symbiotic Organisms Search Algorithm for multilevel thresholding of images, *Expert Syst. Appl.* 147 (2020) 113210.
- [30] R. Monisha, R. Mrinalini, M. Britto, R. Ramakrishnan, V. Rajinikanth, Social group optimization and Shannon's function-based RGB image multi-level thresholding, in: *Smart Intelligent Computing and Applications*, Springer Singapore, Singapore, 2018, pp. 123–132.
- [31] T. Sağ, M. Çunkaş, Color image segmentation based on multiobjective artificial bee colony optimization, *Appl. Soft Comput.* 34 (2015) 389–401.
- [32] K. Sabeena Beevi, M. Nair, G. Bindu, Automatic segmentation of cell nuclei using Krill Herd optimization based multi-thresholding and Localized Active Contour Model, *Biocybern. Biomed. Eng.* 36 (2016) 584–596.
- [33] L. He, S. Huang, An efficient krill herd algorithm for color image multilevel thresholding segmentation problem, *Appl. Soft Comput.* 89 (2020) 106063.
- [34] S. Pare, A. Bhandari, A. Kumar, G. Singh, Rényi's entropy and bat algorithm based color image multilevel thresholding, in: *Advances in Intelligent Systems and Computing*, Springer Singapore, Singapore, 2018, pp. 71–84.
- [35] N. Nimbarte, M. Mushrif, Multi-level thresholding algorithm for color image segmentation, in: *Second International Conference on Computer Engineering and Applications*, Vol. 2, Bali Island, Indonesia, 2010, pp. 231–233.
- [36] W.A. Hussein, S. Sahran, S.N.H.S. Abdullah, A fast scheme for multilevel thresholding based on a modified bees algorithm, *Knowl.-Based Syst.* 101 (2016) 114–134.
- [37] Z. Xing, An improved emperor penguin optimization based multilevel thresholding for color image segmentation, *Knowl.-Based Syst.* 194 (2020) 105570.
- [38] D. Martin, C. Fowlkes, D. Tal, J. Malik,
- [39] M. Mokji, S. Bakar, Adaptive thresholding based on co-occurrence matrix edge information, in: *First Asia International Conference on Modelling & Simulation (AMS'07)*, Phuket, Thailand, 2007, pp. 444–450.
- [40] R. Jensi, G. Jiji, An enhanced particle swarm optimization with levy flight for global optimization, *Appl. Soft Comput.* 43 (2016) 248–261.
- [41] Z. Li, Y. Zhou, S. Zhang, J. Song, Lévy-flight moth-flame algorithm for function optimization and engineering design problems, *Math. Probl. Eng.* 2016 (2016) 1–22.
- [42] R. Salgotra, U. Singh, S. Saha, New cuckoo search algorithms with enhanced exploration and exploitation properties, *Expert Syst. Appl.* 95 (2018) 384–420.
- [43] G. Viswanathan, V. Afanasyev, S. Buldyrev, E. Murphy, P. Prince, H. Stanley, Lévy flight search patterns of wandering albatrosses, *Nature* 381 (1996) 413–415.
- [44] X. Yang, S. Deb, Multiobjective cuckoo search for design optimization, *Comput. Oper. Res.* 40 (2013) 1616–1624.
- [45] D. Oliva, E. Cuevas, G. Pajares, D. Zaldivar, V. Osuna, A multilevel thresholding algorithm using electromagnetism optimization, *Neurocomputing* 139 (2014) 357–381.
- [46] Z. Wang, A. Bovik, H. Sheikh, E. Simoncelli, Image quality assessment: From error visibility to structural similarity, *IEEE Trans. Image Process.* 13 (2004) 600–612.
- [47] L. Zhang, L. Zhang, X. Mou, D. Zhang, FSIM: A feature similarity index for image quality assessment, *IEEE Trans. Image Process.* 20 (2011) 2378–2386.
- [48] P. Ren, Y. Xiao, X. Chang, P.Y. Huang, Z. Li, X. Chen, X. Wang, A comprehensive survey of neural architecture search: Challenges and solutions, 2020, arXiv preprint arXiv:2006.02903.
- [49] X. Chang, F. Nie, S. Wang, Y. Yang, X. Zhou, C. Zhang, Compound rank-k projections for bilinear analysis, *IEEE Trans. Neural Netw. Learn. Syst.* 27 (7) (2015) 1502–1513.