Mathematical Biosciences and Engineering

MBE, 18(6): 7110–7142.
DOI: 10.3934/mbe.2021353
Received: 13 July 2021
Accepted: 20 August 2021
Published: 24 August 2021

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

Research article

Kapur's entropy for multilevel thresholding image segmentation based on moth-flame optimization

Wenqi Ji* and Xiaoguang He

College of Computer Science, Harbin Finance University, Harbin 150030, China

* **Correspondence:** Email: jiwenqi2021@126.com; Tel: +8618800464768.

Abstract: Multilevel thresholding is a reliable and efficacious method for image segmentation that has recently received widespread recognition. However, the computational complexity of the multilevel thresholding method increases as the threshold level increases, which causes the low segmentation accuracy of this method. To overcome this shortcoming, this paper presents a moth-flame optimization (MFO) established on Kapur's entropy to clarify the multilevel thresholding image segmentation. The MFO adjusts exploration and exploitation to achieve the best fitness value. To validate the overall performance, MFO is compared with other algorithms to realize the global optimal solution to maximize the target value of Kapur's entropy. Some critical evaluation indicators are used to determine the segmentation effect and optimization performance of each algorithm. The experimental results indicate that MFO has a faster convergence speed, higher calculation accuracy, better segmentation effect and better stability.

Keywords: multilevel thresholding; image segmentation; moth-flame optimization; Kapur's entropy

1. Introduction

Image segmentation is an important connection from image handling to image investigation, and the intention is to segment a provided image into multiple and unique regions and extricate the objects of interest according to characteristic, color, grain, histogram, grayscale and margin [1–5]. The quality of image segmentation is critical for the accuracy of target feature extraction and target detection, and the processing quality required in image investigation, target identification and machine vision is high. The main image segmentation strategies include thresholding, region, edge, clustering and graphs [6–10]. Compared with other image segmentation strategies, thresholding has the advantages of simple

operation, high computational productivity, small capacity space and strong robustness, which have been utilized to illuminate image segmentation. Many algorithms have been applied to settle image segmentation, including the bat algorithm (BA) [11], flower pollination algorithm (FPA) [12], moth swarm algorithm (MSA) [13], particle swarm algorithm (PSO) [14] and water wave optimization (WWO) [15].

Yan et al. designed an improved water wave optimization algorithm to solve the underwater image segmentation, the proposed algorithm had a better segmentation performance [16]. Li et al. presented a fuzzy c-means method to clarify the image segmentation, the algorithm was practical and efficacious [17]. Bao et al. introduced an alternative crossover algorithm to illuminate the color image segmentation, the proposed algorithm had a better segmentation effect [18]. Gao et al. conducted a study on an improved artificial bee colony algorithm based on the Otsu segmentation method for multi-level threshold image segmentation, the results have demonstrated the productiveness and feasibleness of the method [19]. Akay et al. combined the particle swarm optimization with the artificial bee colony algorithm for multilevel thresholding, the mixed algorithm contained higher calculation precision and better segmentation effect [20]. Pare et al. proposed the cuckoo search algorithm to puzzle out the color image thresholding, the proposed algorithm produced a high feature of the segmented images [21]. Lu et al. designed a neutrosophic C-means clustering method to explain the image segmentation, the algorithm had an excellent effect [22]. Galdran et al. presented a red channel approach to resolve the underwater image restoration, the method handled smoothly falsely enlightened zones and achieved a characteristic color adjustment [23]. Vasamsetti et al. proposed a variational improvement mechanism to unravel the underwater image, the algorithm captured the better segmentation result [24]. Bohat et al. proposed novel thresholding heuristic algorithms to settle the multilevel thresholding of images, these algorithms obtained better fitness value and segmentation effect [25]. Ouadfel et al. applied the blended method to resolve the multilevel thresholding, the proposed algorithm was powerful and achievable [26]. Pare et al. designed the Lévy flight firefly algorithm to perform the color image segmentation, the algorithm had better optimization execution in terms of distinctive constancy parameters and estimation time [27]. Satapathy et al. combined the chaotic bat algorithm with the Otsu method to clarify the image thresholding, the algorithm obtained the optimum thresholds [28]. Emberton et al. recommended a novel algorithm to improve permeability by recognizing and portioning image districts, and the algorithm achieved better optimization execution [29]. Sambandam et al. introduced a self-adaptive dragonfly algorithm to perform the digital image multilevel segmentation, the algorithm gained the threshold values [30]. Sun et al. presented a novel compound algorithm to settle the multi-level thresholding, the proposed algorithm significantly reduced the computational complexity of the given segmented images [31]. Díaz-Cortés et al. employed the dragonfly algorithm to calculate the best thresholds of the segmented image, and the proposed algorithm provided a highly reliable clinical decision support [32]. Shen et al. proposed an adjusted flower pollination algorithm to multi-level image thresholding, the results illustrated the superiority of the algorithm in terms of image quality measures, fitness values and convergence calculation [33]. Hao et al. designed a variational pattern to solve the underwater image restoration and evaluate the effectiveness and robustness [34]. Zhou et al. proposed the moth swarm algorithm to optimize the thresholds and obtain a better segmentation effect [35]. Kalyani et al. introduced an exchange market algorithm to solve the image segmentation, and the results indicate that the proposed algorithm balances exploration and exploitation to find the global threshold values [36]. Duan et al. used a modified cuckoo search algorithm to solve the multilevel thresholding image segmentation, and

the proposed algorithm had strong robustness and stability to obtain the best results [37]. Elaziz et al. combined the improved volleyball premier league algorithm with the whale optimization algorithm for image segmentation, and the hybrid algorithm had the better segmentation accuracy [38]. Li et al. tried to solve the fuzzy multilevel image segmentation using an improved coyote optimization algorithm, and the proposed algorithm had better threshold levels and segmentation effect [39]. Luo et al. applied the enhanced moth swarm algorithm to address the global optimization problem, and the proposed algorithm had a fast convergence speed and higher calculation accuracy [40]. Li et al. presented Lévy-flight moth-flame algorithm to solve the function optimization and engineering design problems, and the proposed algorithm balanced exploration and exploitation to obtain better optimization results [41]. Wang et al. designed eight complex-valued encoding metaheuristic optimization algorithms to solve the function optimization and engineering optimization problems, and the superiority of the algorithms has been proved [42].

Moth-flame optimization (MFO) simulates the transverse orientation navigation mechanism of moths and overcomes premature convergence to attain the global extremum in the optimization area [43]. MFO has some advantages of simple operation, easy implementation, few adjustment parameters, high search efficiency and strong robustness. MFO has a fast convergence speed and high calculation accuracy. MFO based on Kapur's entropy strategy is utilized to achieve multilevel thresholding image segmentation. MFO can efficiently utilize exploration and exploitation to obtain a better segmentation effect and convergence precision. Ten test images are applied to evaluate the overall segmentation performance of the proposed algorithm. To prove the productivity and practicability of the MFO, the MFO is compared with BA, FPA, MSA, PSO and WWO. The experimental results show that MFO consumes less execution time to obtain a better optimization effect, and MFO is an efficient algorithm.

The remaining sections are as follows: Section 2 provides multilevel thresholding. Section 3 surveys MFO. Section 4 describes the MFO-based multilevel threshold method. Section 5 gives the experimental results and analysis. Finally, conclusions and future research are drawn in Section 6.

2. Multilevel thresholding

The bilevel thresholding method separates an allotted image into foreground and background and can only effectively process a simple image that contains an object. However, this is a poor choice to handle a complex image that contains multiple objects. Therefore, the multilevel thresholding method replaces the bilevel thresholding method to perform image segmentation and obtains the best threshold values in the solution space. Kapur's entropy is a valuable and feasible metric for multilevel thresholding segmentation. Kapur's entropy divides the image into different classes, and the size of the entropy determines whether the category is uniform. Kapur's entropy has a low calculation process, easy implementation, strong stability, fast processing speed and high segmentation accuracy. The entropy of the original image is compactness and separateness between different classes. Kapur's entropy finds the best threshold values by maximizing the fitness value. The selection of the segmentation threshold for MFO is manual. Gao et al. designed an effective genetic first-order statistical image segmentation method to solve quantitative crack detection. The proposed method can automatically determine the best statistical feature and threshold selection [44]. Assuming that $[t_1, t_2, ..., t_n]$ are the excellent threshold values, an image is separated into diverse classes [45]. The formula is described as

$$p_i = \frac{h_i}{\sum_{i=0}^{L-1} h(i)} \tag{1}$$

where h_i denotes the number of pixels, i denotes gray level, and L denotes the number of levels.

$$f(t_1, t_2, ..., t_n) = H_0 + H_1 + H_2 + ... + H_n \tag{2}$$

where

$$H_0 = -\sum_{i=0}^{t_1-1} \frac{p_i}{\omega_0} \ln \frac{p_i}{\omega_0}, \quad \omega_0 = \sum_{i=0}^{t_1-1} p_i \tag{3}$$

$$H_1 = -\sum_{i=t_1}^{t_2-1} \frac{p_i}{\omega_1} \ln \frac{p_i}{\omega_1}, \quad \omega_1 = \sum_{i=t_1}^{t_2-1} p_i \tag{4}$$

$$H_2 = -\sum_{i=t_2}^{t_3-1} \frac{p_i}{\omega_2} \ln \frac{p_i}{\omega_2}, \quad \omega_2 = \sum_{i=t_2}^{t_3-1} p_i \tag{5}$$

$$H_n = -\sum_{i=t_n}^{L-1} \frac{p_i}{\omega_n} \ln \frac{p_i}{\omega_n}, \quad \omega_n = \sum_{i=t_n}^{L-1} p_i \tag{6}$$

$H_0, H_1, ..., H_n$ denote the Kapur's entropies of the distinct classes, and $\omega_0, \omega_1, ..., \omega_n$ denote the probabilities of each class.

3. MFO

The unique flight approach of moths at night is called transverse orientation. In MFO, the moths are regarded as a candidate solution, and the positions are regarded as the variables. The moth maintains a fixed flight angle relative to the moon, which is a powerful flight strategy that travels long distances along a straight line. Due to the presence of artificial light sources in nature, the flames are regarded not only as the optimal solutions but also as flags or pins in the search area. The position of each moth is refreshed by the logarithmic spiral movement formula. The transverse orientation is given in Figure 1, and the spiral flying path is given in Figure 2. The corresponding relationship between the algorithm's description and moth-flame's elements is given in Table 1.

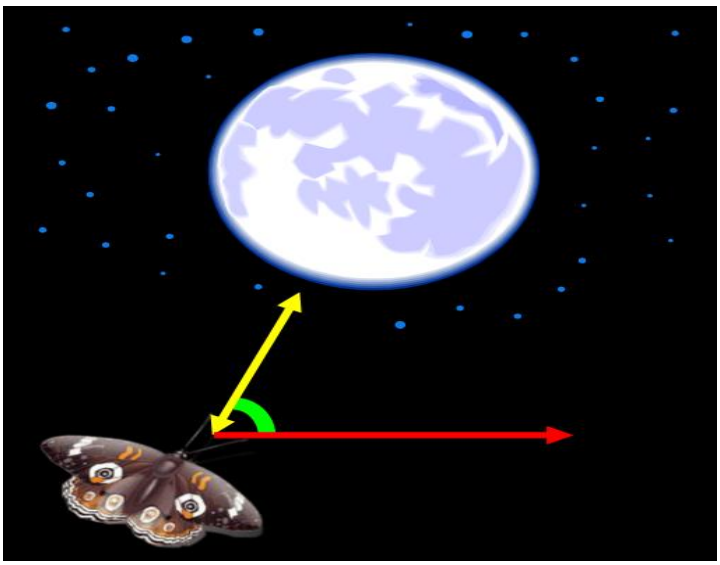


Figure 1. Transverse orientation.

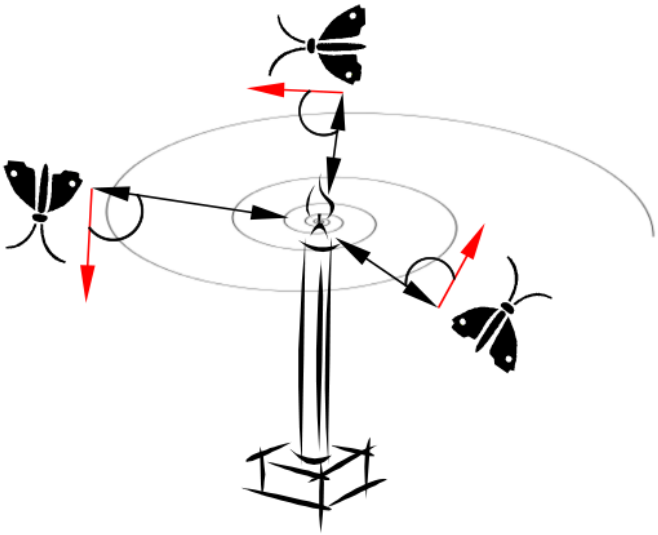


Figure 2. Spiral flying path.

Table 1. Correspondence between algorithm's description and moth-flame's elements.

Algorithm's description	Moth-flame's elements
Decision variable	Moth's position in each dimension
Solutions	Moth's position
Initial solutions	Random positions of moths
Current solutions	Current positions of moths
New solutions	New positions of moths
Best solution	Flame's position
Fitness function	Distance between moth and flame
Process of generating a new solution	Flying in a spiral path toward a flame

3.1. Generate the initial population of moths

The moths fly in one-dimensional, two-dimensional, three-dimensional or n-dimensional space. MFO is a population-based optimization algorithm, the formula is described as

$$M = \begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & \cdots & m_{1,d} \\ m_{2,1} & m_{2,2} & \cdots & \cdots & m_{2,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ m_{n,1} & m_{n,2} & \cdots & \cdots & m_{n,d} \end{bmatrix} \tag{7}$$

where n denotes the number of moths and d denotes the number of dimensions.

The target values of the moths are sorted by an array.

$$OM = \begin{bmatrix} OM_1 \\ OM_2 \\ \vdots \\ OM_n \end{bmatrix} \tag{8}$$

The flames are the core component of the MFO, and its matrix is similar to the moth matrix.

$$F = \begin{bmatrix} F_{1,1} & F_{1,2} & \cdots & \cdots & F_{1,d} \\ F_{2,1} & F_{2,2} & \cdots & \cdots & F_{2,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ F_{n,1} & F_{n,2} & \cdots & \cdots & F_{n,d} \end{bmatrix} \tag{9}$$

The target values of the flames are sorted by an array.

$$OF = \begin{bmatrix} OF_1 \\ OF_2 \\ \vdots \\ OF_n \end{bmatrix} \tag{10}$$

Moths and flames are selected as candidate solutions. The main distinction is that the position update method is different during the iteration process. The moths are chosen as the actual search agents to move in the search area. The flames are chosen as flags or pins that crashed by moths, which is the best solution currently obtained by the moths. Therefore, the moths perform a global search around the flames until the moths find the optimal positions. With the flight mechanism, the moths never lose their best solutions.

3.2. Update the moths' positions

The framework of MFO contains three-tuple approximation functions to realize the best solution. The location is described as

$$MFO = (I, P, T) \tag{11}$$

where I denotes a randomly generated moth swarm and the target function $(I : \phi \to \{M, OM\})$, P denotes the central function of the moth $(P : M \to M)$, and T denotes the end of the search $(T : M \to true, false)$. The I is used to implement the random distribution.

$$M(i, j) = (ub(i) - lb(i)) * rand() + lb(i) \tag{12}$$

where lb and ub denote the lower and upper bounds of the search space, respectively. In MFO, the moths not only use the transverse orientation to fly but also adopt a logarithmic spiral function to update the flight mechanism.

The logarithmic spiral function is described as

$$S(M_i, F_j) = \mathrm{D}_i \cdot e^{bt} \cdot (\cos 2\pi t) + F_j \tag{13}$$

where D_i denotes the length between the i-th moth and the j-th flame, b denotes a fixed value of the logarithmic spiral, and t denotes an arbitrary value in [-1,1].

D is described as

$$\mathrm{D}_i = \left| F_j - M_i \right| \tag{14}$$

where M_i denotes the i-th moth and F_j denotes the j-th flame.

3.3. Update the number of flames

In MFO, n different positions are used to update the positions of the moths, which will reduce the exploitation of the MFO. Therefore, decreasing the number of flames is beneficial to tackle this issue. The formula is described as

$$flame\ no = round(N - l * \frac{N-1}{T}) \tag{15}$$

where N denotes the number of flames, l denotes the current number of iterations, and T denotes the maximum number of iterations.

The MFO is shown in Algorithm 1.

Algorithm 1. MFO.

Randomly initialize the position of moths and the parameters for Moth-flame
While $iteration \le Max\_iteration$ **do**
  Renew flame no utilizing Eq (15)
  $OM = FitnessFunction(M)$
  If $iteration == 1$ **then**
    $F = sort(M)$
    $OF = sort(OM)$

else
$F = sort(M_{t-1}, M_t)$
$OF = sort(M_{t-1}, M_t)$
end if
for $i = 1:n$ **do**
for $j = 1:d$ **do**
Renew r and t
Estimate D utilizing Eq (14)
Renew $M(i, j)$ utilizing Eq (13)
end for
end for
end while
Return the best solution

4. MFO-based multilevel threshold method

The position of each moth denotes the threshold value of the segmented image. The moth revises the position to determine the global optimal solution by the threshold level. The correspondence between the threshold segmentation and MFO is given in Table 2. The MFO-based Kapur's entropy is shown in Algorithm 2. The flowchart of MFO for multilevel threshold is shown in Figure 3.

Table 2. Correspondence between image segmentation and MFO.

Threshold segmentation	MFO
A collection $(x_1, x_2, ..., x_k)$ scheduling schemes	A moth population $(n_1, n_2, ..., n_k)$ moths
An optimal scheme to resolve the image segmentation	An optimal flame's position
The objective function of the image segmentation	The fitness function of the MFO

Algorithm 2. MFO-based Kapur entropy.

Randomly initialize the position of moths and the parameters for Moth-flame
While $iteration \le Max\_iteration$ **do**
Renew flame no utilizing Eq (15)
Estimate the fitness value of each moth utilizing Eq (2) for the Kapur entropy and attain the best solution
$OM = FitnessFunction(M)$
If $iteration == 1$ **then**
$F = sort(M)$
$OF = sort(OM)$
else
$F = sort(M_{t-1}, M_t)$
$OF = sort(M_{t-1}, M_t)$
end if
for $i = 1:n$ **do**
for $j = 1:d$ **do**
Renew r and t

Estimate D utilizing Eq (14)
Renew $M(i, j)$ utilizing Eq (13)
end for
end for
end while
Return the best solution or the optimal threshold value

4.1. Computation complexity of MFO

In this section, the time and spatial complexity of the MFO are studied.

4.1.1. Time complexity

The time complexity of MFO is based on the calculation workload and operational structure of the algorithm, which is used as an important indicator to estimate the optimization efficiency. In MFO, n denotes the number of moths, t denotes the maximum number of iterations, d denotes the number of variables, and the sorting mechanism of flames in each iteration. MFO uses the quicksort method, the best time complexity of MFO is $O(n\log n)$, and the worst time complexity of MFO is $O(n^2)$. Therefore, the total time complexity of MFO is described as

$$O(MFO) = O(t(O(Quick\ sort) + O(position\ update))) \tag{16}$$

$$O(MFO) = O(t(n^2 + n \times d)) = O(tn^2 + tnd) \tag{17}$$

4.1.2. Spatial complexity

The spatial complexity denotes the storage space that requires to be executed by an algorithm. The total spatial complexity of MFO is $O(tn^2 + tnd)$, which is valuable and achievable.

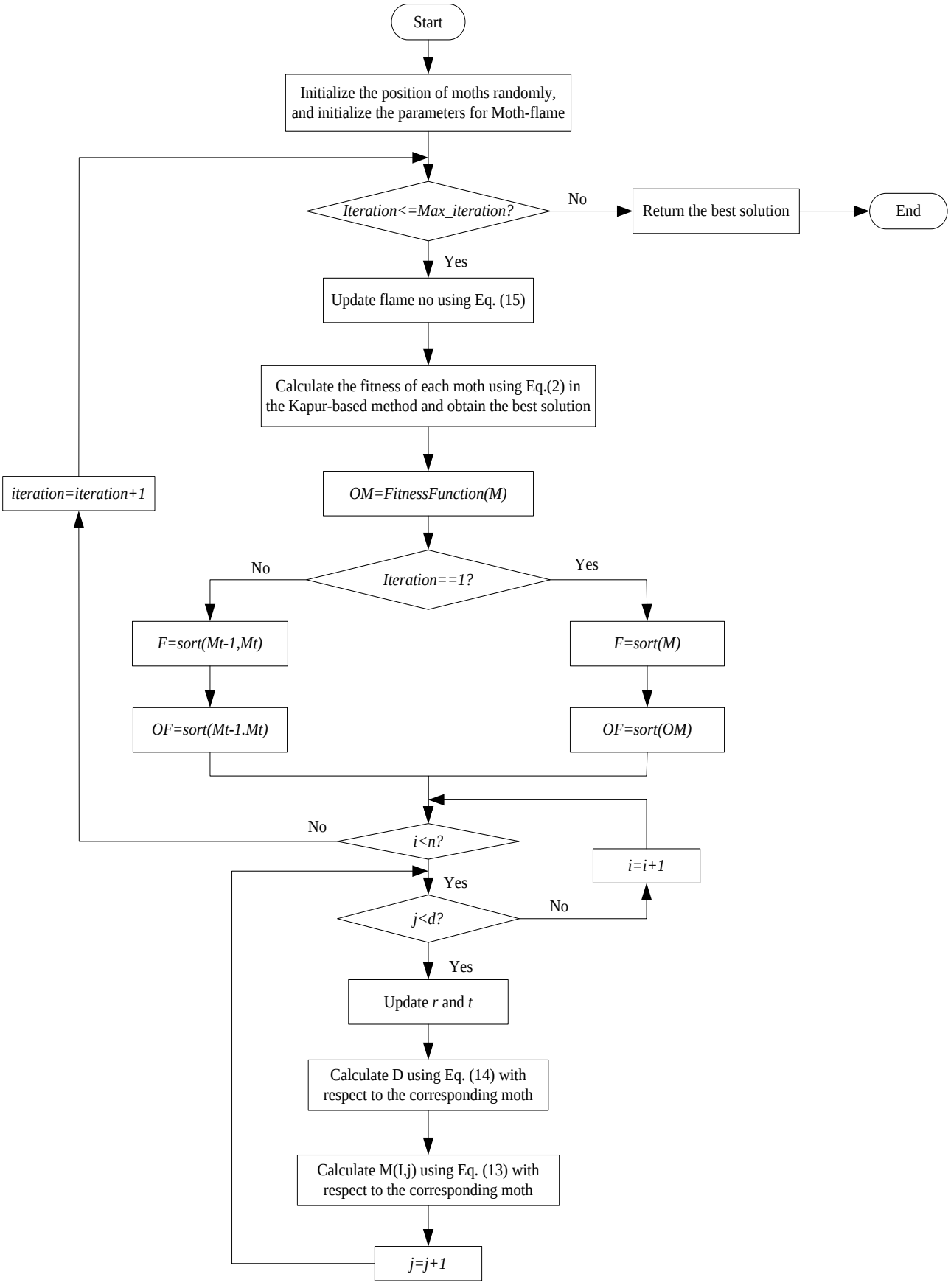


Figure 3. Flowchart of MFO for multilevel threshold.

5. Experimental results and analysis

5.1. Experimental setup

The numerical test is conducted on a computer with a 2.2 GHz Intel Core i7-8750H processor and 8 GB of RAM using MATLAB R2018(b).

5.2. Test images

Image segmentation is an effective method for character extraction and object identification recognition. It is also an important step from image handling to image investigation. Image segmentation segments a provided preprocessed image to obtain a more intuitive target and background. The experiments select ten images to detect the overall segmentation effect, and they are given in Figure 4.

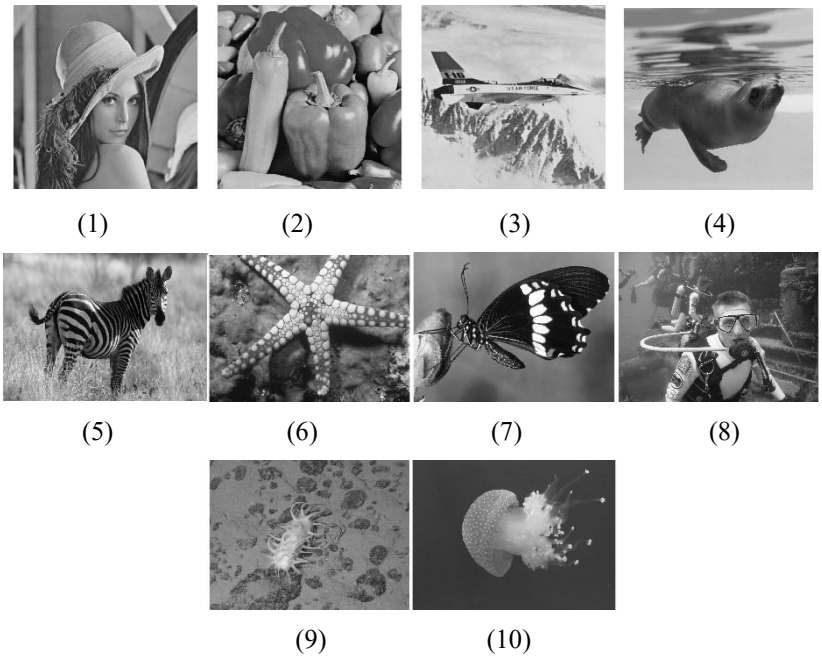


Figure 4. Original test images.

5.3. Parameter setting

To prove the effectiveness and practicability, the MFO is compared with BA, FPA, MSA, PSO and WWO. These control parameters are some representative empirical values and are derived from the original articles. The parameters of all algorithms are given in Table 3.

Table 3. Parameters of all algorithms.

Algorithm	Parameter	Value
BA [11]	Pulse frequency scope f	[0,2]
	Echo loudness A	0.25

Continued on next page

Algorithm	Parameter	Value
	Reducing coefficient γ	0.5
FPA [12]	Switch probability ρ	0.8
MSA [13]	An arbitrary value θ	[-2,1]
	An arbitrary value ε_2	[0,1]
	An arbitrary value ε_3	[0,1]
	An arbitrary value r_1	[0,1]
	An arbitrary value r_2	[0,1]
PSO [14]	Constant inertia ω	0.3
	First acceleration coefficient c_1	1.4962
	Second acceleration coefficient c_2	1.4962
WWO [15]	Wavelength λ	0.5
	Wave height $h_{\max}$	6
	Wavelength attenuation factor α	1.0026
	Breaking factor β	[0.001,0.25]
	Maximum value $k_{\max}$ of breaking directions	$\min(12, D/2)$
MFO [43]	Logarithmic spiral b	1
	An arbitrary value t	[-1,1]

5.4. Segmented image quality measurements

Five important indicators are used to estimate the image segmentation effect of different algorithms as follows:

1) Fitness value. The fitness value shows the calculation precision of each algorithm. The fitness value is proportional to the segmented image information.

2) Execution time. The algorithm consumes less time, which means that the algorithm has a faster calculation process.

3) Peak signal-to-noise ratio (PSNR). The PSNR based on the intensity value is used to measure the variation between the processed image and the reference image. A larger PSNR indicates that the processed image has less distortion. The image segmentation effect of the higher PSNR may be lower than the image segmentation effect of the lower PSNR. The PSNR is described as [46]

$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right) \tag{18}$$

where MSE denotes the mean squared error. It is described as follows:

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left[I(i,j) - K(i,j)\right]^2 \tag{19}$$

where M and N denote the size of the provided image and the processed image, respectively.

4) Structure similarity index (SSIM). The SSIM is a similarity measure between the provided image and the processed image. If the SSIM is close to 1, then the image segmentation result is better. The SSIM is described as [47]

$$SSIM(\mathrm{x},\mathrm{y}) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + \mathrm{c}_2)}{(\mu_x^2 + \mu_y^2 + \mathrm{c}_1)(\sigma_x^2 + \sigma_y^2 + \mathrm{c}_2)} \tag{20}$$

where for the provided image and the processed image, μ_x and μ_y denote the mean intensity. σ_x^2 and σ_y^2 denote the standard deviation. σ_{xy} denotes the covariance. c_1 and c_2 denote constants.

5) Wilcoxon's rank-sum test is utilized to identify whether there is a noteworthy distinction between the two algorithms [48]. There is a noteworthy distinction if the p value is lower than 0.05. There is no noteworthy distinction if the p value is higher than 0.05.

5.5. Results and analysis

To ensure the segmentation effect of the MFO, the population size of each algorithm is set to 30, the maximum number of iterations is set to 100, and the number of independent runs is set to 30. The threshold levels are defined as 2, 3, 4, 5 and 6. The effectiveness and feasibility of the MFO are verified by comparing it with other algorithms. The experimental results of the comparison algorithms are given in Tables 4–9, and the experimental results of some given segmented images are given in Figures 5–14.

Table 4. The optimal fitness of each algorithm.

Images	k	Fitness values						
		BA	FPA	MSA	PSO	WWO	MFO	Rank
Test 1	2	12.3364	12.3198	12.2127	12.3479	12.3112	12.3487	1
	3	15.2194	15.3081	15.2704	15.2018	15.1970	15.3172	1
	4	17.7102	17.7705	17.8821	17.7208	17.8752	17.9921	1
	5	20.3813	20.0295	20.2961	21.1111	20.2899	21.2798	1
	6	22.4683	22.3940	21.9694	22.4280	22.4488	22.6422	1
Test 2	2	12.5842	12.5331	12.5997	12.6346	12.5916	12.6346	1
	3	15.6189	15.5130	15.6003	15.4587	15.4813	15.6531	1
	4	18.1117	18.3233	18.1107	18.3061	18.2180	18.4397	1
	5	20.5250	20.9349	20.9719	20.9611	20.9605	21.0539	1
	6	23.1116	23.3068	23.3549	23.2826	23.2950	23.3837	1
Test 3	2	12.1651	12.1623	12.1384	12.2017	12.0067	12.2061	1

Continued on next page

Images	k	Fitness values						
		BA	FPA	MSA	PSO	WWO	MFO	Rank
	3	15.3462	15.3160	14.9498	15.2155	15.1669	15.5039	1
	4	17.9605	18.1165	18.0610	17.8602	17.7274	18.3107	1
	5	20.0693	20.3889	20.6938	20.7448	20.0828	20.8056	1
	6	22.2665	22.6040	22.7055	23.0135	22.2510	24.1228	1
Test 4	2	11.6162	11.0138	11.6151	11.6170	11.5649	11.6170	1
	3	14.4375	14.1046	14.4277	14.4640	14.4074	14.6363	1
	4	17.1661	17.1351	17.2543	17.2182	16.5648	17.5047	1
	5	19.3315	19.0672	19.5998	19.9435	19.4292	20.0589	1
	6	22.1115	21.7571	21.8045	21.9815	21.7166	22.1257	1
Test 5	2	12.5987	9.17771	12.4965	12.6016	12.5880	12.6016	1
	3	15.8446	15.6981	15.6613	15.7945	15.6387	15.9201	1
	4	18.6330	18.5687	18.6837	18.7647	18.4739	18.8308	1
	5	21.5729	21.4283	21.6418	21.3651	21.5618	21.6809	1
	6	24.1845	24.0339	24.0773	23.7893	23.9359	24.3074	1
Test 6	2	12.9609	12.9185	12.9393	12.9682	12.9668	12.9682	1
	3	15.9835	16.0781	16.0526	16.1034	16.1034	16.1252	1
	4	18.5746	18.7090	18.7443	18.6889	18.7069	19.0165	1
	5	21.3011	21.1524	21.2343	21.0410	20.8741	21.4711	1
	6	24.1403	23.8479	23.9266	23.7607	24.0573	24.1842	1
Test 7	2	12.5779	12.3517	12.5367	12.5936	12.3939	12.5936	1
	3	15.3831	15.4011	15.3516	15.3938	15.0022	15.4274	1
	4	18.3603	18.0115	17.8258	17.9950	18.2520	18.4235	1
	5	20.9092	20.9076	20.6714	20.5292	20.7942	21.0217	1
	6	22.8604	23.3690	23.1421	22.9799	23.0124	23.5484	1
Test 8	2	12.8868	12.8947	12.8930	12.9208	12.8957	12.9208	1
	3	15.8248	15.9079	15.8297	15.8527	15.8992	16.0552	1
	4	18.5661	18.7317	18.6622	18.7469	18.5391	19.0449	1
	5	21.2946	21.1664	21.1186	21.1555	21.5266	21.5719	1
	6	23.5705	23.8786	24.2027	23.8717	23.6695	24.3206	1
Test 9	2	11.9271	11.8657	11.4150	11.9285	11.7951	11.9286	1
	3	14.7762	14.7811	14.7560	14.8629	14.7074	14.9331	1
	4	17.1365	17.2515	17.1703	17.3163	17.3681	17.4062	1
	5	19.8132	19.5984	19.8790	19.7153	19.8422	19.8973	2
	6	21.9515	22.0255	22.1325	21.8282	21.9971	22.2318	1
Test 10	2	11.5248	11.4683	11.4326	11.5288	11.4273	11.5288	1
	3	14.3849	13.9338	14.1944	14.4104	14.3475	14.5453	1
	4	16.8579	17.2591	17.3456	17.3302	17.2842	17.3904	1
	5	19.3943	19.2793	18.9721	19.3799	19.3881	19.8757	1
	6	21.5514	21.6964	21.5692	21.3715	21.8951	22.2800	1

Table 5. The best threshold values of each algorithm.

Images	k	Best threshold values					
		BA	FPA	MSA	PSO	WWO	MFO
Test 1	2	95,168	95,171	113,157	98,165	103,173	97,164
	3	67,121,181	88,134,179	76,133,176	73,124,160	90,139,168	81,126,174
	4	68,110,130,163	60,103,160,200	55,91,139,180	61,83,116,181	79,121,146,175	73,112,147,184
	5	60,98,144,172, 204	46,65,109,142, 176	78,98,135,157, 183	39,100,163,165, 239	63,91,144,168， 187	65,101,125,165, 239
	6	38,58,98,140, 171,195	71,89,127,176, 194,216	44,91,101,133, 155,200	66,106,137,159, 196,224	66,115,140,162, 178,193	77,98,132,156, 181,208
Test 2	2	62,148	64,121	79,138	75,147	66,153	75,147
	3	76,132,174	42,83,158	76,119,158	45,130,169	43,109,146	59,106,170
	4	43,86,144,206	55,108,133,190	62,89,118,192	53,121,152,188	47,118,159,203	50,84,141,179
	5	24,93,134,173, 192	35,93,126,155, 191	32,79,101,136, 188	34,73,95,129, 176	56,84,104,145， 175	50,80,119,163, 204
	6	35,80,106,137, 191,203	56,101,126,160, 177,200	44,76,122,162, 177,196	22,38,82,110, 151,193	65,83,107,145， 179,199	37,52,76,126, 169,195
Test 3	2	82,162	88,164	85,157	69,177	110,163	69,174
	3	87,135,186	62,141,178	102,159,184	73,141,195	81,154,192	69,127,183
	4	89,123,152,189	64,114,151,176	54,95,135,169	68,120,134,185	38,93,148,192	66,106,146,186
	5	78,101,118,133, 186	67,98,135,149, 174	56,80,111,152, 194	34,87,165,182, 231	75,101,132,169, 213	70,104,132,160, 190
	6	82,114,131,154, 174,212	49,80,94,114, 174,198	41,53,73,113, 146,181	55,93,128,153, 183,196	58,71,78,118, 147,181	56,69,127,163, 183,231
Test 4	2	101,174	146,164	95,175	98,174	93,154	95,171
	3	72,136,169	117,177,212	69,115,180	99,152,181	70,113,169	85,137,180
	4	41,84,115,164	50,80,146,183	43,91,124,172	40,86,133,167	63,96,159,172	38,86,138,180
	5	49,82,122,181, 224	29,75,104,120, 205	49,75,105,160, 181	32,91,125,171, 208	37,100,116,152, 210	43,98,125,173, 245
	6	75,102,125,161, 188,237	43,110,129,150, 181,211	79,120,135,167, 221,237	43,59,85,127, 183,212	32,61,78,122, 148,211	28,53,77,114, 146,182
Test 5	2	74,135	110,248	83,157	70,135	64,129	70,135
	3	56,134,215	61,120,184	59,145,230	62,119,197	80,127,180	69,133,204
	4	52,131,206,231	95,145,200,234	70,94,146,207	74,127,186,226	81,114,190,221	70,108,150,216
	5	61,110,168,199, 231	33,71,97,154, 225	37,101,137,179, 217	21,79,150,195, 231	63,93,134,188, 233	41,94,131,177, 225
	6	29,62,117,143, 188,211	23,75,110,142, 208,238	38,61,127,167, 212,233	51,84,124,147, 187,200	32,72,91,116, 192,228	27,77,109,140, 179,225
Test 6	2	84,168	111,175	106,175	91,170	92,171	91,170
	3	55,126,191	84,145,188	85,138,200	71,124,188	68,128,177	75,131,185
	4	37,117,168,211	60,137,164,194	39,77,132,186	94,121,157,214	54,139,178,219	68,105,157,204
	5	50,75,110,184, 206	41,82,122,189, 234	35,78,131,156, 183	37,124,154,182, 208	42,108,120,179, 205	80,111,132,163, 204

Continued on next page

Images	k	Best threshold values					
		BA	FPA	MSA	PSO	WWO	MFO
	6	59,89,114,146, 187,207	32,58,116,146, 186,214	60,80,105,135, 168,189	47,88,111,165, 191,204	44,65,113,146, 170,199	42,84,116,136, 170,209
Test 7	2	97,166	124,164	97,170	96,163	70,156	96,163
	3	75,123,158	53,94,176	54,130,162	80,107,170	62,178,219	81,166,198
	4	71,127,166,218	66,121,145,184	77,129,187,236	43,110,162,181	95,123,166,209	46,104,167,197
	5	46,87,123,161, 225	40,99,130,175, 198	40,59,97,143, 196	24,73,137,166, 214	65,122,146,175, 214	79,107,137,166, 212
	6	37,69,131,145, 174,236	47,63,112,162, 186,223	53,101,125,160, 202,214	62,123,149,174, 190,233	62,110,167,193, 204,233	44,82,132,171, 200,234
Test 8	2	91,170	83,162	100,168	93,161	94,169	93,161
	3	80,164,225	109,165,201	66,102,157	69,105,171	59,114,161	95,159,206
	4	46,104,144,217	73,109,175,225	83,136,177,203	42,109,159,191	47,81,124,156	65,111,160,207
	5	62,120,172,204, 223	36,55,115,157, 217	80,143,175,209, 231	55,112,132,173, 194	68,102,131,164, 223	74,107,159,196, 223
	6	29,47,87,110, 157,212	58,85,113,133, 165,229	52,75,102,131, 160,205	59,74,108,145, 171,204	58,98,144,157, 181,201	50,85,126,158, 202,227
Test 9	2	103,155	93,160	128,163	104,156	92,166	105,156
	3	110,152,213	92,152,212	104,166,207	111,153,193	69,115,167	104,154,203
	4	55,105,135,161	44,86,150,205	55,96,121,174	97,132,155,192	63,99,173,203	57,106,172,201
	5	96,117,150,179, 298	92,121,136,152, 204	56,91,114,165, 220	56,116,153,169, 186	92,129,154,172, 201	61,84,102,163, 197
	6	45,74,86,120, 161,220	66,84,96,128, 156,184	72,95,116,162, 172,209	78,96,107,133, 151,199	53,84,102,111, 165,211	56,82,115,164, 187,198
Test 10	2	74,148	72,138	77,133	75,153	96,168	75,153
	3	86,154,184	75,104,213	84,129,211	71,154,196	97,143,196	83,143,187
	4	76,95,171,222	83,125,164,203	77,123,164,213	74,112,138,177	74,123,156,210	72,116,151,181
	5	94,132,147,188, 222	85,146,168,202, 230	71,88,101,179, 213	50,83,152,179, 248	81,132,174,214, 230	74,105,136,163, 186
	6	70,82,114,172, 194,230	74,110,126,163, 194,203	89,120,133,167, 180,223	64,81,128,162, 202,230	72,126,162,192, 213,233	75,103,144,176, 191,222

The optimal fitness of each algorithm is given in Table 4. The optimization goal is to achieve the best threshold levels by maximizing the desired value of Kapur's entropy. For given segmented images, the threshold levels are set to 2, 3, 4, 5 and 6. The fitness value of each algorithm increases as the threshold level increases, and each algorithm can achieve a better fitness value under a relatively high threshold level. The processed images embody more segmentation information. The ranking based on the fitness value is applied to reflect the optimization performance and segmentation ability of the MFO. The fitness value and ranking of MFO are better than those of the other algorithms, which indicates that MFO can utilize exploration and exploitation to realize the maximum fitness value. The processed images of the MFO have more segmentation information. The best threshold values of each algorithm are given in Table 5. The choice of the threshold level directly affects the effect and quality of the processed image. MFO finds the optimal objective value according to the best threshold level. To summarize, MFO can dodge immature

convergence to achieve better segmentation accuracy and effect.

Table 6. The average execution time of each algorithm.

Images	k	Execution time (in second)						
		BA	FPA	MSA	PSO	WWO	MFO	Rank
Test 1	2	2.0504	2.5449	4.1430	3.2668	2.6219	1.8838	1
	3	2.1806	2.4652	3.6404	3.7600	3.0271	2.1470	1
	4	2.3493	2.5466	3.6591	4.2612	3.5488	2.3476	1
	5	2.3526	2.4754	3.6553	4.4621	4.1037	2.3232	1
	6	2.4534	2.5140	3.6961	4.7721	4.6332	2.4476	1
Test 2	2	2.0546	2.2233	3.5972	3.0943	2.4066	1.9862	1
	3	2.2834	2.4729	3.7247	3.7189	3.0388	2.2371	1
	4	2.4856	2.4565	3.5933	4.0514	3.7730	2.3832	1
	5	2.6244	2.5519	3.6495	13.892	4.2132	2.5896	2
	6	2.7085	2.6344	3.6366	4.6383	4.9811	2.6217	1
Test 3	2	2.0428	2.1661	3.5012	10.081	2.3834	2.0128	1
	3	2.1067	2.2636	3.6970	3.5344	2.9690	2.1931	2
	4	2.3168	2.4309	3.6582	4.1890	3.5014	2.2318	1
	5	2.4964	2.4430	3.6781	4.2050	4.1567	2.4421	1
	6	2.5751	2.5232	3.7181	4.4494	4.7794	2.5211	1
Test 4	2	2.1245	2.2784	3.5689	3.1980	2.4759	1.8874	1
	3	2.2841	2.3142	3.6728	3.9206	3.1605	2.1235	1
	4	2.4620	2.4784	3.7133	4.4914	3.6540	2.4128	1
	5	2.5981	2.5782	3.7064	4.5182	4.2552	2.5252	1
	6	2.5760	2.6113	3.7157	4.6266	4.6624	2.5645	1
Test 5	2	2.1369	2.3193	3.6874	3.3801	2.5160	1.9748	1
	3	2.3770	2.4655	3.8206	3.9882	3.1202	2.1927	1
	4	2.6246	2.5708	3.7468	4.6477	3.7042	2.4725	1
	5	2.6867	2.6756	3.7018	4.8451	4.2924	2.6573	1
	6	3.0574	2.6619	3.7099	5.2076	4.9496	2.6989	2
Test 6	2	2.0659	2.2878	3.7082	3.4298	2.4881	1.8933	1
	3	2.3009	2.4216	3.7141	3.7289	3.1803	2.1388	1
	4	2.5690	2.5297	3.7186	4.4016	3.6365	2.5144	1
	5	2.7012	2.7481	3.8449	4.6251	4.4892	2.5788	1
	6	2.7363	2.6549	3.5908	5.0158	5.0763	2.6390	1
Test 7	2	2.1999	2.2860	3.7677	3.4719	2.5184	1.7930	1
	3	2.3913	2.4617	3.8282	4.4295	3.1260	2.1823	1
	4	2.5758	2.5069	3.6892	4.5830	3.6773	2.5712	2
	5	2.8661	2.6928	3.7420	4.9113	4.3654	2.6331	1
	6	2.8314	2.7552	3.8166	4.7869	4.8823	2.7347	1
Test 8	2	2.1196	2.2375	3.5308	3.2347	2.6986	1.8208	1
	3	2.1790	2.4691	3.6472	4.1744	3.2448	2.0801	1
	4	2.4523	2.5377	3.6976	4.4788	3.7100	2.3825	1

Continued on next page

Images	k	Execution time (in second)						
		BA	FPA	MSA	PSO	WWO	MFO	Rank
	5	2.5949	2.5873	3.7993	4.6852	4.3499	2.5537	1
	6	2.6545	2.6189	3.7323	5.1430	4.7387	2.5936	1
Test 9	2	2.1158	2.2435	3.7132	3.0533	2.3750	1.8010	1
	3	2.2377	2.3223	3.7861	3.8355	2.9901	2.1133	1
	4	2.3979	2.4493	3.6818	4.1490	3.5694	2.3356	1
	5	2.4748	2.4971	3.6920	4.3074	4.2244	2.4209	1
	6	2.4707	2.5587	3.7407	4.6462	4.5649	2.5010	2
Test 10	2	2.0775	2.1429	3.4231	3.0237	2.3839	1.6726	1
	3	2.2550	2.2918	3.5819	3.4502	2.9528	2.2082	1
	4	2.3523	2.3987	3.6269	4.0056	3.4883	2.3009	1
	5	2.3653	2.4887	3.7718	4.1497	4.0963	2.3358	1
	6	2.42209	2.5584	3.7138	4.3912	4.6485	2.4192	1

The average execution time of each algorithm is given in Table 6. The execution time reflects the computational complexity of each algorithm and the realization ability of the problem. At the same threshold level, the algorithm consumes less time and has a better optimization performance. As the segmentation threshold level increases, the time consumption gradually increases. Compared with other algorithms, the execution time of the MFO is better. MFO consumes less time to address image segmentation.

Table 7. The PSNR of each algorithm.

Images	k	PSNR values						
		BA	FPA	MSA	PSO	WWO	MFO	Rank
Test 1	2	54.2052	54.0193	52.6804	53.9194	53.3728	54.2052	1
	3	55.2116	54.7434	55.5773	55.8071	54.6025	56.2391	1
	4	56.1645	56.9214	55.8071	56.8053	55.3545	57.8253	1
	5	56.9214	57.2139	55.4283	67.4037	56.5877	61.6781	2
	6	57.5946	55.9464	63.0931	56.3249	56.3249	68.3558	1
Test 2	2	56.8302	55.6667	55.2570	55.6667	56.4413	56.8496	1
	3	55.5716	55.8390	55.5716	56.9284	58.4671	58.5125	1
	4	56.8302	57.5587	56.8302	57.7738	57.7738	58.6671	1
	5	60.8783	57.6678	59.6180	57.5587	57.4521	58.5125	3
	6	58.7405	57.4521	58.5890	61.3621	56.5356	59.3017	2
Test 3	2	61.6543	60.6947	61.1888	59.1452	57.4414	63.7976	1
	3	60.8653	64.1985	58.3970	63.3619	61.8294	65.2923	1
	4	60.5327	65.0068	64.6841	64.3602	66.0216	66.2020	1
	5	62.3681	65.6347	65.9875	70.5641	62.9357	67.5475	2
	6	61.6543	66.8489	65.6347	66.0853	65.7478	68.2531	1
Test 4	2	53.7298	51.8359	53.9307	53.8281	53.3635	54.0080	1
	3	55.7727	53.2812	56.2939	53.7936	54.4415	56.3931	1

Continued on next page

Images	k	PSNR values						
		BA	FPA	MSA	PSO	WWO	MFO	Rank
	4	69.8724	64.0939	68.4585	70.6684	57.7874	70.6684	1
	5	64.6339	76.5339	64.6339	75.2461	54.0512	72.8879	3
	6	55.3686	68.4585	62.6342	59.8785	71.2461	71.5255	1
Test 5	2	55.9353	54.4501	55.5723	56.1030	56.1944	56.3859	1
	3	56.1489	56.5395	56.6408	56.1489	55.6932	56.7947	1
	4	56.0035	55.0787	56.1030	55.9353	55.6534	56.4888	1
	5	57.1093	57.6783	57.8691	59.3191	56.4366	58.1327	2
	6	58.4300	59.0280	57.8027	57.0566	57.3278	59.0280	1
Test 6	2	52.4037	50.9764	51.2852	52.4037	52.3196	52.9992	1
	3	53.7675	52.9992	52.9146	54.0808	54.3233	55.5443	1
	4	56.9588	55.0405	54.3233	52.1517	55.6405	57.5981	1
	5	55.2454	54.8447	55.8429	57.9588	56.4138	58.3841	1
	6	55.1430	57.0950	54.9352	56.4138	56.7949	59.1946	1
Test 7	2	52.1853	50.1439	52.1853	52.2305	52.2305	53.6248	1
	3	53.1369	52.7825	54.3929	52.8286	54.1279	54.4252	1
	4	53.5238	53.9356	52.9913	54.1279	52.2737	54.9081	1
	5	54.7169	55.1408	54.7169	52.7825	53.9928	57.7724	1
	6	54.4130	54.6643	54.4252	54.4600	54.1279	55.1408	1
Test 8	2	51.2625	51.1401	50.7696	51.1401	51.0819	51.8484	1
	3	52.1217	50.3589	53.9395	53.4810	51.0247	55.1889	1
	4	58.3667	52.9207	51.8484	52.6689	58.1344	59.3198	1
	5	54.6097	54.7941	52.1217	60.9018	53.6305	61.3007	1
	6	62.1126	55.4021	56.8943	55.1889	55.4021	64.4974	1
Test 9	2	54.7684	56.0844	50.9610	54.6524	54.5365	56.2376	1
	3	53.9616	56.2376	54.6524	53.8444	54.6524	61.7530	1
	4	67.6683	73.5879	67.6683	55.5133	58.1149	68.0342	2
	5	62.8685	59.5786	67.1721	67.1721	56.2376	73.5879	1
	6	72.9963	62.8685	68.6413	57.8977	68.6413	79.2437	1
Test 10	2	49.4261	49.4702	49.3850	49.4109	49.2129	49.4809	1
	3	49.3039	49.4109	49.3191	49.3437	49.2012	49.4261	1
	4	49.3976	49.3267	49.3850	49.4261	49.4261	49.5080	1
	5	49.2362	49.3115	49.4261	49.1974	49.3437	49.5080	1
	6	49.5634	49.4261	49.2816	49.3850	49.4702	50.5952	1

The PSNR of each algorithm is given in Table 7. The selection of the threshold level is crucial for the accuracy and effect of image segmentation. The overall segmentation performance improves if the threshold level increases. The PSNR is an effective criterion to determine the distortion between the provided image and the processed image and the segmentation ability of the multilevel thresholding image. The PSNR increases accordingly when the threshold level becomes larger, and the algorithm has lower distortion. To demonstrate the advantage of the MFO, the ranking is based upon the PSNR. For a given image, the threshold levels are defined as 2, 3, 4, 5 and 6; each algorithm has 50 PSNR values; and 43 PSNR values of the MFO are better. The PSNR values and ranking of the MFO are

better than those of other algorithms, which indicates that the MFO has better overall segmentation performance and better superiority. The experimental results indicate that MFO has stable practicability and robustness to obtain a better segmentation effect.

Table 8. The SSIM of each algorithm.

Images	k	SSIM values						
		BA	FPA	MSA	PSO	WWO	MFO	Rank
Test 1	2	0.5811	0.5630	0.4666	0.5589	0.5389	0.5827	1
	3	0.6174	0.6028	0.6291	0.6196	0.5792	0.6677	1
	4	0.6275	0.6713	0.6450	0.6841	0.6040	0.6962	1
	5	0.6706	0.6692	0.6240	0.6779	0.6761	0.7428	1
	6	0.6579	0.6752	0.7295	0.6565	0.6272	0.7853	1
Test 2	2	0.5934	0.5618	0.5486	0.5624	0.5854	0.5951	1
	3	0.5849	0.5892	0.5737	0.6229	0.6440	0.6549	1
	4	0.6118	0.6431	0.6084	0.6336	0.6302	0.6680	1
	5	0.6700	0.6321	0.6721	0.6239	0.6315	0.6953	1
	6	0.6764	0.6305	0.6926	0.7184	0.6484	0.7095	2
Test 3	2	0.7776	0.7700	0.7735	0.7433	0.7326	0.7782	1
	3	0.7694	0.7963	0.7380	0.7321	0.7388	0.8075	1
	4	0.7661	0.8171	0.8107	0.7983	0.7966	0.8286	1
	5	0.8008	0.7352	0.8087	0.7841	0.7682	0.8266	1
	6	0.7468	0.7928	0.8434	0.7941	0.8482	0.8560	1
Test 4	2	0.5682	0.4613	0.5752	0.5724	0.5494	0.5840	1
	3	0.6295	0.5643	0.6447	0.5538	0.5765	0.6573	1
	4	0.7995	0.7457	0.8018	0.7963	0.7050	0.8387	1
	5	0.8021	0.7892	0.7802	0.8431	0.5767	0.8509	1
	6	0.6250	0.8207	0.8056	0.7640	0.8148	0.8493	1
Test 5	2	0.5874	0.5694	0.5478	0.5923	0.5947	0.6025	1
	3	0.6513	0.6167	0.6558	0.6536	0.5760	0.6752	1
	4	0.6629	0.5762	0.6594	0.6147	0.6122	0.6669	1
	5	0.6686	0.6775	0.6918	0.7207	0.6737	0.7221	1
	6	0.7354	0.7299	0.7183	0.6835	0.7114	0.7593	1
Test 6	2	0.3856	0.3073	0.3285	0.3856	0.3823	0.4098	1
	3	0.4862	0.4423	0.4438	0.5025	0.5060	0.5417	1
	4	0.6023	0.5147	0.5425	0.4154	0.5323	0.6141	1
	5	0.5798	0.5765	0.5771	0.5794	0.6314	0.6562	1
	6	0.6170	0.6676	0.6141	0.6412	0.6737	0.7035	1
Test 7	2	0.5412	0.2944	0.5452	0.5357	0.5357	0.6049	1
	3	0.5439	0.5643	0.6308	0.5533	0.6497	0.6713	1
	4	0.5819	0.5867	0.5474	0.6396	0.5011	0.6571	1
	5	0.6752	0.6858	0.6817	0.5044	0.5725	0.7308	1
	6	0.6826	0.6776	0.6360	0.6650	0.5908	0.6938	1
Test 8	2	0.3781	0.3656	0.3403	0.3656	0.3661	0.4100	1

Continued on next page

Images	k	SSIM values						
		BA	FPA	MSA	PSO	WWO	MFO	Rank
	3	0.4250	0.2956	0.4948	0.4927	0.3533	0.5583	1
	4	0.6528	0.4934	0.3981	0.4760	0.6109	0.6803	1
	5	0.5484	0.5568	0.3853	0.4986	0.5249	0.6793	1
	6	0.7181	0.5974	0.6489	0.6071	0.5787	0.7293	1
Test 9	2	0.5502	0.6010	0.2334	0.5494	0.5547	0.6075	1
	3	0.4929	0.5707	0.5475	0.4852	0.5360	0.6251	1
	4	0.5860	0.6949	0.6196	0.4427	0.5586	0.7198	1
	5	0.6465	0.6827	0.7318	0.6505	0.4812	0.7441	1
	6	0.7208	0.6556	0.7322	0.5454	0.7166	0.7698	1
Test 10	2	0.1722	0.1666	0.1571	0.1729	0.1502	0.1733	1
	3	0.1611	0.1719	0.1644	0.1708	0.1475	0.1772	1
	4	0.1822	0.1678	0.1775	0.1753	0.1804	0.1927	1
	5	0.1486	0.1511	0.1786	0.6364	0.1645	0.1915	2
	6	0.2091	0.1812	0.1565	0.1697	0.1755	0.2901	1

The SSIM of each algorithm is given in Table 8. SSIM based on the brightness, contrast and structural information is used to determine the visual similarity between the provided image and the processed image. When the threshold level increases, the SSIM value becomes larger. The optimization algorithm obtains the processed image with less distortion, and the processed image is close to the provided image. The ranking based on the SSIM value is used to detect the segmentation ability. The ranking of the MFO is better, which indicates that the segmented images of the MFO contain more segmentation information. Each algorithm has 50 SSIM values, the first-place ranking of MFO is 48 and the second-place ranking of MFO is 2. The SSIM values and segmentation effect of the MFO are better than those of other algorithms. MFO obtains a segmented image that is close to the original image. The experimental results indicate that MFO has better calculation precision and overall segmentation performance.

Table 9. The *p-value* of Wilcoxon rank-sum.

Images	k	Wilcoxon test				
		MFO vs BA	MFO vs FPA	MFO vs MSA	MFO vs PSO	MFO vs WWO
Test 1	2	4.1486E-08	2.9135E-07	3.7215E-07	1.0905E-02	4.9558E-08
	3	4.1406E-03	**8.8674E-01**	2.7328E-02	5.9956E-04	1.8056E-02
	4	3.9271E-03	6.8298E-03	2.3871E-05	5.2530E-04	5.9054E-05
	5	3.5064E-03	4.4599E-02	7.7394E-06	2.5972E-04	7.7394E-06
	6	1.5429E-04	1.6017E-04	2.1821E-11	1.6475E-04	2.1821E-11
Test 2	2	1.5649E-11	4.1040E-11	4.5618E-11	5.7016E-03	2.6819E-11
	3	3.7061E-02	1.4931E-03	4.5940E-02	1.5957E-02	7.3007E-03
	4	2.1572E-02	7.2382E-04	1.1435E-02	9.2276E-03	1.8454E-02
	5	3.0221E-07	1.2754E-02	4.4680E-10	3.9238E-02	4.4680E-10
	6	2.1287E-05	6.0511E-04	4.8389E-10	1.9143E-02	4.8389E-10

Continued on next page

Images	k	Wilcoxon test				
		MFO vs BA	MFO vs FPA	MFO vs MSA	MFO vs PSO	MFO vs WWO
Test 3	2	1.6510E-11	1.2118E-12	1.2118E-12	2.7793E-03	1.2118E-12
	3	1.7047E-02	7.9323E-05	9.8208E-03	5.1474E-03	3.8910E-02
	4	1.5568E-02	4.8012E-02	6.1697E-05	6.0010E-03	1.8438E-03
	5	1.2075E-04	1.1553E-03	8.9553E-08	8.6976E-06	8.9553E-08
	6	4.2458E-05	1.6691E-02	3.9562E-10	1.4187E-05	3.9562E-10
Test 4	2	1.4119E-10	3.6912E-11	4.5618E-11	2.7605E-02	1.5672E-11
	3	9.6851E-08	1.0617E-03	1.3782E-02	1.8815E-02	2.0296E-11
	4	2.3687E-03	2.6206E-03	3.3040E-02	2.2534E-02	1.8354E-02
	5	1.3187E-02	6.9403E-03	**6.0393E-02**	3.3072E-02	1.3187E-02
	6	1.0490E-02	3.1661E-03	6.3076E-09	6.0661E-05	3.0311E-04
Test 5	2	4.7773E-09	1.4148E-08	1.8780E-09	6.9419E-03	1.0569E-09
	3	1.0226E-06	1.5514E-03	9.3662E-04	1.7753E-02	2.8427E-10
	4	1.2338E-04	6.8037E-03	9.0570E-05	6.0755E-04	6.6120E-03
	5	1.9974E-02	1.5408E-02	1.9284E-05	1.8299E-02	1.1010E-02
	6	2.3675E-02	1.9315E-02	9.3069E-08	2.8298E-02	2.0096E-02
Test 6	2	5.4162E-09	5.4162E-09	2.4736E-08	2.8298E-02	8.3512E-09
	3	1.2298E-07	1.5879E-06	3.6108E-02	7.8961E-03	7.6552E-10
	4	1.3692E-02	1.2204E-02	6.7678E-04	3.3418E-02	N/A
	5	1.5319E-02	1.3228E-05	3.7759E-02	8.2918E-03	8.0670E-03
	6	**7.9518E-01**	7.6105E-05	8.8807E-06	1.6308E-03	7.6285E-03
Test 7	2	1.1999E-12	1.2118E-12	1.2118E-12	5.5398E-03	1.2118E-12
	3	1.0043E-08	3.2704E-04	1.7181E-04	4.5838E-04	1.8783E-11
	4	2.2155E-05	1.3080E-04	2.0545E-03	1.8360E-02	1.8586E-03
	5	9.2817E-05	5.9629E-04	2.6611E-02	2.7791E-02	7.1075E-03
	6	**8.1845E-01**	1.8944E-02	1.0237E-07	7.3815E-04	2.6871E-02
Test 8	2	1.1067E-08	1.2380E-09	1.2377E-09	9.5909E-03	1.2377E-09
	3	5.0104E-03	5.7108E-04	7.4568E-04	2.5069E-02	7.4568E-03
	4	6.0746E-03	7.7156E-05	4.5418E-04	1.2662E-02	7.5196E-03
	5	1.2619E-02	7.7958E-05	4.3177E-10	5.6943E-07	4.3177E-10
	6	1.0049E-02	1.1403E-02	2.5562E-11	9.6312E-05	2.5562E-11
Test 9	2	4.1906E-10	7.2256E-10	1.2384E-09	4.5164E-02	1.1236E-09
	3	2.0185E-02	5.6566E-03	6.7067E-03	8.2284E-02	N/A
	4	1.7717E-02	8.1726E-03	2.2107E-06	2.0015E-02	7.4528E-03
	5	1.2740E-02	3.7302E-05	4.5563E-10	2.6262E-05	4.5563E-10
	6	1.4911E-02	1.7368E-03	4.3348E-10	9.9941E-04	4.3348E-10
Test 10	2	2.5301E-10	1.2384E-09	1.2384E-09	N/A	7.5947E-10
	3	8.1850E-03	1.5376E-02	1.4949E-02	**9.7602E-01**	6.5949E-04
	4	1.3894E-02	1.3909E-02	3.7115E-05	1.2287E-02	1.8341E-03
	5	**9.3482E-01**	1.5129E-02	1.2758E-05	6.5560E-03	3.6908E-04
	6	2.4276E-02	1.5559E-02	3.9225E-08	4.9250E-04	2.3328E-05

The p value of the Wilcoxon rank-sum is given in Table 9. Wilcoxon's rank-sum is utilized to

identify whether there is a noteworthy distinction between the two algorithms. If $p < 0.05$, then there is a noteworthy distinction between MFO and other algorithms. If $p > 0.05$ is expressed in bold, then there is no noteworthy distinction between MFO and other algorithms. There is a noteworthy distinction in most cases, and the experimental data are valid.

Figure 5. Segmentated images of Test 1 for different algorithms using Kapur method at levels 2, 3, 4, 5 and 6.

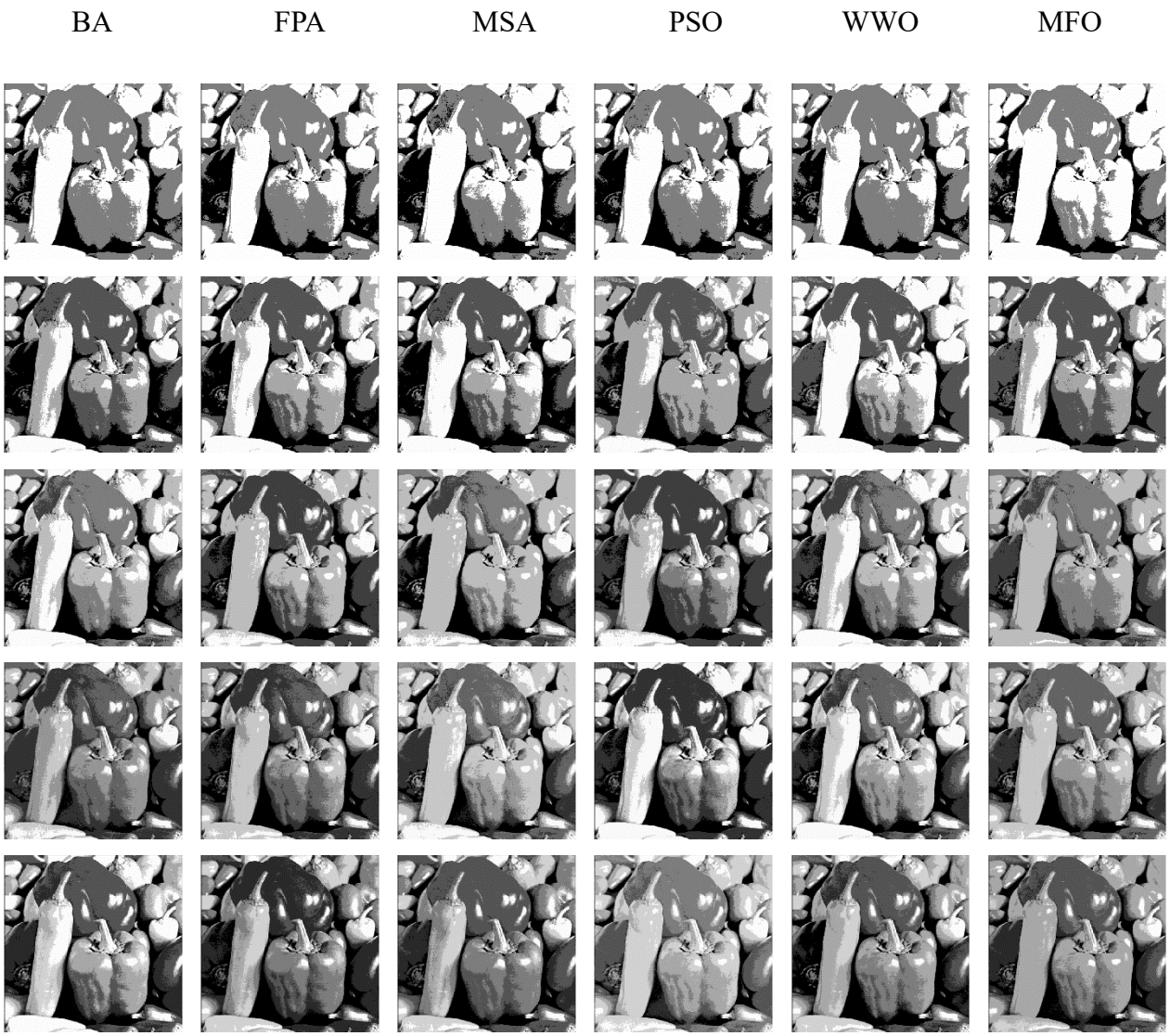
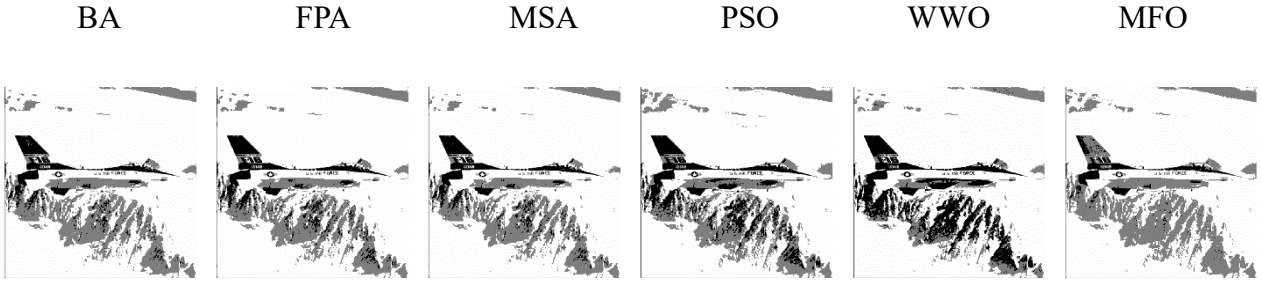


Figure 6. Segmentated images of Test 2 for different algorithms using Kapur method at levels 2, 3, 4, 5 and 6.



continued on next page

Figure 7. Segmentated images of Test 3 for different algorithms using Kapur method at levels 2, 3, 4, 5 and 6.

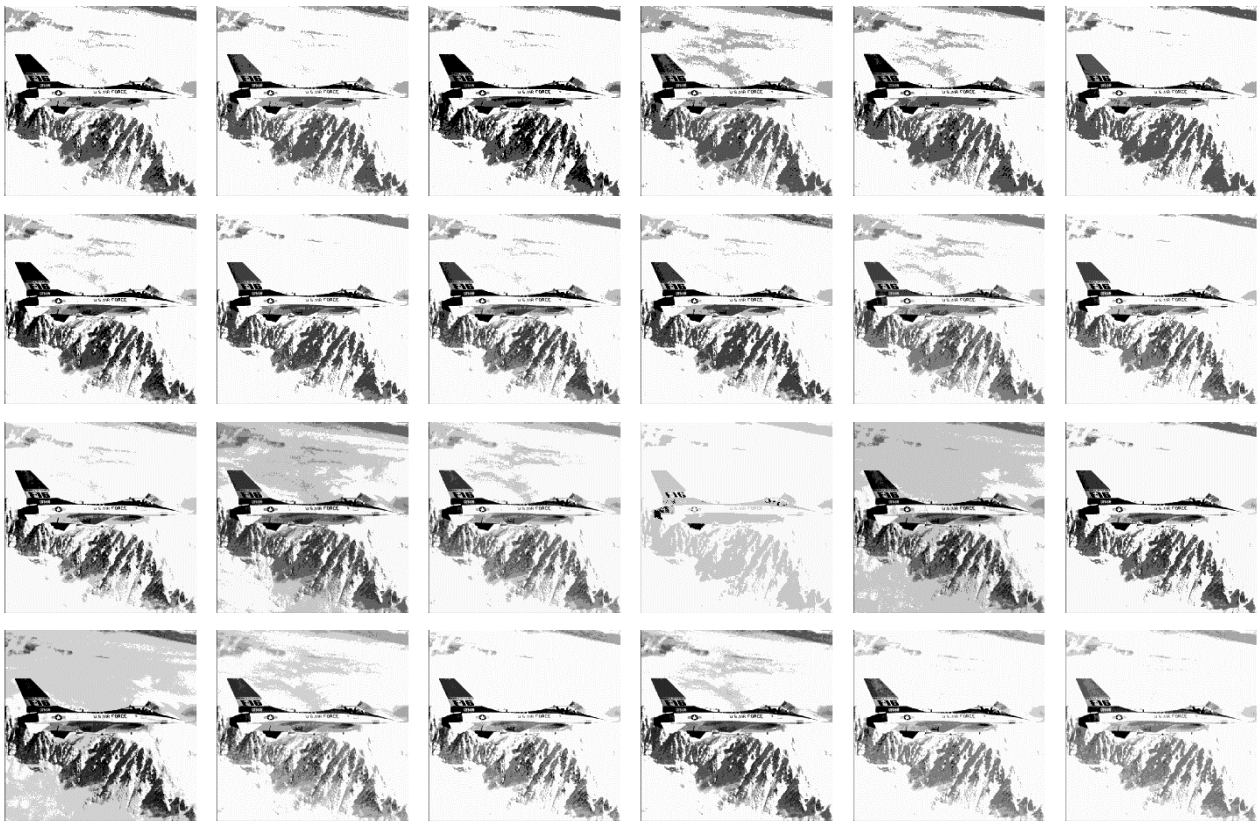
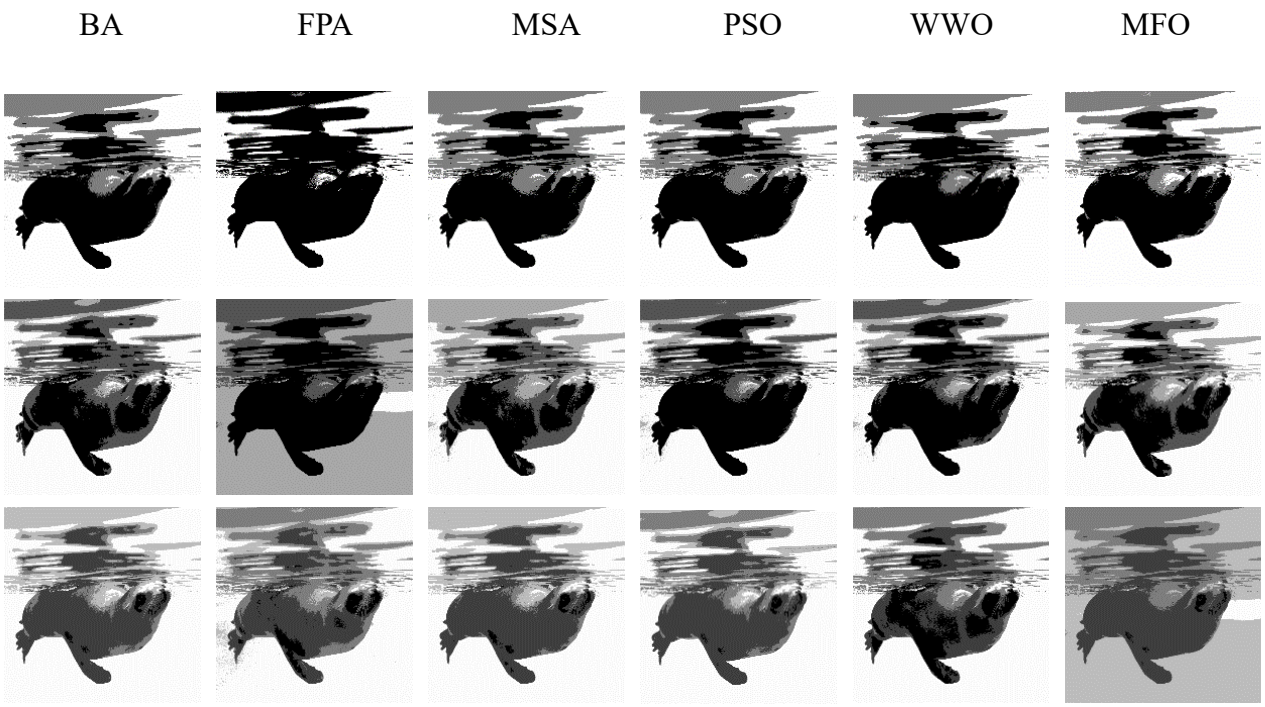


Figure 7. Segmentated images of Test 3 for different algorithms using Kapur method at levels 2, 3, 4, 5 and 6.



continued on next page

Figure 8. Segmentated images of Test 4 for different algorithms using Kapur method at levels 2, 3, 4, 5 and 6.

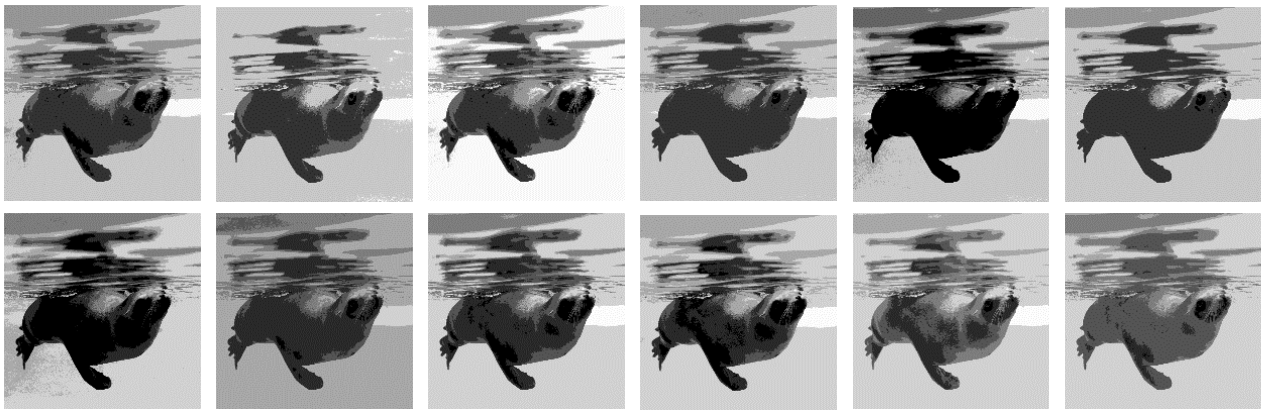


Figure 8. Segmentated images of Test 4 for different algorithms using Kapur method at levels 2, 3, 4, 5 and 6.

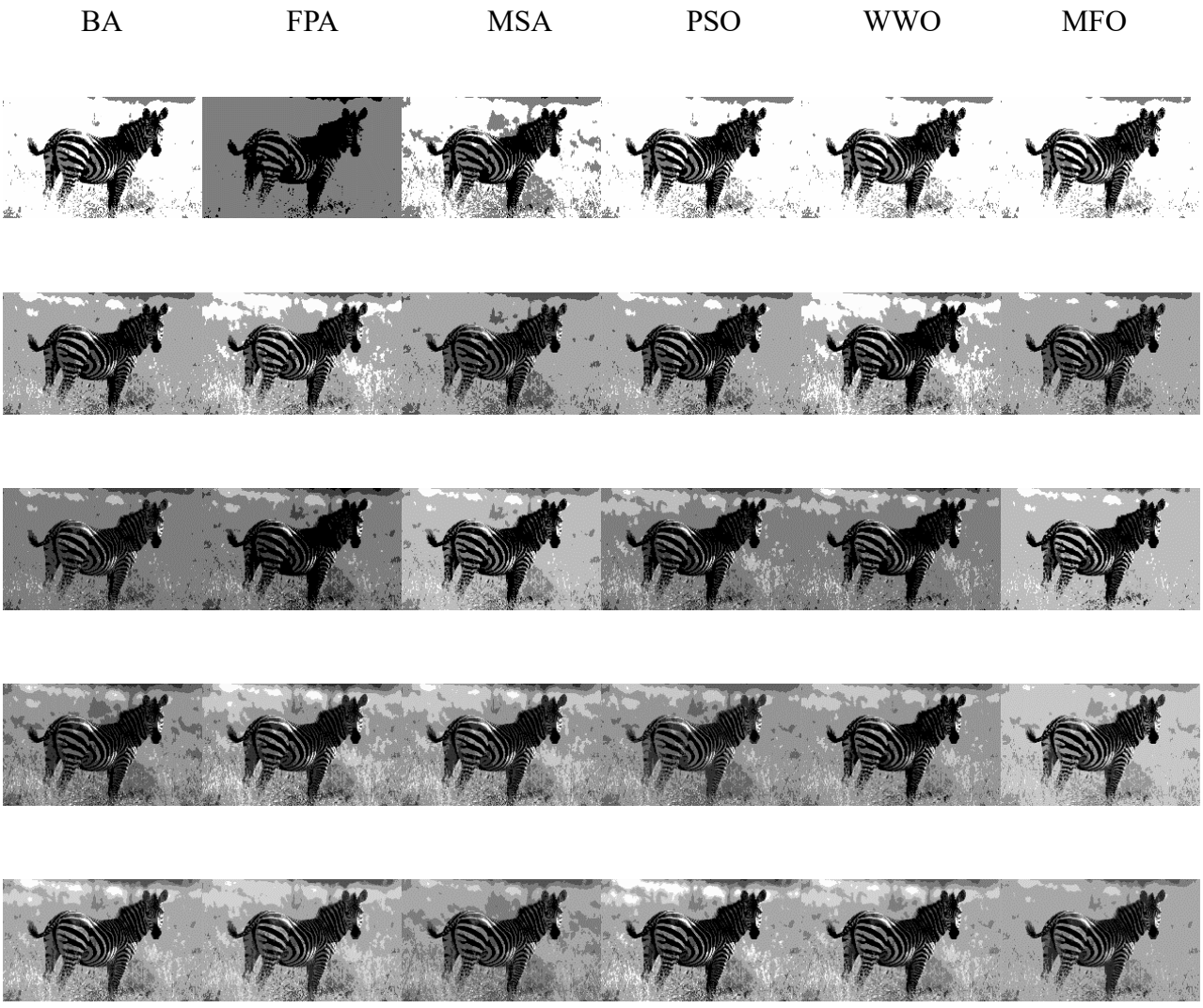


Figure 9. Segmentated images of Test 5 for different algorithms using Kapur method at levels 2, 3, 4, 5 and 6.

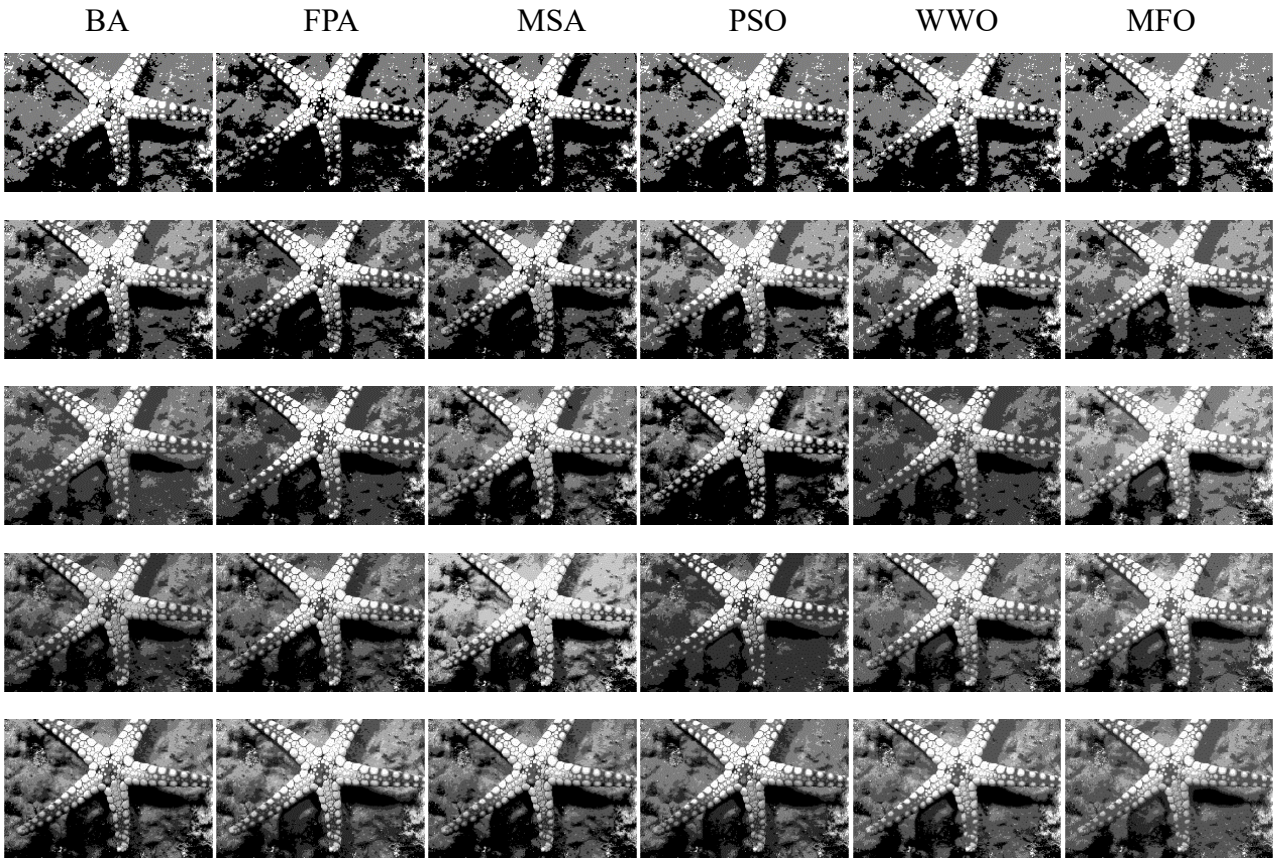
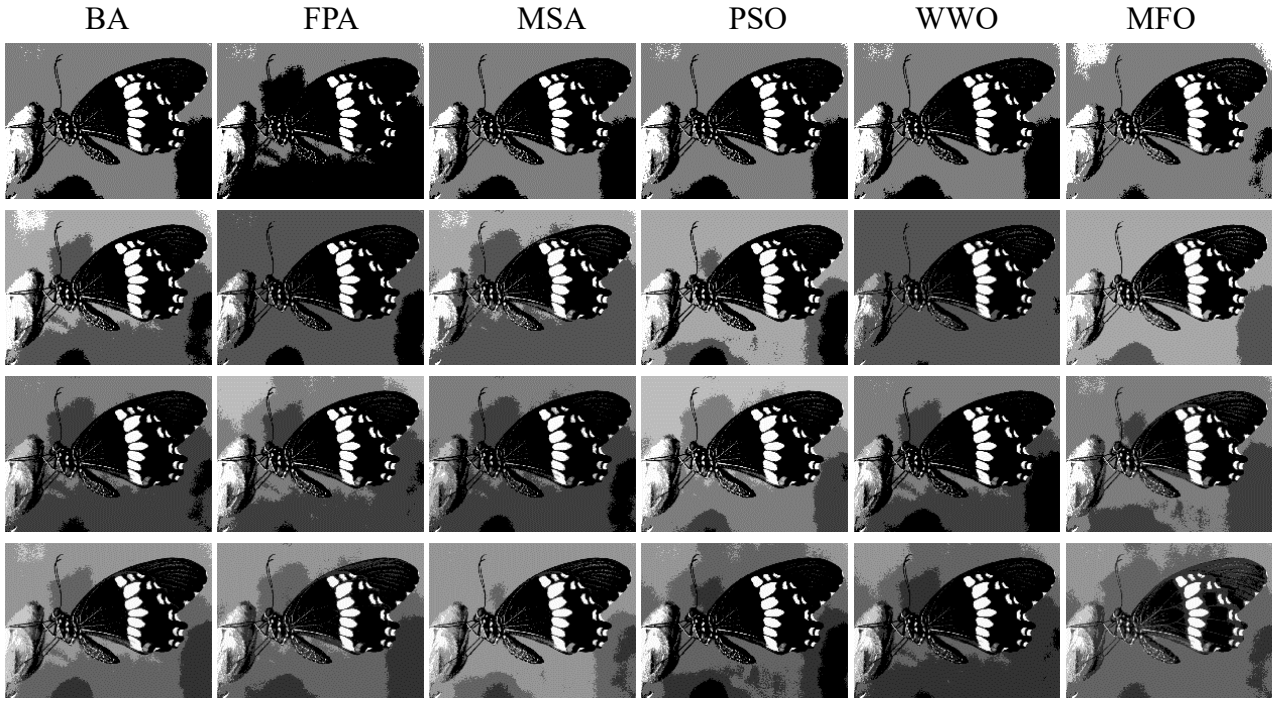


Figure 10. Segmentated images of Test 6 for different algorithms using Kapur method at levels 2, 3, 4, 5 and 6.

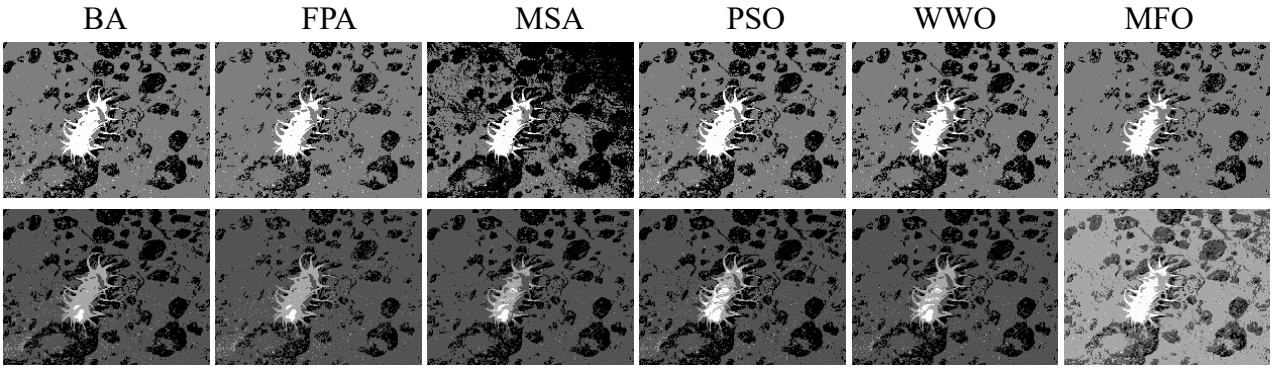


continued on next page

Figure 11. Segmentated images of Test 7 for different algorithms using Kapur method at levels 2, 3, 4, 5 and 6.

Figure 11. Segmentated images of Test 7 for different algorithms using Kapur method at levels 2, 3, 4, 5 and 6.

Figure 12. Segmentated images of Test 8 for different algorithms using Kapur method at levels 2, 3, 4, 5 and 6.



continued on next page

Figure 13. Segmentated images of Test 9 for different algorithms using Kapur method at levels 2, 3, 4, 5 and 6.

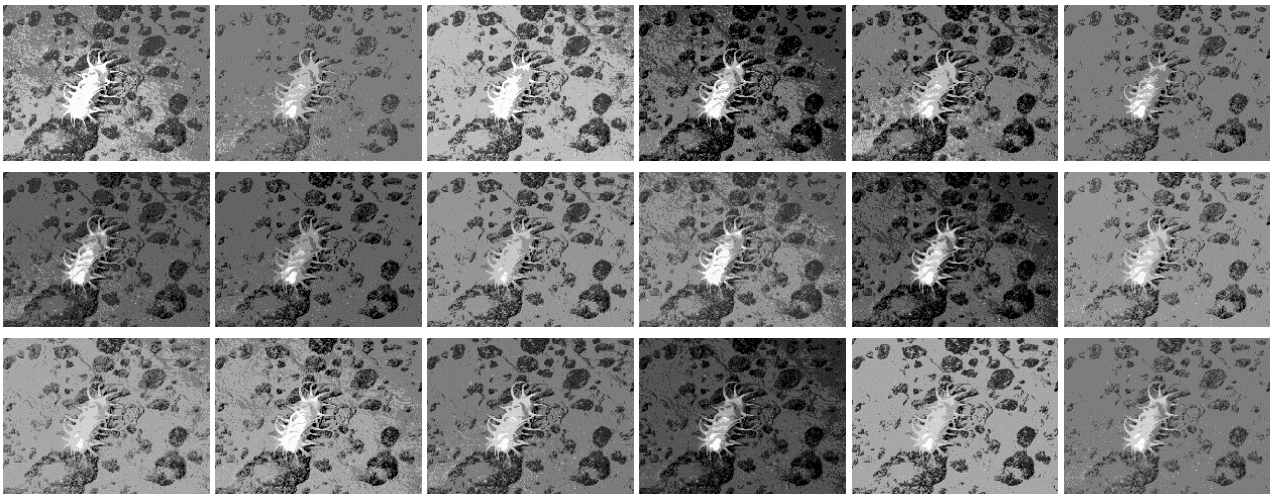


Figure 13. Segmentated images of Test 9 for different algorithms using Kapur method at levels 2, 3, 4, 5 and 6.

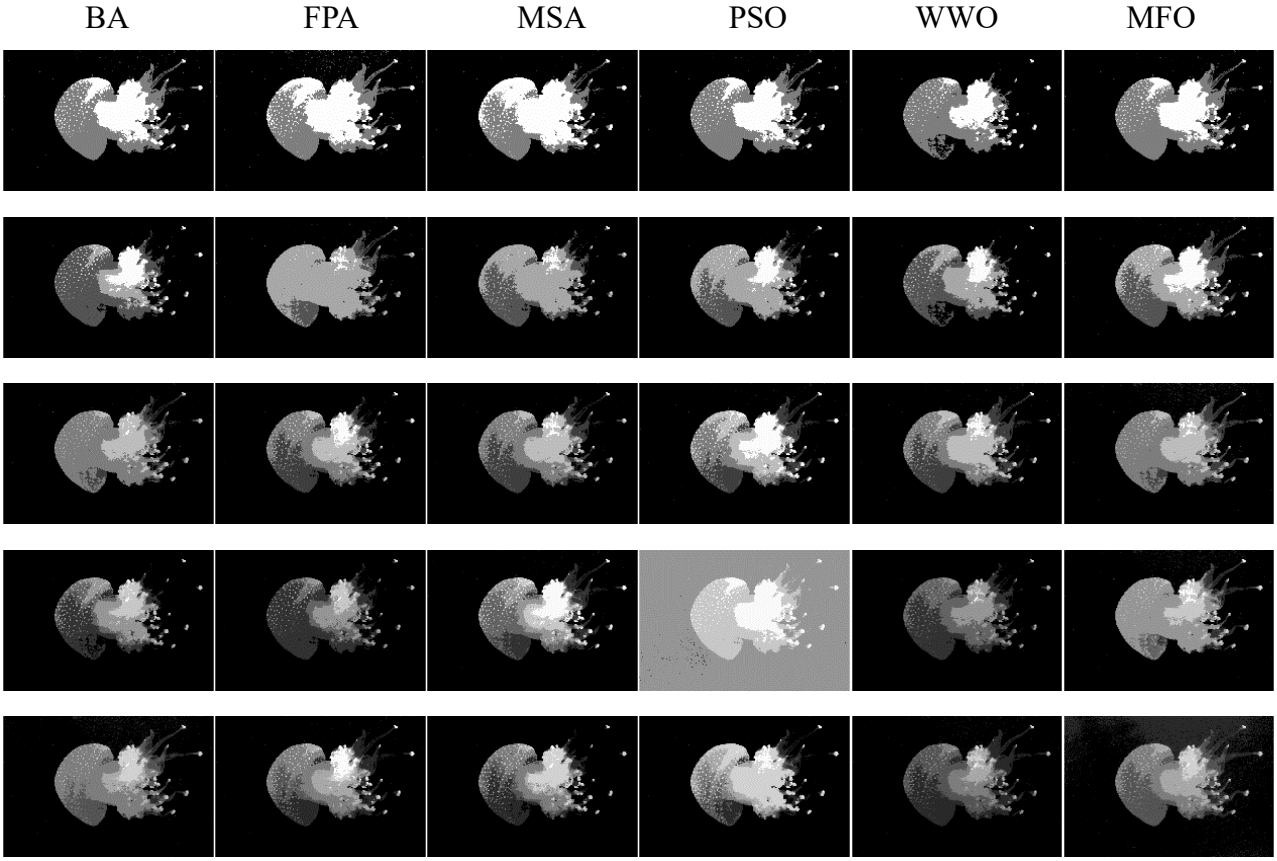


Figure 14. Segmentated images of Test 10 for different algorithms using Kapur method at levels 2, 3, 4, 5 and 6.

The segmented images of Tests 1–10 for different algorithms using the Kapur method at levels 2, 3, 4, 5 and 6 are given in Figures 5–14. The segmentation quality of the provided images is closely related to the threshold level. A segmented image with a higher threshold level includes more segmentation information. MFO has a strong enhancement and optimization performance to obtain

better segmented images that are close to the original images. The MFO can find excellent fitness values and higher calculation accuracy under a given threshold level. The population size of each algorithm is set to 30, the maximum number of iterations is set to 100 and the number of independent runs is set to 30. MFO consumes less execution time to achieve a smaller computational complexity. The PSNR value and SSIM value of the MFO are better than those of other algorithms so that the MFO has better distortion and structural similarity. Wilcoxon's rank-sum can effectively verify the noteworthy distinction between MFO and other algorithms. In summary, MFO has stronger robustness and a better segmentation effect to solve the multilevel thresholding image segmentation problem.

Statistically, the MFO simulates the transverse orientation navigation behavior of the moths to effectively search for the global optimal solution in the search space. MFO resolves multilevel thresholding image segmentation for the following reasons. First, for MFO, the algorithm framework is simple, the amount of calculation is small, the control parameters are few and the algorithm is easy to implement. The MFO has better search ability and strong robustness. Second, MFO has a better position update strategy and overall optimization performance. MFO can effectively avoid falling into a local optimal solution or premature convergence. MFO balances the global search ability and the local search ability in the optimization space to improve the convergence speed and calculation accuracy. Third, the combination of MFO and Kapur's entropy method realizes complementary advantages to enhance the segmentation effect and optimization performance. To summarize, MFO is the optimal choice to solve the multilevel thresholding image segmentation problem.

6. Conclusions and future research

The objective of image segmentation is to consume less time to attain the best threshold values by maximizing the objective value of Kapur's entropy. This paper proposes an MFO based on Kapur's entropy to resolve multithreshold image segmentation. The MFO simulates the transverse orientation navigation mechanism of moths to perform global search optimization. MFO has high search efficiency and strong robustness to avoid immature convergence or falling into the local solution, which balances exploration and exploitation to find the best solution in the optimization space. For image segmentation, as the threshold level increases, the difference between the MFO and other algorithms is significant. To verify the overall segmentation performance, compared with other algorithms, MFO has a faster convergence rate and higher calculation precision to obtain segmented images that contain more useful segmentation information. The experimental results indicate that the MFO has strong stability and a better segmentation effect in terms of the fitness value, threshold values, execution time, PSNR, SSIM and Wilcoxon's rank-sum. Meanwhile, MFO has stronger robustness and better practicality to effectively achieve the image segmentation problem.

In future research, we will develop a system that can automatically detect the optimal number of levels for the input image. MFO will be applied to settle complicated high-threshold or color image segmentation. Meanwhile, different segmentation methods will be introduced into the MFO such as Tsallis entropy, Renyi entropy, cross entropy, fuzzy entropy and Otsu. These research works will further demonstrate the effectiveness and feasibility of MFO.

Acknowledgments

The authors declare that they have no known competing financial interests or personal

relationships that could have appeared to influence the work reported in this paper. This work was partially funded by the Research on the Application of Internet of Things Technology based on Smart Chain in University Management under Grant No. 2021-KYYWF-E005.

Conflict of interest

The authors declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

References

1. M. A. Elaziz, D. Oliva, A. A. Ewees, S. Xiong, Multi-level thresholding-based grey scale image segmentation using multi-objective multi-verse optimizer, *Expert Syst. Appl.*, **125** (2019), 112–129.
2. K. S. Fu, J. K. Mui, A survey on image segmentation, *Pattern recognit.*, **13** (1981), 3–16.
3. L. He, S. Huang, Modified firefly algorithm based multilevel thresholding for color image segmentation, *Neurocomputing*, **240** (2017), 152–174.
4. S. Hinojosa, K. G. Dhal, M. A. Elaziz, D. Oliva, E. Cuevas, Entropy-based imagery segmentation for breast histology using the Stochastic Fractal Search, *Neurocomputing*, **321** (2018), 201–215.
5. S. H. Lee, H. I. Koo, N. I. Cho, Image segmentation algorithms based on the machine learning of features, *Pattern Recognit. Lett.*, **31** (2010), 2325–2336.
6. F. Breve, Interactive image segmentation using label propagation through complex network, *Expert Syst. Appl.*, **123** (2019), 18–33.
7. W. Chen, H. Yue, J. Wang, X. Wu, An improved edge detection algorithm for depth map inpainting, *Opt. Laser Eng.*, **55** (2014), 69–77.
8. Y. Li, X. Bai, L. Jiao, Y. Xue, Partitioned-cooperative quantum-behaved particle swarm optimization based on multilevel thresholding applied to medical image segmentation, *Appl. Soft. Comput.*, **56** (2017), 345–356.
9. N. Tang, F. Zhou, Z. Gu, H. Zheng, Z. Yu, B. Zheng, Unsupervised pixel-wise classification for Chaetoceros image segmentation, *Neurocomputing*, **318** (2018), 261–270.
10. D. H. M. P. Van, S. C. De Lange, A. Zalesky, A. Zalesky, C. Seguin, B. T. Yeo, Proportional thresholding in resting-state fMRI functional connectivity networks and consequences for patient-control connectome studies: Issues and recommendations, *Neuroimage*, **152** (2017), 437–449.
11. X. S. Yang, X. S. He, Bat algorithm: literature review and applicationsm, *Int. J. Bio-Inspired Comput.*, **5** (2013), 141–149.
12. X. Yang, Flower pollination algorithm for global optimization, in *International Conference on Unconventional Computation*, (2012), 240–249.
13. A. A. A. Mohamed, Y. S. Mohamed, A. A. Elgaafary, A. M. Hemeida, Optimal power flow using moth swarm algorithm, *Electr. Power Syst. Res.*, **142** (2017), 190–206.
14. J. Kennedy, R. C. Eberhart, Particle Swarm Optimization, in *International Conference on Networks*, **4** (2002), 1942–1948.
15. Y. J. Zheng, Water wave optimization: a new nature-inspired metaheuristic, *Comput. Oper. Res.*, **55** (2015), 1–11.

16. Z. Yan, J. Zhang, J. Tang, Modified water wave optimization algorithm for underwater multilevel thresholding image segmentation, *Multimed. Tools Appl.*, **79** (2020), 32415–32448.
17. X. Li, J. Song, F. Zhang, X. Ouyang, S. U. Khan, MapReduce-based fast fuzzy c-means algorithm for large-scale underwater image segmentation, *Future Gener. Comp. Syst.*, **65** (2016), 90–101.
18. X. Bao, H. Jia, C. Lang, A novel hybrid harris hawks optimization for color image multilevel thresholding segmentation, *IEEE Access*, **7** (2019), 76529–76546.
19. H. Gao, Z. Fu, C. M. Pun, H. Hu, R. Lan, A multi-level thresholding image segmentation based on an improved artificial bee colony algorithm, *Comput. Electr. Eng.*, **70** (2018), 931–938.
20. B. Akay, A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding, *Appl. Soft. Comput.*, **13** (2013), 3066–3091.
21. S. Pare, A. Kumar, V. Bajaj, G. K. Singh, An efficient method for multilevel color image thresholding using cuckoo search algorithm based on minimum cross entropy, *Appl. Soft. Comput.*, **61** (2017), 570–592.
22. Z. Lu, Y. Qiu, T. Zhan, Neutrosophic C-means clustering with local information and noise distance-based kernel metric image segmentation, *J. Vis. Commun. Image Rep.*, **58** (2019), 269–276.
23. A. Galdran, D. Pardo, A. Picón, A. Alvarez-Gila, Automatic red-channel underwater image restoration, *J. Vis. Commun. Image Rep.*, **26** (2015), 132–145.
24. S. Vasamsetti, N. Mittal, B. C. Neelapu, H. K. Sardana, Wavelet based perspective on variational enhancement technique for underwater imagery, *Ocean Eng.*, **141** (2017), 88–100.
25. V. K. Bohat, K. V. Arya, A new heuristic for multilevel thresholding of images, *Expert Syst. Appl.*, **117** (2019), 176–203.
26. S. Ouadfel, A. Taleb-Ahmed, Social spiders optimization and flower pollination algorithm for multilevel image thresholding: a performance study, *Expert Syst. Appl.*, **55** (2016), 566–584.
27. S. Pare, A. K. Bhandari, A. Kumar, G. K. Singh, A new technique for multilevel color image thresholding based on modified fuzzy entropy and Lévy flight firefly algorithm, *Comput. Electr. Eng.*, **70** (2018), 476–495.
28. S. C. Satapathy, N. S. M. Raja, V. Rajinikanth, A. S. Ashour, N. Dey, Multi-level image thresholding using Otsu and chaotic bat algorithm, *Neural Comput. Appl.*, **29** (2018), 1285–1307.
29. S. Emberton, L. Chittka, A. Cavallaro, Underwater image and video dehazing with pure haze region segmentation, *Comput. Vis. Image Underst.*, **168** (2018), 145–156.
30. R. K. Sambandam, S. Jayaraman, Self-adaptive dragonfly based optimal thresholding for multilevel segmentation of digital images, in *Journal of King Saud University-Computer and Information Sciences*, **30** (2018), 449–461.
31. G. Sun, A. Zhang, Y. Yao, Z. Wang, A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding, *Appl. Soft. Comput.*, **46** (2016), 703–730.
32. M. A. Díaz-Cortés, N. Ortega-Sánchez, S. Hinojosa, D. Oliva, E. Cuevas, R. Rojas, A. Demin, A multi-level thresholding method for breast thermograms analysis using Dragonfly algorithm, *Infrared Phys. Technol.*, **93** (2018), 346–361.
33. L. Shen, C. Fan, X. Huang, Multi-level image thresholding using modified flower pollination algorithm, *IEEE Access*, **6** (2018), 30508–30519.
34. G. Hou, Z. Pan, G. Wang, H. Yang, J. Duan, An efficient nonlocal variational method with application to underwater image restoration, *Neurocomputing*, **369** (2019), 106–121.

35. Y. Zhou, X. Yang, Y. Ling, J. Zhang, Meta-heuristic moth swarm algorithm for multilevel thresholding image segmentation, *Multimed. Tools Appl.*, **77** (2018), 23699–23727.
36. R. Kalyani, P. D. Sathya, V. P. Sakthivel, Multilevel thresholding for image segmentation with exchange market algorithm, *Multimed. Tools Appl.*, **80** (2021), 27553–27591.
37. L. Duan, S. Yang, D. Zhang, Multilevel thresholding using an improved cuckoo search algorithm for image segmentation, *J. Supercomput.*, **77** (2021), 6734–6753.
38. M. A. Elaziz, N. Nabil, R. Moghdani, A. A. Ewees, E. Cuevas, S. Lu, Multilevel thresholding image segmentation based on improved volleyball premier league algorithm using whale optimization algorithm, *Multimed. Tools Appl.*, **80** (2021), 12435–12468.
39. L. Li, L. Sun, Y. Xue, S. Li, R. F. Mansour, Fuzzy multilevel image thresholding based on improved coyote optimization algorithm, *IEEE Access*, **9** (2021), 33595–33607.
40. Q. Luo, X. Yang, Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, *Math. Comput. Simul.*, **159** (2019), 57–92.
41. Z. Li, Y. Zhou, S. Zhang, J. Song, Lévy-flight moth-flame algorithm for function optimization and engineering design problems, *Math. Probl. Eng.*, (2016), 1–22.
42. P. Wang, Y. Zhou, Q. Luo, C. Han, M. Lei, Complex-valued encoding metaheuristic optimization algorithm: A comprehensive survey, *Neurocomputing*, **407** (2020), 313–342.
43. S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, *Knowledge-based Syst.*, **89** (2015), 228–249.
44. B. Gao, X. Q. Li, W. L. Woo, G. Y. Tian, Physics-based image segmentation using first order statistical properties and genetic algorithm for inductive thermography imaging, *IEEE Trans. Image Process.*, **27** (2017), 2160–2175.
45. J. N. Kapur, P. K. Sahoo, A. K. C. Wong, A new method for gray-level picture thresholding using the entropy of the histogram, in *Computer vision, graphics, and image processing*, **29** (1985), 273–285.
46. A. Aldahdooh, E. Masala, G. Van Wallendael, M. Barkowsky, Framework for reproducible objective video quality research with case study on PSNR implementations, *Digit. Signal Prog.*, **77** (2018), 195–206.
47. Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment: from error visibility to structural similarity, *IEEE Trans. Image Process.*, **13** (2004), 600–612.
48. F. Wilcoxon, Individual Comparisons by Ranking Methods, *Biom. Bull.*, **1** (1945), 80–83.

AIMS Press