Retraction Notice

The Editor-in-Chief and the publisher have retracted this article, which was submitted as part of a guest-edited special section. An investigation uncovered evidence of systematic manipulation of the publication process, including compromised peer review. The Editor and publisher no longer have confidence in the results and conclusions of the article.

BDS and SKG agree with the retraction.

Hybrid whale optimization algorithm-Levy flight approach for multilevel thresholding image segmentation

Basu Dev Shivahare[a,*] **and Sanjai Kumar Gupta**[b]
[a]Dr. A.P.J. Abdul Kalam Technical University, Department of Computer Science and Engineering, Lucknow, Uttar Pradesh, India
[b]Bundelkhand Institute of Engineering and Technology Jhansi, Department of Computer Science and Engineering, Jhansi, Uttar Pradesh, India

Abstract. Determining the optimal threshold for multilevel image segmentation is a stimulating task. Image segmentation has many applications, such as content-based image retrieval, medical imaging, object detection, object recognition, machine vision, etc. In multilevel thresholding, multiple thresholds are used to segment complex images, and the images are segmented into multiple levels to extract meaningful information for further image analysis. Where threshold number is small, p-tile, Otsu, moment preserving, and entropy thresholding methods attain good accuracy. These classical thresholding methods are time-consuming, computation expensive, and unable to produce good segmentation accuracy with increased threshold numbers. To overcome these problems, the classical thresholding methods are utilized as objective functions with nature-inspired metaheuristic algorithms such as whale optimization algorithm (WOA), simplified swarm optimization (SSO), sine cosine algorithm (SCA), BAT algorithm, WOA-thresholding, and black widow optimization (BWO) to determine optimal multiple thresholds. Nature-inspired metaheuristic algorithms are widely used to search for an optimal solution for global optimization problems. WOA is widely used to find the optimal solution in search space. WOA suffers from entrapment into local optima due to premature convergence behavior. We introduce hybridization of WOA with Levy flight trajectory and named as hybrid whale optimization algorithm-Levy flight (HWOAL), which is utilized to find optimal multiple thresholds for multilevel image segmentation. Levy flight trajectory is utilized to increase diversity in the swarm population. The efficacy of HWOAL is tested on 23 benchmark optimization functions (F1 to F23) and compared with WOA, SSO, SCA, and BAT algorithms. Experiment arms that HWOAL is efficient and makes WOA faster, enhances the ability of exploitation and exploration phase, and can avoid getting stuck into local optima. The segmentation performance of the proposed HWOAL method has been compared with other algorithms, such as WOA, SSO, SCA, BAT, and other two recent segmentation algorithms, such as WOA-TH and BWO on several benchmark images (BSD 300). The experimental result is carried in 30 trials and analyzed based on objective fitness value, optimal multilevel threshold values, segmentation quality measures, such as mean square error, peak signal-to-noise ratio, structural similarity index, average difference, and computation time to compute optimal multilevel threshold values. The experiment conducted by the authors shows that the proposed HWOAL method is efficient, produces better fitness value, and segmentation metrics for multilevel image segmentation than other algorithms. © *2022 SPIE and IS&T* [DOI: 10.1117/1.JEI.31.5.051420]

Keywords: black widow optimization; differential evolution; Levy flight; multilevel image segmentation; optimization algorithms; whale optimization algorithm.

Paper 220072SS received Feb. 11, 2022; accepted for publication May 26, 2022; published online Jun. 17, 2022; retracted Jul. 25, 2023.

*Address all correspondence to Basu Dev Shivahare, bdshivahare@gn.amity.edu; basuiimt@gmail.com

1 Introduction

To determine optimal thresholds for segmenting an image into multiple regions is a stimulating task and an NP-hard problem.[1] Images can be segmented into two or more levels. In bilevel thresholding image segmentation, the image is partitioned into two levels and in multilevel image segmentation, the image is partitioned into more than two levels.[2,3] Complex images are segmented into multilevel with multiple threshold values[4] for further image analysis. An image is segmented into $k+1$ classes or levels for k number of thresholds. Complex images are segmented into multiple regions using multilevel thresholding to get information from target regions.[5]

Otsu's maximum between-class variance or minimum within-class variance criteria is utilized to compute optimal thresholds.[6] Otsu's criteria are extended to compute multilevel thresholds. The classical thresholding methods such as p-tile, moment preserving, Otsu, and entropy are computationally inefficient, time-consuming, and do not produce good segmentation accuracy due to multiple peaks and valleys in the histogram with increased number of thresholds.[7] Nature-inspired metaheuristic algorithms have drawn attention in recent years and are widely used to solve global optimization problems. To overcome these problems, nature-inspired metaheuristic algorithms with classical image thresholding method are utilized to compute optimal multiple thresholds for multilevel image segmentation.

Mirjalili and Lewis[8] proposed WOA, which enhanced local search capability of unimodal functions and better explored multiple local search solution. WOA was able to jump out from local optima and no use of derivative makes WOA faster than PSO, GSA, DE, and fast evolutionary programming. Positions of search agents are updated in each iteration of WOA and updated positions are fed as input in next iteration to achieve best solution.[8–10]

Global optimization problems were effectively solved by Levy flight trajectory-based WOA (LWOA). LWOA was proposed by Ling et al.,[11] in which positions of whales are updated using Levy flight. LWOA was tested on 23 benchmark optimization functions (F1 to F23) and compared with "moth flame optimization (MFO), WOA, particle swarm optimization gravitational search algorithm (PSOGSA), BAT, and artificial bee colony (ABC)." Dimensions of objective functions F15, F18, F19, F20 were set as 4, F14, F16, F17, F21-F23 were set as 2, and F1-F13 were set as 50. LWOA has shown better results than WOA and other methods, avoids premature convergence and able to jump out from local optima. LWOA has achieved fitness score close to defined optimal value for 14 functions out of 23 functions.

Akay[12] used optimal multiple thresholds to segment the images into multiple regions. Optimal thresholds were computed by PSO, ABC algorithm separately by Otsu's maximum between-class variance criteria and maximizing Kapur's entropy as objective function. For higher threshold number $k > 2$, ABC has shown better performance than PSO and Otsu algorithms.

Ali[13] determined optimal multiple thresholds by Cuckoo search, PSO, firefly FA, ABC swarm algorithms by utilizing Otsu's maximum between-class variance criteria as objective function. Experiment was carried out in 25 trials, and 500 iterations were set under each trial. PSO has produced better fitness score and efficient better than other algorithms.

MWOA has solved high-dimension (100 to 1000 dimensions) global optimization problems efficiently. WOA may stuck into local optima while solving high-dimension problems, such as scientific engineering problems, aerospace design, etc. Modified WOA based on Levy flight and quadratic interpolation (MWOA) was proposed by Sun et al.[14] to overcome this problem. MWOA was tested on standard 25 benchmark functions and range of dimensions were set as 100 to 1000. MWOA is stable, more accurate, achieves faster convergence, makes WOA faster, well-utilized exploitation, and exploration phases than exploration-enhanced gray wolf optimization, best-so-far ABC algorithm, and LWOA.

Pruthi and Gupta[15] improved segmentation accuracy, segmentation metrics result by genetic algorithm (GA) with Otsu method. El Aziz et al.[16] performed multilevel image segmentation by optimal thresholds computed by proposed whale optimization algorithm (WOAMOP). WOAMOP has utilized Otsu and Kapur's entropy as a single fitness function to compute effective optimal multilevel thresholds. Computation time, average fitness value, best threshold values, peak signal-to-noise ratio (PSNR), and structural similarity index (SSIM) were

determined over 30 trials, and 100 iterations were set under each trial. WOAMOP has been compared with WOA, social spider optimization (SSO), firefly algorithm (FA), and firefly algorithm and social spider optimization algorithm (FASSO). The proposed method has achieved better SSIM and PSNR at threshold numbers $k = 2$, 3, 4, and 5. El Aziz et al.[17] computed optimal set of thresholds using MFO and WOA to perform multilevel image segmentation. Otsu's maximum between criteria was used as objective function. Computation time, segmentation quality metrics such as SSIM and PSNR, best fitness score were computed by MFO, WOA, and other algorithms, such as SSO, FA, FASSO, SCA, and harmony search. MFO and WOA have achieved better results than other algorithms.

Talal et al.[18] proposed EMD-discrete wavelet transform (DWT) method to improve the sharpening of multiband satellite images by preserving the spectral and spatial quality of images with efficient computational time than empirical mode decomposition (EMD), DWT, Brovey, high pass filtering, intensity hue saturation, and multiplicative methods.

Attiya et al.[19] proposed modified central force optimization MCFO method and applied it with various fusion methods such as IHS, DWT, HPF, to enhance the SPOT-4, Landsat-8, and QuickBird satellite images visually and quantitatively along such variables as PSNR, RMSE, edge intensity, UIQI, Std, correlation coefficient, and entropy.

Bohat and Arya[20] proposed an improvement to WOA and named it WOA thresholding (WOA-TH) to compute optimal multiple thresholds and compared it with WOA, PSO, PSO-TH, gray wolf optimization (GWO), and GWO thresholding (GWO-TH) technique. WOA-TH has more promising results as compared to WOA.

Houssein et al.[21] proposed black widow optimization (BWO) to perform multilevel image segmentation. The performance of BWO was superior to WOA, SCA, salp swarm algorithm (SSA), MFO, GWO, and equilibrium optimization (EO). BWO was further applied in image segmentation and has better PSNR and SSIM than other methods in most cases.

Shivahare and Gupta[22] computed ideal multiple threshold value for multilevel COVID-19 CT scan image segmentation by improved whale optimization algorithm (IWOA). Better segmentation mask and segmentation accuracy were obtained for IWOA-based segmented images as compared to WOA, SSA, and SCA-based segmented images. IWOA has been proved as better automatic clustering algorithm to compute the optimal thresholds as compared to other methods.

This paper introduces hybridization of a variant of the WOA with Levy flight trajectory and named as hybrid whale optimization algorithm-Levy flight trajectory approach (HWOAL). The efficacy of HWOAL is tested on 23 benchmark optimization functions (F1-F23) and compared with WOA, SSO, SCA, and BAT algorithms (explained in Sec. 4.1). Experimentation shows that HWOAL is efficient, makes WOA faster and more capable of exploiting local and global search space, and avoids entrapment into local optima. Several benchmark images from Berkeley Image segmentation dataset(BSD300) are selected to perform multilevel image segmentation at various threshold numbers $k = 2$, 3, 4, and 5 by using HWOAL, WOA,[8] SSO,[23] SCA,[24] BAT,[25] WOA-TH[20] and BWO[21] over 30 trials and 500 iterations were set under each trial (as multilevel image segmentation performed previously by Akay,[12] Bhandari et al.,[2] and Ewees et. al.[17]) All considered algorithms utilized Otsu thresholding method as the fitness function and step length from Levy(dim) is computed and utilized in exploitation and exploration phase of WOA to update the whales' position. Segmentation quality metrics such as PSNR, SSIM, mean square error (MSE), average difference (AD), average computation time, best fitness score, and optimal threshold values of each image computed by considered algorithms and performance is analyzed with other thresholding methods. The experiment has shown that the proposed HWOAL method is efficient and reports better segmentation metrics, fitness value, and optimal threshold values for multilevel image segmentation than other algorithms.

2 Materials and Methods

2.1 *Problem Formulation*

Image is segmented into $k+1$ regions/levels/classes for thresholds, i.e., $\{TH_1, TH_2, TH_3, \ldots, TH_K\}$ thresholds where $k = 1,2,3,\ldots,k$. The threshold values range of each class is defined as

$$\text{CLASS}_1 = \{f(x,y)\varepsilon G | 0 \le G(x,y) \le TH_1\}$$
$$\text{CLASS}_2 = \{f(x,y)\varepsilon G | TH_1 + 1 \le G(x,y) \le TH_2\}$$
$$\ldots$$
$$\text{CLASS}_{k+1} = \{f(x,y)\varepsilon G | TH_k + 1 \le G(x,y) \le L - 1\}.$$

To determine efficient multilevel threshold for good visual quality, multilevel image segmentation is stimulating task. Multilevel thresholds can be determined by classical Otsu thresholding method. To compute optimal multiple thresholds for efficient multilevel image segmentation, Otsu's maximizing between class variance criteria is utilized as objective/fitness function for the proposed method HWOAL and with other considered methods. HWOAL computes the fitness at various threshold number $k = 2$, 3, 4, and 5.

The pixel intensity $f(x,y)$ of gray level image G varies between 0 and 255, i.e., $f(x,y)\varepsilon$ [0,255].

Multiple threshold values are computed by Otsu's maximum between class variance criteria and denoted by following Eq. (1):

$$TH_1^*, TH_2^*, \ldots TH_k^* = \text{Maximize}\{TH_1, TH_{2,} \ldots TH_k\} \text{Objective}(TH_1, TH_{2,} \ldots TH_k), \quad (1)$$

where objective $(TH_1, TH_{2,} \ldots TH_k)$ is expressed as Eq. (2).

$$\text{Objective}(TH_1, TH_{2,} \ldots TH_k) = \sum_{i=TH_1}^{TH_k} Z_i (M_i - M_T)^2, \quad (2)$$

where Z_i, M_i represent weight and mean of i'th class:

$$Z_i = \sum_{j=TH_i}^{TH_{i+1}-1} p_j. \quad (3)$$

Equation (4) describes the probability of j'th class

$$p_j = f(j)/P, \quad (4)$$

where $f(j)$ denotes frequency of j'th gray level and P denotes total number of pixels.

Equations (5) and (6) represent the class means M_i and M_T

$$M_i = \sum_{j=TH_i}^{TH_{i+1}-1} i \frac{p_j}{Z_j}, \quad (5)$$

$$M_T = \sum_{j=0}^{L-1} j p_j. \quad (6)$$

Equations (7) and (8) must be satisfied to meet Otsu's condition

$$\sum_{i=1}^{L+1} Z_i M_i = M_T, \quad (7)$$

$$\sum_{i=1}^{L+1} Z_i = 1. \quad (8)$$

2.2 *Whale Optimization Algorithm*

WOA, introduced by Mirjalili and Lewis,[8] is an efficient algorithm and finds global solutions by enhancing the capability of the exploitation and exploration phases of WOA. The important

feature of WOA is that computation of derivation is not required. For $p < 0.5$ condition, whales' positions are updated to achieve better solution under exploitation $|A| < 1$ or exploration phase $|A| \geq 1$. WOA searches for a better solution by exploiting local search area or explores the search space by random whale.

2.2.1 *Exploitation phase*

The bubble creation by humpback whale is performed in two ways.

Shrinking encircling prey. For $|A| < 1$ under $p < 0.5$ condition and $p \geq 0.5$, WOA searches the better solution by exploiting the promising areas of local search space. Humpback whales/search agents sense the target prey and encircle the prey. After that, search agents try to move closer to target prey by the shrinking encircle technique. The mathematical expression of shrinking encircle is shown as

$$\vec{U}(t+1) = \vec{H}_b(t) - \vec{A}.\vec{D}, \tag{9}$$

where $\vec{U}(t+1)$ represents updated positions of search agents at current iteration (t) and $\vec{A}$ is the coefficient vector and expressed as

$$\vec{A} = 2\vec{a} \cdot \mathrm{p1} - \vec{a}. \tag{10}$$

Distance $\vec{D}$ is computed between best search agent $\vec{H}_b(\mathrm{t})$ and current search agent $\vec{H}(\mathrm{t})$ from search space at iteration t. Distance $\vec{D}$ is represented as

$$\vec{D} = |\vec{C}.\vec{H}_b(t) \sim \vec{H}(t)|. \tag{11}$$

Search agents utilize adjustment factor $\vec{C}$ to catch promising local regions of search space. $\vec{C}$ is represented as

$$\vec{C} = 2 \cdot p2. \tag{12}$$

The value of a is shrinking from 2 to 0 in successive iterations. The random range of $\vec{A}$ is $[-1, 1]$. $p1$, $p2$ ε rand(0,1)

Spiral updating position. Spiral formation helps humpback whale to reach toward target prey. This phase is used to create spiral form by whales and expressed as

$$\vec{U}(t+1) = D' \cdot e^{b\ell} \cdot \cos(2\pi l) + \vec{H}_b(t). \tag{13}$$

Logarithmic spiral shape is defined by constant variable b and parameter ℓ is mathematically expressed as

$$\ell = (a2 - 1) * \mathrm{rand} + 1. \tag{14}$$

The value of $a2$ is decreasing from -1 to -2 over successive iteration and expressed as

$$a2 = -1 + t * ((-1)/\mathrm{MaxItr}). \tag{15}$$

Distance D' is computed between best search agent $\vec{H}_b(t)$ and search agent $\vec{H}(t)$ at current iteration (t) and represented as

$$D' = |\vec{H}_b(t) - \vec{H}(t)|. \tag{16}$$

The encircling and spiral updating phase under exploitation is mathematical modeled in Eq. (17):

$$\vec{U}(\mathrm{t}+1)=\begin{cases}\text{Encircling Eq}\,(9), & p<0.5\\ \text{Spiral Eq.}\,(13), & p\geq 0.5\end{cases}. \tag{17}$$

The range of $p\ \varepsilon$ rand(0,1).

2.2.2 *Exploration phase*

For $|A|\geq 1$ under random value ($p<0.5$) condition, positions of a randomly chosen humpback whale $\vec{H}_{\text{rand}}$ from search space is utilized to compute the distance $\vec{D}$. Distance $\vec{D}$ is further used to update the positions of whale. The exploration phase of WOA is mathematically expressed in Eqs. (18) and (19) as

$$\vec{D}=|\vec{C}.\vec{H}\,\text{rand}-\vec{H}(t)|, \tag{18}$$

$$\vec{U}(t+1)=\vec{H}\,\text{rand}-\vec{A}.\vec{D}, \tag{19}$$

where $\vec{H}_{\text{rand}}$ is randomly picked whale, $\vec{H}(t)$. is the search agent's position in current iteration t, and updated position of search agent/whale at iteration t is represented by $\vec{U}(t+1)$ (Algorithm 1).

Algorthim 1 Whale optimization algorithm[8]

1: Input: Initialize whales population (S_m) within [LB,UB] in n dimension, $m=1,2,...,n$
2: Output: H^* (global position of best whale H_b)
3: Calculate the objective function value of each whale and determine best search agent (H^*)
4: while ($t<\text{Max}_{\text{iter}}$) do
5: Check and limit the population of whales within [LB,UB], calculate the objective function value of each whale and determine best search agent ($H*$)
6: Iterate each whale do
7: Calculate p, $\vec{a}$, $\vec{A}$, and $\vec{C}$, ℓ
8: Condition ($p\geq 0.5$) under exploitation
9: Find updated positions of whales using Eq. (13) under spiral phase
10: Condition ($p<0.5$)
11: Condition check $|A|<1$ under exploitation
12: Find updated positions of whales using encircling phase Eq. (9)
13: Condition check $|A|\geq 1$ under exploration
14: Determine location of random whale (H_{rand})
15: Find updated positions of whales using Eq. (19)
16: end step 11
17: Step 8 terminate
18: Step 6 terminate
19: Increment t
20: Step 4 terminate
21: Output is best whale leader position

2.3 *Levy Flight Trajectory*

Levy flight trajectory-based whale optimization algorithm (LWOA), proposed by Ling,[11] has solved global optimization problems effectively. In LWOA, the best positions (S_i) of each search agent is obtained through an iteration of WOA. After that, Levy flight trajectory approach is applied to update the positions (S_i) of search agents to achieve a better solution. This is fed as input to the next iteration of WOA. Levy flight utilizes random walk approach and enhances the searching ability of WOA, population's diversity, and avoids entrapment into local optima. "LWOA, WOA, MFO, BAT, and PSOGSA" were tested on 23 benchmark functions (F1-F23). Dimension of F15, F18, F19, F20, F14, F16, F17, F21-F23, and (F1-F13) functions were set as 4, 2, and 50, respectively. In LWOA, the positions of whales are updated using Levy flight. LWOA outperformed in 14 functions out of 23 functions, except for F5, F6, F12, F14, F16, F17, F18, F20, and F23. LWOA avoids premature convergence jumping out of local optima and makes WOA faster. The experiment was carried out in 30 trials. Population size and number of iterations were set as 20 and 1000, respectively, under each trial.

Step length S is computed from Levy(dim) for $\beta = 1.5$ and $\Upsilon = 1$ and expressed in Eq. (20), where dim represents dimension of search agent:

$$S = \frac{y}{z^{1/\beta}}, \tag{20}$$

$$y = \text{rand}(1, \text{dim}) * \sigma, \tag{21}$$

$$z = \text{rand}(1, \text{dim}), \tag{22}$$

$$\sigma = \left[\frac{\Upsilon(1+\beta)x\ \sin(\pi x\beta/2)}{\Upsilon\left(1+\frac{\beta}{2}\right)x\beta x 2^{(\beta-1)/2}}\right]^{1/\beta}. \tag{23}$$

3 Proposed Method: Hybrid Whale Optimization—Levy Flight Algorithm

In this section, hybridization of a variant of the WOA by including randomization in the spiral step of WOA with Levy flight trajectory is proposed and named as HWOAL approach. Fitness score and optimal multiple thresholds are computed by HWOAL by utilizing Otsu's maximum between class variance criteria Eq. (2) as an objective function. In HWOAL, step length (S) of Levy flight trajectory is calculated in each iteration of WOA when $p < 0.5$ condition is satisfied. This step length (S) value is used to calculate the distance either in exploration or exploitation phase of WOA depending on the value of $|A|$ to achieve best solution/best positions of search agents. The efficiency of HWOAL is described in Sec. 4.1. HWOAL has attained optimal value close to $f_{\min}$ in 15 functions and attained at least second rank in four functions as compared with other methods. Average computation time, segmentation metrics, best and worst fitness values, and optimal set of thresholds for multilevel image segmentation are computed and compared with WOA, SSO, SCA, BAT, WOA-TH, and BWO. For random value $p < 0.5$ condition of WOA, step length S is computed from Levy(dim) that is used to compute the distance under exploitation $|A| < 1$ or exploration phase $|A| \geq 1$. The computed distance is further used to update the positions of whales either in exploitation or exploration phase.

Flowchart in Fig. 3 shows the working mechanism of HWOAL. In flowchart, E1 denotes to Eq. (13), E2 denotes to Eqs. (24), (25), E3 denotes to Eqs. (26) and (27).

If the best search agent is not moving close to target prey while the search for prey is performed by humpback whale under the exploitation and exploration phase, then WOA may suffer from being stuck in local optima due to premature convergence.

To jump out from local optima or to avoid entrapment in local optima, the following approaches are used:

(i) For $|A| \geq 1$ and $|A| < 1$ under random condition ($p < 0.5$) of HWOAL, step length S is computed from Levy(dim) as discussed in Sec. 2.3. S is utilized with the exploitation and

exploration phase to update the positions of search agents. Hence, HWOAL may get a better solution from the search agent over the course of iterations than other methods and better utilize the exploitation and exploration capability of WOA.

E2 is mathematically modeled in Eqs. (24) and (25)

$$\vec{D} = \mathrm{S} \oplus (\vec{C}.\vec{H}\mathrm{b}(t) \sim \vec{H}t(\mathrm{sa}, :)), \tag{24}$$

$$\vec{U}(t+1) = \vec{H}\mathrm{b}(t) - \vec{A}.\vec{D}. \tag{25}$$

E3 is mathematically modeled in Eqs. (26) and (27):

$$\vec{D} = S \oplus (\vec{C}.\vec{H}_{\mathrm{rand}} - \vec{H}t(sa, :), \tag{26}$$

$$\vec{U}(\mathrm{t}+1) = \vec{H}_{\mathrm{rand}} - \vec{A}.\vec{D}, \tag{27}$$

where S is the step length, $\oplus$ implies elementwise multiplication, $\vec{H}_b(t)$ is the search agent's global position in the current iteration. $\vec{H}t(sa, :)$ is the row-wise positions of the search agent in the dim dimension instead of row- and columnwise positions, and $\vec{H}_{\mathrm{rand}}$ is a randomly picked whale.

In E1, $\vec{H}(t)$ denotes (row, column) position of whale.

(ii) In WOA, the value of $a2$ parameter is linearly decreasing toward -1 to -2 over the course of iterations. The $a2$ parameter attains the value -1.5 after crossing half of the declared iterations and atttains -2 at last iteration. The value of ℓ parameter depends on $a2$ parameter. The parameter ℓ is utilized to make spiral form of WOA as shown in Eq. (13). If random value of $a2$ is dynamically selected in the range between $[-1, -2]$ as shown in Fig. 1, then there maybe chances that $a2$ may get value -1.5 to -2 in starting few iterations. In general, parameter ℓ utilizes $a2$ parameter in Eq. (14), which influences the spiral updating phase of WOA and may enhance better the exploitation or local search area. $a2$ parameter is expressed in as

$$a2 = -1 + (-2 + 1) * \mathrm{rand}(). \tag{28}$$

Optimal threshold values at various threshold number are computed by HWOAL and shown in Fig. 2 (Algorithm 2). Figure 3 illustrates the flowchart of the proposed method (HWOAL) to compute optimal threshold values. In the flowchart, E1 denotes to Eq. (13), E2 denotes to Eqs. (24), (25), and E3 denotes to Eqs. (26) and (27).

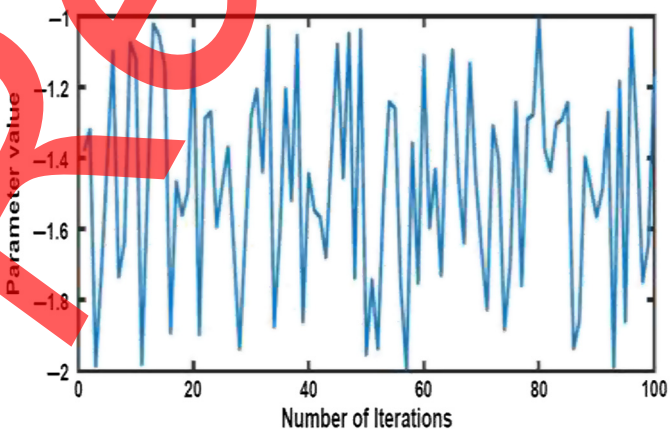

Fig. 1 Variation of $a2$ parameter.

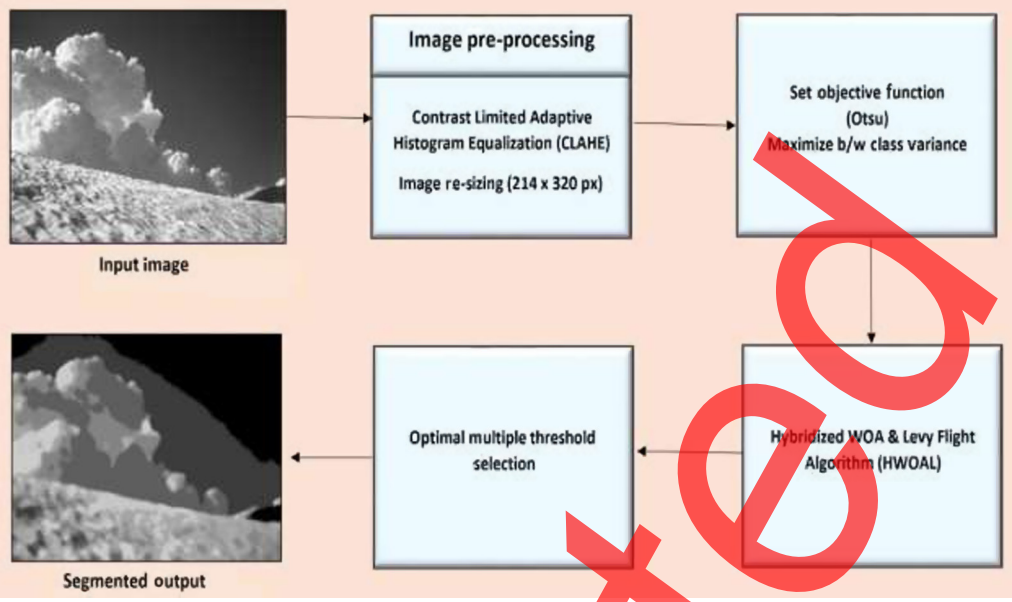

Fig. 2 Optimal multiple thresholds computed by proposed algorithm HWOAL.

Algorithm 2 Hybrid whale optimization-Levy flight algorithm.

1: Input: Initialize whales population (S_m) within [LB,UB] in n dimension, $m = 1, 2, \ldots, n$
2: Output: H^* (global position of best whale H_b)
3: Calculate the objective function value of each whale and determine best search agent (H^*)
4: while ($t < \text{Max}_{\text{iter}}$) do
5: Check and limit the population of whales within [LB,UB], Calculate the objective function value of each whale and determine best search agent (H^*)
6: Iterate each whale do
7: Calculate p, $\vec{a}$, $\vec{A}$, and $\vec{C}$, ℓ
8: Condition ($p \geq 0.5$) under exploitation
9: Find updated positions of whales using E1 under spiral phase
10: Condition ($p < 0.5$)
11: Calculate step length (S) from Levy flight(dim)
12: Condition $|A| < 1$ under exploitation
13: Find updated positions of whales using E2 under encircling phase
14: Condition $|A| \geq 1$ under exploration
15: Determine location of random whale ($\vec{H}_{\text{rand}}$)
16: Find updated positions of whales using E3
17: end step 12
18: Step 8 terminate
19: Step 6 terminate
20: Increment t
21: Step 4 terminate
22: Output is best whale leader position

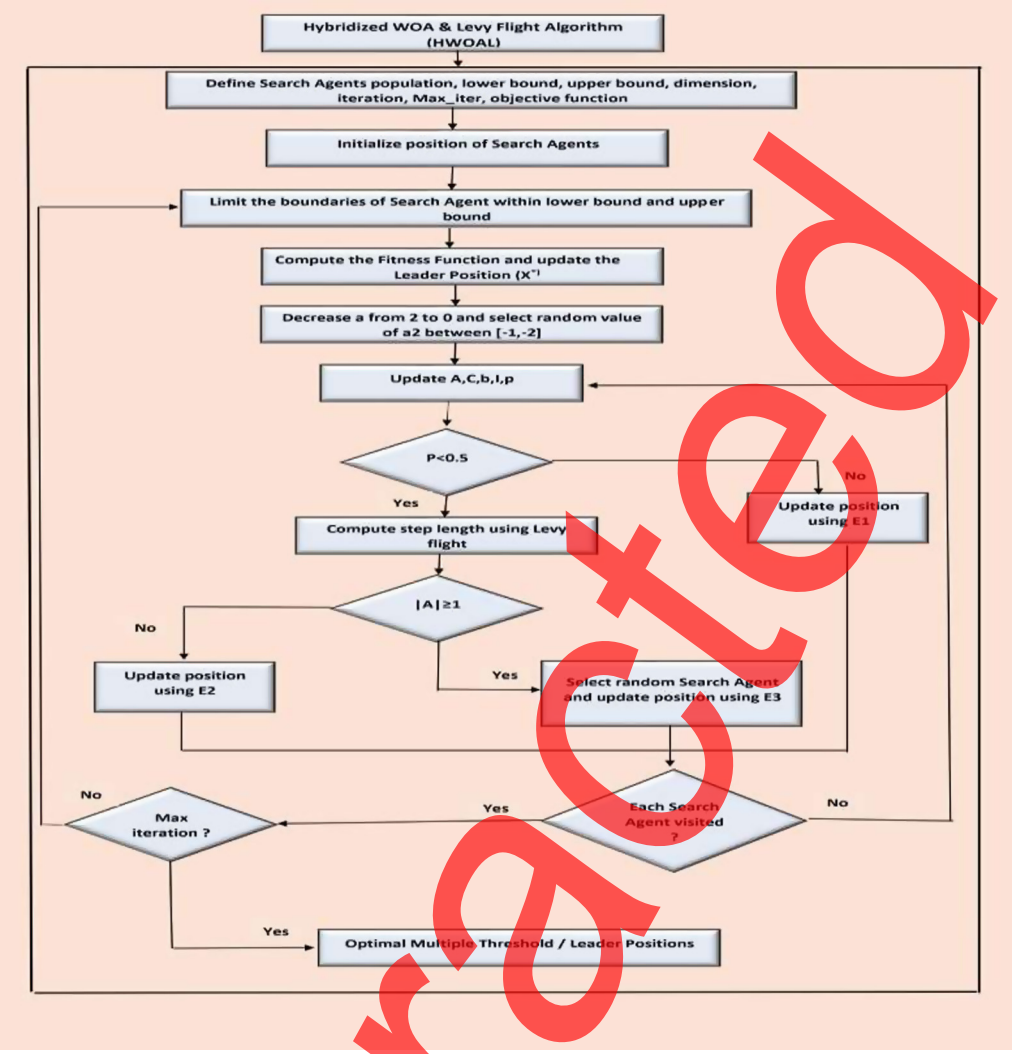

Fig. 3 Flow chart of proposed method: HWOAL.

4 Experiments and Results

This paper introduces hybridization of variant of WOA with Levy flight trajectory (HWOAL). In WOA variant, $a2$ parameter is randomized in the spiral step of WOA. Experiments by all considered algorithms were performed in MATLAB 2016a on a 64-bit Windows 8 machine with Intel Core i3 @ 2 GHz CPU and 8 GB RAM. Parameters and values of each algorithm are listed in Table 1.

4.1 *Performance of HWOAL on Standard Benchmark Problems*

The efficiency of proposed method (HWOAL) and WOA,[8] SSO,[23] SCA,[24] BAT[25] algorithms are tested for each of the 23 well-known benchmark optimization functions (F1–F23). The experiment is carried out in 30 runs. Whales' population size and iterations are set as 30 and 500, respectively, under each run. The optimal value $f_{\min}$ of these 23-benchmark optimization function is quoted in Tables 2–4 and taken from the literature.[8] The best or optimal fitness value under each run computed by every considered algorithm is recorded. Best fitness value over 30 runs, worst fitness value over 30 runs, mean of best fitness value over 30 runs, and standard deviation (Std) of best fitness value over 30 runs are computed and reported in Tables 2–4.

Table 1 Value setting of each algorithm's parameter.

Parameters	Value	Algorithm/method
ℓ	[−1, 1]	WOA[8]
B	1	
$a1$	2 to 0	
$a2$	−1 to −2	
ℓ	[−1,1]	Proposed method HWOAL
B	1	
$a1$	2 to 0	
$a2$	[−1, −2]	
β	1.5	
Υ	1	
C_g	0.55	SSO[23]
C_w	0.95	
C_p	0.75	
a	2	SCA[24]
A	0.5	BAT algorithm[25]
Υ	0.5	
Initial pulse emission rate r_0	0.001	

Tables 2–4 indicate that HWOAL method has achieved better mean, standard deviation for unimodal function F1 to F7 except F2, for multimodal function F9, F10, F12 except F8, F11, F13 and for fixed-dimension multimodal function F15, F17, F18, F19, F20, F23 except F14, F16, F21, F22 as compared with WOA, SSO, SCA, and BAT methods. Tables 2–4 indicate that HWOAL has attained optimal value close to $f_{\min}$ in 15 functions and attained at least second rank in four functions (F2, F11, F21, F22) as compared with other methods. Hence, HWOAL enhances the exploitation capability of WOA for unimodal function (F1-F7) except F2 and enhances exploration capability of multimodal function (F8-F23) except F8, F11, F13, F14, F16, F21, and F22. Rank of considered algorithm is determined by mean and standard deviation value over 30 runs. Experiments have shown that HWOAL is efficient and produces better results in most cases.

The convergence behavior of HWOAL, WOA, SSO, SCA, and BAT for unimodal, multimodal, and fixed-dimension multimodal functions are shown in Fig. 4–19. Average best score indicates the mean of best fitness values obtained over 30 runs.

HWOAL method has achieved faster convergence behavior for unimodal functions F1, F3, F4, F5, F6, F7 except F2 as compared with WOA, SSO, SCA, and BAT methods. Hence, HWOAL enhances the exploitation capability of WOA for unimodal functions except F2.

HWOAL method has achieved faster convergence for multimodal functions F9, F10, F11, F12 except F8 and F13 as compared with WOA, SSO, SCA, and BAT methods.

HWOAL method has shown faster convergence capability for fixed-dimension multimodal functions F15, F17, F18, F19, F20, F23 except F14, F16, F21, and F22 as compared with WOA, SSO, SCA, and BAT methods. Thus, HWOAL enhances the exploration capability of WOA for multimodal functions (F8-F23) except F8, F13, F14, F16, F21, and F22.

Table 2 Fitness value obtained for unimodal function over 30 runs.

Fitness function	Fitness value	WOA	SSO	SCA	BAT	HWOAL	Standard F_{min} for fitness function
*F*1	Best	3.58E-73	66.63271	0.039435	1.15	1.39E-78	0
	Worst	6.46E-66	413.7332	78.36779	9.765132	2.10E-72	
	Mean	3.96E-69	196.5886	17.05463	2.354642	**1.73E-73**	
	Std.	1.39E-70	72.2627	24.28538	3.170699	4.33E-73	
	Rank	2	5	4	3	1	
*F*2	Best	4.59E-60	22.64445	0.000119	0.98	4.12E-43	0
	Worst	1.94E-49	38.78648	0.273677	14.7771	1.03E-39	
	Mean	**1.14E-50**	31.77351	0.029282	5.168878	1.57E-40	
	Std.	4.35E-50	4.12659	0.056539	4.911508	2.43E-40	
	Rank	1	5	3	4	2	
*F*3	Best	12369.43	21439.55	2227.884	1.23	9.18E-50	0
	Worst	72305.43	54160.64	21836.77	239.3373	1.97E-37	
	Mean	40813.38	31576.29	8123.352	17.65816	**1.25E-38**	
	Std.	14252.32	7066.124	4924.717	44.3297	4.75E-38	
	Rank	5	4	3	2	1	
*F*4	Best	0.315638	20.29459	9.868343	0.3249	3.19E-33	0
	Worst	21.19467	36.7803	33.00709	0.891029	8.2E-30	
	Mean	3.66779	28.01121	27.64469	0.475747	**5.65E-31**	
	Std.	4.21594	4.385434	4.32129	0.267751	1.52708E-30	
	Rank	3	5	4	2	1	
*F*5	Best	26.95386	65651.64	58.39609	28.70701	24.16433	0
	Worst	28.71	1220023	786884.8	1239.935	28.13	
	Mean	27.83237	367066.1	79008.53	324.2428	**25.63911**	
	Std.	0.656585	348099.5	170437.2	320.0795	0.429735	
	Rank	2	5	4	3	1	
*F*6	Best	0.076431	121.0129	4.183592	4.41E-4	2.59E-05	0
	Worst	0.992686	394.9743	284.5569	11.05089	0.878497	
	Mean	0.433898	261.4799	21.82515	6.21508	**0.262128**	
	Std.	0.2450	70.58526	51.25678	3.604231	0.2010	
	Rank	2	5	4	3	1	
*F*7	Best	1.93E-05	9.389133	0.014017	5.67E-05	1.67E-05	0
	Worst	0.015632	51.43659	0.716842	106.5623	0.004845	
	Mean	0.003459	26.64258	0.145287	13.59889	**0.001412**	
	Std.	0.004043	11.12958	0.192939	24.85043	0.001014	
	Rank	2	5	3	4	1	

Table 3 Fitness value obtained for multimodal function over 30 runs.

Fitness function	Value	WOA	SSO	SCA	BAT	HWOAL	Standard F_{min} for fitness function
*F*8	Best	−12,569.4	−8118.32	−4216.16	−117.054	−9224.7	−418.9829 x5
	Worst	−8325.84	−6798.19	−3162.08	−115.922	−5657.59	
	Mean	−10914.1	−7389.75	**−3684.42**	−116.598	−6580.65	
	Std.	1741.837	330.5568	244.3589	0.289816	933.1494	
	Rank	5	4	1	2	3	
*F*9	Best	0	152	0.032883	0	0	0
	Worst	5.68E-14	233.3424	215.5125	59.84877	5.78E-14	
	Mean	1.89E-15	183	40.09466	44.28364	**1.76E-15**	
	Std.	1.12E-14	20	16.29255	26.14505	**1.04E-14**	
	Rank	2	5	3	4	1	
*F*10	Best	8.88E-16	6.992379	0.015002	8.88E-16	4.32E-16	0
	Worst	7.99E-15	11.69898	20.33152	4.373819	1.33	
	Mean	3.73E-15	9.169105	14.6485	1.681832	**3.67E-15**	
	Std.	2.7E-15	1.035365	8.765546	1.345951	**1.78E-16**	
	Rank	2	4	5	3	**1**	
*F*11	Best	0	1.026991	0.115496	0	0	0
	Worst	0.172976	1.113118	6.67495	0.414842	0.077715	
	Mean	**0.000191**	1.062125	1.172177	0.107881	0.005346	
	Std.	0.001581	0.021544	1.145998	0.129104	0.016101	
	Rank	**1**	4	5	3	2	
*F*12	Best	0.004571	8.518291	0.479021	0.093389	3.21E-06	0
	Worst	0.083131	42.73567	894457.8	1.258442	0.03245	
	Mean	0.023681	13.79744	61642.71	0.374436	0.007868	
	Std.	0.016459	5.993287	186967.5	0.280795	0.008231	
	Rank	2	4	5	3	**1**	
*F*13	Best	0.125504	6.740981	2.438714	0.337756	0.10973	0
	Worst	0.907314	41176.53	8353455	0.933688	1.12198	
	Mean	0.538561	4028.57	535565.4	0.625236	0.75783	
	Std.	0.185441	9384.341	1732226	0.200583	0.24050	
	Rank	1	4	5	2	3	

HWOAL avoids premature convergence, attains faster convergence for 16 functions toward global optima as shown in Fig. 4–19, achieves promising results (mean, std) and first rank in 15 functions out of 23 functions and attains at least second rank four times for F2, F11, F21, F22, as shown in Tables 2–4. Thus, HWOAL enhances the balance between exploitation and exploration capability of WOA, avoids jumping out of local optima, and makes WOA faster.

Table 4 Fitness value obtained for fixed-dimension multimodal function over 30 runs.

Fitness function	Value	WOA	SSO	SCA	BAT	HWOAL	Standard F_{min} for fitness function
*F*14	Best	0.998004	1.992522	0.998004	12.67051	0.998004	1
	Worst	10.76318	14.1534	2.982105	12.67052	12.67051	
	Mean	2.923478	9.471074	**1.664093**	12.67051	3.454282	
	Std.	3.612162	4.09649	0.948096	2.55E-06	3.982231	
	Rank	2	4	1	5	3	
*F*15	Best	0.000321	0.001493	0.000336	0.000466	0.000307	0.00030
	Worst	0.001565	0.11298	0.001574	0.01833	0.001223	
	Mean	0.000683	0.020439	0.001032	0.006245	**0.000401**	
	Std.	0.000347	0.026508	0.000377	0.004854	0.000182	
	Rank	2	5	3	4	1	
*F*16	Best	−1.03163	−1.03105	−1.03163	−1.03091	−1.03163	−1.0316
	Worst	−1.03163	0.435913	−1.03136	−0.0044	−1.03163	
	Mean	−1.03163	−0.71419	−1.03157	−0.73569	**−1.03163**	
	Std.	5.6106E-09	0.33895686	6.5717E-05	0.31935482	**1.47968E-11**	
	Rank	2	5	3	4	1	
*F*17	Best	0.397887	0.400038	0.39791	0.398395	0.398052	0. 398
	Worst	0.397887	1.893892	0.414974	1.302787	0.397912	
	Mean	0.397887	1.249946	0.400389	0.585417	**0.397998**	
	Std.	2.76E-05	0.37102	0.003635	0.23963	**1.03E-09**	
	Rank	2	5	3	4	1	
*F*18	Best	3	5.138474	3	3.809839	3	3
	Worst	3.000684	97.30669	3.000602	102.7454	3.000381	
	Mean	3.000093	42	3.000086	76.02673	**3.000009**	
	Std.	4.12E-15	28.27224	0.000134	29.299	**3.78E-15**	
	Rank	2	4	3	5	1	
*F*19	Best	−3.8627	−3.85401	−3.862	−3.85679	−3.8627	-3.86
	Worst	−3.74379	−3.06756	−3.85038	−0.9842	−3.85541	
	Mean	−3.85487	−3.4002	−3.85067	−2.62646	**−3.86253**	
	Std.	0.002455	0.175125	0.00334	0.774945	0.001345	
	Rank	2	4	3	5	1	
*F*20	Best	−3.32195	−3.22351	−3.17591	−2.15726	−3.322	−3.32
	Worst	−2.43076	−1.83399	−1.91195	−0.04459	−3.07792	
	Mean	−3.20284	−2.80429	−2.92185	−0.7074	**−3.24642**	
	Std.	0.201794	0.375788	0.254231	0.630825	0.12594	
	Rank	2	4	3	5	1	

Table 4 (*Continued*).

Fitness function	Value	WOA	SSO	SCA	BAT	HWOAL	Standard F_{min} for fitness function
*F*21	Best	−10.1523	−8.04485	−6.63419	−4.82946	−10.1532	−10.1532
	Worst	−2.63046	−0.9614	−0.49649	−1.3995	−2.63047	
	Mean	**−7.49315**	−3.43158	−2.44638	−2.95927	−7.29305	
	Std.	2.68161	2.984104	3.250118	3.056289	2.782262	
	Rank	1	3	5	4	2	
F22	Best	−10.3981	−9.73316	−5.48505	−4.79699	−10.4028	−10.4028
	Worst	−1.83492	−1.47629	−0.90364	−1.65388	−2.7659	
	Mean	**−7.03725**	−3.98053	−3.48802	−3.84904	−6.85212	
	Std.	3.024245	3.566223	3.895837	3.699609	3.378602	
	Rank	1	3	5	4	2	
*F*23	Best	−10.5331	−8.69231	−7.69364	−4.3076	−10.5363	−10.5363
	Worst	−5.41866	−1.35284	−0.93987	−1.58014	−5.42173	
	Mean	−10.0465	−2.90481	−5.54963	−3.99727	**−10.13636**	
	Std.	2.479445	2.912206	2.696529	2.785189	2.39991	
	Rank	2	5	3	4	1	

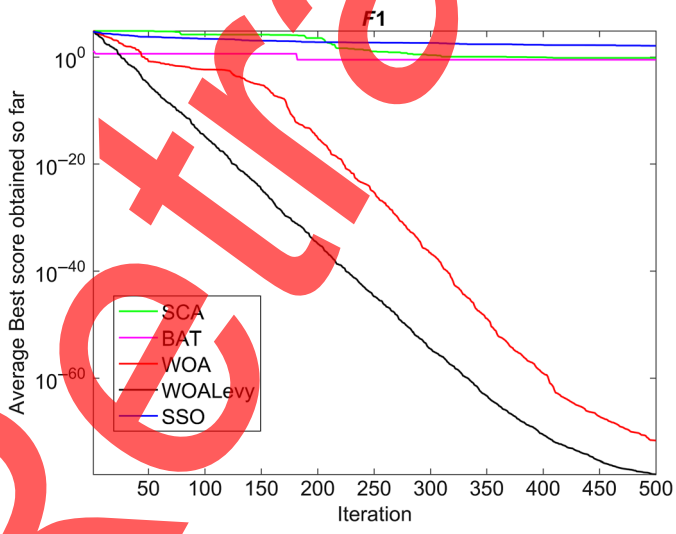

Fig. 4 *F*1 convergence behavior.

4.2 *Performance Analysis of HWOAL for Multilevel Image Segmentation*

HWOAL has shown efficient performance on standard benchmark functions (F1-F23) in most cases as described in Sec. 4.1. Further HWOAL, WOA,[8] SSO,[23] SCA,[24] BAT,[25] and other two recent multilevel thresholding techniques WOA-TH[20] and BWO[21] are utilized to find optimal set of thresholds for multilevel image segmentation on the Berkeley segmentation dataset and benchmark (BSDS300) and SIPI image database.

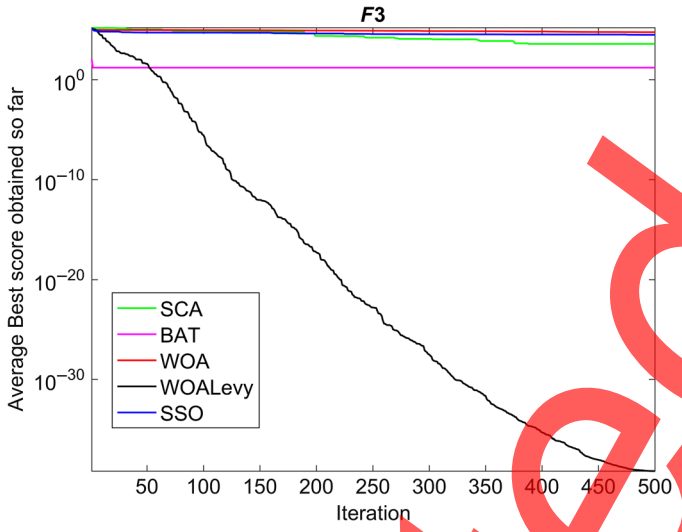

Fig. 5 *F*3 convergence behavior.

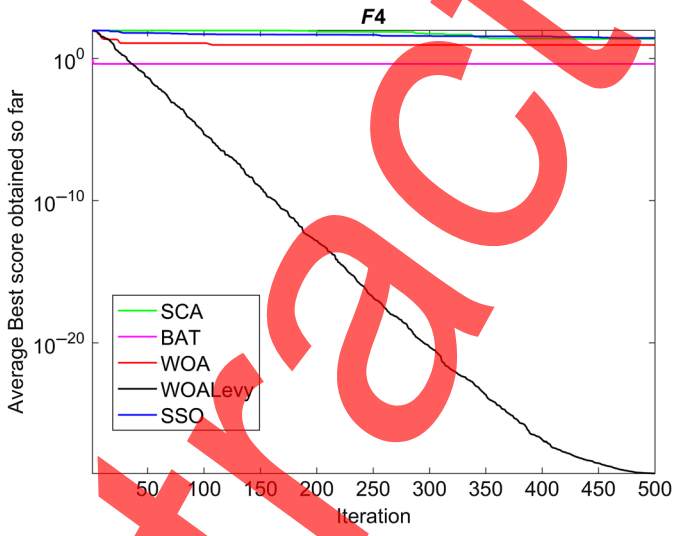

Fig. 6 *F*4 convergence behavior.

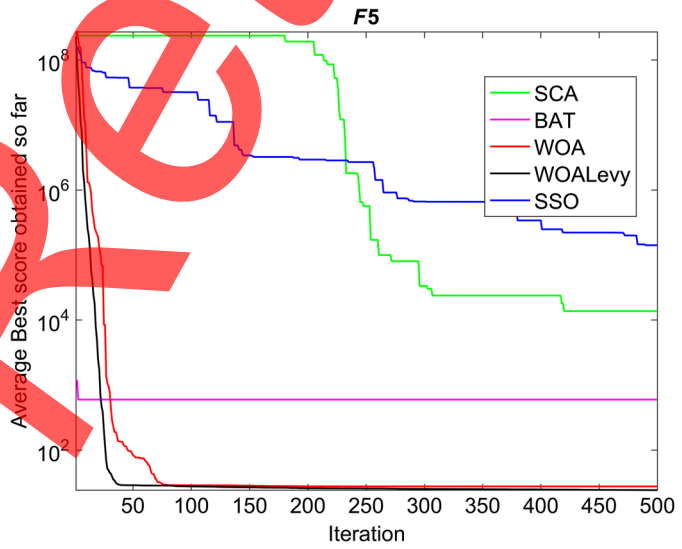

Fig. 7 *F*5 convergence behavior.

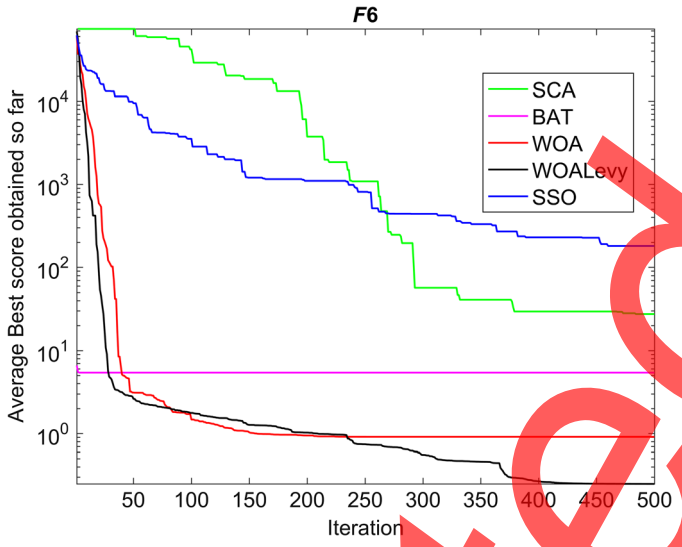

Fig. 8 *F*6 convergence behavior.

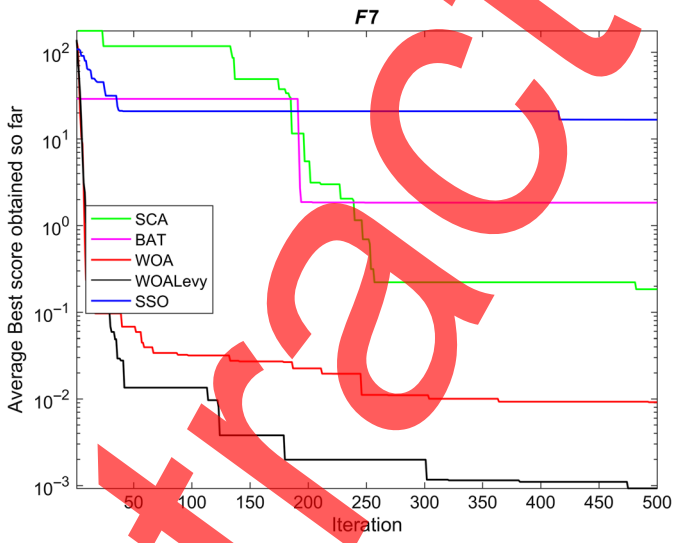

Fig. 9 *F*7 convergence behavior.

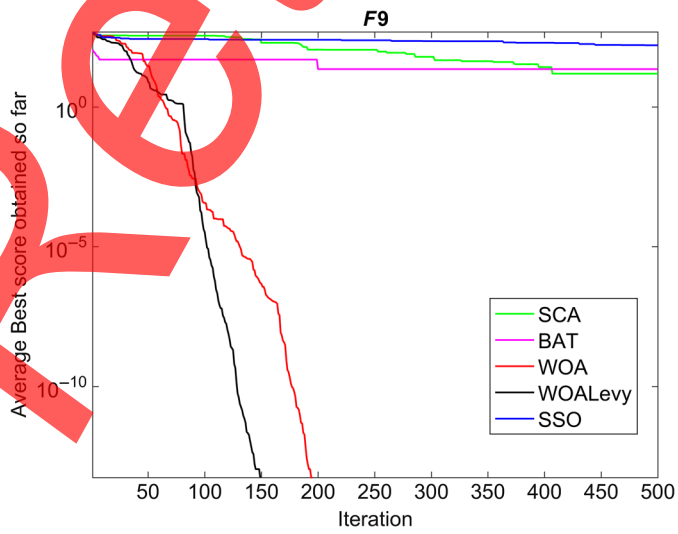

Fig. 10 *F*9 convergence behavior.

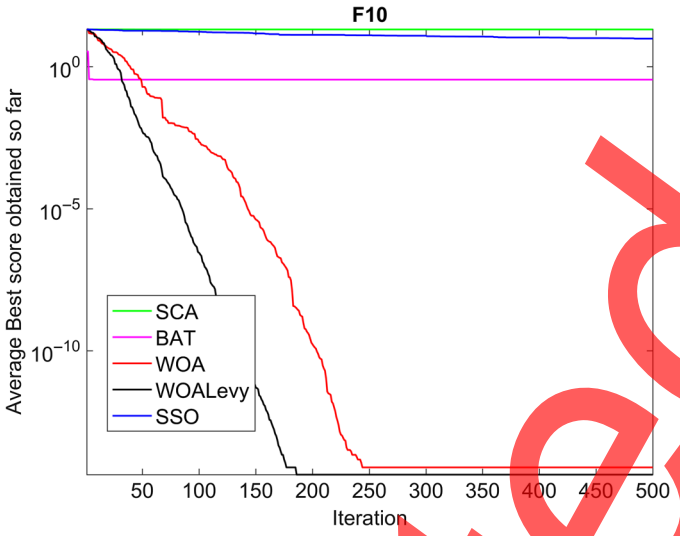

Fig. 11 *F*10 convergence behavior.

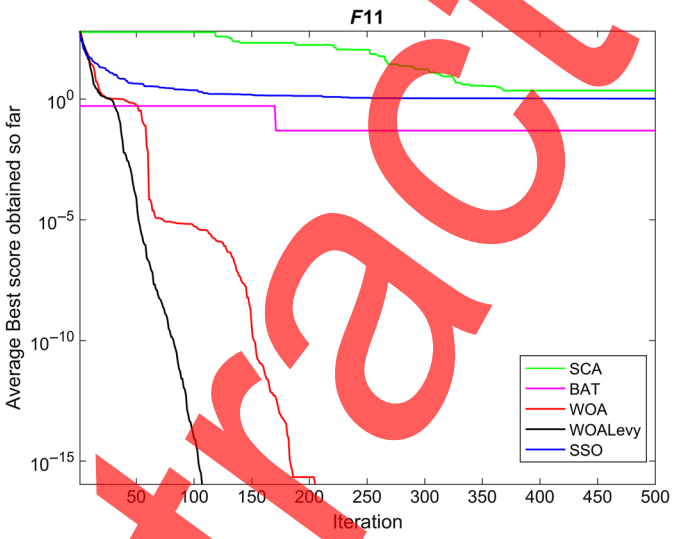

Fig. 12 *F*11 convergence behavior.

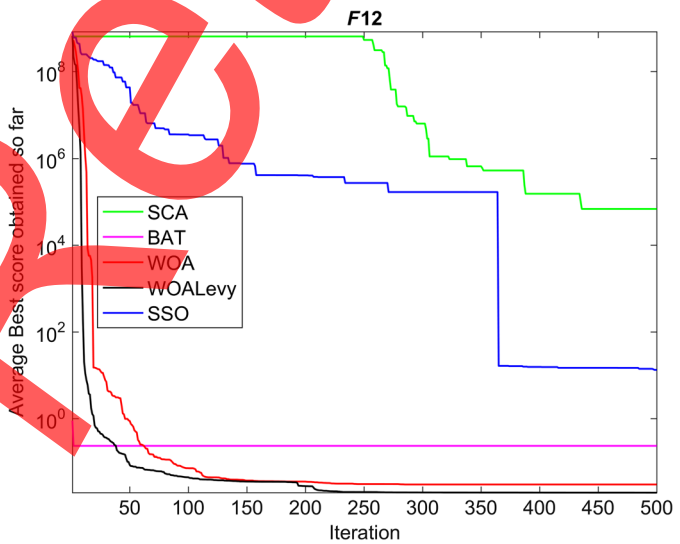

Fig. 13 *F*12 convergence behavior.

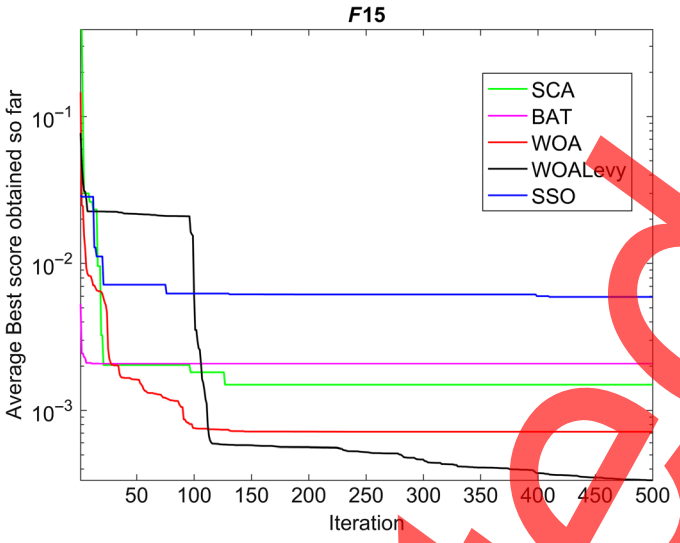

Fig. 14 *F*15 convergence behavior.

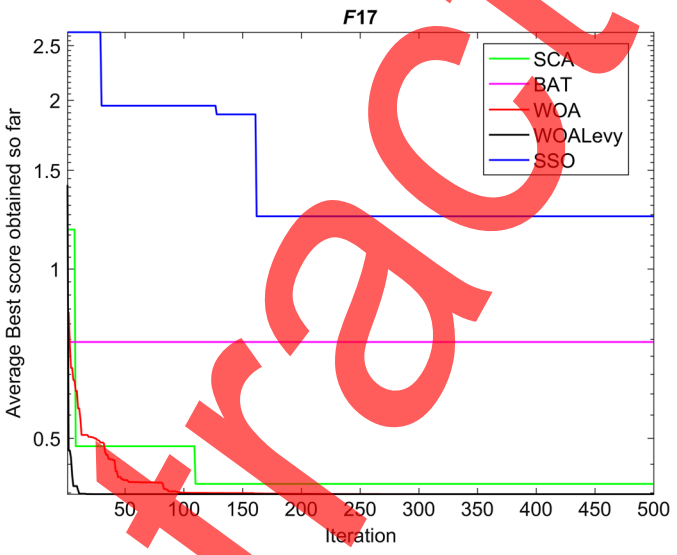

Fig. 15 *F*17 convergence behavior.

4.2.1 *Benchmark Images*

Five images from Berkeley segmentation dataset and Benchmark (BSD 300)[26] and TestImage5 from SIPI image database[27] are randomly selected to evaluate the performance of the proposed method (HWOAL) and WOA,[8] SSO,[23] SCA,[24] BAT,[25] WOA-TH,[20] and BWO[21] at various threshold numbers $k = 2$, 3, 4, and 5 for multilevel image segmentation (As multilevel thresholds achieved previously by Akay,[12] Bhandari et al.,[2] Ewees et al.,[17] Bohot et al.,[20] Houssein et al.[21]). Each image is resized to human compiled ground truths (214×320). Human compiled ground truths are used to compute segmentation metrics (SSIM, PSNR, MSE, and AD) results. Randomly selected images are shown in Fig. 20.

4.2.2 *Experimental setting for multilevel image segmentation*

Experiment is carried out in 30 trials. Population size and iterations were set as 25 and 100 in each trial at various threshold number $k = 2$, 3, 4, and 5. The average computation time, segmentation quality metrics, best and worst fitness values, best optimal thresholds to perform multilevel image segmentation are computed by HWOAL and compared with other six image segmentation methods: WOA,[8] SSO,[23] SCA,[24] BAT algorithm,[25] WOA-TH,[2],0 and BWO.[21]

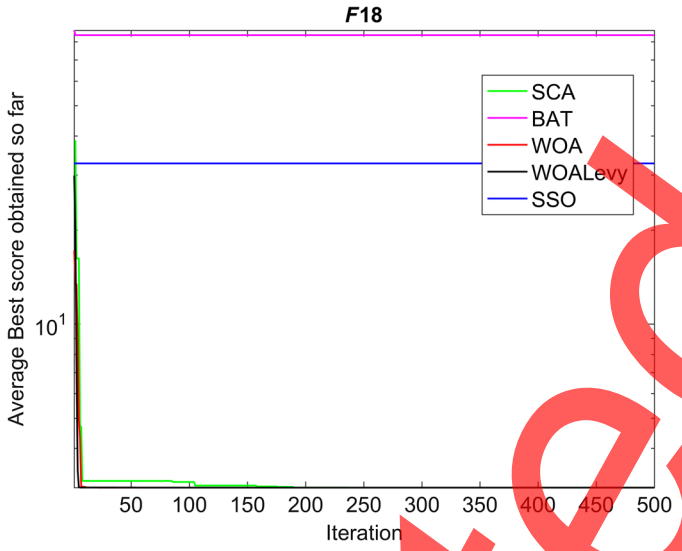

Fig. 16 *F*18 convergence behavior.

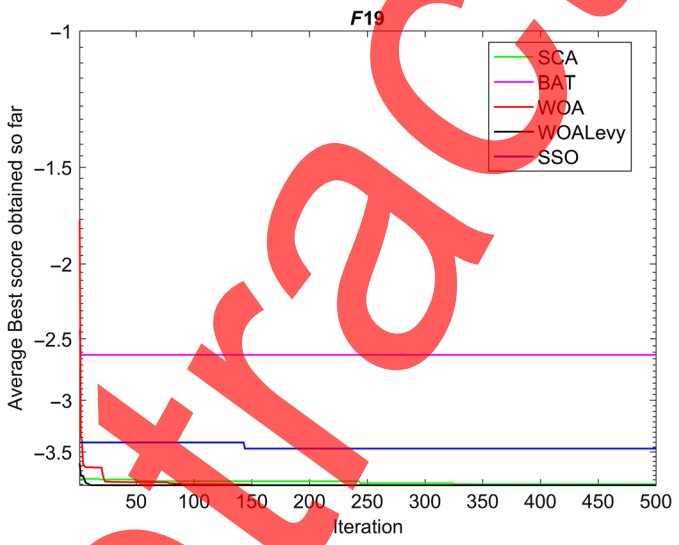

Fig. 17 *F*19 convergence behavior.

4.3 *Segmentation Quality Metrics*

Quantitative metrics of segmented image are described as follows:[28–30]

(i) MSE: It is average squared difference between the ground truth image (G) and segmented image (S) and expressed as

$$\mathrm{MSE} = \frac{1}{XY}\sum_{p=1}^{X}\sum_{q=1}^{Y}|G(p,q) - S(p,q)|^2. \tag{29}$$

(ii) PSNR: PSNR is represented as ratio between maximum power of signal and MSE and mathmatically expressed as

$$\mathrm{PSNR} = 10\log_{10}\frac{S^2}{\mathrm{MSE}}. \tag{30}$$

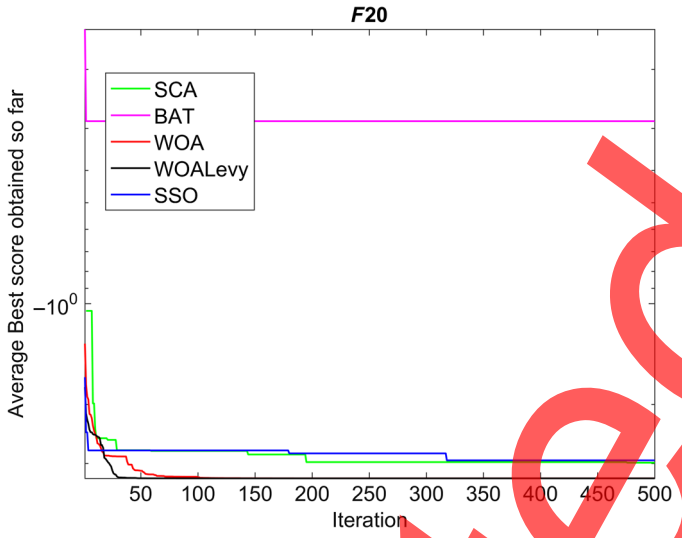

Fig. 18 *F*20 convergence behavior.

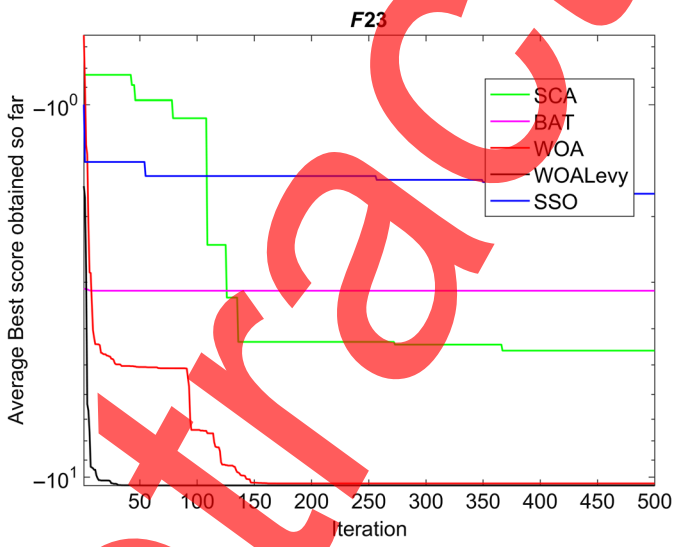

Fig. 19 *F*23 convergence behavior.

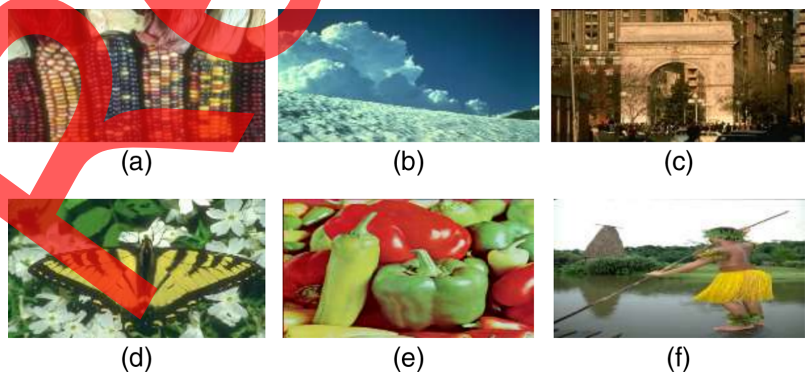

Fig. 20 Benchmark sample images (BSD300), from top-left (a) TestImage1, (b) TestImage2, (c) TestImage3, (d) TestImage4, (e) TestImage5, and (f) TestImage6.

Table 5 Best and worst fitness values of considered algorithms at various thresholds over 30 runs.

Image name	Threshold number (k)	Best fitness values (Max)						
		WOA	HWOAL	SSO	SCA	BAT	WOA-TH[20]	BWO[21]
TestImage1	2	3478	3478	3478	3478	3470	3478	3478
	3	3633	3633	3630	3630	3623	3633	3633
	4	3708	**3710**	3707	3705	3698	3708	3709
	5	3750	**3752**	3750	3742	3739	3751	3751
	k	**Worst fitness values (min)**						
	2	3478	3478	3476	3476	3359	3478	3478
	3	3633	**3633**	3628	3616	3570	3633	3633
	4	3705	**3710**	3700	3693	3605	3706	3707
	5	3749	**3752**	3748	3703	3668	3750	3751
TestImage2	***k***	**Best fitness values (max)**						
	2	3213	3213	3213	3213	3211	3213	3213
	3	3330	**3332**	3330	3326	3313	3331	3331
	4	3391	**3393**	3387	3389	3366	3392	3391
	5	3435	**3436**	3433	3429	3410	3435	3435
	k	**Worst fitness values (min)**						
	2	3213	3213	3212	3212	3116	3213	3213
	3	3329	**3332**	3328	3325	3215	3330	3331
	4	3387	**3393**	3377	3331	3338	3390	3392
	5	3432	**3436**	3427	3387	3354	3433	3434
TestImage3	***k***	**Best fitness values (max)**						
	2	3951	3951	3948	3950	3950	3951	3951
	3	4165	4165	4162	4162	4134	4165	4165
	4	4262	**4263**	4261	4260	4215	4262	4262
	5	4312	**4313**	4310	4300	4260	4313	4313
	k	**Worst fitness values (min)**						
	2	3948	3951	3945	3950	3850	3949	3949
	3	4163	**4165**	4160	4158	4066	4164	4164
	4	4261	**4263**	4258	4246	4120	4262	4262
	5	4309	**4313**	4307	4288	4243	4311	4312

Table 5 (*Continued*).

Image name	Threshold number (k)	Best fitness values (Max)						
		WOA	HWOAL	SSO	SCA	BAT	WOA-TH[20]	BWO[21]
TestImage4	***k***	**Best fitness values (max)**						
	2	3802	3802	3802	3801	3800	3802	3802
	3	3914	3914	3912	3909	3899	3914	3914
	4	3973	**3976**	3970	3968	3940	3974	3974
	5	4014	**4016**	4013	4007	4005	4015	4015
	k	**Worst fitness values (min)**						
	2	3800	**3802**	3800	3800	3758	3800	3800
	3	3914	3914	3910	3904	3797	3914	3914
	4	3971	**3976**	3967	3963	3903	3974	3973
	5	4012	**4016**	4004	3986	3966	4014	4015
TestImage5	***k***	**Best fitness values (max)**						
	2	2437	2437	2432	2436	2429	2437	2437
	3	2588	**2590**	2581	2578	2580	2588	2588
	4	2654	2657	2653	2625	2643	2656	2657
	5	2696	**2698**	2694	2652	2695	2696	2697
	k	**Worst fitness values (Min)**						
	2	2436	2436	2430	2434	2427	2436	2436
	3	2586	**2589**	2580	2576	2579	2586	2589
	4	2652	2655	2652	2622	2640	2654	2655
	5	2695	**2697**	2692	2650	2693	2695	2696
TestImage6	***k***	**Best fitness values (max)**						
	2	5070	**5078**	5049	5063	5061	5074	5069
	3	5231	**5255**	5227	5160	5221	5238	5240
	4	5310	**5324**	5291	5297	5279	5316	5318
	5	5336	**5361**	5332	5334	5312	5355	5357
	k	**Worst fitness values (min)**						
	2	5070	**5077**	5049	5061	5060	5072	5067
	3	5229	**5253**	5226	5158	5217	5236	5238
	4	5308	**5323**	5290	5295	5276	5314	5316
	5	5333	**5359**	5331	5332	5309	5352	5356

Table 6 Best threshold values computed by various methods over 30 runs at various threshold number (k).

Image	k	WOA	HWOAL	SSO	SCA
TestImage1	2	92 167	92 167	91 168	91 162
	3	77 129 188	77 129 187	76 128 186	73 126 185
	4	66 105 151 200	67 105 151 200	60 93 139 193	67 108 146 193
	5	60 95 131 181 213	43 92 128 168 210	56 87 123 163 207	59 97 125 176 217
	k	**BAT**	**WOA-TH[20]**	**BWO[21]**	
	2	102 171	92 167	91 162	
	3	65 120 200	75 128 191	77 129 188	
	4	65 124 159 214	65 104 150 203	63 112 162 204	
	5	58 101 143 181 210	43 92 128 168 210	52 94 123 176 214	
TestImage2	**k**	**WOA**	**HWOAL**	**SSO**	**SCA**
	2	95 166	95 166	95 165	96 166
	3	92 145 192	91 146 193	92 144 190	97 141 194
	4	60 99 146 193	60 98 147 194	60 99 149 198	64 107 159 207
	5	60 95 131 181 213	60 94 130 166 204	58 93 132 166 204	48 83 132 155 202
	k	**BAT**	**WOA-TH[20]**	**BWO[21]**	
	2	96 167	95 166	95 166	
	3	85 127 180	92 144 190	92 145 192	
	4	71 103 141 177	69 99 121 169	71 98 131 187	
	5	59 84 137 185 210	48 83 123 186 211	43 95 144 176 206	
TestImage3	**k**	**WOA**	**HWOAL**	**SSO**	**SCA**
	2	63 145	63 145	64 145	62 146
	3	48 96 161	46 97 161	46 96 160	44 100 165
	4	45 88 138 189	43 88 138 189	43 88 134 182	50 89 139 200
	5	58 101 143 181 210	43 79 138 168 205	45 67 103 149 194	46 64 109 155 200
	k	**BAT**	**WOA-TH[20]**	**BWO[21]**	
	2	61 139	63 145	63 145	
	3	69 126 196	67 127 198	18 134 178	
	4	51 85 171 220	45 88 138 189	45 88 138 189	
	5	36 74 110 131 195	37 85 120 174 208	33 87 111 154 198	

Table 6 (*Continued*).

Image	*k*	WOA	HWOAL	SSO	SCA
TestImage4	***k***	**WOA**	**HWOAL**	**SSO**	**SCA**
	2	85 155	85 155	84 155	81 156
	3	82 145 200	82 145 200	80 136 190	77 149 201
	4	77 121 167 206	76 121 167 207	65 97 148 200	67 108 151 201
	5	59 89 128 171 208	60 89 128 171 208	55 85 122 165 206	54 90 130 181 212
	k	**BAT**	**WOA-TH**[20]	**BWO**[21]	
	2	73 165	72 166	75 178	
	3	82 142 196	80 133 186	81 125 179	
	4	17 88 155 209	19 76 145 198	17 74 133 187	
	5	52 84 121 169 218	48 77 129 178 212	49 76 131 178 211	
TestImage5	***k***	**WOA**	**HWOAL**	**SSO**	**SCA**
	2	70 137	70 137	71 137	71 138
	3	65 121 168	64 123 165	65 123 187	62 118 169
	4	58 69 116 165	39 58 116 165	47 79 129 188	47 93 147 189
	5	58 84 116 142 167	25 58 116 165 188	16 76 140 245 251	19 87 103 140 233
	k	**BAT**	**WOA-TH**[20]	**BWO**[21]	
	2	75 138	54 151	70 137	
	3	70 118 165	77 118 175	63 127 176	
	4	58 116 136 165	58 116 136 165	58 116 142 167	
	5	9 19 72 194 231	58 107 116 165 183	6 58 116 128 165	
TestImage6	***k***	**WOA**	**HWOAL**	**SSO**	**SCA**
	2	101 190	103 178	101 198	96 179
	3	107 146 187	91 136 221	107 168 230	83 152 212
	4	82 134 170 198	82 134 198 217	59 112 187 211	82 13 161 198
	5	17 82 134 152 198	4 36 82 134 198	11 45 89 110 212	10 129 230 239 253
	k	**BAT**	**WOA-TH**[20]	**BWO**[21]	
	2	103 197	95 172	106 198	
	3	56 134 187	87 132 210	67 176 212	
	4	18 133 223 241	82 134 177 198	82 134 198 246	
	5	6 34 150 234 246	4 45 77 146 243	15 24 86 127 252	

(iii) SSIM: It finds the similar structural variation between ground truth image (G) and segmented image (S). It is expressed as Eq. (31)

$$\mathrm{SSIM}(G,S) = (2\mu_{G,S} * \mu_{G,S} \ + c1)(2\sigma_{G,S} + c2)/({\mu_o}^2 + {\mu_1}^2 + C_1)({\sigma_o}^2 + {\sigma_1}^2 + C_2). \tag{31}$$

(iv) AD: It is represented as difference between segmented image (S) and ground truth image (G). It is expressed as

$$\mathrm{AD} = \frac{1}{XY}\sum_{p=1}^{X}\sum_{q=1}^{Y}|S(p,q) - G(p,q)|. \tag{32}$$

Table 7 Average PSNR of various segmentation methods.

		Average PSNR values						
Image name	Threshold number (k)	WOA	HWOAL	SSO	SCA	BAT	WOA-TH[20]	BWO[21]
TestImage1	2	13.79	13.79	13.78	13.79	13.31	**13.83**	13.79
	3	15.46	15.46	15.45	15.32	15.30	**15.48**	15.41
	4	16.93	**16.97**	16.70	16.46	16.22	16.88	16.92
	5	18.17	18.20	18.17	17.62	17.49	18.19	**18.22**
TestImage2	2	14.06	14.06	14.04	14.03	13.90	14.06	14.06
	3	14.85	**14.87**	14.86	14.84	14.84	14.86	14.85
	4	18.12	18.26	17.09	16.87	16.49	**18.28**	18.25
	5	18.92	**19.12**	18.99	18.45	17.87	19.09	19.11
TestImage3	2	15.51	15.51	15.46	15.50	14.97	15.61	**15.64**
	3	17.92	**17.97**	17.83	17.84	16.70	17.90	17.93
	4	19.81	**19.82**	19.81	19.58	18.34	19.80	19.81
	5	21.24	21.27	21.13	20.86	19.30	21.25	21.26
TestImage4	2	14.28	14.28	14.26	14.26	**14.35**	14.31	14.33
	3	15.38	**15.41**	15.39	15.32	14.75	14.98	14.84
	4	16.42	16.50	**16.55**	16.39	16.12	16.44	16.46
	5	19.00	**19.04**	18.45	17.55	17.45	19.02	19.02
TestImage5	2	15.76	16.22	15.98	16.26	15.29	16.26	**16.30**
	3	18.32	**18.43**	18.12	17.76	17.16	18.40	18.39
	4	20.87	**20.95**	20.23	18.37	18.23	20.92	20.87
	5	22.43	**22.64**	21.11	20.98	20.92	22.51	22.46
TestImage6	2	13.65	13.68	12.99	**13.89**	12.79	13.85	13.83
	3	18.11	18.13	17.11	17.23	16.72	18.15	**18.17**
	4	19.76	19.95	19.32	19.25	18.11	19.95	**19.98**
	5	20.11	**20.69**	19.46	19.35	19.39	20.43	20.61

4.4 *Experiment Results*

Best and worst fitness values for each image are computed by considered algorithms over 30 runs at various threshold number k and results are displayed in Table 5. The best values are represented in bold text.

Best or optimal threshold values for each image are computed by all considered algorithms over 30 runs at threshold number $k = 2$, 3, 4, and 5 and results are shown in Table 6.

SSIM, PSNR, MSE, and Avg. difference are important metrics to evaluate segmentation accuracy measures and performance. These metrics are computed for each algorithm's based segmented image at various threshold number $k = 2$, 3, 4, and 5 over 30 runs. Tables 7–10 indicate the average segmentation metrics as average of 30 runs.

Table 8 Average SSIM of various segmentation methods.

		Average SSIM values						
Image name	Threshold number (k)	WOA	HWOAL	SSO	SCA	BAT	WOA-TH[20]	BWO[21]
TestImage1	2	0.4430	0.4430	0.4415	0.4429	0.4043	**0.4434**	0.4431
	3	0.5400	**0.5403**	0.5391	0.5313	0.5332	0.5401	0.5398
	4	0.6170	**0.6197**	0.6046	0.5922	0.5786	0.6179	0.6192
	5	0.6740	**0.6752**	0.6745	0.6507	0.6495	0.6745	0.6751
TestImage2	2	0.4140	0.4143	0.4141	0.4140	0.3856	0.4139	0.4140
	3	0.4630	**0.4639**	0.4628	0.4631	0.4398	0.4628	0.4630
	4	0.6320	0.6452	0.5810	0.5656	0.5366	**0.6457**	0.6454
	5	0.6620	**0.6697**	0.6642	0.6594	0.6164	0.6690	0.6695
TestImage3	2	0.5340	0.5345	0.5368	0.5364	0.5345	0.5366	0.5370
	3	0.6450	0.6451	0.6454	**0.6462**	0.5779	0.6450	0.6451
	4	0.6690	**0.6799**	0.6708	0.6701	0.6334	0.6690	0.6690
	5	0.7050	0.7057	0.7033	**0.7109**	0.6944	0.7053	0.7055
TestImage4	2	0.5280	0.5285	0.5281	**0.5296**	0.5228	0.5290	0.5293
	3	0.5540	**0.5567**	0.5554	0.5542	0.5346	0.5392	0.5399
	4	0.6100	0.6141	**0.6166**	0.6077	0.6015	0.6112	0.6128
	5	0.7080	**0.7090**	0.6842	0.6439	0.6369	0.7083	0.7083
TestImage5	2	0.7412	0.7505	0.6900	0.6028	0.6226	0.7498	**0.7585**
	3	0.7929	**0.8067**	0.7041	0.7581	0.6546	0.8016	0.8023
	4	0.8287	**0.8511**	0.7345	0.7989	0.7321	0.8498	0.8478
	5	0.8445	**0.8776**	0.7701	0.8013	0.7541	0.8713	0.8710
TestImage6	2	0.6212	0.6110	0.5902	**0.6321**	0.5546	0.6047	0.6019
	3	0.7331	0.7402	0.6820	0.6743	0.6432	0.7391	**0.7412**
	4	0.8000	0.8008	0.7511	0.7112	0.7865	0.8003	**0.8011**
	5	0.8053	**0.8259**	0.8035	0.7512	0.7948	0.8212	0.8216

Table 9 Average MSE of various segmentation methods.

		Average MSE values						
Image name	Threshold number (k)	WOA	HWOAL	SSO	SCA	BAT	WOA-TH[20]	BWO[21]
TestImage1	2	2714	2714	2719	2714	3032	**2708**	2713
	3	1846	1846	1853	1912	1924	**1839**	1851
	4	1315	**1304**	1394	1470	1584	1319	1313
	5	990	984	989	1130	1171	987	**978**
TestImage2	2	2551	2551	2560	2568	2671	2551	2552
	3	2119	**2115**	2117	2127	2147	2117	2119
	4	1002	968	1339	1418	1526	**963**	970
	5	835	**795**	821	944	1117	801	797
TestImage3	2	1825	1825	1848	1831	2084	1823	**1819**
	3	1040	**1037**	1070	1067	1390	1044	1039
	4	679	**677**	678	715	966	682	680
	5	487	**484**	501	533	765	482	482
TestImage4	2	2421	2421	2426	2430	**2394**	2419	2416
	3	1880	**1867**	1878	1913	2198	1963	1978
	4	1481	1453	**1437**	1494	1567	1476	1467
	5	815	**810**	955	1191	1262	813	812
TestImage5	2	1978	1897	1934	1876	1946	1877	**1856**
	3	1367	**1327**	1384	1396	1412	1342	1346
	4	997	**975**	1011	1123	1143	988	996
	5	597	**567**	613	657	668	574	589
TestImage6	2	2698	2691	2712	**2654**	2733	2671	2679
	3	1945	1939	2052	2043	2176	1932	**1917**
	4	787	767	801	812	833	769	**754**
	5	511	**461**	533	551	546	496	472

Tables 11–14 indicate the minimum and maximum computation time taken by each algorithm to perform multilevel image segmentation of each image over 30 runs at threshold number $k = 2, 3, 4, 5$. TestImage1 to TestImage6 are renamed as TI1 to TI6, respectively.

Average computation time to perform multilevel image segmentation at threshold number $k = 2, 3, 4, 5$ of each image taken by each algorithm as average of 30 runs are illustrated in Table 15.

Figures 21 and 22 show six-level gray color image segmentation of TestImage2, TestImage3, and their respective histogram. Optimal threshold values are computed by HWOAL and other considered methods are used to perform segmentation. It can be observed from histogram that HWOAL-based segmented image as shown for TestImage2 and TestImage3 has better visual quality and greater number of pixel intensities that belong to foreground regions at $k = 5$. TestImage2 that is segmented using HWOAL has good visual quality, clearly defined boundaries and is well segmented at $k = 5$.

Table 10 Average avg. difference of various segmentation methods.

Average avg. difference values								
Image name	Threshold number (k)	WOA	HWOAL	SSO	SCA	BAT	WOA-TH[20]	BWO[21]
TestImage1	2	47.23	47.23	47.27	47.21	49.39	**47.11**	47.18
	3	38.16	38.16	38.26	38.56	38.66	**38.08**	38.69
	4	31.31	**31.17**	32.21	33.17	34.33	31.21	31.28
	5	26.41	**26.33**	26.39	28.40	28.98	26.37	26.39
TestImage2	2	45.72	45.72	45.87	45.97	45.76	45.73	45.72
	3	40.67	**40.49**	40.60	40.83	40.80	40.66	40.58
	4	26.59	26.50	30.49	31.37	32.38	**26.46**	26.53
	5	23.81	**23.11**	23.47	25.68	27.28	23.34	23.42
TestImage3	2	35.66	35.67	35.84	35.72	37.73	35.65	**35.61**
	3	26.86	**26.72**	27.18	27.10	30.67	26.82	26.89
	4	22.22	**22.19**	22.17	22.50	25.25	22.23	22.21
	5	18.71	**18.69**	18.94	19.08	22.32	18.73	18.73
TestImage4	2	44.32	44.32	44.36	44.44	43.58	44.30	**44.27**
	3	37.94	**37.85**	37.98	38.43	40.23	38.98	40.11
	4	32.77	32.47	**32.34**	32.94	33.77	32.59	32.53
	5	23.44	**23.34**	25.16	28.09	28.39	23.40	23.38
TestImage5	2	39.59	39.54	39.61	**39.48**	39.63	39.56	39.55
	3	30.49	30.49	30.76	30.51	30.55	30.48	**30.45**
	4	20.48	**20.37**	20.68	20.64	20.71	20.41	20.43
	5	16.67	**16.54**	16.71	16.75	16.92	16.61	16.64
TestImage6	2	46.48	46.52	47.13	**46.19**	47.20	46.29	46.23
	3	38.67	38.64	39.11	39.02	39.49	38.59	**38.51**
	4	32.41	32.32	32.48	32.67	32.86	32.36	**32.25**
	5	20.11	**21.78**	22.34	22.42	22.39	21.98	21.85

Table 11 Minimum and maximum elapsed time taken by each thresholding method to perform three-level segmentation.

	Images											
	Minimum computation time(s) at $k = 2$						Maximum computation time(s) at $k = 2$					
Methods	TI1	TI2	TI3	TI4	TI5	TI6	TI1	TI2	TI3	TI4	TI5	TI6
WOA	11.02	**10.57**	10.55	11.50	10.83	10.98	12.40	12.64	12.09	12.55	12.21	12.13
HWOAL	11.71	10.89	**10.48**	11.81	10.68	**10.71**	**12.37**	11.43	**11.68**	**11.93**	12.01	11.90
SSO	11.89	10.96	10.76	11.92	11.02	11.12	12.93	**11.15**	11.98	11.96	12.68	12.10
SCA	**10.76**	10.85	10.70	**10.89**	10.98	11.04	12.39	11.20	11.78	12.18	11.97	11.79
BAT	10.90	10.95	10.98	11.03	10.96	11.12	13.67	12.03	12.39	12.59	12.01	12.24
WOA–TH[20]	10.97	10.62	10.52	11.43	**10.45**	10.78	12.40	11.22	11.79	11.99	**11.43**	11.89
BWO[21]	10.93	10.60	10.50	11.36	10.78	10.74	12.39	11.19	11.72	11.96	11.74	**11.76**

Table 12 Minimum and maximum elapsed time taken by each thresholding method to perform four-level segmentation.

	Images											
	Minimum computation time(s) at $k = 3$						Maximum computation time(s) at $k = 3$					
Method	TI1	TI2	TI3	TI4	TI5	TI6	TI1	TI2	TI3	TI4	TI5	TI6
WOA	11.10	**10.63**	10.53	11.68	11.14	11.22	12.49	12.68	12.32	12.90	12.23	12.32
HWOAL	11.46	10.70	**10.47**	**11.62**	11.01	10.98	**12.25**	**11.39**	**11.76**	**11.95**	12.11	11.67
SSO	11.59	11.12	11.55	12.00	11.98	11.15	12.73	12.14	12.28	12.46	12.54	12.01
SCA	10.90	11.10	11.10	11.98	11.76	11.12	12.45	11.59	12.01	12.30	12.49	11.98
BAT	**10.89**	10.78	11.12	12.28	11.81	11.12	13.03	11.98	12.68	12.96	12.51	12.20
WOA-TH[20]	11.03	10.68	10.50	11.64	**10.98**	10.95	12..33	11.49	11.88	11.99	**12.05**	11.56
BWO[21]	10.98	10.70	10.49	11.66	11.01	**10.87**	12.29	11.46	11.79	11.97	12.09	**11.49**

Table 13 Minimum and maximum elapsed time taken by each thresholding method to perform five-level segmentation.

	Images											
	Minimum computation time(s) at $k = 4$						Maximum computation time(s) at $k = 4$					
Methods	TI1	TI2	TI3	TI4	TI5	TI6	TI1	TI2	TI3	TI4	TI5	TI6
WOA	11.12	10.57	10.76	11.59	11.03	11.20	12.90	11.90	12.36	13.05	12.11	12.13
HWOAL	11.68	**10.43**	**10.59**	**11.55**	10.97	10.94	12.62	**11.34**	**11.45**	**12.93**	12.01	11.53
SSO	11.86	11.03	10.87	11.87	11.90	11.12	12.98	11.15	11.65	13.00	12.34	11.94
SCA	11.87	11.04	10.75	11.74	11.69	11.10	**12.59**	11.11	11.56	12.99	12.29	11.81
BAT	12.00	11.14	10.88	12.12	11.72	11.08	13.01	12.42	11.84	13.09	12.30	11.96
WOA-TH[20]	11.08	10.49	10.67	11.57	**10.91**	10.90	12.66	11.54	11.53	12.97	**11.98**	11.54
BWO[21]	**11.03**	10.52	10.63	11.57	10.93	**10.83**	12.69	11.58	11.55	12.98	12.01	**11.46**

Figures 23–26 show five-level RGB color image segmentation of TestImage1 and TestImage4 through TestImage6 by all considered methods.

Figures 27–30 show six-level RGB color image segmentation of TestImage1 and TestImage4 through TestImage6 by all considered methods.

Figures 31–34 show five-level gray color image segmentation of TestImage1 and TestImage4 through TestImage6 by all considered methods.

Figures 35–38 show six-level gray color image segmentation of TestImage1 and TestImage4 through TestImage6 by all considered methods.

5 Experiment Results Discussion

Each algorithm has computed best and worst fitness values, optimal threshold values at threshold number ($k = 2$, 3, 4, and 5) are computed for every considered image by all algorithm, quantitative segmentation metrics computed for the proposed method and other algorithms based segmented images, minimum computation time, maximum computation time, average

Table 14 Minimum and maximum elapsed time taken by each thresholding method to perform six-level segmentation.

	Images											
	Minimum computation time(s) at $k = 5$						Maximum computation time(s) at $k = 5$					
Methods	TI1	TI2	TI3	TI4	TI5	TI6	TIM1	TI2	TI3	TI4	TI5	TI6
WOA	11.14	11.30	11.60	11.76	11.06	11.18	12.68	13.00	12.19	13.10	12.10	12.11
HWOAL	**11.10**	**11.00**	**11.01**	**11.67**	10.95	10.97	**12.40**	**12.31**	**11.62**	**12.72**	11.98	11.49
SSO	11.20	11.07	11.65	12.06	11.92	11.14	12.76	12.76	12.03	12.99	12.23	11.87
SCA	11.16	11.05	11.09	11.96	11.72	11.20	12.54	12.60	11.64	12.91	12.21	11.78
BAT	11.25	11.25	11.17	12.18	11.78	11.22	12.92	12.82	12.34	13.20	12.27	11.90
WOA-TH[20]	11.12	11.24	11.04	11.71	**10.92**	10.96	12.48	12.34	11.63	12.79	**11.93**	11.53
BWO[21]	11.08	11.28	11.06	11.72	10.96	**10.89**	12.52	12.38	11.64	12.81	11.99	**11.44**

Table 15 Average elapsed time taken by each thresholding method to perform segmentation.

	Images											
Methods	TI1	TI2	TI3	TI4	TI5	TI6	TI1	TI2	TI3	TI4	TI5	TI6
	Average computation time(s) at *k*= 2						**Average computation time(s) at *k*= 3**					
WOA	11.47	11.67	11.54	11.98	11.33	11.36	11.63	11.52	11.12	12.23	11.44	11.62
HWOAL	12.02	11.04	**10.77**	11.84	11.21	**11.11**	11.57	**10.92**	**10.98**	**11.73**	11.32	11.17
SSO	12.23	11.05	11.53	11.99	11.38	11.40	12.03	11.98	11.96	12.20	12.12	11.52
SCA	**11.39**	**11.02**	11.50	**11.62**	11.35	11.34	**11.55**	11.24	11.49	12.12	12.02	11.50
BAT	12.01	11.85	11.56	11.67	11.36	11.41	12.12	11.31	12.42	12.44	12.10	11.50
WOA-TH[20]	11.43	11.52	10.96	11.77	**10.78**	11.14	11.58	10.94	11.03	11.79	**11.12**	11.10
BWO[21]	11.45	11.47	10.92	11.73	10.96	11.13	11.56	10.94	11.01	11.77	11.28	**11.01**
	Average computation time(s) at *k*= 4						**Average computation time(s) at *k*= 5**					
WOA	11.69	10.98	11.59	12.22	11.46	11.49	11.97	12.25	11.96	12.33	11.45	11.52
HWOAL	11.93	**10.78**	**10.94**	**12.03**	**11.34**	11.21	11.63	**12.01**	**11.34**	**12.11**	**11.27**	11.28
SSO	12.23	11.07	11.12	12.16	12.11	11.33	11.72	12.23	11.90	12.38	12.07	11.44
SCA	12.09	11.05	11.07	12.09	11.98	11.29	11.69	12.17	11.43	12.24	11.97	11.40
BAT	12.45	11.42	11.34	12.59	12.02	11.26	11.92	12.24	11.79	13.14	11.99	11.46
WOA-TH[20]	11.58	10.92	10.98	12.18	11.37	11.17	**11.58**	12.11	11.39	12.19	11.32	11.27
BWO[21]	**11.53**	10.94	10.99	12.11	11.41	**11.14**	11.63	12.08	11.37	12.16	11.29	**11.19**

computation time to perform multilevel image segmentation after getting optimal threshold values and results are displayed in Tables 5–15, respectively. The experiment related to image segmentation is carried out in 30 runs. Under each run, population size and iterations were set as 25 and 100 iterations.

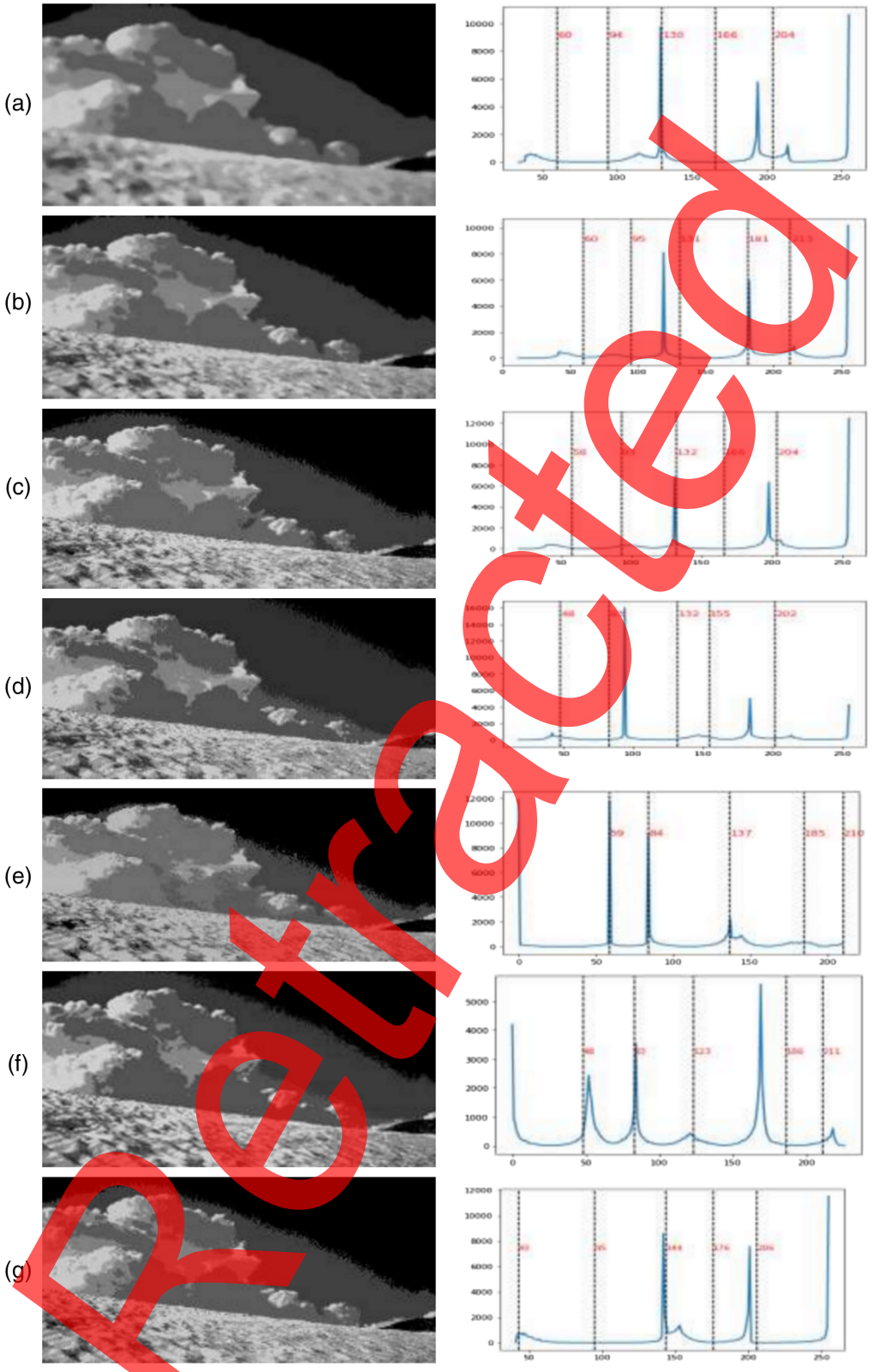

Fig. 21 Six-level segmentation performed on TestImage2 using optimal thresholds computed by (a) HWOAL, (b) WOA, (c) SSO, (d) SCA, (e) BAT, (f) WOA-TH, and (g) BWO methods and their respective histogram.

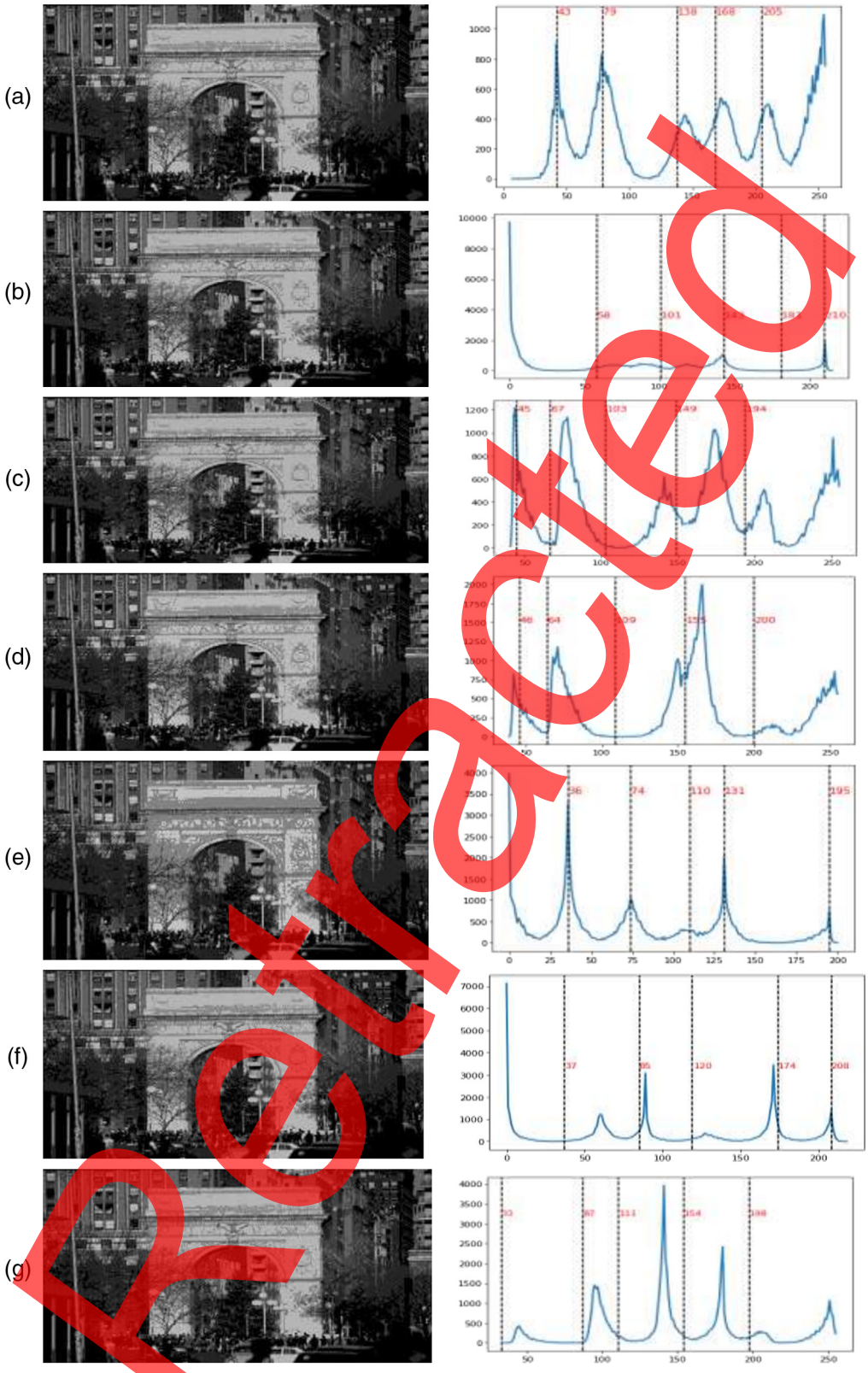

Fig. 22 Six-level segmentation performed on TestImage3 using optimal thresholds computed by (a) HWOAL, (b) WOA, (c) SSO, (d) SCA, (e) BAT, (f) WOA-TH, and (g) BWO methods and their respective histogram.

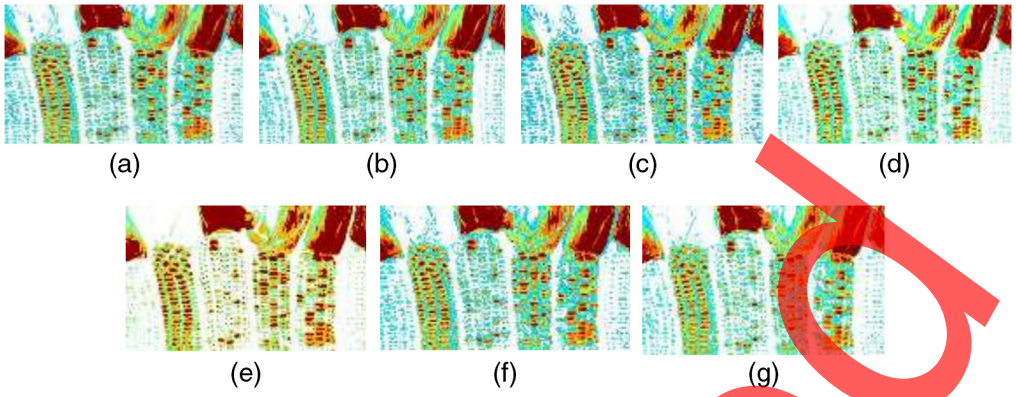

Fig. 23 Five-level segmentation performed on TestImage1 using optimal thresholds computed by (a) HWOAL, (b) WOA, (c) SSO, (d) SCA, (e) BAT, (f) WOA-TH, and (g) BWO methods.

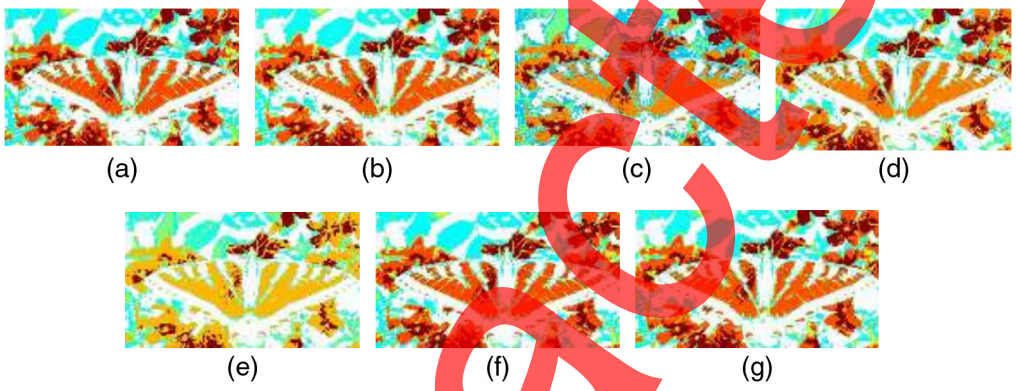

Fig. 24 Five-level segmentation performed on TestImage4 using optimal thresholds computed by (a) HWOAL, (b) WOA, (c) SSO, (d) SCA, (e) BAT, (f) WOA-TH, and (g) BWO methods.

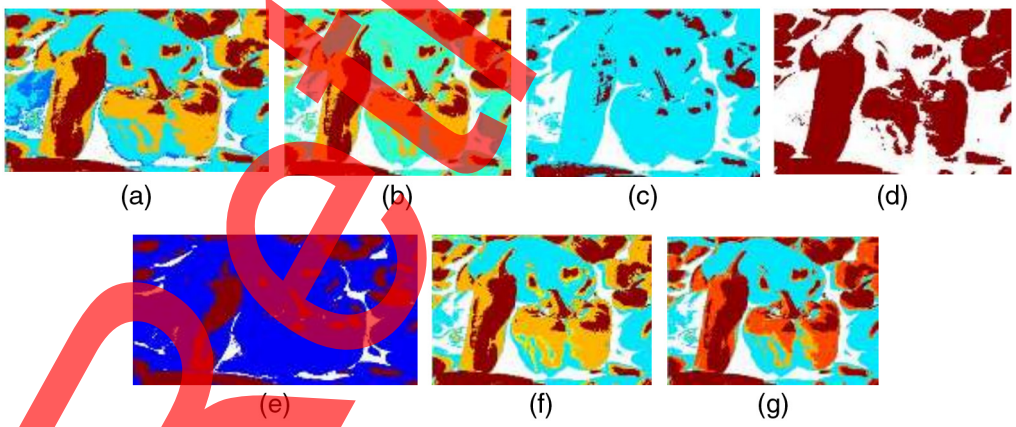

Fig. 25 Five-level segmentation performed on TestImage5 using optimal thresholds computed by (a) HWOAL, (b) WOA, (c) SSO, (d) SCA, (e) BAT, (f) WOA-TH, and (g) BWO methods.

Table 5 indicates that at higher threshold number ($k = 5$), the proposed method HWOAL has reported best value of fitness function (maximum fitness value) of each image as compared with other algorithms.

Threshold values at various levels of each image by all considered method are computed over 30 runs and best threshold value at various level of each image for every method is reported in Table 6. Best fitness value corresponds best threshold value.

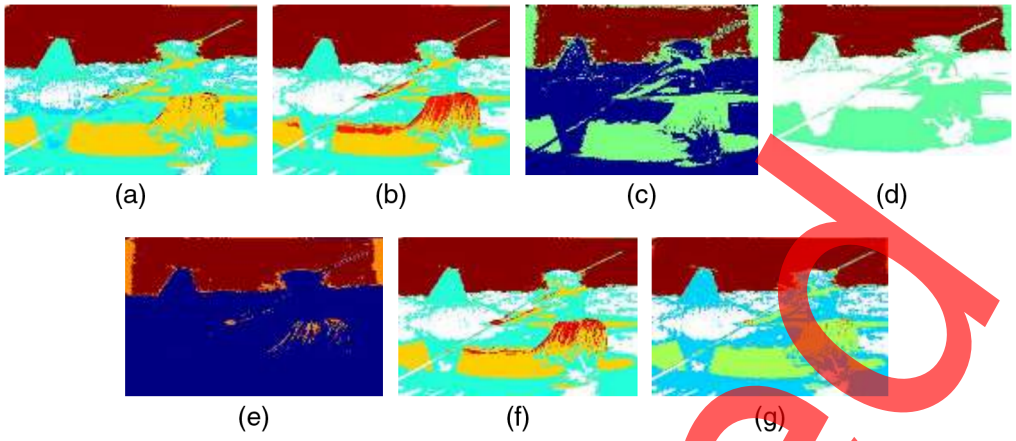

Fig. 26 Five-level segmentation performed on TestImage6 using optimal thresholds computed by (a) HWOAL, (b) WOA, (c) SSO, (d) SCA, (e) BAT, (f) WOA-TH, and (g) BWO methods.

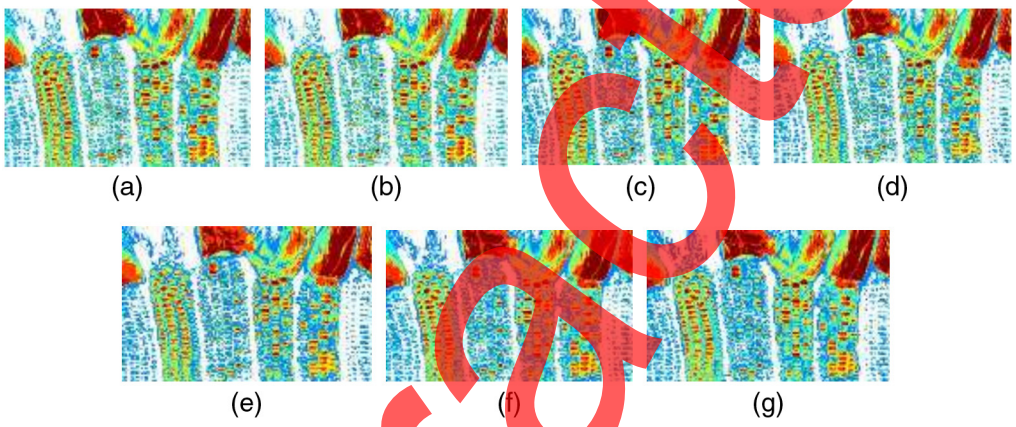

Fig. 27 Six-level segmentation performed on TestImage1 using optimal thresholds computed by (a) HWOAL, (b) WOA, (c) SSO, (d) SCA, (e) BAT, (f) WOA-TH, and (g) BWO methods.

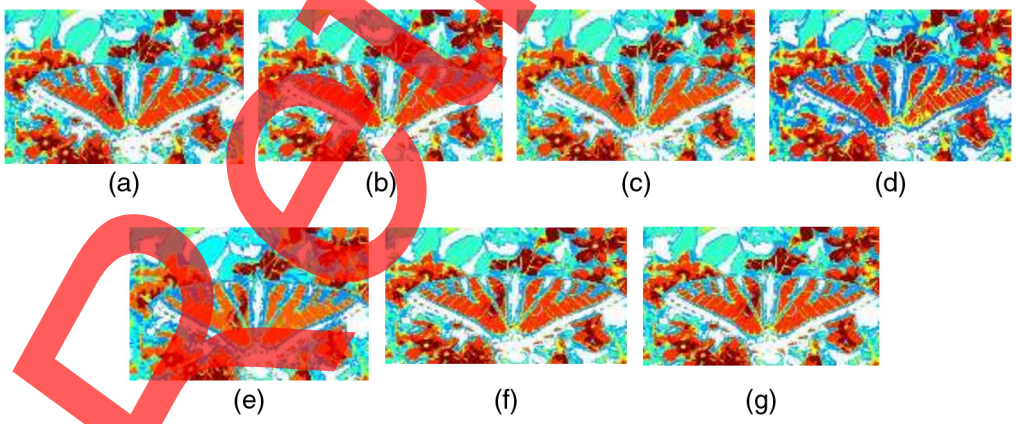

Fig. 28 Six-level segmentation performed on TestImage4 using optimal thresholds computed by (a) HWOAL, (b) WOA, (c) SSO, (d) SCA, (e) BAT, (f) WOA-TH, and (g) BWO methods.

Tables 7–10 indicate the average segmentation quality metrics (PSNR, SSIM, MSE, Avg. difference) by taking the average of 30 runs. HWOAL has achieved better average segmentation quality measures at higher threshold numbers ($k = 5$) of each image as compared to other algorithms.

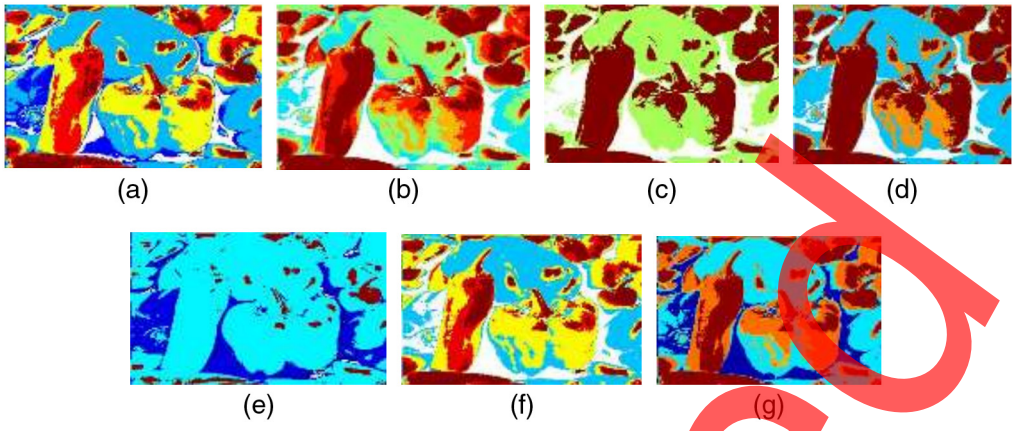

Fig. 29 Six-level segmentation performed on TestImage5 using optimal thresholds computed by (a) HWOAL, (b) WOA, (c) SSO, (d) SCA, (e) BAT, (f) WOA-TH, and (g) BWO methods.

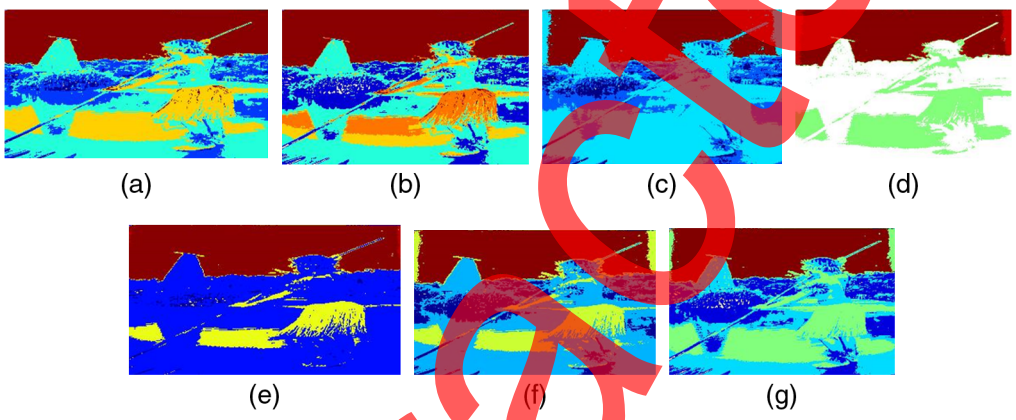

Fig. 30 Six-level segmentation performed on TestImage6 using optimal thresholds computed by (a) HWOAL, (b) WOA, (c) SSO, (d) SCA, (e) BAT, (f) WOA-TH, and (g) BWO methods.

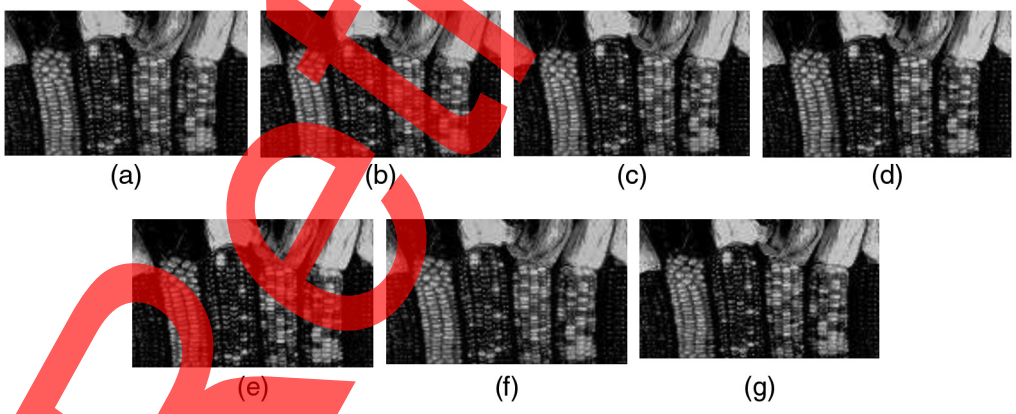

Fig. 31 Five-level gray color segmentation performed on TestImage1 using optimal thresholds computed by (a) HWOAL, (b) WOA, (c) SSO, (d) SCA, (e) BAT, (f) WOA-TH, and (g) BWO methods.

From Table 14, it can be inferred that the proposed method HWOAL is efficient and takes less time to perform multilevel image segmentation using multiple optimal threshold than other methods at higher threshold number $k = 4$ and 5. The maximum time is also less as compared with other thresholding methods to perform multilevel segmentation at threshold number $k = 4$ and 5.

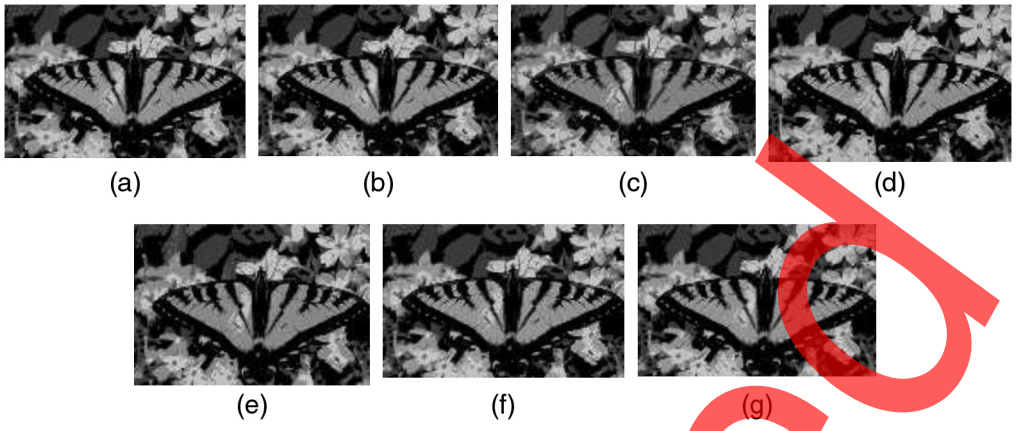

Fig. 32 Five-level gray color segmentation performed on TestImage4 using optimal thresholds computed by (a) HWOAL, (b) WOA, (c) SSO, (d) SCA, (e) BAT, (f) WOA-TH, and (g) BWO methods.

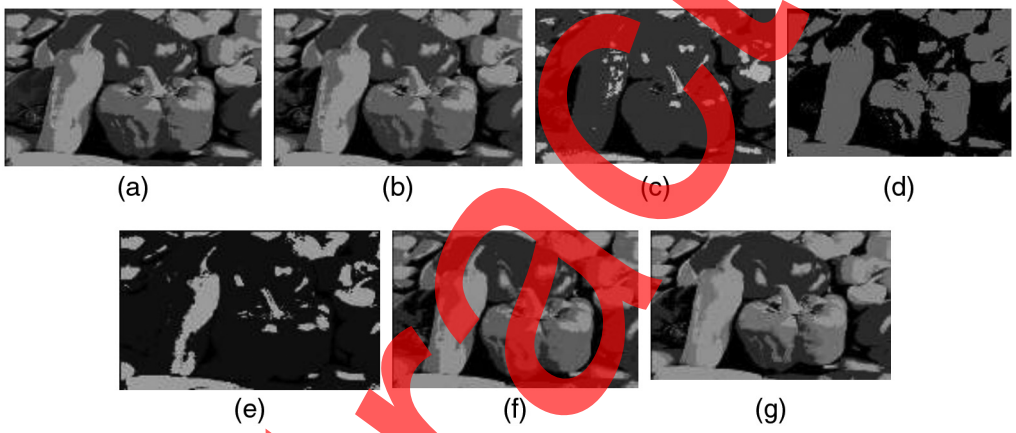

Fig. 33 Five-level gray color segmentation performed on TestImage5 using optimal thresholds computed by (a) HWOAL, (b) WOA, (c) SSO, (d) SCA, (e) BAT, (f) WOA-TH, and (g) BWO methods.

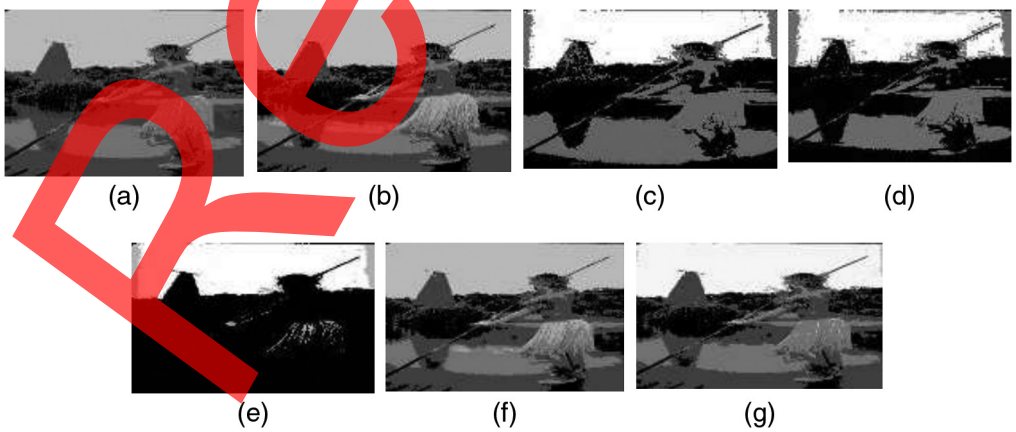

Fig. 34 Five-level gray color segmentation performed on TestImage6 using optimal thresholds computed by (a) HWOAL, (b) WOA, (c) SSO, (d) SCA, (e) BAT, (f) WOA-TH, and (g) BWO methods.

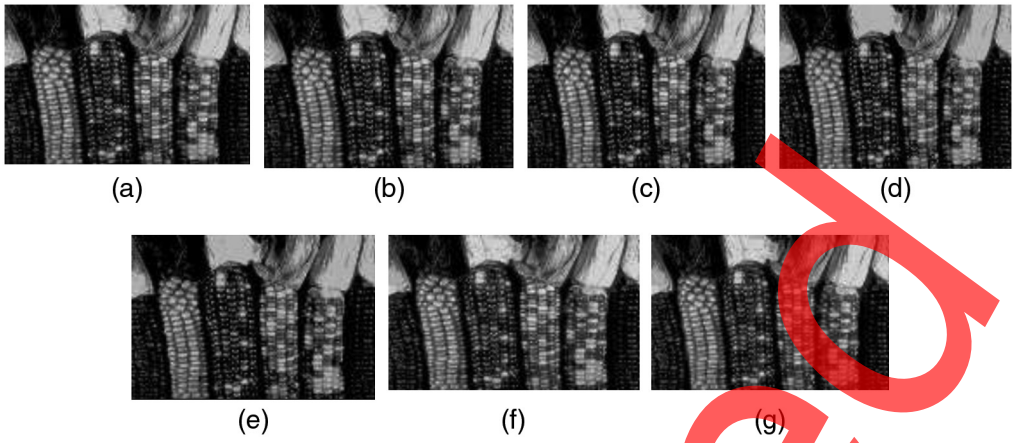

Fig. 35 Six-level gray color segmentation performed on TestImage1 using optimal thresholds computed by (a) HWOAL, (b) WOA, (c) SSO, (d) SCA, (e) BAT, (f) WOA-TH, and (g) BWO methods.

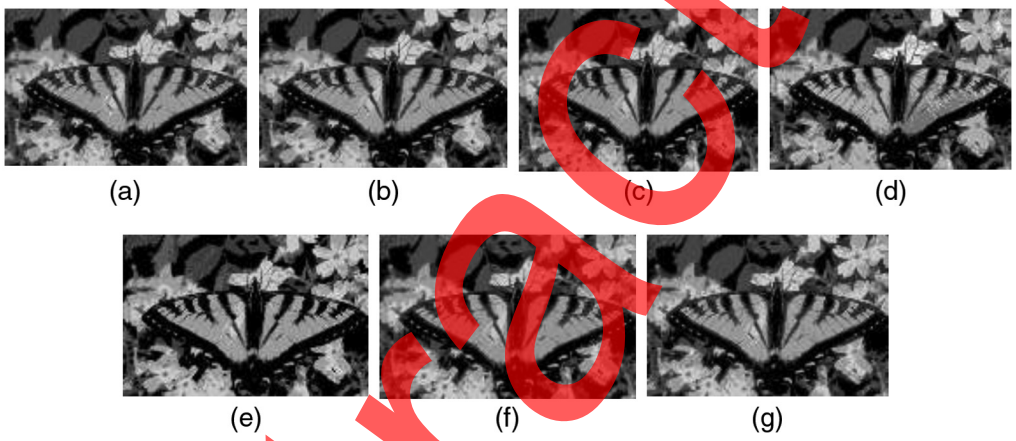

Fig. 36 Six-level gray color segmentation performed on TestImage4 using optimal thresholds computed by (a) HWOAL, (b) WOA, (c) SSO, (d) SCA, (e) BAT, (f) WOA-TH, and (g) BWO methods.

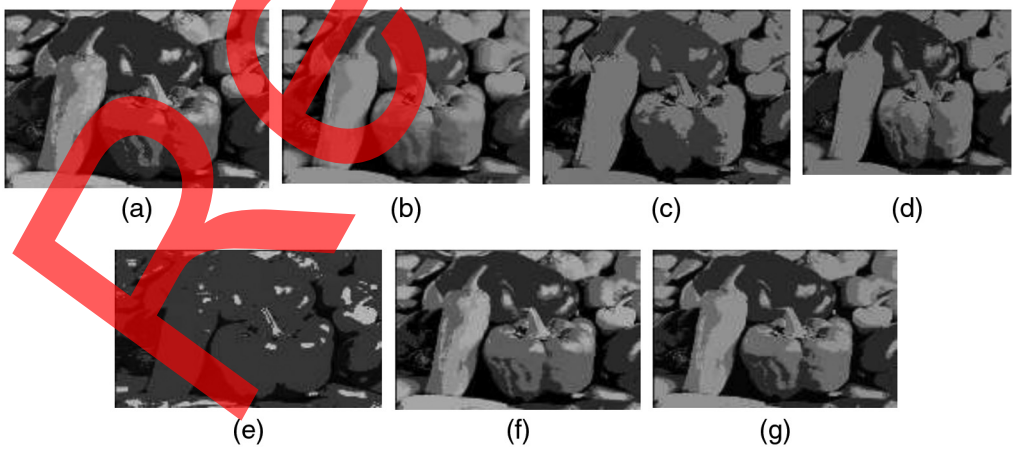

Fig. 37 Six-level gray color segmentation performed on TestImage5 using optimal thresholds computed by (a) HWOAL, (b) WOA, (c) SSO, (d) SCA, (e) BAT, (f) WOA-TH, and (g) BWO methods.

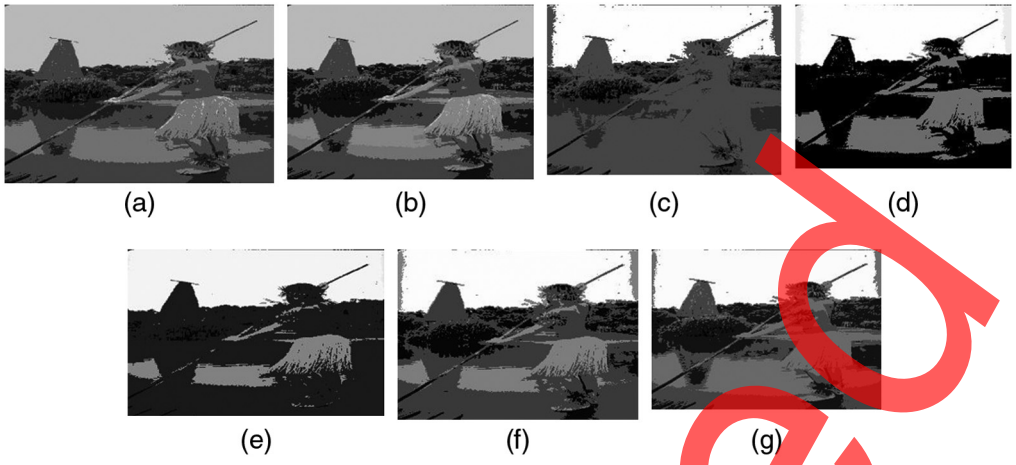

Fig. 38 Six-level gray color segmentation performed on TestImage6 using optimal thresholds computed by (a) HWOAL, (b) WOA, (c) SSO, (d) SCA, (e) BAT, (f) WOA-TH, and (g) BWO methods.

From Table 15, it can be inferred that HWOAL is a computationally efficient algorithm in terms of average computation time over 30 runs. HWOAL was the algorithm with minimum average computation time for four out of six test images at threshold values $k = 4$ and 5.

From Figs. 21 and 22, it can be seen that HWOAL-based six-level segmented images (TestImage2 and TestImage3) have superior visual quality as compared with other methods based segmented images. From HWOAL-based histogram, it can be seen that foreground regions are well segmented and fine edges are present in segmented images. Figures 23–26 show five-level RGB color image segmentation of TestImage1 and TestImage4 through TestImage6 by all considered methods.

In Figs. 23 and 24, it is observed that values of PSNR, SSIM achieved by HWOAL are higher although the images are not visually distinguishable.

For Fig. 25, objects in the image segmented by HWOAL have clear boundaries as compared with images segmented by WOA-TH and BWO. Objects in the image segmented using SSO are not distinguishable.

In Fig. 26, HWOAL produced a better visual quality image with clearly defined boundaries and minimum noise as compared with WOA-TH and BWO. Image segmented by WOA has more noise than image segmented by WOA-TH.

In Fig. 28, by increasing the threshold number (k) to 5, boundaries have become more detectable in all the images. It is also observed that values of PSNR and SSIM achieved by HWOAL are higher although the images are not visually distinguishable.

In Fig. 29, TestImage5 segmented by HWOAL has the clearest boundary and objects are distinguishable as compared with WOA-TH and BWO.

In Fig. 30, with increase in threshold number $k = 5$ (six-level segmentation) and due to reduced noise, image segmented by HWOAL has clear regions as compared to WOA-TH and BWO.

It can be observed that the proposed method HWOAL produces better fitness score, segmentation results than other algorithms at higher threshold number. Initially, when the number of thresholds is small, say, $k = 2$, the difference between algorithms for segmentation metrics is small, whereas when the number of thresholds is increased, it is clear that the proposed method (HWOAL) has a better fitness score of the objective function and segmentation results than other algorithms. For higher threshold value ($k = 5$), images segmented by HWOAL have better visual quality, fine edges, and better segmentation metrics as compared with other methods.

6 Conclusion and Future Scope

The authors have validated the proposed variant, HWOAL on 23 benchmark functions along with WOA, SSO, SCA, and BAT algorithms. This experiment showed that HWOAL has

achieved optimal value close to $f_{\min}$ for 15 functions and achieved second rank for four other functions. HWOAL has avoided premature convergence, achieved faster convergence, and avoided entrapment into local optima. Since the results produced by HWOAL were very promising, the authors have applied the HWOAL algorithm in multilevel image segmentation. HWOAL approach achieves better fitness score, segmentation quality metrics (such as MSE, PSNR, SSIM, AD), and average CPU time to perform multilevel image segmentation than other metaheuristic algorithms, such as WOA, SSO, SCA, BAT, WOA-TH, and BWO at a higher number of thresholds $k = 3, 4, 5$ in most cases. It can be said that HWOAL improves the ability of WOA to bypass local optima and obtains a better balance between exploitation and exploration phase of WOA. HWOAL has been used to find optimal multiple threshold values for multilevel image segmentation. The experimental result is carried in 30 trials and several benchmark images from BSD 300 and SIPI image dataset are randomly selected. In the future, the proposed HWOAL method will be applied to segment and detect tumors from brain MRI images, to detect Covid-19 disease from chest CT scan/x-ray images and for satellite image analysis. Classification metrics comparison for the same will be done.

Acknowledgments

The authors declare no conflict of interest (financial) or otherwise.

References

1. A. Marciniak et al., "Swarm intelligence algorithms for multi-level image thresholding," in *Intelligent Systems in Technical and Medical Diagnostics*, J. Korbicz and M. Kowal, Eds., pp. 301–311, Springer, Berlin, Heidelberg (2014).
2. A. K. Bhandari et al., "Cuckoo search algorithm and wind driven optimization based study of satellite image segmentation for multilevel thresholding using Kapur's entropy," *Expert Syst. Appl.* **41**(7), pp. 3538–3560 (2014).
3. A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognit. Lett.* **31**, 651–666 (2010).
4. C. Guo and H. Li, "Multilevel thresholding method for image segmentation based on an adaptive particle swarm optimization algorithm," in *Aust. Joint Conf. Artif. Intell.*, Springer, pp. 654–658 (2007).
5. Y. Zhang and L. Wu, "Optimal multi-level thresholding based on maximum Tsallis entropy via an artificial bee colony approach," *Entropy* **13**(4), 841–859 (2011).
6. N. Otsu, "A threshold selection method from gray level histograms," *IEEE Trans. Syst. Man Cybern.* **9**(1), 62–66 (1979).
7. J. N. Kapur, P. K. Sahoo, and A. K. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," *Comput. Vision, Graphics, Image Process.* **29**(3), 273–285 (1985).
8. S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Software* **95**, 51–67 (2016).
9. H. Mohammed, U. Shahla, and T. Rashid, "A systematic and meta-analysis survey of whale optimization algorithm," *Comput. Intell. Neurosci.* **2019**, 8718571 (2019).
10. F. S. Gharehchopogh and H. Gholizadeh, "A comprehensive survey: whale optimization algorithm and its applications," *Swarm Evol. Comput.* **48**, 1–24 (2019).
11. Y. Ling, Y. Zhou, and Q. Luo, "Lévy flight trajectory-based whale optimization algorithm for global optimization," *IEEE Access* **5**, 6168–6186 (2017).
12. B. Akay, "A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding," *Appl. Soft Comput.* **13**(6), 3066–3091 (2013).
13. L. Ali, "Multilevel thresholding in image segmentation using swarm algorithms," in *Advances in Intelligent Systems and Computing*, S. Satapathy et al., Eds., Vol. 338, Springer, Cham (2015).
14. Y. Sun et al., "A modified whale optimization algorithm for large-scale global optimization problems," *Expert Syst. Appl.* **114**, 563–577 (2018).

15. J. Pruthi and G. Gupta, "Image segmentation using genetic algorithm and Otsu," in *Proc. Fifth Int. Conf. Soft Comput. for Prob. Solving*, Springer, pp. 473–480 (2016).
16. M. A. El Aziz, A. A. Ewees, and A. E. Hassanien, "Multi-objective whale optimization algorithm for multilevel thresholding segmentation," in *Advances in Soft Computing and Machine Learning in Image Processing*, A. Hassanien and D. Oliva, Eds., pp. 23–39, Springer, Cham (2018).
17. M. A. El Aziz, A. A. Ewees, and A. E. Hassanien, "Whale optimization algorithm and moth-flame optimization for multilevel thresholding image segmentation," *Expert Syst. Appl.* **83**, 242–256 (2017).
18. T. Talal et al., "Efficient hybrid method for pan-sharpening enhancement of multiband satellite images," *Menoufia J. Electron. Eng. Res.* **29**, 21–30 (2020).
19. T. M. Talal et al., "Satellite image fusion based on modified central force optimization," *Multimedia Tools Appl.*, 1–26 (2020).
20. V. K. Bohat and K. V. Arya, "A new heuristic for multilevel thresholding of images," *Expert Syst. Appl.* **117**, 176–203 (2019).
21. E. Houssein et al., "A novel black widow optimization algorithm for multilevel thresholding image segmentation," *Expert Syst. Appl.* **167**, 114159 (2021).
22. B. D. Shivahare and S. K. Gupta, "Efficient COVID-19 CT scan image segmentation by automatic clustering algorithm," *J. Healthc. Eng.* **2022**, 9009406 (2022).
23. C. Bae et al., "A new simplified swarm optimization (SSO) using exchange local search scheme" (2012).
24. S. Mirjalili, "SCA: a sine cosine algorithm for solving optimization problems," *Knowl.-Based Syst.* **96**, 120–133 (2016).
25. X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature-Inspired Metaheuristic Algorithms*, J. R. González et al., Eds., Vol. 284, Springer, Berlin, Heidelberg (2010).
26. "The Berkeley segmentation dataset and benchmark," https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/.
27. "The USC-SIPI image database," https://sipi.usc.edu/database/.
28. A. M. Eskicioglu and P. S. Fisher, "Image quality measures and their performance," *IEEE Trans. Commun.* **43**(12), 2959–2965 (1995).
29. B. D. Shivahare and S. K. Gupta, "Multi-level image segmentation using randomized spiral-based whale optimization algorithm," *Recent Patents Eng.* **14**, 1 (2020).
30. Q. Huynh-Thu and M. Ghanbari, "Scope of validity of PSNR in image/video quality assessment," *Electron. Lett.* **44**(13), 800–801 (2008).

Basu Dev Shivahare received his BTech (CSE) degree from Uttar Pradesh Technical University, Lucknow, Uttar Pradesh, India, in 2006, and his MTech (CS) degree from BIT MESRA, Ranchi, in 2012. Currently, he is working as an assistant professor in Department of Computer Science and Engineering, at Amity University Greater Noida, Uttar Pradesh, India, and pursuing his PhD (CSE) from AKTU, Lucknow, Uttar Pradesh, India. He has published more than 10 research papers in peer-reviewed international journals and conferences. His research area is image processing. His areas of interest are big data, Hadoop, data structure, and algorithms. He is UGC-NET qualified. He has more than 14 years of teaching experience.

Sanjai Kumar Gupta is an associate professor in Department of Computer Science and Engineering at Bundelkhand Institute of Engineering and Technology (BIET) Jhansi, Uttar Pradesh, India, and member of ISTE. He received his BTech (CSE) from HBTI, Kanpur, MTech (CSE) from MNREC Allahabad, PhD (CSE) in 2010 from Bundelkhand University Jhansi, Uttar Pradesh, India. His area of interest is image processing, compiler design, operating system. He has published more than 30 research papers in international journals and conferences. He worked as a joint controller of examination, Mahamaya Technical University, Noida, Uttar Pradesh, India, since March, 2011 to December, 2013. He has guided more than 15 PhD students. He has more than 25 years of teaching experience.