Multimedia Tools and Applications (2023) 82:40701–40743
https://doi.org/10.1007/s11042-023-15129-y

Improving the segmentation of digital images by using a modified Otsu's between-class variance

Simrandeep Singh[1,2] · Nitin Mittal[3] · Harbinder Singh[4] · Diego Oliva[5]

Received: 22 December 2021 / Revised: 8 October 2022 / Accepted: 13 March 2023 /
Published online: 31 March 2023

Abstract

Image segmentation is a critical stage in the analysis and pre-processing of images. It comprises dividing the pixels according to threshold values into several segments depending on their intensity levels. Selecting the best threshold values is the most challenging task in segmentation. Because of their simplicity, resilience, reduced convergence time, and accuracy, standard multi-level thresholding (MT) approaches are more effective than bi-level thresholding methods. With increasing thresholds, computer complexity grows exponentially. A considerable number of metaheuristics were used to optimize these problems. One of the best image segmentation methods is Otsu's between-class variance. It maximizes the between-class variance to determine image threshold values. In this manuscript, a new modified Otsu function is proposed that hybridizes the concept of Otsu's between class variance and Kapur's entropy. For Kapur's entropy, a threshold value of an image is selected by maximizing the entropy of the object and background pixels. The proposed modified Otsu technique combines the ability to find an optimal threshold that maximizes the overall entropy from Kapur's and the maximum variance

✉ Harbinder Singh
harbinder.ece@gmail.com

Simrandeep Singh
staff.simrandeep.singh@iitrpr.ac.in; simrandeepsinghcgc@gmail.com

Diego Oliva
diego.oliva@cucei.udg.mx

[1] Department of Computer Science and Engineering, AWaDH, IIT Ropar, Rupnagar 140001, India

[2] Department of Electronics & Communication Engineering, UCRD, Chandigarh University, Gharuan, Punjab, India

[3] Department of Skill Faculty of Science and Technology, Shri Vishwakarma Skill University, Palwal, Haryana 121102, India

[4] Department of Electronics & Communication Engineering, Chandigarh Engineering College, Landran, Punjab, India

[5] Depto. de Innovación Basada en la Información y el Conocimiento, Universidad de Guadalajara, CUCEI, Guadalajara, Jal, Mexico

value of the different classes from Otsu. The novelty of the proposal is the merging of two methodologies. Clearly, Otsu's variance could be improved since the entropy (Kapur) is a method used to verify the uncertainty of a set of information. This paper applies the proposed technique over a set of images with diverse histograms, which are taken from Berkeley Segmentation Data Set 500 (BSDS500). For the search capability of the segmentation methodology, the Arithmetic Optimization algorithm (AOA), the Hybrid Dragonfly algorithm, and Firefly Algorithm (HDAFA) are employed. The proposed approach is compared with the existing state-of-art objective function of Otsu and Kapur. Qualitative experimental outcomes demonstrate that modified Otsu is highly efficient in terms of performance metrics such as PSNR, mean, threshold values, number of iterations taken to converge, and image segmentation quality.

Keywords Image segmentation · Multi-level thresholding · Metaheuristics · Otsu · Kapur's entropy

1 Introduction

In the domain of computer vision and image processing, image segmentation is a key component. A single image is always better than thousand words. Segmentation aims to extract features and track information of an image. Image segmentation divides the image into a set of non-overlapping contours based on certain properties like texture, color, homogeneity, and structure [37, 39]. An automatic image segmentation process always remains a very complex procedure in image processing. This process becomes more sophisticated when test images are natural, realistic, and degraded. An algorithm may be called a good segmentation technique if it is able to differentiate among different classes of images or frontiers. The major issue in segmentation is identifying the scene elements successfully in an image. Further application of image thresholding is image recognition. Image thresholding can be categorized into, i.e. single-level and multi-level thresholding methods. In single-level thresholding segmentation processes, only one threshold value is needed, while in multi-level, multiple thresholds are required. The main problem is to find the appropriate values for each picture based on the histogram [31]. Image segmentation can be used in a wide variety of applications, including different fields like biomedical, satellite, infrared (IR), surveillance, agricultural images, etc. These other modalities have unique information, and therefore it is a complex process to make a generalized automatic thresholding technique.

In 1979, Otsu proposed a thresholding method that maximizes the between-class variance and minimizes intraclass variance to achieve optimum threshold values and generates better results [1]. In order to perform segmentation based on image thresholding, Otsu's between class variance and other maximum entropy methods like Kapur's entropy [13], Renyi's entropy [10], and Tsallis entropy [30] have been developed. This approach combines information theory successfully, but the probability of a gray level value being shown primarily affects the methodologies. Another reason for influencing or affecting the segmentation results is to ignore the gray level value of the pixels. The brightness and contrast of an image are not affected by the 1-D Otsu method. Furthermore, it is a procedure with less computation cost for a small number of thresholds. Nevertheless, the algorithm primarily considers only the gray level value; for that reason, it fails to produce optimum results in the case of noisy images.

Tsallis in [2] suggested using the idea of moment-preserving to create a threshold approach for a robust gray image. Kapur, Sahoo, and Wong utilized histogram entropy in a method termed Kapur entropy to discover optimal thresholds [21] and commonly employed the approach to detect the problem of picture threshold segmentation. Cross-entropy reduces the cross-entropy of a picture and the segmented image [10, 26], and the optimal threshold is also utilized for detecting the ideal thresholds. Entropy-based methods are more common among all the methods listed above. Several thresholding methods have been included in the literature [48].

In addition, these approaches may be easily applied to the segmentation of thresholding using multi-levels. For many thresholds, however, the computational time will increase exponentially as they hunt for the best threshold values to improve objective features, resulting in increasingly longer calculation times. Separating objects from backgrounds is the hardest and most complex task in image processing [1]. Background and foreground can be differentiated using the segmentation process. To define multiple regions of an image, multi-level thresholding techniques have been developed, but the computational cost also increases with an increasing number of thresholds. Given this, it is necessary to adopt algorithms that help to search for the optimal threshold values. The most preferred and cost-effective procedure for thresholding is using metaheuristic or optimization algorithms [13, 37].

On the other hand, metaheuristic algorithms (MA) are used to define the process of finding the optimal solution out of available solutions while considering different constraints [10, 30]. This property of the optimization algorithm is used to find the optimal thresholding value used for image thresholding. Different objective functions like Otsu, Kapur, Renyi, Tsallis, etc. [2, 21, 26] are defined to calculate optimal threshold values. Some recently proposed MA are the Hybrid Dragonfly algorithm (DA) and Firefly Algorithm (FA) (HDAFA) [39], the Starling Murmuration Optimizer [48], the Conscious Neighborhood-based Crow Search Algorithm (CCSA) [46], the Quantum-based avian navigation optimizer algorithm (QANA) [47], the Opposition-based Moth Swarm Algorithm (OBMSA) [32], the Archimedes optimization algorithm (AOA) [14], to mention some.

In the related literature, metaheuristic algorithms and their improved versions are used for multi-level segmentation. Such as artificial bee colony (ABC) [44], genetic algorithms (GA) [40], honey bee mating optimization (HBMO) [17], modified firefly algorithm (MFA) [15], particle swarm optimization (PSO) [30], bacterial foraging algorithm (BFA) [45], differential evolution (DE) algorithms [25], wind-driven optimization(WDO) [24], cuckoo search (CS) algorithms [31], ant colony algorithm (ACO) [11], grasshopper optimization algorithm (GOA) [27], self-adaptive parameter optimization (SAPO) [8], electromagnetism-like optimization (EMO) algorithm [16], and glowworm swarm optimization (GSO) algorithms [29]. Some other interesting approaches that successfully segment digital images are based on modern MA as the Coronavirus Optimization Algorithm combined with Harris Hawks Optimizer [18], the improved modified Differential Evolution (MDE) [35], the directional mutation and crossover boosted ant colony optimization (XMACO) [34], the Harris Hawks Optimizer (HHO) [36], the Mutated Electromagnetic Field Optimization (MEFO) [4] or the Altruistic Harris Hawks Optimizer [5]. From these methods it is possible to see that the use of MA for thresholding also benefits field as medicine, where the proper analysis of the images is crucial for a good diagnosis.

The paper aims to present a methodology for image segmentation using the Arithmetic Optimization algorithm (AOA) [3] and Hybrid Dragonfly algorithm (DA) and Firefly

Algorithm (FA) HDAFA using modified Otsu's function. The results of modified Otsu's are compared with the standard Otsu and Kapur's methods. The suggested approach combines local properties performed using Otsu with the entropy of Kapur. In this context, the maximum variance value from multiple Otsu classes is coupled with the maximum total entropy computed from Kapur's entropy method. The novelty of the proposed hybrid objective function is the combination of the between-class variance with the entropy. Since entropy is a tool that helps to measure the uncertainty of a set of information, it clearly improves the segmentation by selecting the appropriate thresholds that reduce such uncertainties by maximizing the entropy and the variance at the same time.

On the other hand, due to its simple and easy application, AOA has been applied to address various real-time problems, like encompassing the lifetime of the RFID network, photovoltaic systems, and range-based wireless node localization [41]. In this paper, the application of AOA is extended to multi-level digital image segmentation by thresholding. Although randomization and static swarm behavior have a great worldwide search capability for AOA, its local search capabilities are limited and result in local optima capturing. At the original phases, the iteration level hybridization method guarantees exploration capability and exploitation capability in the subsequent phases and ensures an enhanced accuracy of the global optimum [41].

It uses common mathematical operations such as Division (D), Addition (A), Multiplication (M), and Subtraction (S), which are applied and modeled to execute optimization in a wide variety of search fields. Population-based algorithms (PBA) [7] commonly launch their improvement processes by randomly selecting several candidate solutions. A defined solution is enhanced incrementally by a set of optimization rules and analyzed sequentially by a particular objective function, which is the basis of optimization techniques. Although PBA is stochastically trying to find some efficient strategy for optimization problems, a single-run solution is not guaranteed. However, a large set of possible solutions and optimization simulations improve the chance of an optimum global solution to the problem c. The major contributions of this paper may be summarized as:

- Propose a modified Otsu method and use it as an objective function for image segmentation.
- The local properties of Otsu's between-class variance, like the maximum variance value achieved from multiple Otsu classes, are combined with maximum total entropy calculated from the Kapur method.
- The effectiveness and quality of solutions generated by modified Otsu's are evaluated using arithmetic optimization algorithm (AOA), hybrid dragonfly algorithm, and firefly algorithm HDAFA.
- The performance of the modified Otsu has been investigated on images widely used in the image processing literature and compared with a basic Otsu method.
- Quantitative analysis is carried out using different threshold values, PSNR, mean, STD, and number of iterations.

The structure of this manuscript is as follows, Section 2, presents the basic concepts of image segmentation and the modified Otsu approach. Section 3 includes AOA, motivation, exploration, and exploitation stages. Section 4, provides statistical testing/ qualitative parameters, performance evaluation and comparison, experimental results, and related comparative analysis, among other state-of-art. Finally, the paper is concluded in section 5.

2 Image segmentation

Each local region of a picture has a distinct threshold based on its features. Histogram-based thresholding is the most popular approach for segmenting digital pictures. The automated separation of pictures and context is the most active and intriguing field of image processing and pattern recognition [37]. However, it can also provide original forecasts or pre-processing for more complicated phases.

Depending on the threshold values needed to segment the image, thresholding may be defined by bi-level (BT) or multi-level (MT) thresholding defined by Eq. 1. Bi-level thresholding divides an image into two distinct regions, while MT creates multiple regions in an image.

$$b(x,y) = \left\{ \begin{array}{c} a_1 \leftarrow b(x,y) \;\; if \; 0 \leq b(x,y) < T \\ a_2 \leftarrow b(x,y) if \;\; T \leq b(x,y) < L{-}1 \end{array} \right\} \tag{1}$$

For a given image $b(x,\ y)$ having a size $(m\ \times\ n)$ with L intensity levels, a_1 and a_2 are two different classes depending on the threshold value T. While opting for bilevel thresholding, extra care should be seen to find a suitable correct threshold value (T). To improve segmentation results, MT is used in many cases [7]. Besides, the traditional techniques are calculatedly costly owing to the multimodality of the histogram if many thresholds are necessary. This is a difficult question; thus, scholars have identified several ways for the pre-eminent gray threshold [24].

2.1 Otsu's between class variance

The between-class variance proposed by Otsu is a nonparametric automatic method for image segmentation that employs threshold values [19]. In Otsu's method, the intraclass variance is calculated and it provides optimum threshold values. a_1, a_2, …. , a_n are different classes of image with different threshold values. In this sense to employ the Otsu's between class variance, it is necessary to compute the probability distribution p_i that is given as:

$$p_i = \frac{n_i}{N} \tag{2}$$

where n_i is a number of a pixel having grey level *i,* and N is the total number of pixels. The average grey level of an image I is given by

$$\mu_I = \sum_{i=0}^{L-1} iP_i \tag{3}$$

$$a_1 = \frac{P_0}{\omega_a}, \frac{P_1}{\omega_a}, \frac{P_2}{\omega_a} \ldots\ldots \frac{P_T}{\omega_a} \tag{4}$$

$$a_2 = \frac{P_{T+1}}{\omega_b}, \frac{P_{T+2}}{\omega_b} \ldots\ldots \frac{P_{L-1}}{\omega_b} \tag{5}$$

Where, $\omega_a = \sum_{i=0}^{T} p_i, \;\; \omega_b = \sum_{i=T+1}^{L-1} p_i$

The average levels of μ_a and μ_b for two classes of a_1 and a_2 are as follows:

$$\mu_a(k) = \sum_{i=0}^{T} \frac{ip_i}{\omega_a}, \mu_b(k) = \sum_{i=T+1}^{L-1} \frac{ip_i}{\omega_b} \tag{6}$$

If the mean intensity of the image is given by μ_I then

$$\omega_a\mu_a + \omega_b\mu_b = \mu_I, \qquad \omega_a + \omega_b = 1 \tag{7}$$

Between class variance σ_B^2 needs to be maximized to find the optimal thresholding value using the following equation:

$$\sigma_B^2 = \omega_a(\mu_a - \mu_I)^2 + \omega_b(\ \mu_b - \mu_I)^2 \tag{8}$$

Bilevel thresholding generates two separate regions based on the intensity of a single threshold value. A further extension from bi-level to multi-level thresholding may be carried out for Otsu. In addition, multi-level thresholding generates several regions [a_1, a_2, a_3, a_i, ……. . a_n] based on the following rules:

$$\begin{array}{c} a_1 \leftarrow b(x,y)\ \mathit{if}\ 0 < b(x,y) < T_1 \\ a_2 \leftarrow b(x,y)\ \mathit{if}\ T_1 < b(x,y) < T_2 \\ a_3 \leftarrow b(x,y)\ \mathit{if}\ T_2 < b(x,y) < T_3 \\ a_i \leftarrow b(x,y)\ \mathit{if}\ T_{i-1} < b(x,y) < T_{i+1} \\ . \\ . \\ a_n \leftarrow b(x,y)\ \mathit{if}\ T_{n-1} < b(x,y) < L-1 \end{array} \tag{9}$$

where n defines the number of classes like [a_1, a_2, a_3, a_i, ……. . a_n] considering i as a certain class for a given image $b(x,\ y)$ having L gray levels (1,2,…, L) in the range [0, L-1]. Extended between the class value is given by $f(k)$ defined as the between-class variance.

$$f(k) = \sum_{i=1}^{M} \omega_i(\mu_i - \mu_T)^2 \tag{10}$$

while considering the above classes, the Otsu method can be easily extended to multi-level thresholding for M-1 thresholding levels. Where ω_i is a zeroth-order cumulative moment for i^{th} class and μ_T is mean intensity for hole image. In 2001 Liao proposed a simple and less complex alternative method given by the following equation for k thresholds [28].

$$f(k) = \omega_a(\mathrm{k}){\mu_a}^2(k) + \omega_b(k){\mu_b}^2(k) \tag{11}$$

From Eq. 11, μ_a and μ_b are average levels for classes a_1 and a_2 already explained. If the between-class variance has a maximum value, then within class variable will always have a minimum value. This methodology is described in Eq. 12

$$f_{OTSU}(T) = \varnothing_o = \mathrm{Arg} \quad \max(f(k)), \qquad 0 \le k \le L-1 \tag{12}$$

where f_{OTSU} represents fitness function, and maximizing this would correspond to optimal intensity threshold levels. The fitness function considering i multi-level threshold values is given by Eq. 13.

$$f_{OTSU}(T_i) = \varnothing_o = \text{Arg}\max(f(\text{k}_i)), \qquad 0 \le k \le L-1,\ \ i = 1, 2, \ldots\ldots.., \text{T} \tag{13}$$

2.2 Kapur's entropy

The basic concept underlying Kapur's entropy approach is Shannon entropy. Shannon proposed an entropy function based on the idea that the probability of occurrence is inversely proportional to information [20]. The Shannon entropy for a system H is defined in Eq. 14.

$$H = -\sum_{i=1}^{n} P_i log_2 P_i \tag{14}$$

Where H is Shannon entropy, P_i is the probability of the i^{th} gray level, and n is the total number of pixels. Kapur's entropy is implemented using the probability distribution of the gray-level histogram. The maximal value of Kapur's entropy is obtained from the optimal threshold value. To define it, let us consider $p_a, p_b, \ldots.\ p_s$ as the probability distribution of gray level. Two distinct distributions for object and background are derived from discrete values in the range of a to m, and another value proceeds from $1 + m$ to n; these are symbolized by A and B and are defined as follows:

$$\boldsymbol{A} : \frac{\boldsymbol{p_a}}{\boldsymbol{p_m}}, \frac{\boldsymbol{p_b}}{\boldsymbol{p_m}}, \ldots\ldots \frac{\boldsymbol{p_s}}{\boldsymbol{p_m}} \tag{15}$$

$$\boldsymbol{B} : \frac{\boldsymbol{p_{m+1}}}{\boldsymbol{1-p_m}}, \frac{\boldsymbol{p_{m+2}}}{\boldsymbol{1-p_m}}, \ldots\ldots \frac{\boldsymbol{p_n}}{\boldsymbol{1-p_m}} \tag{16}$$

The following equations define the corresponding entropies for the probability distributions A and B:

$$\boldsymbol{H}(\boldsymbol{A}) = -\sum_{i=a}^{s} \frac{\boldsymbol{p_a}}{\boldsymbol{p_m}} \boldsymbol{ln} \frac{\boldsymbol{p_a}}{\boldsymbol{p_m}} = -\frac{\boldsymbol{1}}{\boldsymbol{p_m}} \left[\sum_{i=a}^{m} \boldsymbol{p_a ln\, p_a - p_a ln\, p_a} \right] = \boldsymbol{ln\, p_m} + \frac{\boldsymbol{H_m}}{\boldsymbol{p_m}} \tag{17}$$

$$\begin{aligned}\boldsymbol{H}(\boldsymbol{B}) &= -\sum_{i=1+m}^{n} \frac{\boldsymbol{p_i}}{\boldsymbol{1-p_m}} \boldsymbol{ln} \frac{\boldsymbol{p_i}}{\boldsymbol{1-p_m}} = -\frac{\boldsymbol{1}}{\boldsymbol{1-p_m}} \left[\sum_{i=m+1}^{n} \boldsymbol{p_i ln\, p_i - (1-p_m) ln\ (1-p_m)} \right] \\ &= \boldsymbol{ln(1-p_m)} + \frac{\boldsymbol{H_n - H_m}}{\boldsymbol{1-p_m}}\end{aligned} \tag{18}$$

In order to find the optimal threshold value, a sum of these two entropies must have attained maximum value, and the final expression is defined in Eq. 19.

$$\begin{aligned}\varphi_s &= \boldsymbol{H_A} + \text{H}_{\boldsymbol{B}} = \boldsymbol{ln\, p_m} + \frac{\boldsymbol{H_m}}{\boldsymbol{p_m}} + \boldsymbol{ln(1-p_m)} + \frac{\boldsymbol{H_n - H_m}}{\boldsymbol{1-p_m}} \\ \varphi_m &= \boldsymbol{ln\, p_m(1-p_m)} + \frac{\boldsymbol{(H_n - H_m)p_m + H_m(1-p_m)}}{\boldsymbol{p_m(1-p_m)}}\end{aligned} \tag{19}$$

Notice that $\boldsymbol{Max}(\boldsymbol{\varphi_s})$ gives optimal threshold value with the well-separated object and background details.

Multi-level thresholding using Kapur's entropy Adopting the entropy-based segmentation can only be beneficial if multi-level segmentation is performed. Kapur's entropy method can be stretched from two-level to MT. The entropy of an image is a measure of its compactness and addresses how separation may be carried out among different classes. An image may be divided into different segments using multiple threshold values, as explained in the next equations.

$$\begin{aligned} H(a) &= -\sum_{i=a}^{t_1-1}\frac{p_i}{p_a}ln\frac{p_i}{p_a}, & p_0 &= \sum_{i=a}^{t_1-1}p_i \\ H(b) &= -\sum_{i=a}^{t_2-1}\frac{p_i}{p_b}ln\frac{p_i}{p_b}, & p_1 &= \sum_{i=a}^{t_2-1}p_i \\ H(c) &= -\sum_{i=a}^{t_3-1}\frac{p_i}{p_c}ln\frac{p_i}{p_c}, & p_2 &= \sum_{i=a}^{t_3-1}p_i \\ H(j) &= -\sum_{i=t_j}^{t_{j+1}-1}\frac{p_i}{p_j}ln\frac{p_i}{p_j}, & p_j &= \sum_{i=t_j}^{t_{j+1}-1}p_i \\ H(m) &= -\sum_{i=t_m}^{L-1}\frac{p_i}{p_m}ln\frac{p_i}{p_m}, & p_m &= \sum_{i=t_m}^{L-1}p_i \end{aligned} \tag{20}$$

Where, $\boldsymbol{p_i}$ is the probability that event i occurs, H denotes entropy, m represents dimensions and $\boldsymbol{p_a}$,$\boldsymbol{p_b}$, …,$\boldsymbol{p_m}$ represents the probability of grey level for different areas in a multi-dimensional way. The optimal threshold value is chosen analogously, such the objective function is maximized [43]. The Eq. 21 defines the Kapur's objective function.

$$f_{Kapur}(th) = \varnothing_k = argmax\sum_{i=0}^{m}H_i(th),\quad 0\le th\le L-1 \tag{21}$$

To extend the above expression for multi-level thresholding Kapur's objective function is defined as follows:

$$f_{Kapur}(TH) = f_{Kapur}(th_i),\quad i = 1,2,3,\ldots,k \tag{22}$$

where different thresholds are represented by $TH\,[th_1,\ th_2,\ th_3,\ \ldots,\ th_{k-1}]$ and i correspond to a specific class.

2.3 A modified Otsu's between class variance

In Kapur's entropy, if $H(A)$ and $H(B)$ are identical or similar, i.e., variance between the classes is less than both classes are same as given below:

$$H(A) = H(B) \tag{23}$$

$$ln\,p_s + \frac{H_s}{p_s} = ln(1-p_s) + \frac{H_n-H_s}{1-p_s} \tag{24}$$

In general terms, a value of $\varnothing_k$ will be maximum at $s = \frac{1}{2}n$ for symmetrical distribution, where n is the total number of grey levels. However, this concept cannot be useful for images having multiple objects superimposed on the same background. If an image has two different objects with the same histogram level to be segmented, that image can be segmented using the proposed multi-level thresholding method [6, 28]. In the proposed methodology, the local properties carried out using Otsu's variance are combined with Kapur's entropy. In this case, the maximum variance value obtained from different classes of Otsu is combined with the maximum overall entropy calculated from Kapur's entropy. The proposed method presented in Eq. 25 is the proposed modification of Otsu. Here Kapur's entropy function is used as a weight function for Otsu.

$$\psi_i = f_{OTSU}(T_i) \cdot f_{Kapur}(th_i) \tag{25}$$

Otsu and Kapur's entropy objective functions fall under the optimization problems used in image segmentation. Otsu maximizes the between-class variance, and Kapur's entropy maximizes posterior entropy. As discussed in subsequent sections, the experimental results prove that the proposed modified algorithm can produce better segmentation results.

3 Optimizing the proposed improved Otsu's method

This section explains how the different metaheuristic algorithms could be used to optimize the modified Otsu's method proposed in the previous section. Here two algorithms are discussed that will be used to generate an optimum value of multi-level threshold value using various objective functions like modified Otsu, the standard version Otsu, and Kapur's method. Basically, the thresholding techniques are treated as optimization problems. To exemplify the implementation of metaheuristics to search for the best configuration of thresholds, they are used the Arithmetic Optimization algorithm (AOA) [3] and the Hybrid Dragonfly algorithm and Firefly Algorithm (HDAFA) [39]. The basic concepts, such as optimization techniques, are explained in this section. Finally, their implementation for image thresholding is also described.

3.1 AOA to optimize the modified Otsu's function

Considering the variations among meta-heuristic methods in population-based approaches, the optimization process comprises two cycles: exploitation vs. exploration. The previous examples of extensive coverage are search fields utilizing search agents to bypass local solutions. Above is the increase in the performance of solutions achieved during the exploration process, as shown in Fig. 1.

The AOA algorithm intends to explore the search space position (optimum threshold value) for the multi-level threshold issue that maximizes the objective function considered in Eqs. 13, 22, and 25. The intake to this method is an image, and the output indicates the optimum threshold. As discussed in Eq. 25, ψ_i represents modified Otsu fitness function, and maximizing this would correspond to optimal intensity threshold levels.

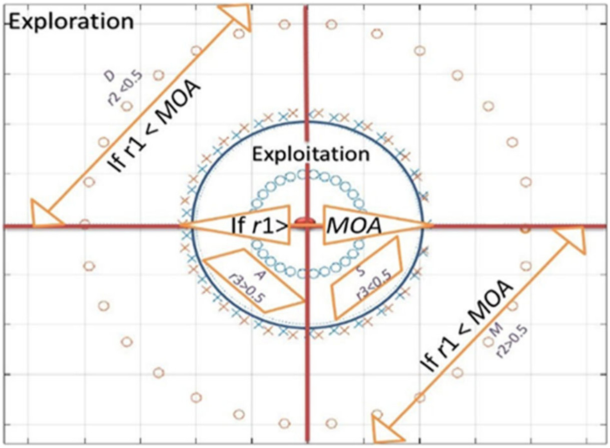


Fig. 1 AOA search phases [3]

$$\psi_i(T) = \varnothing_o = \max(\psi_i(k)), \quad 0 \le k \le L-1 \tag{26}$$

The following equations give the fitness for multi-level thresholding.

$$\psi_i(T_i) = \varnothing_o = \max(\psi_i(k_i)), \quad 0 \le k \le L-1, \qquad i = 1, 2, \ldots\ldots.., \mathrm{T} \tag{27}$$

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & .. & .. & a_{1,j} & a_{1,1} & a_{1,T} \\ a_{2,1} & a_{2,2} & .. & .. & a_{2,j} & .. & a_{2,T} \\ a_{3,1} & a_{3,2} & .. & .. & .. & .. & .. \\ .. & .. & .. & .. & .. & .. & .. \\ a_{N-1,1} & .. & .. & .. & a_{N-1,j} & .. & a_{N-1,T} \\ a_{N,1} & .. & .. & .. & a_{N,j} & a_{N,T-1} & a_{N,T} \end{bmatrix} \tag{28}$$

The optimization method starts with selected sets denoted by A as in Eq. 28, A denotes the whole population, and N is the number of elements in the population. The ideal set in every iteration is created randomly and is taken as the optimum threshold value. A grayscale image with a pixel value between 0 and 255 has the probability of a threshold value between the above-said limit. Therefore, lower and upper limits for matrix element $a_{1,j}$ is 0 and 255, respectively. Exploitation/Exploration should be carefully chosen at the start of AOA for image segmentation. The coefficient of math optimizer accelerated (MOA) is defined in Eq.29.

$$MOA\ (C_{iter}) = Min + C_{iter} \times \left(\frac{Max - Min}{M_{iter}} \right) \tag{29}$$

where, MOA (C_{iter}) = i^{th} corresponds to the iteration function value, M_{iter} is the maximum number of iterations, Max and Min is the accelerated function of Max. and Min. Values C_{iter} is the current iteration (within 1 and M_{iter}).

Exploration stage The exploratory nature of AOA is discussed as per the AO mathematical calculations, whether Division (D) or Multiplication (M) operators have obtained high distribution values or decisions that contribute to an exploration search method. However, as opposed to other operators, these D and M operators never easily reach the objective due to the high distribution of S and A operators. AOA exploration operators exploit the whole image arbitrarily through many regions based on the modified Otsu values and seek a better alternative (threshold values) dependent on two fundamental search techniques M and D search techniques, as shown in Eq. 30.

$$a_{i,j}(C_{iter}+1) = \begin{cases} besta_j \div (MOP \div \varepsilon) \times \left(\left(UB_j – LB_j \right) \times \mu + LB_j \right), r_2 < 0.5 \\ besta_j \times MOP \times \left(\left(UB_j – LB_j \right) \times \mu + LB_j \right), otherwise \end{cases} \tag{30}$$

Where, $a_i(C_{iter} + 1) = i^{th}$ is a solution for the next iteration, $a_{i,j}(C_{iter} + 1) = j^{th}$ is the position in the current iteration, μ is a control parameter that must have a value lower or equal to 0.5. LB_j and UB_j are the lower and upper bound limits, ε is the smallest integer value, and finally, $besta_j$ is the j^{th} position of the optimal solution (threshold value) so far. From Eq. 28 the Math Optimizer Probability coefficient (MOP) is computed as follows:

$$MOP(C_{iter}) = 1 – \frac{C_{iter}{}^{\frac{1}{\alpha}}}{M_{iter}{}^{\frac{1}{\alpha}}} \tag{31}$$

where, $MOP(C_{iter}) = i^{th}$ is the iteration function value, C_{iter} corresponds to the current iteration, M_{iter} is the maximum number of iterations.

Exploitation stage To apply AOA to image processing applications, the exploitation nature of AOA makes use of AO mathematical formulas, whether using addition (A) or subtraction (S) as they provide high-density results. AOA exploitation operators exploit the search field deeply through many regions of an image and seek a better threshold value dependent on two key search techniques A and S search techniques, as shown in Eq. 32.

$$a_{i,j}(C_{iter}+1) = \begin{cases} besta_j – MOP \times \left(\left(UB_j – LB_j \right) \times \mu + LB_j \right), r_3 < 0.5 \\ besta_j + MOP \times \left(\left(UB_j – LB_j \right) \times \mu + LB_j \right), otherwise \end{cases} \tag{32}$$

The description of AOA is presented in Algorithm 1, and the flow chart of the proposed technique implemented using AOA is discussed in Fig. 2.

Algorithm 1. Description of the steps of the AOA.

```
By initializing position of solutions randomly $(i = 1,2,3,4,\dots\dots\dots,N)$ and parameters of AOA
(α & μ).

    While $C_{iter} < M_{iter}$
      Find fitness values from solution using eq. 27
      Determine optimum solution
      Update MOA & MOP from Eq.29 & 31.
for (i=1 for solutions) & (j=1 for positions)
Random no. b/t (0,1) $(r_1, r_2, \dots\dots\dots)$
if $r_1 > MOA$ then
          Exploration Stage
        if $r_2 > 0.5$ then
          (i)Use Division $(D \div)$ then update $i^{th}$value in Eq. 30 (1st rule)
        else
          (ii)Use Multiplication $(M \times)$ then update $i^{th}$value in Eq. 30 (2nd rule)
        end if
else
        Exploitation stage
        if $r_3 > 0.5$ then
          (i)Use Subtraction $(S-)$ then update $i^{th}$value in Eq. 32 (1st rule)
        else
          (ii)Use Addition $(A+)$ then update $i^{th}$value in Eq. 32 (2nd rule)
        end if
end if
end for
    $C_{iter} = C_{iter} + 1$
end while
return optimum threshold value solutions(A)
```

3.2 HDAFA to optimize the modified Otsu function

Dragonfly (DF) behavior follows concepts of separation, harmonization, cohesiveness, the distraction of the opponent, and the attraction of food. The search space is the answer to every dragon flying in the swarm. A swarm motion of dragonflies is determined by five separate operators such as separation, alignment, cohesiveness, food attraction, and the diversion towards hostile sources. Separation (Si) pertains to static collision prevention between individuals and persons living in the area. Alignment (Ai) is about the pace of people in the neighbourhood matching to others. Cohesion (Ci) concerns people's tendency to the mass centre of the neighbourhood. The proper weights are assigned for each operator and adapted for the convergence of DF to the best solution. The nearby range of the DF also improves with the progress of the optimization technique. The mathematical application of DA can be explained below. Considering the population of the *N* dragonfly Eq. 33 is the location of the i^{th} dragonfly [39].

$$X_i = \left(x_i^1, x_i^2, \dots, x_i^d, \dots, x_i^N\right) \tag{33}$$

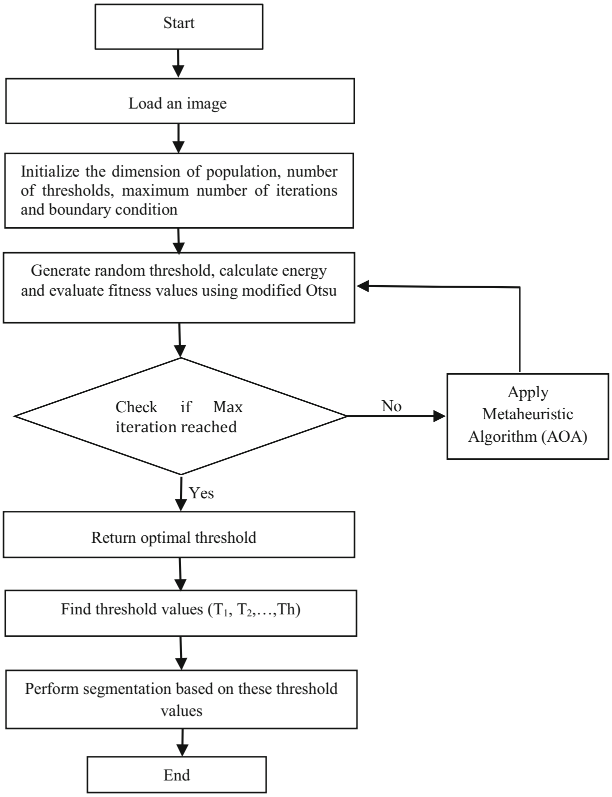

Start
Load an image
Initialize the dimension of population, number of thresholds, maximum number of iterations and boundary condition
Generate random threshold, calculate energy and evaluate fitness values using modified Otsu
Check if Max iteration reached
No
Apply Metaheuristic Algorithm (AOA)
Yes
Return optimal threshold
Find threshold values (T1, T2,…,Th)
Perform segmentation based on these threshold values
End

Fig. 2 Flow chart of segmentation using AOA

If the search space of the dragonfly $i = 1, 2, 3, \ldots, \mathrm{N}, x_i^d$ and the search agent number N. is the search agent. The objective function is assessed based on the starting position values altered between the variables' lower and higher bounds. Weighing rands (s), lines (a), cohesiveness (c), feed (f), and adversary (e) are randomly initialized for every dragonfly. The position and speed of dragonfly separation are estimated with Eqs. 34–36, alignment and cohesiveness factors.

$$S_i = -\sum_{j=1}^{N} X - X_i \tag{34}$$

$$A_i = \frac{\sum_{j=1}^{N} V_i}{N} \tag{35}$$

$$C_i = \frac{\sum_{j=1}^{N} X_i}{N} - X \tag{36}$$

Where X_i refers to the position and V_i is the speed of the person. X refers to the current condition of persons, and N refers to the number of neighboring persons. Eqs. 37 and 38, respectively, are computed for F_i and E_i adversary diversionfFood source attraction.

$$F_i = X^{+} - X \tag{37}$$

$$E_i = X^{-} + X \tag{38}$$

Here X represents the current position of the person and X^{+} shows the food supply and X^{-} shows the enemy's source. The distance from the locality is computed with the N of Euclid between all the dragonflies calculated and selected. The distance r_{ij} is calculated by Eq. 39.

$$r_{ij} = \sqrt{\sum_{k=1}^{d} \left(x_{i,k} - x_{j,k} \right)^2} \tag{39}$$

If there is at least a DF in the area, the speed of the DF is arranged following Particle Swarm Optimization (PSO) speed equation Eq. 40 [23]. The location of the DF is updated using Eq. 41, and this equation is comparable to PSO's position equation.

$$\Delta X_{t+1} = (sS_i + aA_i + cC_i + fF_i + eE_i) + w\Delta X_t \tag{40}$$

$$X_{t+1} = X_t + \Delta X_{t+1} \tag{41}$$

Suppose there is no DF in the surrounding radius. In that case, the location of the DF is revised using Levy Flight [9] as described in Eq. 42. This improves the unpredictability, messiness, and search capacity of DF globally.

$$X_{t+1} = X_t + Levy(d)X_t \tag{42}$$

This methodology fuses the Dragonfly algorithm with the Firefly algorithm (FA). In FA, fireflies emit flashlights, and their mates attract them. The objective function is then evaluated based on the current position and speed. The position update procedure continues until the end condition is fulfilled.

Objective (attractiveness) and light intensity fluctuation are the major FA development issues. The luminosity of each firefly is affected by the type of the encrypted cost function or simply by the illumination of the fitness value or objective function. This is the problem of maximization, where the objective function is maximized to find the optimal solution. In this case, it is required to increase the light intensity emitted by these flies, and obviously, light intensity decreases with an increase in distance. Eq. 43 may be used to show the intensity of light at many distances:

$$I(r) = I_0 \exp\left(-\gamma \cdot r^2\right) \tag{43}$$

Where I is the provider of the luminance with distance r from the firefly and I_0 is the initial light intensity $r = 0$; and where γ is the factor of the absorption of light, which describes the

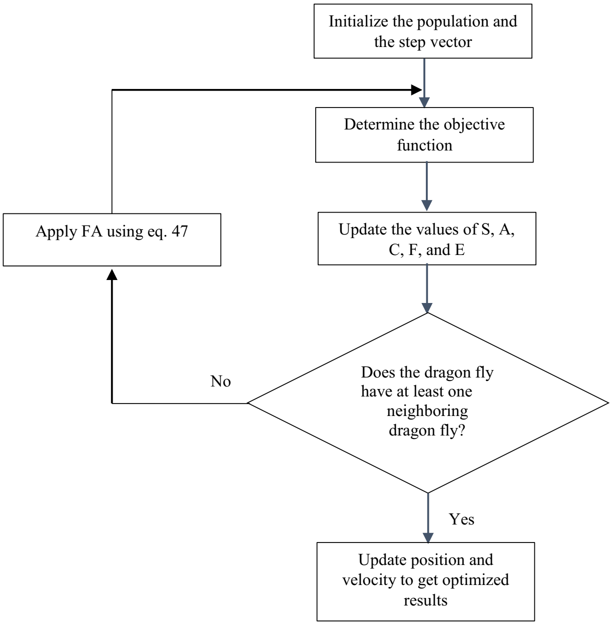
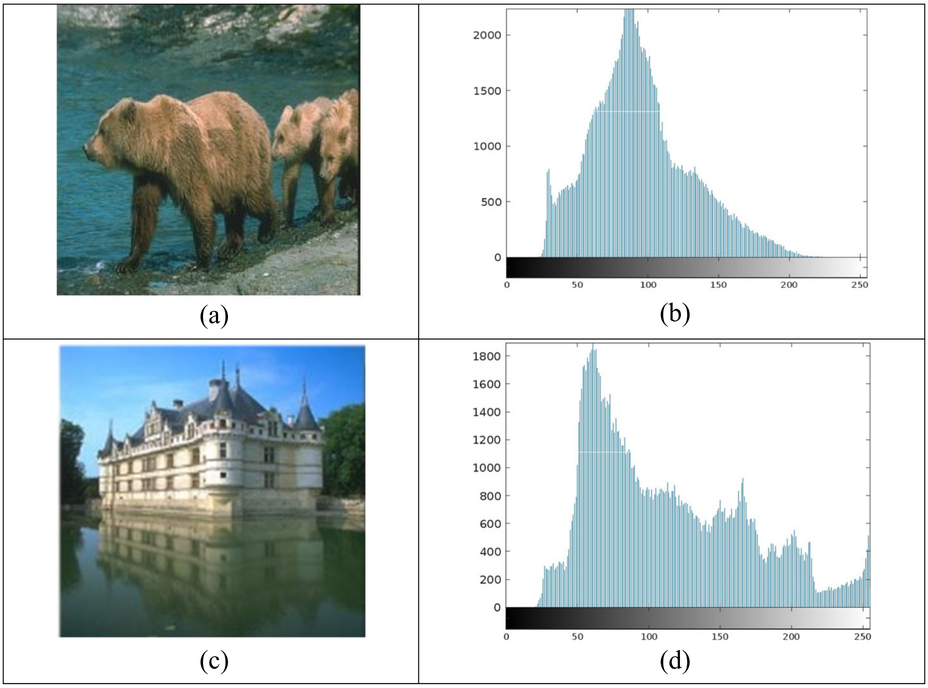

Initialize the population and the step vector
Determine the objective function
Update the values of S, A, C, F, and E
Apply FA using eq. 47
Does the dragon fly have at least one neighboring dragon fly?
No
Yes
Update position and velocity to get optimized results

Fig. 3 Flowchart of HDAFA



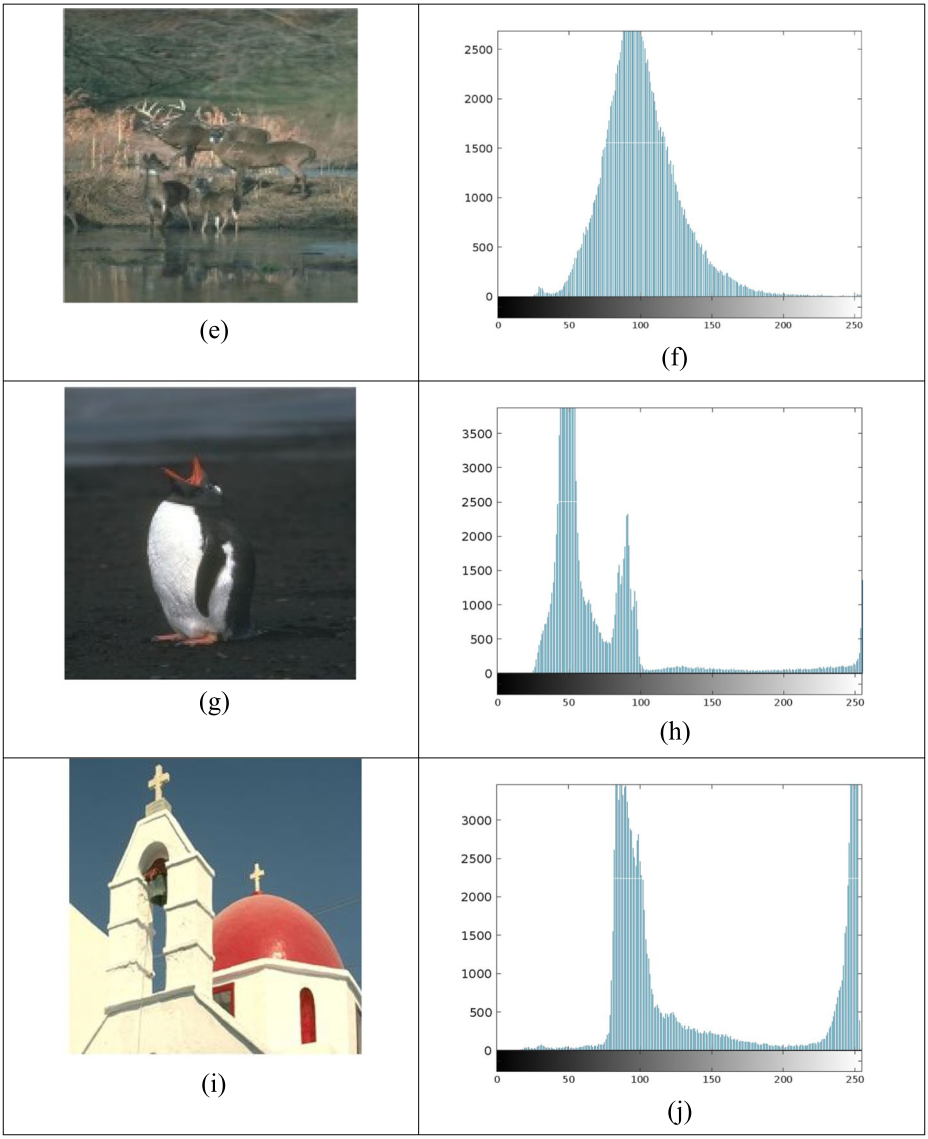
(a)

(b)

(c)

(d)

Fig. 4 **a** Bear, **b** Bear’s Histogram, **c** Building, **d** Building’s Histogram, **e** Deer, **f** Deer’s Histogram, **g** Penguin, **h** Penguin’s Histogram, **i** Dome and **j** Dome’s Histogram



(e)

(f)

(g)

(h)

(i)

(j)

Fig. 4 (continued)

change of the attractiveness and the speed of convergence, and general FA effects. γ varies typically from 0.1 to 10. As the appeal of a firefly is related to the intensity of light perceived by adjoining fireflies, the attraction may be shown as follows (at a Cartesian distance r from the firefly:

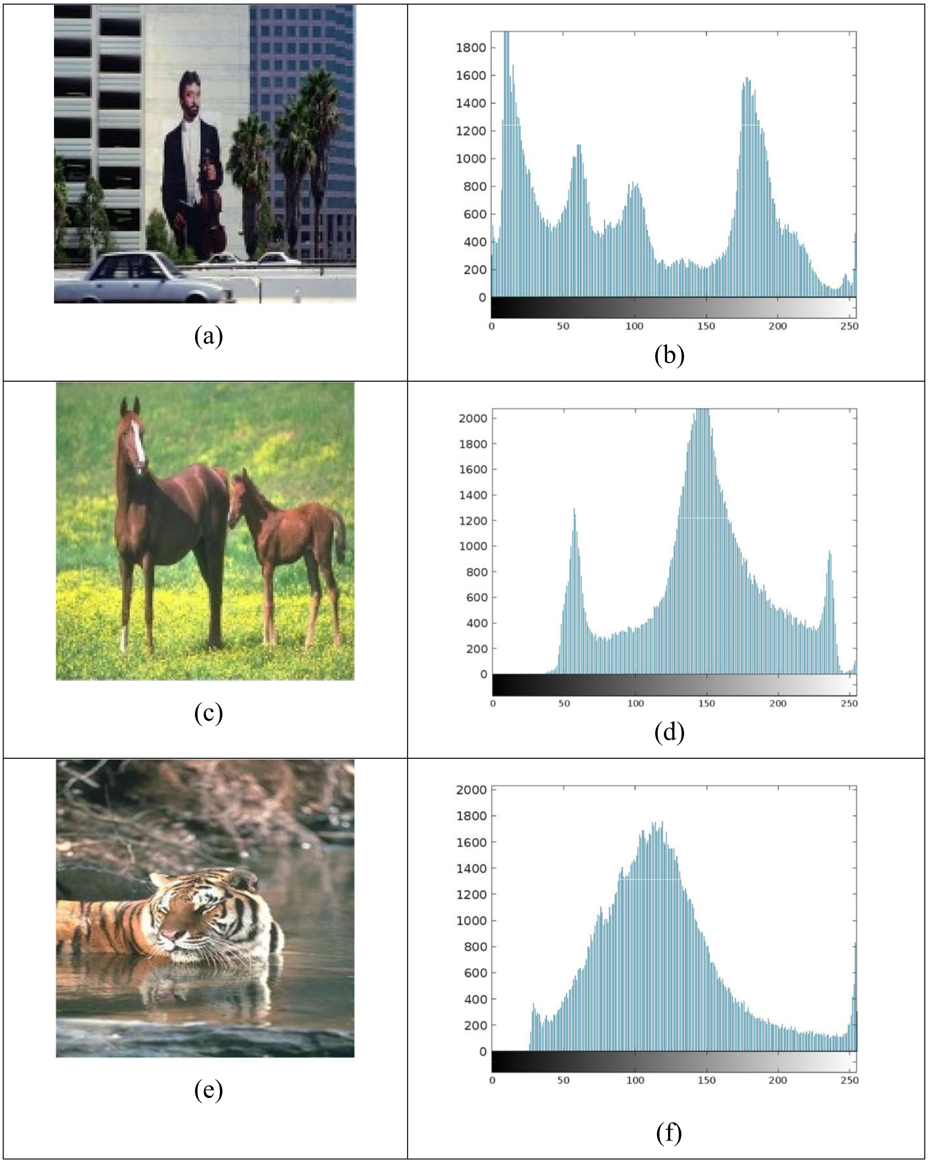
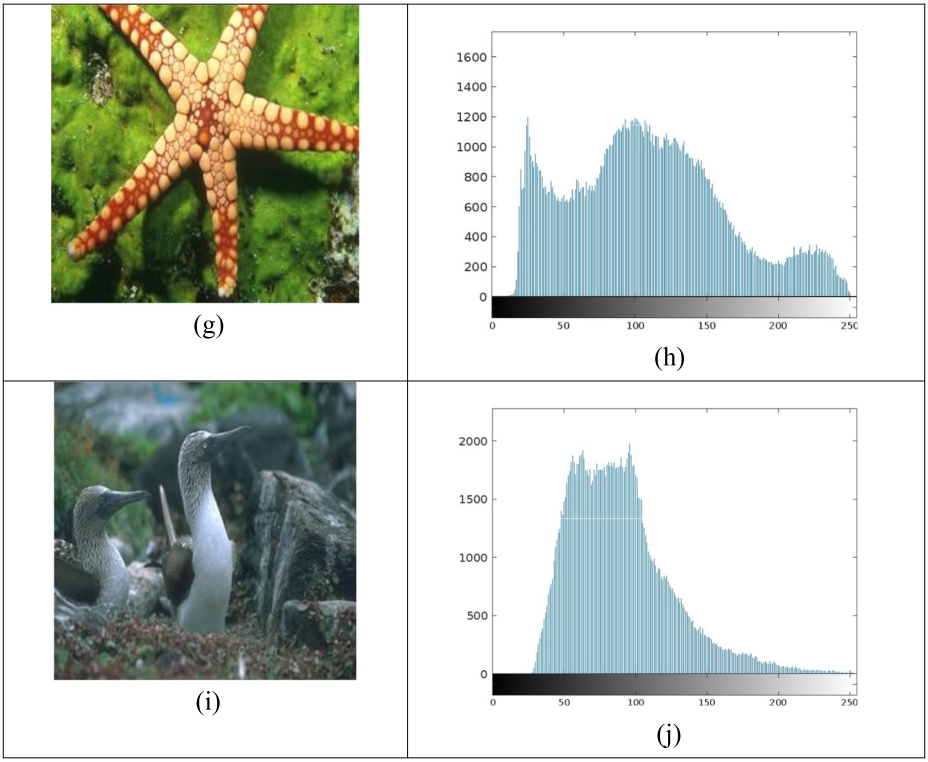

(a)
(b)
(c)
(d)
(e)
(f)

Fig. 5 **a** Gentleman, **b** Gentleman’s Histogram, **c** Horse, **d** Horse’s Histogram, **e** Tiger, **f** Tiger’s Histogram, **g** Sea star, **h** Sea star’s Histogram, **i** White bird and **j** White bird’s Histogram

$$\beta = \beta_0 \exp\left(-\gamma \,.\, r^2\right) \tag{44}$$

Range attraction $r = 0$ is represented by the function β_0. In the same way, the intensity of light I and attractiveness factor β are synonyms. Intensity is an objective measurement of the light emitted, whereas attractiveness is a comparative measure of the light perceived by fireflies and evaluated by other mates. Any two fireflies i and j can be separated by the Cartesian distance, defined as the distance between them at x_i and x_j, respectively, as in Eq. 47.



(g)

(h)

(i)

(j)

Fig. 5 (continued)

$$r_{ij} = \left\| x_i - x_j \right\| \tag{45}$$

Alternatively, firefly i can be attracted to another brighter firefly j as follows:

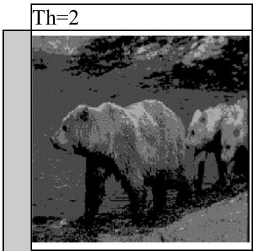
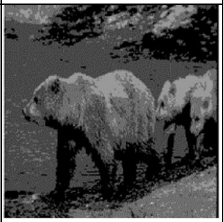
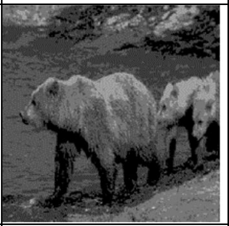
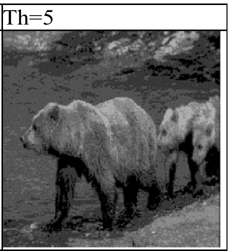
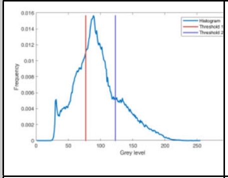
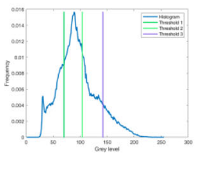
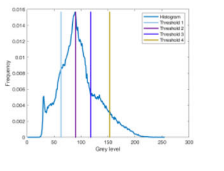
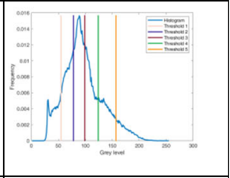
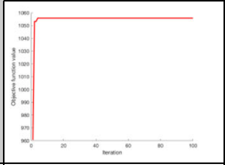
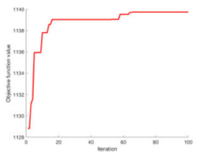
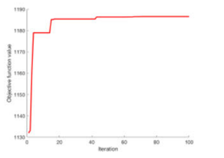
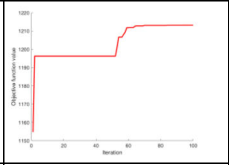
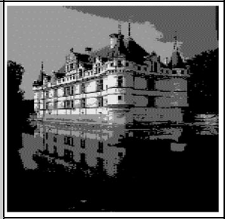
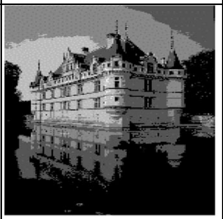
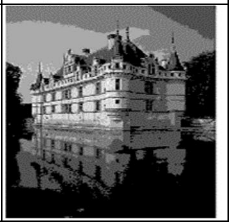
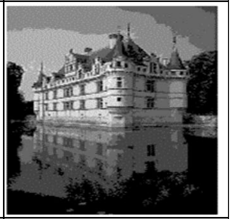
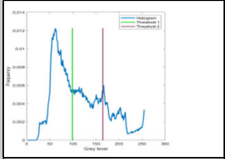
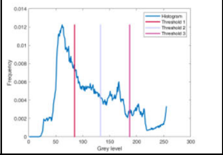
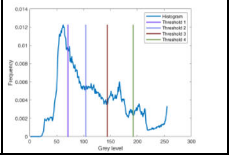
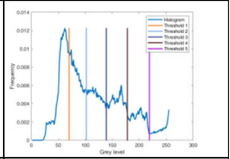
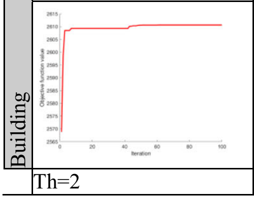
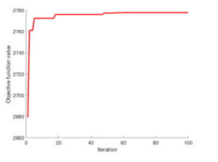
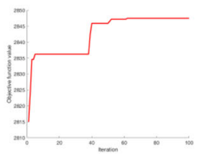
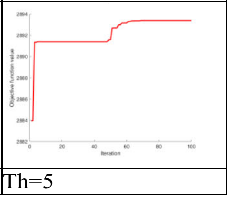
$$\Delta x_i = \beta_0 e^{-\gamma . r_{ij}^2} \left(x_j^t - x_i^t \right) + \propto (N_{rand} - 0.5) \tag{46}$$

From Eq. 46 t is the number of iterations in the loop. Besides, the first term in Eq. 46, which is attributable to the appeal, appears in the equation. $\propto (N_{rand} - 0.5)$ is the random sampling term, and the search space can be widened by using it. Randomization coefficient ($\propto$) and number vector (N_{rand}) from Gaussian distribution ([0, 1]). Random number generator in $\propto \in [0, 1]$. N_{rand}'s value is an evenly distributed random number generator. This is Firefly i's next move given by Eq. 47:

$$x_i^{t+1} = x_i^t + \Delta x_i = x_i^t + \beta_0 e^{-\gamma . r_{ij}^2} \left(x_j^t - x_i^t \right) + \propto (N_{rand} - 0.5) \tag{47}$$

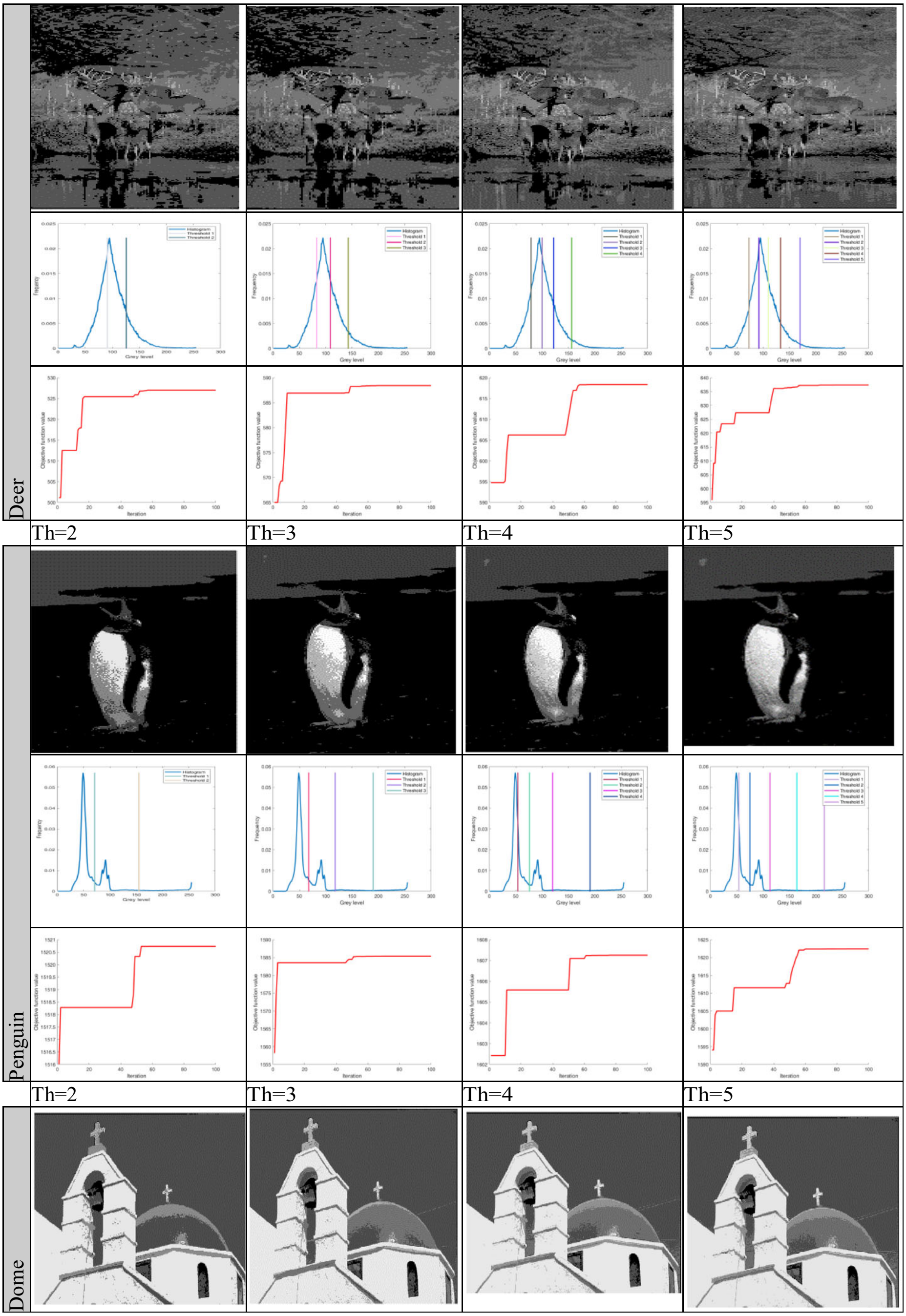
The initial phase is followed by a continuous series of performances, which continue until the optimization technique is complete. Any optimization algorithm must balance exploring and

Table 1 Results after applying the AOA using Otsu's method as an objective function over the selected benchmark images

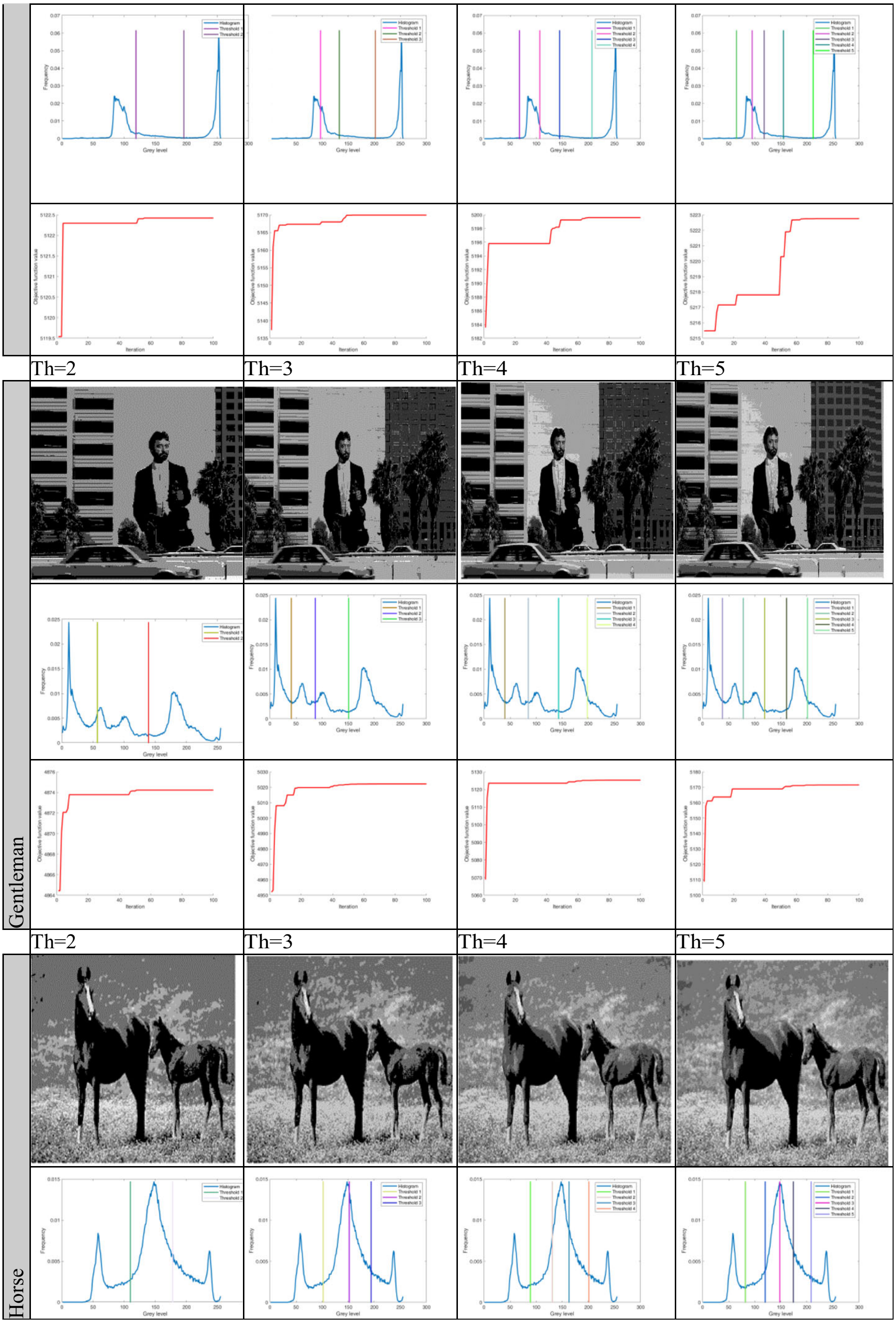
	Th=2	Th=3	Th=4	Th=5
Bear				
				
				
	Th=2	Th=3	Th=4	Th=5
Building				
				
				
	Th=2	Th=3	Th=4	Th=5

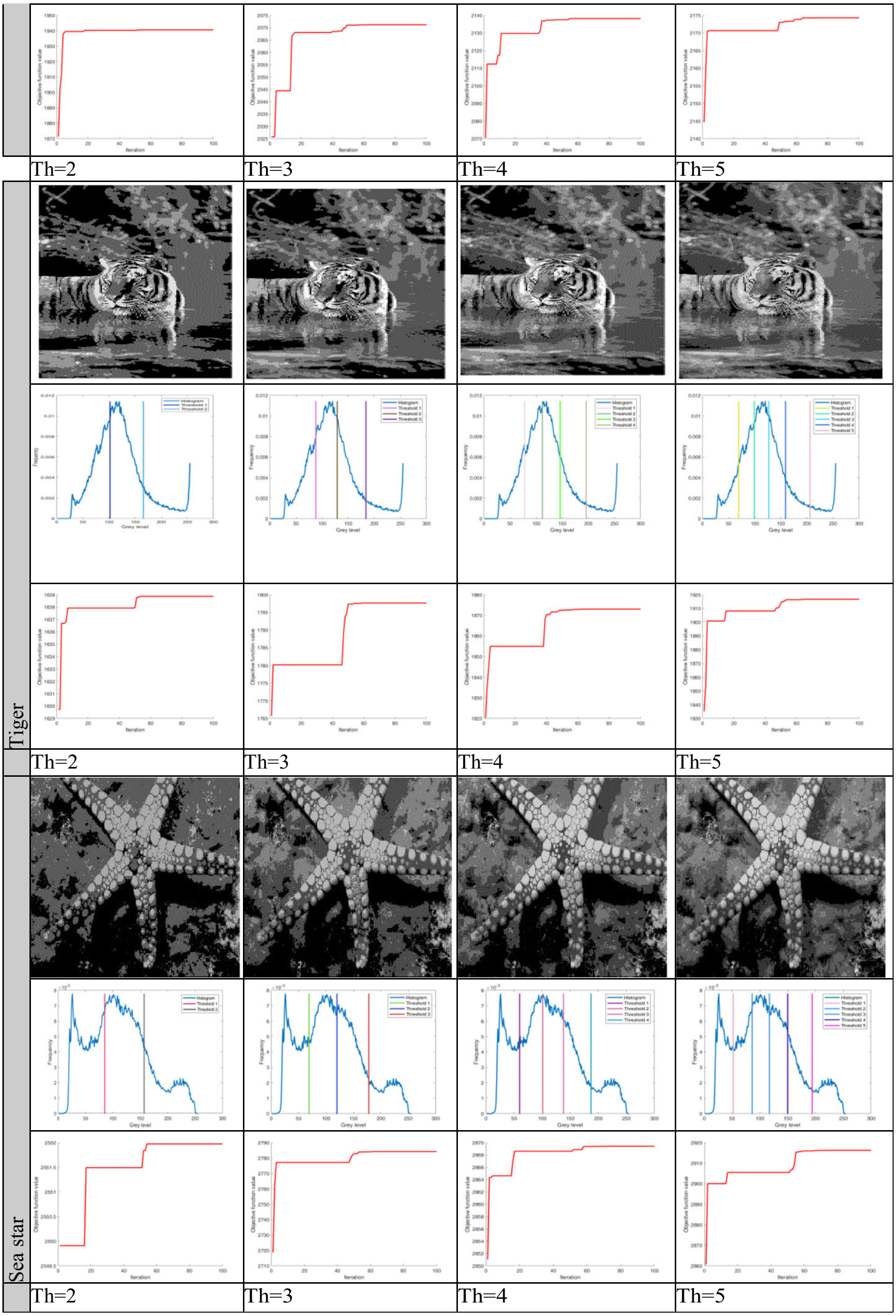
exploiting the search space to arrive at an optimal global solution. A worldwide search in the search space is called exploration. Exploitation or intensification is a local search based on the best available solution. Inefficient algorithms suffer from too much exploration and exploitation, which increases the chance of local optima [12].

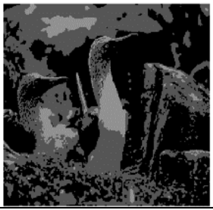
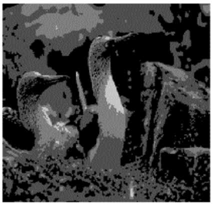
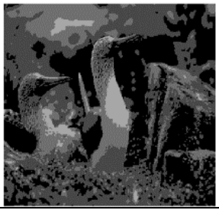
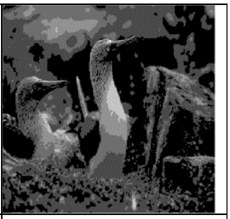
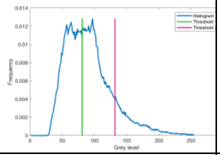
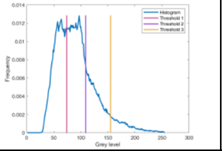
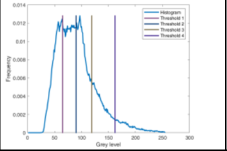
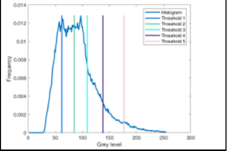
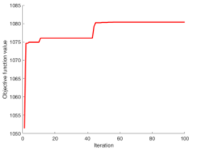
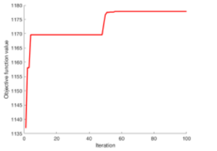
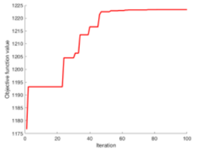
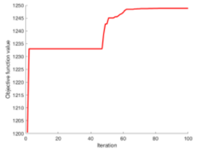
Dragonflies traverse the search space via Levy flying. It increases the number of possible answers and boosts the algorithm's exploration capabilities. These factors may be adjusted



with very few parameters, and adaptive tuning helps to balance the swarming algorithm's local and global search capabilities.

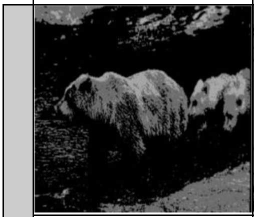
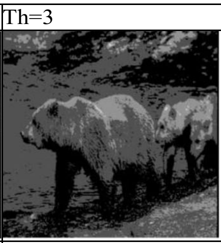
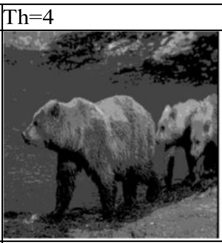
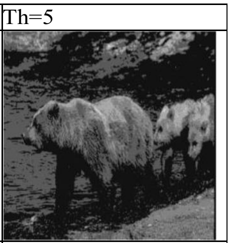
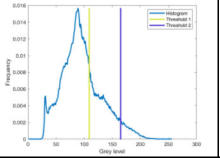
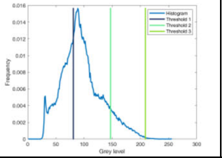
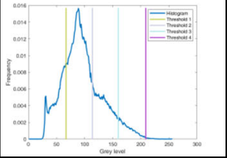
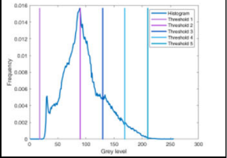
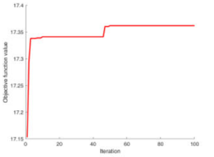
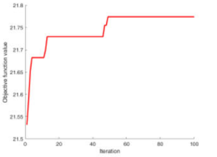
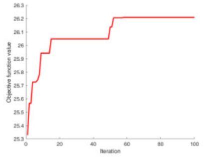
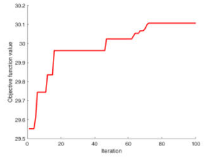

Th=2
Th=3
Th=4
Th=5
Th=2
Th=3
Th=4
Th=5
Gentleman
Horse
Histogram
Threshold 1
Threshold 2
Threshold 3
Threshold 4
Threshold 5
Frequency
Grey level
Objective function value
Iteration

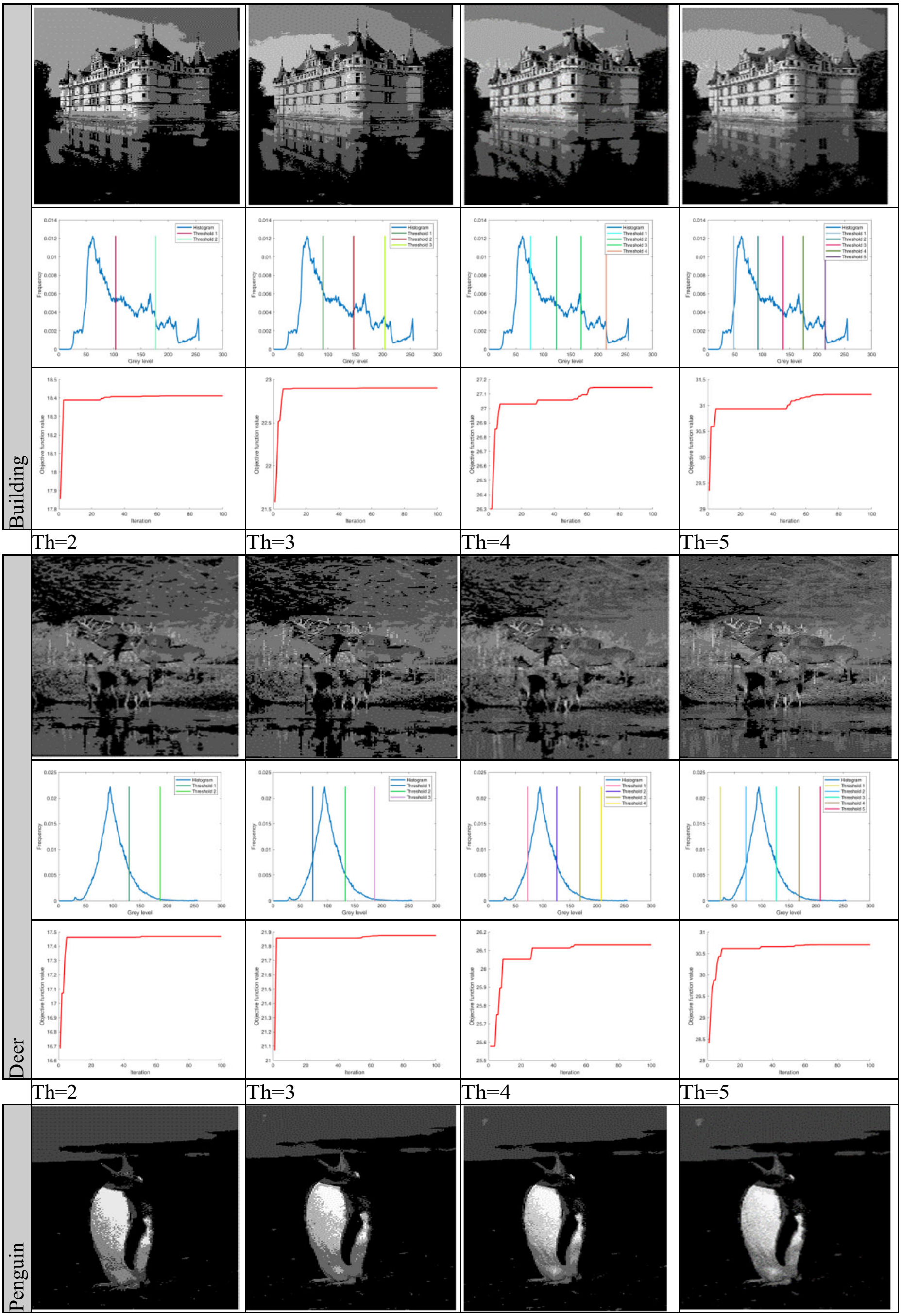

Th=2
Th=3
Th=4
Th=5
Tiger
Sea star
Objective function value
Iteration
Frequency
Grey level
Histogram
Threshold 1
Threshold 2
Threshold 3
Threshold 4
Threshold 5

White bird				
				
				

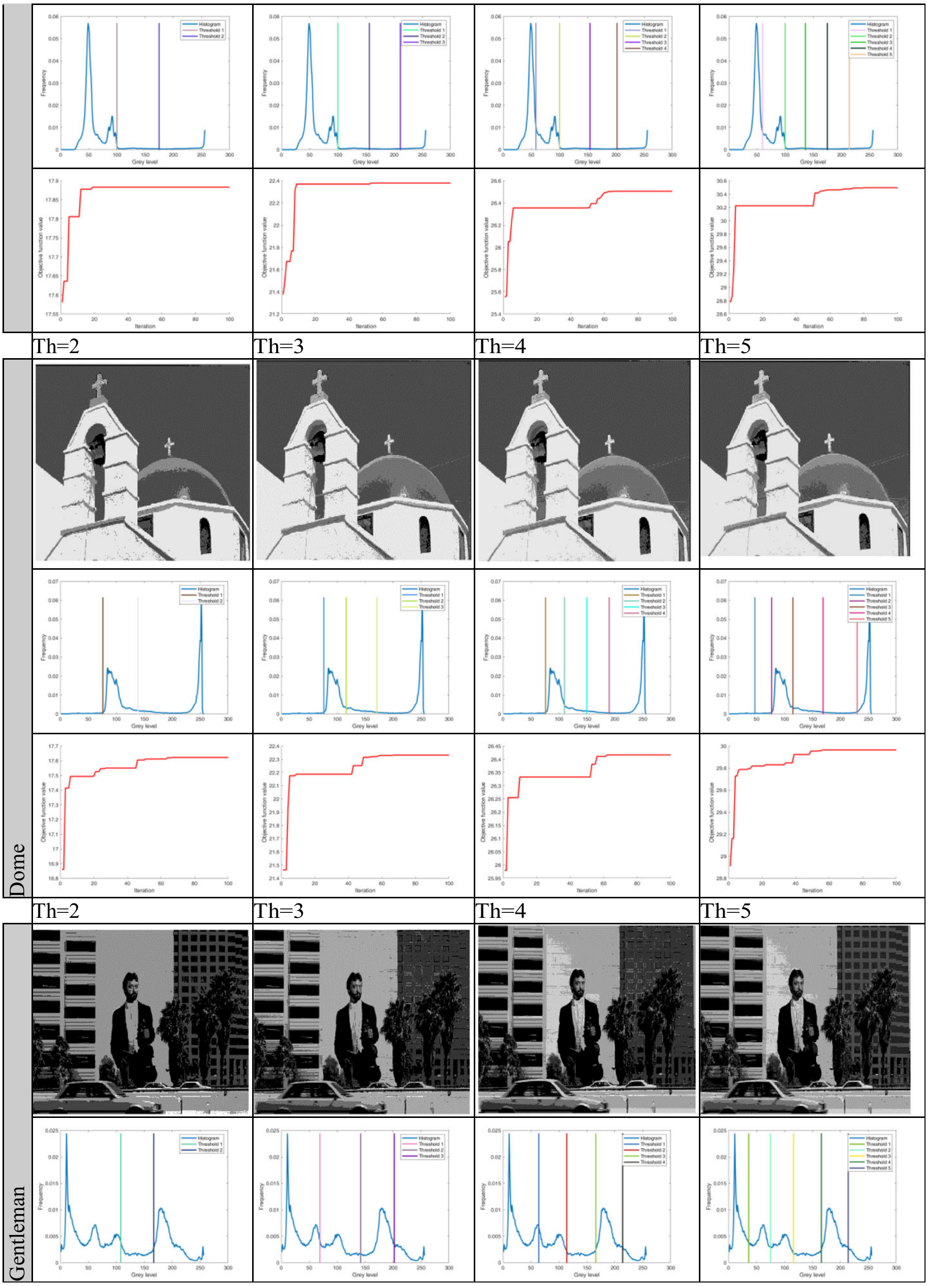
There are several advantages to using the conventional DA method; however, there are also some downsides, such as sluggish convergence. FA, too, looks to be a little restricted based on the convergence rate As a result, both ideas are scheduled to be blended in a way that answers the difficulties of optimization with greater convergence, and the DA algorithm impacts FA.

Table 2 Results after applying the AOA using Kapur's method as an objective function over the selected benchmark images

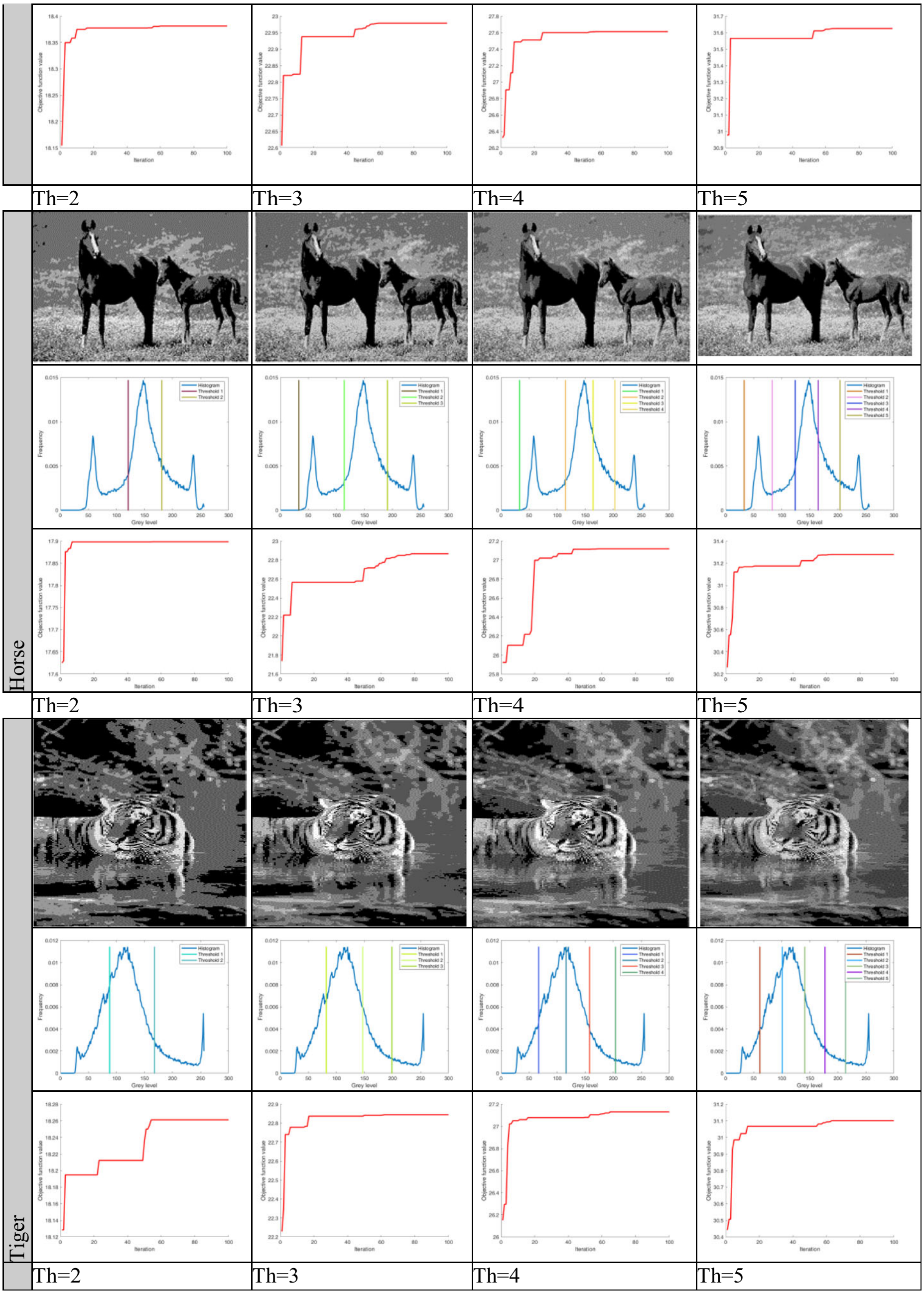
	Th=2	Th=3	Th=4	Th=5
White bird				
				
				
	Th=2	Th=3	Th=4	Th=5

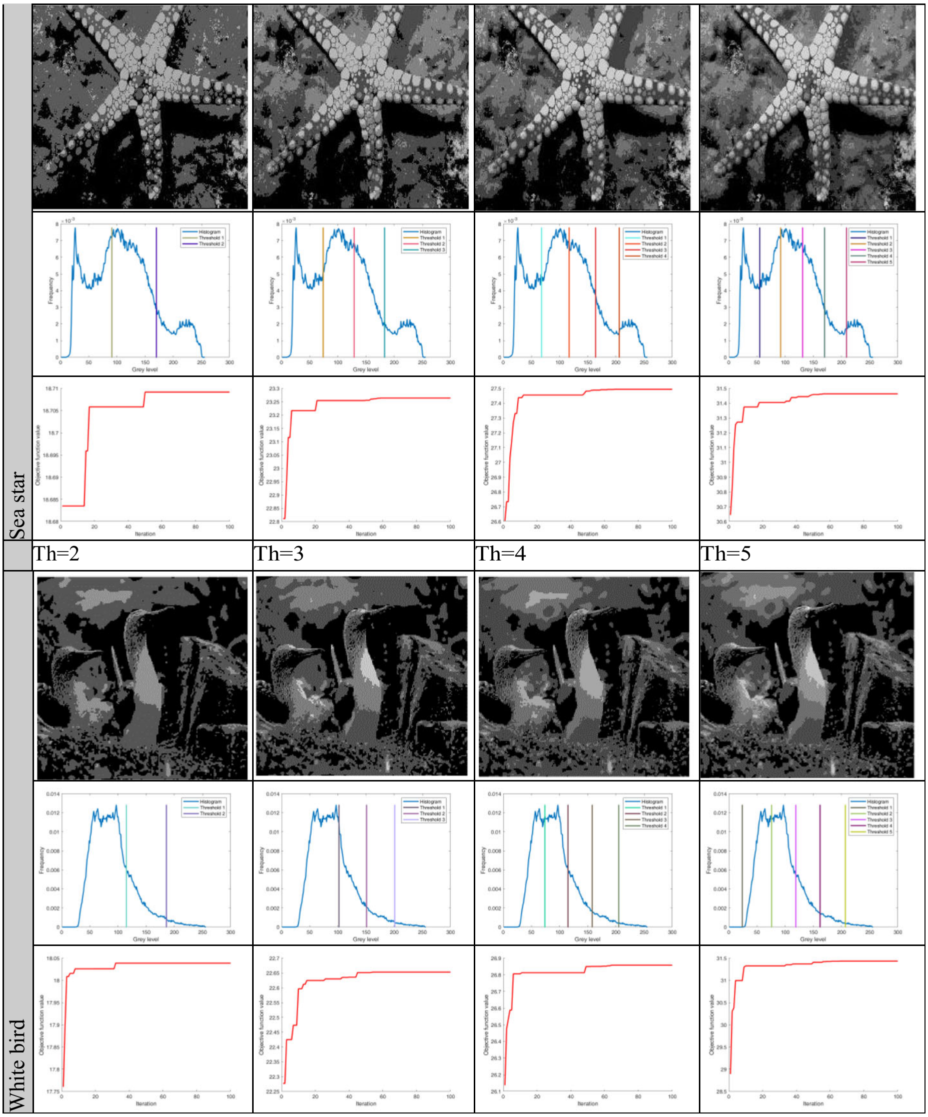


Eq. 40 will be used if the dragonfly does not include any neighborhoods. Algorithm HDAFA evaluates the FA position update given in Eq. 45. Due to the poor convergence features of DA



and FA, this change will assist overcome the current disadvantages of slow convergence characteristics (Fig. 3).


Th=2
Th=3
Th=4
Th=5
Horse
Tiger
Objective function value
Iteration
Frequency
Grey level
Histogram
Threshold 1
Threshold 2
Threshold 3
Threshold 4
Threshold 5

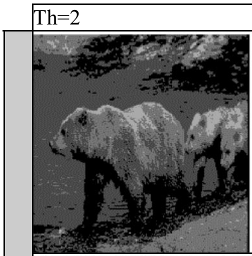
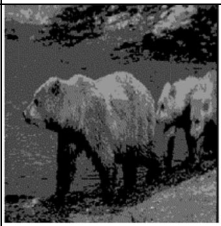
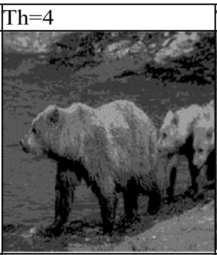
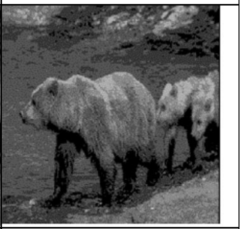
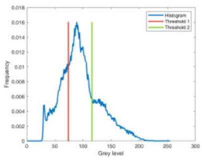
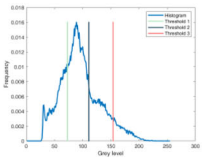
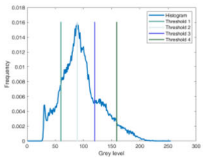
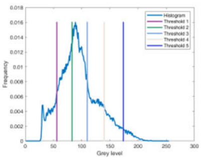
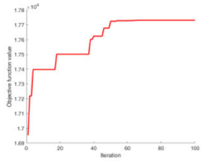
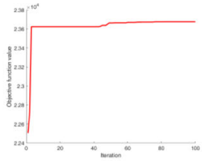
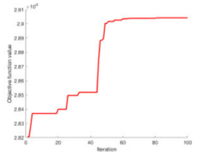
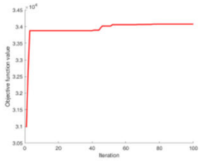
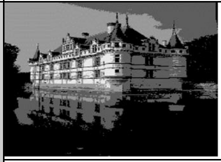
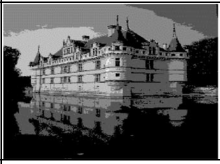
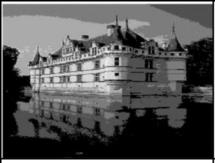
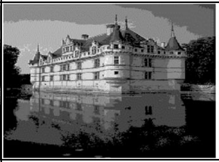
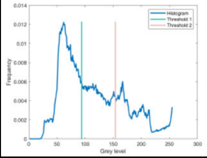
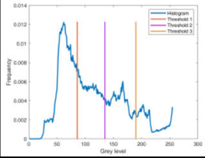
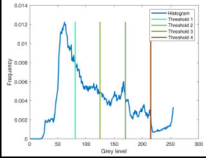
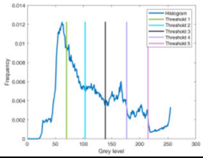
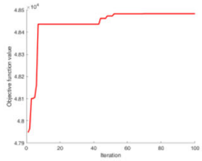
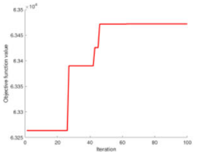
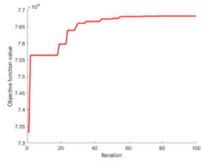
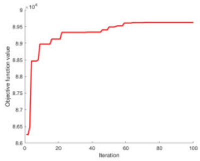
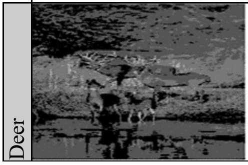
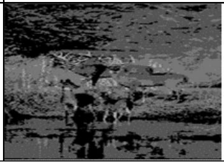
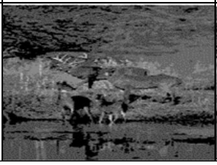
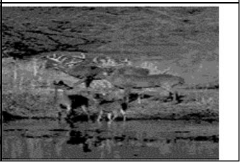


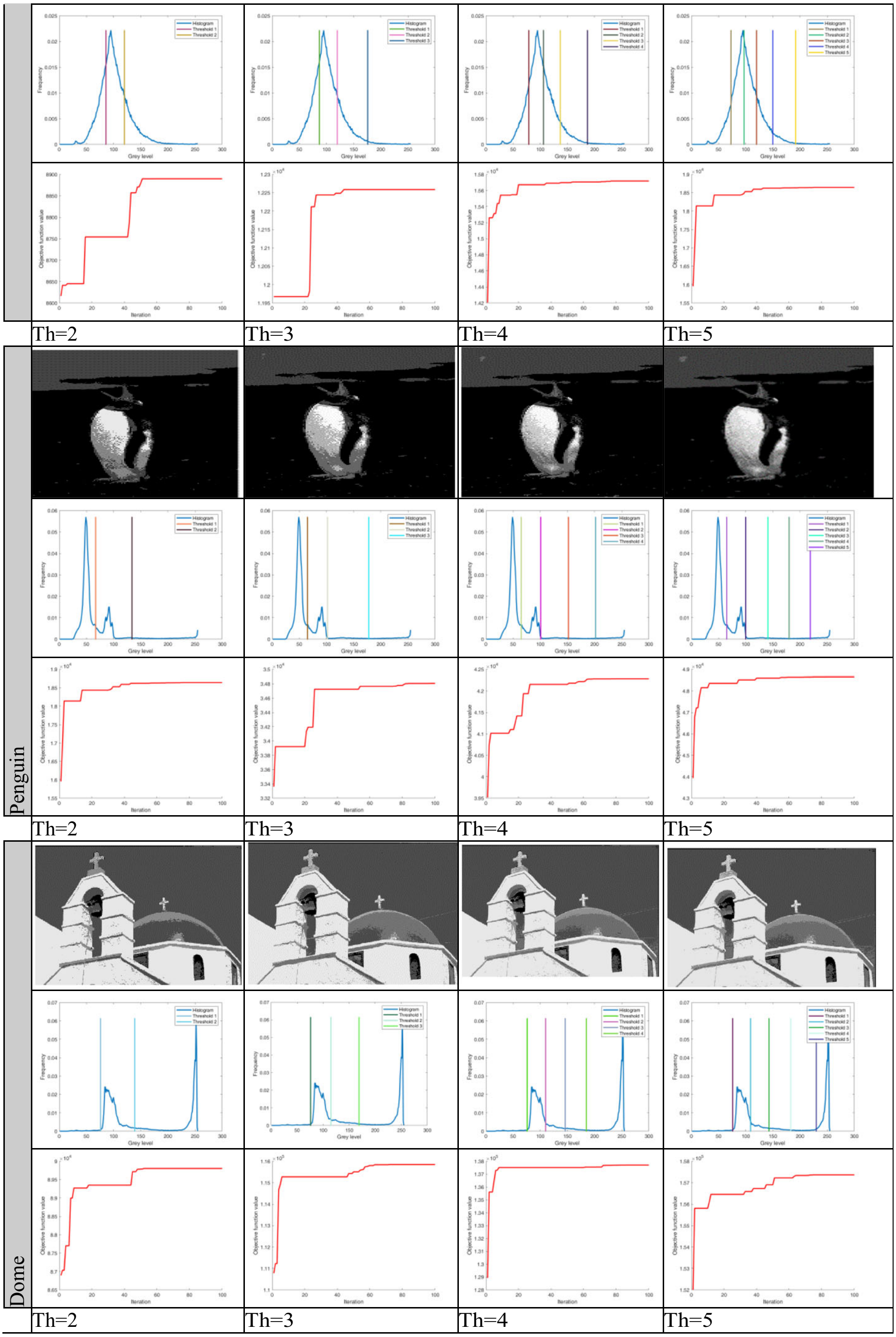
4 Experimental results

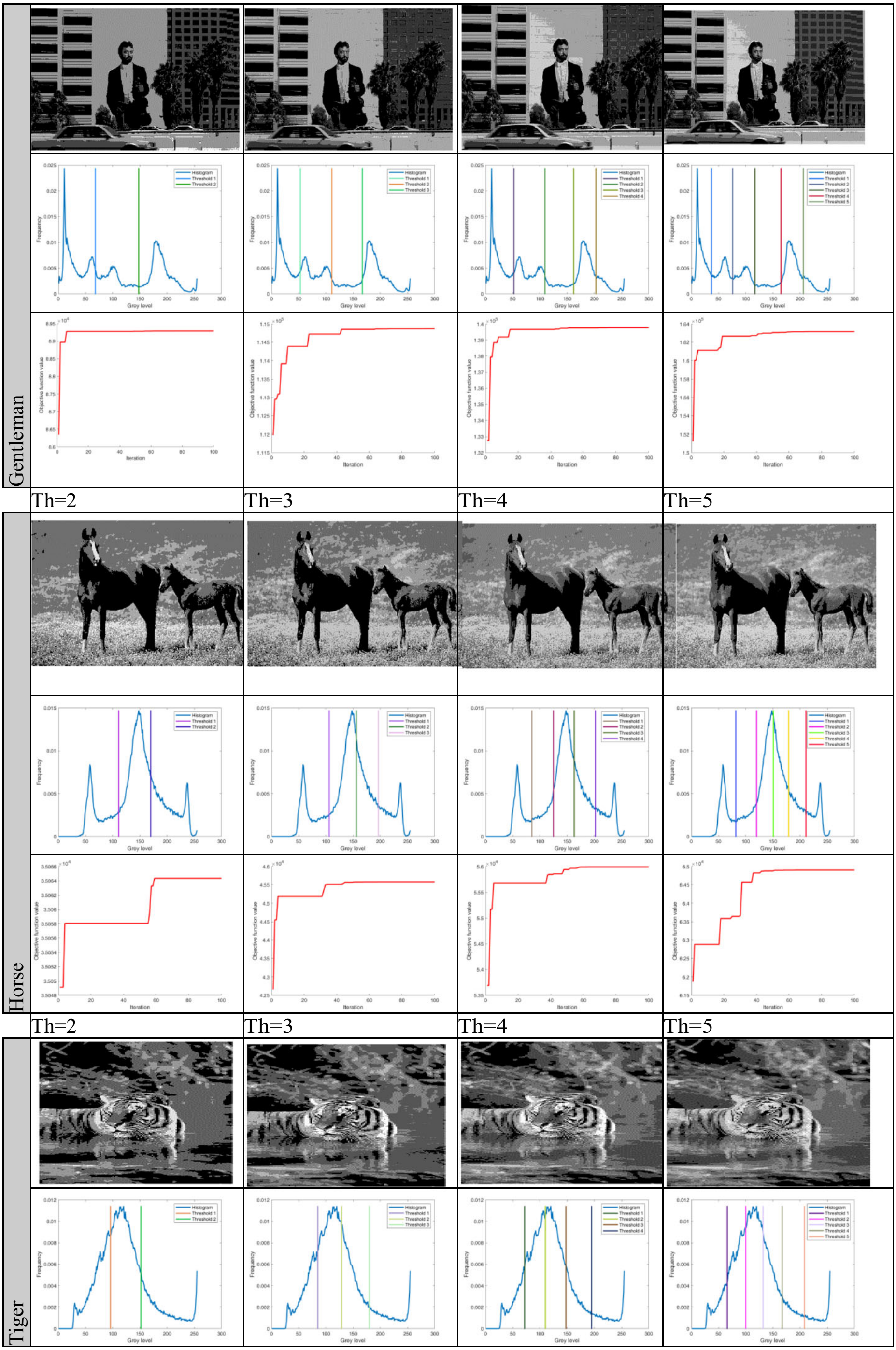
4.1 Experimental setup

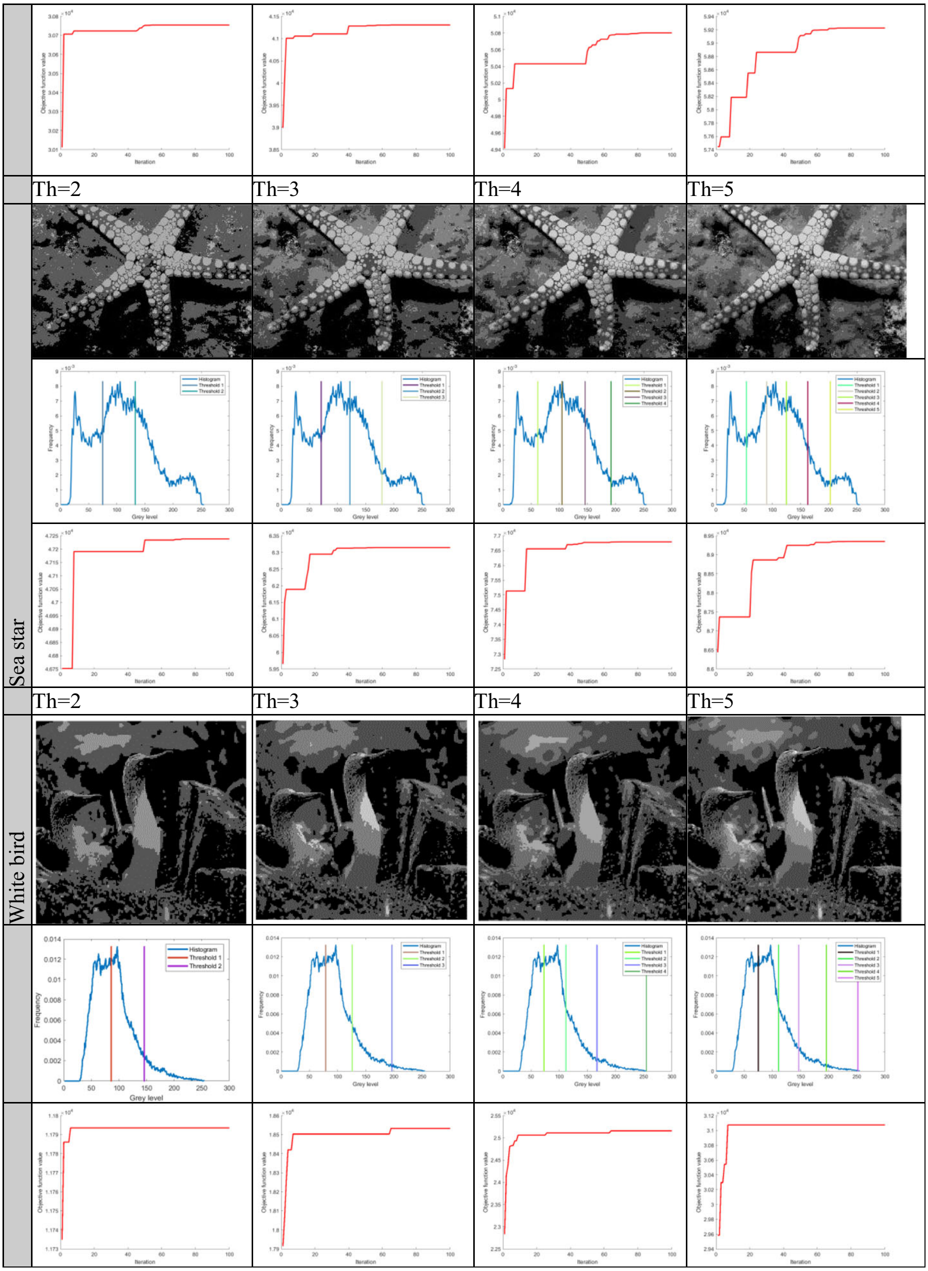
The widely used ten benchmark images (Bear, Building, Deer, Penguin, Dome, Gentleman, Horse, Tiger, Sea-star, and White bird) are selected to verify the correctness of the presented approach. All the selected images are shown in Figs. 4 and 5. It can be judged from these figures that all the images have distinct patterns and shapes of the histograms. A wide variety of histogram images are selected to verify the diversity of the presented image. The proposed algorithms are implemented in MATLAB 2021 using Intel(R) Core (TM) i7-3520 M CPU @

Table 3 Results after applying the AOA using modified Otsu's method as an objective function over the selected benchmark images

	Th=2	Th=3	Th=4	Th=5
Bear				
				
				
	Th=2	Th=3	Th=4	Th=5
Building				
				
				
	Th=2	Th=3	Th=4	Th=5
Deer				


Th=2
Th=3
Th=4
Th=5
Penguin
Dome
Histogram
Threshold 1
Threshold 2
Threshold 3
Threshold 4
Threshold 5
Frequency
Grey level
Objective function value
Iteration


Gentleman
Th=2
Th=3
Th=4
Th=5
Horse
Th=2
Th=3
Th=4
Th=5
Tiger
Histogram
Threshold 1
Threshold 2
Threshold 3
Threshold 4
Threshold 5
Frequency
Grey level
Objective function value
Iteration



2.90GHz 16 GB RAM machine. The results presented by optimization algorithms are unstable because the required parameters depend on the random numbers and are stochastic methods. To verify the consistency of the presented techniques, all the algorithms are tested 50 times for different threshold values $Th = 2, 3, 4$ *and* 5.

Peak signal-to-noise ratio (PSNR) indicates the amount of noise present in the resultant image as compared to the original image [38]. The PSNR between original or ground truth I_G and the segmented image I_{th} is calculated as mentioned in Eq. 48.

$$PSNR = 20\ log_{10} \frac{255}{RMSE}\ (dB) \tag{48}$$

$$\text{Where RMSE} = \sqrt{\frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(I_G - I_{th})}{M\ X\ N}} \tag{49}$$

Table 4 Various performance metrics for Otsu's method using AOA for different threshold values

Image	Th	Thresholds	PSNR	Mean	STD	Iteration
Bear	2	77 123	16.2435	1056	4.61E-13	13
	3	70 104 142	17.3682	1140	5.90E-03	21
	4	63 90 118 153	19.8349	1187	5.11E-02	25
	5	55 78 99 124 157	20.0268	1213	5.75E-02	28
Building	2	99 165	13.1969	2611	4.61E-13	10
	3	85 133 187	14.8753	2778	1.30E-03	17
	4	71 104 144 192	17.0078	2848	5.80E-02	24
	5	70 102 139 178 219	17.5042	2893	4.28E-02	26
Deer	2	91 126	14.3741	527	15.7E-02	15
	3	83 109 143	16.9119	588	6.92E-13	25
	4	79 100 122 156	18.4367	618	2.40E-03	11
	5	73 92 110 133 170	20.7031	637	5.84E-02	22
Penguin	2	71 155	14.5829	1521	2.31E-12	12
	3	68 118 190	15.2107	1585	2.22E-02	19
	4	54 76 120 191	16.5445	1607	1.93E-02	25
	5	54 75 113 164 216	16.8059	1622	1.60E-03	30
Dome	2	119 196	10.6292	5122	0.00E+00	10
	3	97 133 202	12.7253	5170	1.38E-12	15
	4	68 107 145 207	18.2683	5200	4.61E-13	33
	5	65 95 118 155 212	19.7144	5223	2.03E+00	25
Gentleman	2	57 139	15.5726	4874	1.65E-02	10
	3	41 87 151	17.7619	5022	2.16E-02	27
	4	40 85 143 198	19.2084	5125	1.58E-02	24
	5	38 78 119 161 201	21.31	5172	5.75E-02	20
Horse	2	110 178	14.5189	1941	1.40E-12	15
	3	102 152 194	16.1829	2071	3.07E-01	11
	4	89 131 163 201	18.232	2138	8.40E-03	44
	5	82 120 148 174 208	19.3001	2174	2.12E+00	12
Tiger	2	102 166	13.3475	1639	0.00E+00	11
	3	88 129 184	15.9458	1798	2.64E-02	20
	4	78 112 146 196	18.025	1873	1.76E-02	33
	5	69 99 127 159 206	20.147	1917	6.64E-01	27
Sea-star	2	85 157	14.8134	2552	6.92E-13	22
	3	69 120 178	17.3218	2784	7.66E-01	21
	4	60 101 138 187	19.117	2869	2.65E-02	45
	5	52 86 117 150 194	20.7072	2916	4.86E-02	27
White Bird	2	81 132	14.826	1080	9.68E-02	12
	3	74 109 155	16.3037	1178	1.15E-12	14
	4	65 90 119 162	18.3029	1223	3.38E+00	26
	5	62 85 109 138 177	19.2958	1249	2.86E+00	25

Where an image size is $M \times N$, a higher PSNR value is desirable, and it represents less amount of noise that has been added during the processing [22].

Another performance metric is Standard Deviation (STD); its ideal value should be zero. It denotes the level of variation or deviation from its mean value and is given by Eq. 50 [42]. A lower value of STD will denote higher stability, and a higher value signifies an unstable algorithm.

$$STD = \sqrt{\sum_{i=1}^{k} \frac{(\sigma_i - \mu)^2}{k}} \tag{50}$$

Table 5 Various performance metrics for the Kapur method using AOA for different threshold values

Image	*Th*	Thresholds	PSNR	Mean	STD	Iteration
Bear	2	109 165	11.4512	17.3619	0.0031	11
	3	81 147 209	15.3368	21.7743	0.175	17
	4	67 114 160 209	17.9764	26.2098	0.2707	22
	5	18 90 130 169 210	16.4334	30.1074	0.3048	24
Building	2	104 177	12.8308	18.4119	0.0373	26
	3	91 147 204	14.2961	22.9043	0.1782	31
	4	77 124 169 215	15.9943	27.1463	0.3513	28
	5	48 92 138 175 215	20.3525	31.2115	0.4789	42
Deer	2	130 187	9.0497	17.4704	0.0138	14
	3	73 133 187	17.4092	21.8765	0.2715	24
	4	73 126 169 208	17.9108	26.1298	0.3819	18
	5	24 71 127 169 208	18.833	30.703	0.4418	20
Penguin	2	100 175	12.8199	17.8834	0.047	15
	3	100 156 211	12.9247	22.3797	0.193	15
	4	58 100 154 202	15.8618	26.5063	0.6478	33
	5	60 100 136 175 214	15.8531	30.4966	1.6202	27
Dome	2	76 139	19.9068	17.6228	0.0426	9
	3	76 116 171	21.6724	22.3318	0.3586	17
	4	76 110 150 190	22.5036	26.4166	0.6253	21
	5	47 77 115 169 230	22.1242	29.9646	0.5968	35
Gentleman	2	108 167	14.5599	18.3815	0.0257	9
	3	69 142 202	16.9051	22.979	0.1538	30
	4	64 114 166 214	18.7849	27.614	0.3287	23
	5	36 75 116 166 214	21.092	31.6256	0.8569	19
Horse	2	121 181	14.301	17.8988	0.0879	20
	3	33 114 191	15.9477	22.8677	0.1236	16
	4	33 115 164 203	17.96	27.1198	0.1897	33
	5	33 83 124 165 204	20.5256	31.28	0.3647	20
Tiger	2	88 168	14.4075	18.2613	0.0129	11
	3	82 147 199	15.8568	22.8454	0.2987	20
	4	67 116 158 204	18.6805	27.1328	0.7964	28
	5	61 101 141 177 214	20.3121	31.1014	1.9871	29
Sea-star	2	91 170	14.3456	18.7092	0.0232	14
	3	74 129 183	17.0393	23.264	0.1455	15
	4	68 117 164 206	18.2243	27.4945	0.2232	30
	5	55 92 131 170 209	20.264	31.4637	0.4589	24
White Bird	2	115 186	11.086	18.0389	0.0049	11
	3	102 151 201	12.228	22.6531	0.11	19
	4	74 115 158 205	16.365	26.8582	0.2594	26
	5	24 76 119 162 207	19.2801	31.4348	0.3048	35

Where k represents *Max. iter* and in the presented approach *Maximum iteration* value is chosen as 150, σ_i represents a value of an objective function for i^{th} run, and μ is the mean value.

4.2 Performance evaluation and comparison

To demonstrate that the modified Otsu's is an interesting alternative for multi-level thresholding, the proposed algorithm is compared with other similar implementations. For each image, the PSNR, STD, and the mean of the objective function values are computed. Moreover, the entire test is performed using Otsu's, Kapur's, and modified Otsu's objective

Table 6 Various performance metrics for modified Otsu's method using AOA for different threshold values

Image	Th	Thresholds	PSNR	Mean	STD	Iteration
Bear	2	79 122	17.0545	1822	3.74E-02	8
	3	74 116 155	17.5968	1423	2.38E-02	12
	4	72 99 124 164	19.9789	3236	3.32E-04	18
	5	62 92 124 164 254	20.3422	3577	3.45E-03	20
Building	2	95 155	13.8643	8483	4.11E-02	42
	3	94 144 193	14.7953	3472	4.61E-13	45
	4	85 104 137 192	15.5873	6819	2.21E-13	32
	5	84 100 122 154 205	17.5043	9625	3.72E-11	58
Deer	2	68 132	15.8681	8890	2.45E-01	12
	3	66 100 173	15.599	2259	3.12E-03	22
	4	62 111 148 224	18.2051	5719	3.79E-14	25
	5	69 106 138 186 252	20.3761	8649	2.28E-01	18
Penguin	2	61 141	15.2029	5796	2.68E-03	18
	3	67 104 171	15.4113	4807	3.43E-03	11
	4	65 102 152 201	15.5829	2360	3.83E-05	41
	5	65 100 142 191 232	15.6302	8654	2.28E-02	24
Dome	2	76 139	20.1046	9807	2.98E-02	8
	3	76 116 171	21.6779	5850	3.88E-02	18
	4	76 110 160 190	22.5583	7710	3.44E-06	9
	5	77 107 145 169 230	23.1847	7440	3.09E-04	45
Gentleman	2	68 147	16.2232	9286	2.69E-15	7
	3	56 114 166	18.8644	5150	3.04E-03	32
	4	64 114 166 214	19.6141	9770	3.21E-03	21
	5	36 75 116 166 214	21.4032	3170	3.43E-03	17
Horse	2	121 168	15.1626	5064	3.65E-04	25
	3	104 154 191	15.9618	5577	9.59E-14	20
	4	82 125 164 203	18.2909	5990	3.43E-03	21
	5	83 123 150 175 214	19.4964	4907	3.65E-04	27
Tiger	2	98 152	14.5195	0755	2.38E-02	10
	3	82 127 179	16.5166	1308	2.89E-04	20
	4	67 116 148 194	19.1166	2802	3.42E-02	22
	5	61 101 131 157 214	20.53	9570	2.35E-13	31
Sea-star	2	71 137	16.8248	8434	2.32E-03	6
	3	64 119 183	17.3927	4724	3.52E-02	9
	4	68 102 144 196	18.8662	8636	2.16E-02	15
	5	55 92 131 160 209	20.8329	1477	3.01E-07	20
White Bird	2	85 156	15.2077	9171	4.02E-04	9
	3	72 125 201	16.1141	6055	0.00356	8
	4	74 115 158 205	18.6366	2262	2.36E-02	10
	5	74 119 142 192 247	19.9358	7862	7.66E-01	11

functions. To analyze the results of the proposed approach, different comparisons are conducted. The first experiment is conducted to compare the modified Otsu's, Otsu's function, and Kapur as criteria using the AOA technique. The second experiment is performed using HDAFA as an optimization method using modified Otsu's, Otsu's, and Kapur's as objective functions. Several performance parameters like PSNR, mean, Std. Dev., and 35 iterations are selected to verify its performance and computational effort. All algorithms run 35 times over each benchmark image to ensure the correct results.

Tables 1, 2, and 3 show the results after applying the AOA using Otsu's, Kapur's, and modified Otsu's method as an objective function over the selected 10 benchmark images. The results present the segmented images considering four different threshold points Th = 2, 3, 4, and 5. Tables 1, 2, and 3, provide the segmented images, histogram along with threshold

Table 7 Various performance metrics for Otsu's method using HDAFA for different threshold values

Image	Th	Thresholds	PSNR	Mean	STD	Iteration
Bear	2	77 123	16.2435	1493	2.65E-02	15
	3	70 104 142	18.068	1588	4.86E-02	25
	4	63 90 118 153	20.0129	1574	2.31E-12	32
	5	54 77 98 124 157	22.4513	1633	2.22E-02	15
Building	2	99 165	12.412	2631	1.93E-02	32
	3	85 133 187	13.8643	2836	1.60E-03	25
	4	71 104 144 192	14.9609	2856	9.68E-02	25
	5	69 101 139 178 219	15.3049	3089	1.15E-12	13
Deer	2	91 126	12.5328	339	3.38E+00	12
	3	83 109 143	15.8681	422	2.86E+00	36
	4	78 99 121 155	17.1915	545	1.65E-02	24
	5	71 91 109 132 168	18.2071	643	2.16E-02	13
Penguin	2	71 155	12.7269	916	1.58E-02	15
	3	68 118 190	15.2029	1211	4.61E-13	45
	4	54 76 120 191	15.4087	1466	5.90E-03	12
	5	53 74 112 161 213	15.5805	1676	5.11E-02	34
Dome	2	119 196	10.7073	3163	5.75E-02	23
	3	97 133 202	12.7253	4034	4.61E-13	45
	4	68 107 145 207	18.2683	4764	1.30E-03	67
	5	65 95 118 155 212	19.7144	5423	5.80E-02	22
Gentleman	2	57 139	16.4194	3139	4.28E-02	12
	3	41 87 151	16.2232	4007	15.7E-02	45
	4	40 84 143 198	18.8644	4828	6.92E-13	56
	5	38 77 118 161 201	19.6457	5614	2.40E-03	32
Horse	2	57 139	14.3376	1333	5.84E-02	19
	3	41 87 151	15.1325	1666	0.00E+00	16
	4	40 85 143 198	17.7745	2032	1.38E-12	34
	5	38 77 118 161 201	18.3161	2060	4.61E-13	24
Tiger	2	102 166	12.9429	1078	2.03E+00	12
	3	88 129 184	14.5195	1169	1.40E-12	15
	4	77 111 145 196	16.5166	1258	3.07E-01	22
	5	67 97 125 157 204	18.9816	1496	8.40E-03	18
Sea-star	2	85 157	14.1464	1605	2.12E+00	6
	3	69 120 178	16.8248	1703	0.00E+00	5
	4	60 101 138 187	17.2935	1768	2.64E-02	9
	5	52 86 117 150 194	18.9737	1864	1.76E-02	16
White Bird	2	81 132	14.0106	750	6.64E-01	63
	3	74 108 154	15.2077	908	6.92E-13	37
	4	65 90 119 162	16.1153	1062	2.65E-02	39
	5	62 84 108 136 175	16.8453	1137	4.86E-02	28

values, and convergence graphs showing the number of iterations required to achieve stabilization. The segmented images provide evidence that the outcome is better with the 4 and the 5; however, if the segmentation task does not require to be extremely accurate, then it is possible to select the 3. It is depicted from the results that most of the time, modified Otsu's as objective function stabilizes or achieves convergence under 50 iterations only. Tables 1, 2, and 3 gather segmented images for qualitative analysis and from each technique to graphically contrast them. It is feasible to note that the algorithm's overall performance using modified Otsu's as an objective function for thresholding is more accurate to a human expert's assessment of the findings in Table 6.

Tables 4, 5, 6, 7, 8, and 9 show threshold values, PSNR, Mean, STD, and iteration required to achieve convergence using Otsu's, Kapur's entropy and modified Otsu's method for AOA

Table 8 Various performance metrics for the Kapur method using HDAFA for different threshold values

Image	Th	Thresholds	PSNR	Mean	STD	Iteration
Bear	2	109 165	11.3464	17.3636	0.0031	12
	3	81 147 209	15.3383	22.7765	0.1750	19
	4	67 114 160 209	17.9782	26.2124	0.2707	20
	5	18 90 130 169 210	16.4350	32.1104	0.3048	25
Building	2	104 177	12.8321	18.4137	0.0373	27
	3	91 147 204	14.2975	22.9066	0.1782	31
	4	77 124 169 215	15.9959	26.1490	0.3513	28
	5	48 92 138 175 215	20.3545	31.2146	0.4789	45
Deer	2	130 187	9.0506	18.4721	0.0138	14
	3	73 133 187	17.4109	21.9787	0.2715	24
	4	73 126 169 208	17.9126	26.9324	0.3819	18
	5	24 71 127 169 208	18.8349	31.7061	0.4418	20
Penguin	2	100 175	12.8212	17.8852	0.0470	15
	3	100 156 211	12.9260	22.3819	0.1930	17
	4	58 100 154 202	15.8634	26.5090	0.6479	33
	5	60 100 136 175 214	15.8547	30.4996	1.6204	27
Dome	2	76 139	19.9088	17.6246	0.0426	19
	3	76 116 171	21.6746	22.3340	0.3586	17
	4	76 110 150 190	22.5059	26.4192	0.6254	21
	5	47 77 115 169 230	22.1264	29.9676	0.5969	35
Gentleman	2	108 167	14.5614	18.3833	0.0257	11
	3	69 142 202	16.9068	22.9813	0.1538	30
	4	64 114 166 214	18.7868	27.6168	0.3287	23
	5	36 75 116 166 214	21.0941	31.6288	0.8570	21
Horse	2	121 181	14.3024	17.9006	0.0879	20
	3	33 114 191	15.9493	22.8700	0.1236	16
	4	33 115 164 203	17.9618	27.1225	0.1897	33
	5	33 83 124 165 204	20.5277	31.2831	0.3647	23
Tiger	2	88 168	14.4089	18.2631	0.0129	11
	3	82 147 199	15.8584	22.8477	0.2987	20
	4	67 116 158 204	18.6824	27.1355	0.7965	28
	5	61 101 141 177 214	20.3141	31.1045	1.9873	29
Sea-star	2	91 170	14.3470	18.7111	0.0232	14
	3	74 129 183	17.0410	23.2663	0.1455	15
	4	68 117 164 206	18.2261	27.4972	0.2232	34
	5	55 92 131 170 209	20.2660	31.4668	0.4589	24
White Bird	2	115 186	11.0871	18.0407	0.0049	10
	3	102 151 201	12.2292	22.6554	0.1100	19
	4	74 115 158 205	16.3666	26.8609	0.2594	26
	5	24 76 119 162 207	19.2820	31.4379	0.3048	33

and HDAFA algorithm for 10 test images. These tables show significant conclusions concerning the PSNR and mean value for modified Otsu's method over other objective functions. It can be noticed that many approaches find the same threshold values, especially during segmentation with a small number of thresholds. Image thresholding's goal is to provide high-quality pictures that have a specific number of thresholds. The PSNR is a quality statistic that is frequently used to evaluate the quality of a processed signal relative to the original, as mentioned in the previous subsection. To assess multi-dimensional signals, in this case, images, PSNR has been expanded. In Tables 4, 5, 6, 7, 8, and 9, a higher mean PSNR value denotes better image segmentation when considering the threshold values of a specific algorithm.

Table 9 Various performance metrics for modified Otsu's method using HDAFA for different threshold values

Image	Th	Thresholds	PSNR	Mean	STD	Iteration
Bear	2	77 123	16.9108	1712	2.70E-02	10
	3	70 104 142	18.0863	1621	5.83E-02	15
	4	63 90 118 153	21.5983	2624	2.77E-12	20
	5	54 77 98 124 157	23.0286	3234	2.66E-02	10
Building	2	99 165	13.1969	2197	2.32E-02	22
	3	85 133 187	14.8753	8483	1.92E-03	25
	4	71 104 144 192	17.0078	3564	1.16E-01	12
	5	69 101 139 178 219	17.6531	3564	1.38E-12	8
Deer	2	91 126	14.3741	5515	4.06E+00	12
	3	83 109 143	16.9119	8890	3.43E+00	22
	4	78 99 121 155	18.7545	2262	1.98E-02	14
	5	71 91 109 132 168	21.2919	5688	2.59E-02	8
Penguin	2	71 155	14.5829	6493	1.90E-02	7
	3	68 118 190	15.2107	5796	5.53E-13	11
	4	54 76 120 191	16.5445	4806	7.08E-03	32
	5	53 74 112 161 213	17.0817	2354	6.13E-02	23
Dome	2	119 196	10.6292	8827	6.90E-02	4
	3	97 133 202	20.1046	9807	5.53E-13	15
	4	68 107 145 207	21.6779	5850	1.56E-03	9
	5	65 95 118 155 212	22.413	7690	6.96E-02	22
Gentleman	2	57 139	15.5726	9512	5.14E-02	7
	3	41 87 151	17.7619	9286	1.88E-01	26
	4	40 84 143 198	19.2022	5150	8.30E-13	16
	5	38 77 118 161 201	21.2983	9710	2.88E-03	17
Horse	2	57 139	15.5726	3177	7.01E-02	5
	3	41 87 151	17.7619	5063	0.00E+00	16
	4	40 85 143 198	19.2084	5522	1.66E-12	17
	5	38 77 118 161 201	21.2983	5901	5.53E-13	24
Tiger	2	102 166	13.3475	9665	2.44E+00	10
	3	88 129 184	15.9458	2555	1.68E-12	15
	4	77 111 145 196	18.1711	1895	3.68E-01	22
	5	67 97 125 157 204	20.4013	8794	1.01E-02	18
Sea-star	2	85 157	14.8134	7649	2.54E+00	6
	3	69 120 178	17.3218	4434	0.00E+00	5
	4	60 101 138 187	19.117	6418	3.17E-02	9
	5	52 86 117 150 194	20.7072	1868	2.11E-02	16
White Bird	2	81 132	14.826	8204	7.97E-01	9
	3	74 108 154	16.3092	9171	8.30E-13	8
	4	65 90 119 162	18.3029	6052	3.18E-02	10
	5	62 84 108 136 175	19.3053	2235	5.83E-02	11

In contrast to the mean value, a lower STD value is preferred because it shows less range in the outcomes produced by each approach. The STD value often rises when the number of threshold values has been increased. Most sample images demonstrate that modified Otsu's method outperforms Otsu's and Kapur's entropy functions. During experimentation, it is found that the result for two-level thresholding is always better for all source images in AOA and HDAFA. It is noticeably observed for the entire dataset the modified Otsu's method achieved the best metric values for maximum cases.

The number of iterations in Table 10 provides evidence that the AOA and HDAFA using modified Otsu's stabilizes the system in less time. Modified Otsu's using both AOA and HDAFA optimization requires a low number of iterations depending on the dimension of the

Table 10 Comparison of the number of iterations required to converge

Image	Th	AOA (Modified Otsu's)	AOA (Otsu's)	AOA (Kapur)	HDFA (Modified Otsu's)	HDFA (Otsu's)	HDFA (Kapur)
Bear	2	**8**	13	15	10	15	13
	3	**12**	21	23	15	25	20
	4	**18**	25	27	20	32	26
	5	20	28	30	**10**	15	13
Building	2	42	**10**	12	22	32	27
	3	**17**	**17**	19	25	25	25
	4	32	24	26	**12**	25	19
	5	**8**	26	28	**8**	13	11
Deer	2	**12**	15	17	**12**	12	12
	3	**22**	25	27	**22**	36	29
	4	25	**11**	13	14	24	19
	5	**8**	22	24	**8**	13	11
Penguin	2	18	12	14	**7**	15	11
	3	**11**	19	21	**11**	45	28
	4	41	25	27	32	**12**	22
	5	**23**	30	32	**23**	34	29
Dome	2	**8**	10	12	11	23	14
	3	**14**	15	17	15	45	30
	4	**9**	33	35	**9**	67	38
	5	45	25	27	**22**	**22**	**22**
Gentleman	2	**7**	10	12	8	12	10
	3	32	27	29	**26**	45	36
	4	21	24	26	**16**	56	36
	5	**17**	20	22	**17**	32	25
Horse	2	**5**	15	17	**5**	19	12
	3	20	**11**	13	16	18	17
	4	21	44	46	**17**	34	26
	5	27	**12**	14	24	24	24
Tiger	2	**10**	11	13	**10**	12	11
	3	20	20	22	**15**	**15**	**15**
	4	**22**	33	35	**22**	**22**	**22**
	5	**16**	27	29	18	18	18
Sea- star	2	**6**	22	24	**6**	8	9
	3	**9**	21	23	15	10	14
	4	**15**	45	47	16	12	13
	5	21	27	29	**16**	18	21
White bird	2	**9**	12	14	**9**	63	36
	3	**8**	14	16	**8**	37	23
	4	**10**	26	28	12	39	25
	5	14	25	27	**11**	28	20

Bold values represent minimum number of iterations required to converge problem provided. Under such circumstances, it is demonstrated that the computational cost of modified Otsu's is lower than Otsu's and Kapur's entropy for multi-level thresholding problems. Out of the total of forty experiments, twenty-four times modified Otsu's using AOA, and HDAFA optimization achieved the best score. It is clearly concluded from the evaluation of metric scores proposed that modified Otsu's method outperforms in terms of the number of iterations required to achieve optimum value.

PSNR provides information related to the peak signal-to-noise ratio, and it is a measure to represent peak error. It is used to verify the similarity that exists between the original and the segmented image. To compute the PSNR, it is necessary to use the Root Mean Square Error (RMSE) pixel to pixel. PSNR results in Table 11 provide evidence that the

Table 11 Comparison for PSNR

Image	Th	AOA (Modified Otsu's)	AOA (Otsu's)	AOA (Kapur)	HDFA (Modified Otsu's)	HDFA (Otsu's)	HDFA (Kapur)
Bear	2	17.0545	16.2435	11.4512	16.9108	16.2435	11.3464
	3	17.5968	17.3682	15.3368	18.0863	18.068	15.3383
	4	19.9789	19.8349	17.9764	21.5983	20.0129	17.9782
	5	20.3422	20.0268	16.4334	23.0286	22.4513	16.4350
Building	2	13.8643	13.1969	12.8308	13.1969	12.412	12.8321
	3	14.7953	14.8753	14.2961	14.8753	13.8643	14.2975
	4	15.5873	17.0078	15.9943	17.0078	14.9609	15.9959
	5	17.5043	17.5042	20.3525	17.6531	15.3049	20.3545
Deer	2	15.8681	14.3741	9.0497	14.3741	12.5328	9.0506
	3	15.599	16.9119	17.4092	16.9119	15.8681	17.4109
	4	18.2051	18.4367	17.9108	18.7545	17.1915	17.9126
	5	20.3761	20.7031	18.833	21.2919	18.2071	18.8349
Penguin	2	15.2029	14.5829	12.8199	14.5829	12.7269	12.8212
	3	15.4113	15.2107	12.9247	15.2107	15.2029	12.9260
	4	15.5829	16.5445	15.8618	16.5445	15.4087	15.8634
	5	15.6302	16.8059	15.8531	17.0817	15.5805	15.8547
Dome	2	20.1046	10.6292	19.9068	10.6292	10.7073	19.9088
	3	21.6779	12.7253	21.6724	20.1046	12.7253	21.6746
	4	22.5583	18.2683	22.5036	21.6779	18.2683	22.5059
	5	23.1847	19.7144	22.1242	22.413	19.7144	22.1264
Gentleman	2	16.2232	15.5726	14.5599	15.5726	16.4194	14.5614
	3	18.8644	17.7619	16.9051	17.7619	16.2232	16.9068
	4	19.6141	19.2084	18.7849	19.2022	18.8644	18.7868
	5	21.4032	21.31	21.092	21.2983	19.6457	21.0941
Horse	2	15.1626	14.5189	14.301	15.5726	14.3376	14.3024
	3	15.9618	16.1829	15.9477	17.7619	15.1325	15.9493
	4	18.2909	18.232	17.96	19.2084	17.7745	17.9618
	5	19.4964	19.3001	20.5256	21.2983	18.3161	20.5277
Tiger	2	14.5195	13.3475	14.4075	13.3475	12.9429	14.4089
	3	16.5166	15.9458	15.8568	15.9458	14.5195	15.8584
	4	19.1166	18.025	18.6805	18.1711	16.5166	18.6824
	5	20.53	20.147	20.3121	20.4013	18.9816	20.3141
Sea- star	2	16.8248	14.8134	14.3456	14.8134	14.1464	14.3470
	3	17.3927	17.3218	17.0393	17.3218	16.8248	17.0410
	4	18.8662	19.117	18.2243	19.117	17.2935	18.2261
	5	20.8329	20.7072	20.264	20.7072	18.9737	20.2660
White bird	2	15.2077	14.826	11.086	14.826	14.0106	11.0871
	3	16.1141	16.3037	12.228	16.3092	15.2077	12.2292
	4	18.6366	18.3029	16.365	18.3029	16.1153	16.3666
	5	19.9358	19.2958	19.2801	19.3053	16.8453	19.2820

Table 12 Comparison of the mean of the objective function values

Image	Th	AOA (Modified Otsu's)	AOA (Otsu's)	AOA (Kapur)	HDFA (Modified Otsu's)	HDFA (Otsu's)	HDFA (Kapur)
Bear	2	1822	1056	17.3619	1712	1493	17.3636
	3	1423	1140	21.7743	1621	1588	22.7765
	4	3236	1187	26.2098	2624	1574	26.2124
	5	3577	1213	30.1074	3234	1633	32.1104
Building	2	8483	2611	18.4119	2197	2631	18.4137
	3	3472	2778	22.9043	8483	2836	22.9066
	4	6819	2848	27.1463	3564	2856	26.1490
	5	9625	2893	31.2115	3564	3089	31.2146
Deer	2	8890	527	17.4704	5515	339	18.4721
	3	2259	588	21.8765	8890	422	21.9787
	4	5719	618	26.1298	2262	545	26.9324
	5	8649	637	30.703	5688	643	31.7061
Penguin	2	5796	1521	17.8834	6493	916	17.8852
	3	4807	1585	22.3797	5796	1211	22.3819
	4	2360	1607	26.5063	4806	1466	26.5090
	5	8654	1622	30.4966	2354	1676	30.4996
Dome	2	9807	5122	17.6228	8827	3163	17.6246
	3	5850	5170	22.3318	9807	4034	22.3340
	4	7710	5200	26.4166	5850	4764	26.4192
	5	7440	5223	29.9646	7690	5423	29.9676
Gentleman	2	9286	4874	18.3815	9512	3139	18.3833
	3	5150	5022	22.979	9286	4007	22.9813
	4	9770	5125	27.614	5150	4828	27.6168
	5	3170	5172	31.6256	9710	5614	31.6288
Horse	2	5064	1941	17.8988	3177	1333	17.9006
	3	5577	2071	22.8677	5063	1666	22.8700
	4	5990	2138	27.1198	5522	2032	27.1225
	5	4907	2174	31.28	5901	2060	31.2831
Tiger	2	0755	1639	18.2613	9665	1078	18.2631
	3	1308	1798	22.8454	2555	1169	22.8477
	4	2802	1873	27.1328	1895	1258	27.1355
	5	9570	1917	31.1014	8794	1496	31.1045
Sea- star	2	8434	2552	18.7092	7649	1605	18.7111
	3	4724	2784	23.264	4434	1703	23.2663
	4	8636	2869	27.4945	6418	1768	27.4972
	5	1477	2916	31.4637	1868	1864	31.4668
White bird	2	9171	1080	18.0389	8204	750	18.0407
	3	6055	1178	22.6531	9171	908	22.6554
	4	2262	1223	26.8582	6052	1062	26.8609
	5	7862	1249	31.4348	2235	1137	31.4379

outcome is better with the 4th and the 5th level of thresholding. Similarly, a comparison of the mean among modified Otsu's, Otsu's, and Kapur using AOA and HDAFA is presented in Table 12. All algorithms run 35 times independent runs of the same algorithm over each benchmark image, and the average value is reported to ensure the correct results.

Two different metaheuristic approaches test these three objective functions and in maximum cases proposed objective function produces not only quantitatively superior images but also good in terms of qualitatively results.

5 Conclusion

In this study, modified Otsu's method is proposed for the multi-level thresholding approach using metaheuristic algorithms AOA and HDAFA for image segmentation. Otsu's as an objective function is modified by hybridizing the features of Otsu's and Kapur's entropy algorithms. The proposed modified Otsu's function combines the capability to find the optimum threshold value that maximizes the overall entropy from Kapur's and the maximum variance value of the different classes from Otsu's method. The efficiency of the proposed objective function is checked over existing Otsu's and Kapur's entropy functions to find optimum multi-level threshold values required for image segmentation using AOA and HDAFA optimization algorithms. The experiment uses four levels of threshold values, i.e., 2,3,4, and 5, over ten standard benchmark images widely used for segmentation. The quantitative analysis for the maximum sample images demonstrates that modified Otsu's method outperforms Otsu's and Kapur's entropy functions using both metaheuristic methods. During experimentation, it is found that the result for two-level thresholding is always better for all source images in both AOA and HDAFA. Moreover, using the proposed modified Otsu's function, the results for higher threshold values are superior using AOA as compared to HDAFA. The modified Otsu's algorithm may be implemented for many image processing applications, for example, surveillance, biomedical imaging, industrial implementations, and pattern recognition. In future work, more complex concepts of entropy algorithms may be explored to make them durable for colored images. Also, the algorithm may be able to segment biomedical images like MRI and other thermal images.

Acknowledgments This work was supported by DST, Government of India, for the Technology Innovation Hub at the IIT Ropar in the National Mission on Interdisciplinary Cyber-Physical Systems framework.

Data availability The datasets generated during and/or analysed during the current study are available in the Berkeley Segmentation Data Set 500 (BSDS500) repository,
https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html

Declarations

Conflict of interest The authors declare that there is no conflict of interest regarding the publication of this manuscript.

References

1. Abak AT, Baris U, Sankur B (1997) The performance evaluation of thresholding algorithms for optical character recognition. Proc Fourth Int Conf Doc Anal Recognit 2:10–13. https://doi.org/10.1109/ICDAR.1997.620597
2. Abdel-Khalek S, Ben Ishak A, Omer OA, Obada ASF (2017) A two-dimensional image segmentation method based on genetic algorithm and entropy. Optik (Stuttg) 131:414–422. https://doi.org/10.1016/j.ijleo.2016.11.039
3. Abualigah L, Diabat A, Mirjalili S, Abd Elaziz M, Gandomi AH (2021) The arithmetic optimization algorithm. Comput Methods Appl Mech Eng 376:113609. https://doi.org/10.1016/j.cma.2020.113609
4. Aranguren I, Valdivia A, Pérez-Cisneros M, Oliva D, Osuna-Enciso V (2022) Digital image thresholding by using a lateral inhibition 2D histogram and a mutated electromagnetic field optimization. Multimed Tools Appl 81:10023–10049. https://doi.org/10.1007/s11042-022-11959-4

5. Bandyopadhyay R, Kundu R, Oliva D, Sarkar R (2021) Segmentation of brain MRI using an altruistic Harris hawks' optimization algorithm. Knowledge-Based Syst 232:107468. https://doi.org/10.1016/j.knosys.2021.107468
6. Bhandari AK, Kumar IV (2019) A context sensitive energy thresholding based 3D Otsu function for image segmentation using human learning optimization. Appl Soft Comput J 82:105570. https://doi.org/10.1016/j.asoc.2019.105570
7. Bohat VK, Arya KV (2019) A new heuristic for multilevel thresholding of images. Expert Syst Appl 117:176–203. https://doi.org/10.1016/j.eswa.2018.08.045
8. Brest J, Žumer V, Maučec MS (2006) Self-adaptive differential evolution algorithm in constrained real-parameter optimization. In: 2006 IEEE congress on evolutionary computation, CEC 2006. pp 215–222
9. Chaves AS (1998) A fractional diffusion equation to describe Lévy flights. Phys Lett Sect A Gen At Solid State Phys 239:13–16. https://doi.org/10.1016/S0375-9601(97)00947-X
10. Elaziz MA, Lu S (2019) Many-objectives multilevel thresholding image segmentation using knee evolutionary algorithm. Expert Syst Appl 125:305–316. https://doi.org/10.1016/j.eswa.2019.01.075
11. Feng Y, Wang Z (2014) Ant Colony optimization for image segmentation. https://doi.org/10.5772/14269
12. Gandomi AH, Yang XS, Talatahari S, Alavi AH (2013) Firefly algorithm with chaos. Commun Nonlinear Sci Numer Simul 18:89–98. https://doi.org/10.1016/j.cnsns.2012.06.009
13. Garg A, Mittal N, Singh S, Sharma N (2020) TLBO Algorithm for Global Optimization : Theory , Variants and Applications with Possible Modification. Int J Adv Sci Technol 29:1701–1728
14. Hashim FA, Hussain K, Houssein EH, Mabrouk MS, al-Atabany W (2021) Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems. Appl Intell 51:1531–1551. https://doi.org/10.1007/s10489-020-01893-z
15. He L, Huang S (2017) Modified firefly algorithm based multilevel thresholding for color image segmentation. Neurocomputing. 240:152–174. https://doi.org/10.1016/j.neucom.2017.02.040
16. Hemeida A, Mansour R, Hussein ME (2019) Multilevel thresholding for image segmentation using an improved electromagnetism optimization algorithm. Int J Interact Multimed Artif Intell 5:102. https://doi.org/10.9781/ijimai.2018.09.001
17. Horng M-H (2010) A multilevel image thresholding using the honey bee mating optimization. Appl Math Comput 215:3302–3310
18. Hosny KM, Khalid AM, Hamza HM, Mirjalili S (2022) Multilevel segmentation of 2D and volumetric medical images using hybrid coronavirus optimization algorithm. Comput Biol Med 150:106003. https://doi.org/10.1016/j.compbiomed.2022.106003
19. Houssein EH, Helmy BE-d, Oliva D et al (2021) A novel black widow optimization algorithm for multilevel thresholding image segmentation. Expert Syst Appl 167:114159. https://doi.org/10.1016/j.eswa.2020.114159
20. Ji W, He X (2021) Kapur's entropy for multilevel thresholding image segmentation based on moth-flame optimization. Math Biosci Eng 18:7110–7142. https://doi.org/10.3934/mbe.2021353
21. Kapur JN, Sahoo PK, Wong AKC (1985) A new method for gray-level picture thresholding using the entropy of the histogram. Comput Vision, Graph Image Process 29:273–285
22. Kaur R, Singh S (2017) An artificial neural network based approach to calculate BER in CDMA for multiuser detection using MEM. Proc 2016 2nd Int Conf next Gener Comput Technol NGCT 2016 450–455. https://doi.org/10.1109/NGCT.2016.7877458
23. Kennedy J, Eberhart RC (1995) Particle swarm optimization. Neural networks, 1995 proceedings, IEEE Int Conf 4:1942–1948 vol.4. https://doi.org/10.1109/ICNN.1995.488968
24. Kumar A, Kumar V, Kumar A, Kumar G (2014) Cuckoo search algorithm and wind driven optimization based study of satellite image segmentation for multilevel thresholding using Kapur ' s entropy. Expert Syst Appl 41:3538–3560. https://doi.org/10.1016/j.eswa.2013.10.059
25. Kumar A, Misra RK, Singh D, Das S (2019) Testing a multi-operator based differential evolution algorithm on the 100-digit challenge for single objective numerical optimization. 2019 IEEE Congr Evol Comput CEC 2019 - proc 34–40. https://doi.org/10.1109/CEC.2019.8789907
26. Lei B, Fan J, Fan J (2019) Adaptive Kaniadakis entropy thresholding segmentation algorithm based on particle swarm optimization. Soft Comput 2:. https://doi.org/10.1007/s00500-019-04351-2, 24, 7305, 7318
27. Liang H, Jia H, Xing Z, Ma J, Peng X (2019) Modified grasshopper algorithm-based multilevel thresholding for color image segmentation. IEEE Access 7:11258–11295. https://doi.org/10.1109/ACCESS.2019.2891673
28. Liao PS, Chen TS, Chung PC (2001) A fast algorithm for multilevel thresholding. J Inf Sci Eng 17:713–727. https://doi.org/10.6688/JISE.2001.17.5.1
29. Liao WH, Kao Y, Li YS (2011) A sensor deployment approach using glowworm swarm optimization algorithm in wireless sensor networks. Expert Syst Appl 38:12180–12188. https://doi.org/10.1016/j.eswa.2011.03.053

30. Nadimi-Shahraki MH, Taghian S, Mirjalili S, Zamani H, Bahreininejad A (2022) GGWO: gaze cues learning-based grey wolf optimizer and its applications for solving engineering problems. J Comput Sci 61:101636. https://doi.org/10.1016/j.jocs.2022.101636
31. Oliva D, Abd Elaziz M, Hinojosa S (2019) Multilevel thresholding for image segmentation based on metaheuristic algorithms. In: Studies in Computational Intelligence. pp 59–69
32. Oliva D, Esquivel-Torres S, Hinojosa S et al (2021) Opposition-based moth swarm algorithm. Expert Syst Appl 184:115481. https://doi.org/10.1016/j.eswa.2021.115481
33. Ouyang HB, Gao LQ, Kong XY, Li S, Zou DX (2016) Hybrid harmony search particle swarm optimization with global dimension selection. Inf Sci 346:318–337. https://doi.org/10.1016/j.ins.2016.02.007
34. Qi A, Zhao D, Yu F, Heidari AA, Wu Z, Cai Z, Alenezi F, Mansour RF, Chen H, Chen M (2022) Directional mutation and crossover boosted ant colony optimization with application to COVID-19 X-ray image segmentation. Comput Biol Med 148:105810. https://doi.org/10.1016/j.compbiomed.2022.105810
35. Ren L, Zhao D, Zhao X, Chen W, Li L, Wu TS, Liang G, Cai Z, Xu S (2022) Multi-level thresholding segmentation for pathological images : optimal performance design of a new modified differential evolution. Comput Biol Med 148:105910. https://doi.org/10.1016/j.compbiomed.2022.105910
36. Rodríguez-Esparza E, Zanella-Calzada LA, Oliva D et al (2020) An efficient Harris hawks-inspired image segmentation method. Expert Syst Appl:155. https://doi.org/10.1016/j.eswa.2020.113428
37. Singh S, Mittal N, Singh H (2020) A multilevel thresholding algorithm using LebTLBO for image segmentation. Neural Comput Applic 32:16681–16706. https://doi.org/10.1007/s00521-020-04989-2
38. Singh S, Mittal N, Singh H (2020) Multifocus image fusion based on multiresolution pyramid and bilateral filter. IETE J Res 68:2476–2487. https://doi.org/10.1080/03772063.2019.1711205
39. Singh S, Mittal N, Singh H (2021) A multilevel thresholding algorithm using HDAFA for image segmentation. Soft Comput 25:10677–10708. https://doi.org/10.1007/s00500-021-05956-2
40. Singh S, Mittal N, Thakur D, Singh H, Oliva D, Demin A (2021) Nature and biologically inspired image segmentation techniques. Arch Comput Methods Eng 29:1415–1442. https://doi.org/10.1007/s11831-021-09619-1
41. Singh S, Singh H, Mittal N, Singh H, Hussien AG, Sroubek F (2022) A feature level image fusion for night-vision context enhancement using arithmetic optimization algorithm based image segmentation. Expert Syst Appl 209:118272. https://doi.org/10.1016/j.eswa.2022.118272
42. Singh S, Mittal N, Singh H (2022) A feature level image fusion for IR and visible image using mNMRA based segmentation. Neural Comput Applic 34:8137–8154. https://doi.org/10.1007/s00521-022-06900-7
43. Sreeja P, Hariharan S (2018) An improved feature based image fusion technique for enhancement of liver lesions. Biocybern Biomed Eng 38:611–623. https://doi.org/10.1016/j.bbe.2018.03.004
44. Sun X (2019) Multilevel image threshold segmentation using an improved Bloch quantum artificial bee colony algorithm. Multimed Tools Appl 79:2447–2471. https://doi.org/10.1007/s11042-019-08231-7
45. Verma OP, Parihar AS (2017) An optimal fuzzy system for edge detection in color images using bacterial foraging algorithm. IEEE Trans Fuzzy Syst 25:114–127. https://doi.org/10.1109/TFUZZ.2016.2551289
46. Zamani H, Nadimi-Shahraki MH, Gandomi AH (2019) CCSA: conscious neighborhood-based crow search algorithm for solving global optimization problems. Appl Soft Comput J 85:105583. https://doi.org/10.1016/j.asoc.2019.105583
47. Zamani H, Nadimi-Shahraki MH, Gandomi AH (2021) QANA: quantum-based avian navigation optimizer algorithm. Eng Appl Artif Intell 104:104314. https://doi.org/10.1016/j.engappai.2021.104314
48. Zamani H, Nadimi-Shahraki MH, Gandomi AH (2022) Starling murmuration optimizer: a novel bio-inspired algorithm for global and engineering optimization. Comput Methods Appl Mech Eng 392:114616. https://doi.org/10.1016/j.cma.2022.114616

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.